TDK-dolgozat

2023.

Solymár-Müller László Róbert

Budapesti Gazdasági Egyetem - Pénzügyi és Számviteli Kar

Examining potential ways to find Causality through Machine Learning Instruments

Dr Csongrádi Gyöngyi

Dr Kovács Endre

Table of Contents

1. Introduction

1.1. Research Motivation:

The primary catalyst for embarking on this research was the observed correlations between gold prices and various economic indices, coupled with a paucity of comprehensive research establishing causality. The quest to unravel the underlying causal web tying economic variables, notably inflation rates, to gold prices transcends mere academic curiosity. All the broader economic paradigms, investment strategies, and policy formulations are dependent on a nuanced understanding of gold price dynamics.

In the context of the transforming economic landscape, the imperative for the development of precise predictive models for gold prices is growing exponentially, with corresponding implications for investors, policy-makers, and the academic community. To enhance the predictive acumen and decipher the causal relationships, various regression models were explored, followed by a foray into neural networks.

The journey through the intricate web of machine learning models, economic theories, and a vast repository of financial data is fueled by the aspiration to provide a substantial amount of knowledge to the field of financial econometrics. I am not as interested in the explanation of gold price behavior, but rather in establishing a methodological framework that could be extrapolated to other financial assets and economic phenomena. This research project combines machine learning, economic analysis, and empirical validation to bridge the gap between correlation and causality and make it easier to understand the complicated economic picture.

Furthermore, the potential for implementing the findings in real-world scenarios, ranging from well-informed financial selections to elucidated economic policy formulations, lends this investigation a pragmatic character. The anticipation of uncovering vital hints that could be crucial in enhancing forecasting models, comprehending financial interdependencies, and ultimately contributing to a more stable financial environment, fuels this in-depth examination of the complex phenomena of gold price swings and their economic implications.

1.2. Abstract

This study provides a comprehensive investigation into the dynamics governing gold prices, utilizing a corpus of financial data spanning from 2007 to the present day, encompassing 86 different currency pairs and indices. The focus of this inquiry is the development and validation

of predictive models, primarily the Extra Trees Regressor model and neural network models, to find potential causality between the target variable and the selected explanatory variables.

An iterative methodology was adopted, commencing with a scrutiny of about 15 regression models, which culminated in the selection of the Extra Trees Regressor model. Employing a lagging method, predictions on gold prices were generated, providing a substantive foundation for the subsequent deployment of neural network models. The endeavor to refine the predictive capability led to the incorporation of five distinct types of inflation into both regression and neural network models. An analysis of the absolute mean deviation revealed that each inflation type augmented the predictive prowess of the models, albeit to varying degrees.

The findings posit a compelling narrative of not only correlation but also a hint towards causality, particularly with the inflation type that manifested the most significant improvement in predictive accuracy. This narrative is fortified through rigorous hypothesis testing and a comparative analysis of different models based on their strengths and weaknesses.

The broader implications of this research are manifold. Firstly, it proposes a robust framework for analyzing the impact of economic variables on financial assets, potentially extending beyond gold to other commodities or financial instruments. Secondly, it presents a methodological blueprint for intertwining machine learning, economic analysis, and empirical validation to bridge the gap between correlation and causality. Lastly, it beckons a foray into practical applications, ranging from informed investment strategies to nuanced economic policy formulations, thus contributing a pragmatic dimension to the theoretical discourse.

The fusion of machine learning models, economic analysis, and a rich dataset has not only provided insights into the multifaceted world of gold price dynamics but also laid down a pathway for future research endeavors aiming to unravel the complex economic interdependencies in the global financial ecosystem.

1.3. Purpose of the Research

The main goal of this study is to find a way to prove causality through creating empirically feasible models with the tools of machine learning. I chose to achieve this via the example of how gold prices are connected to different economic factors, especially inflation. Through the lens of machine learning models and empirical analysis, I'm trying to unravel the correlations and explore the realm of causality that may govern these relationships. The quest for understanding the dynamics of gold prices is twofold: to contribute to the academic discourse in financial econometrics and to propose a pragmatic framework for causality testing.

Specifically, the research aims are as follows:

1. To develop and test models that illustrate how agents behave when certain economic variables change.

2. To find out how different types of inflation affect the accuracy of the models, and see if inflation rates and gold prices are connected.

3. To conduct a comparative examination of various models to discover their strengths and weaknesses, fostering a deeper comprehension of their capabilities and how they affect the conclusions.

4. To test hypotheses to prove the claims of correlation and causality, giving a solid empirical foundation to the theories.

5. To potentially conclude economic laws—in this case regarding inflation—given that the explanatory power of my method is large and diverse enough to indicate real measurable differences between the inflation indexes.

6. The final objective is to provide a methodological blueprint for future research endeavors in financial economics, especially when it comes to exploring the dynamics of other financial assets, economic variables, and potential agents in other fields.

By combining artificial intelligence, economic research, and empirical verification, this investigation aims at the gap between correlation and causality, hopefully adding to the already numerous attempts to fill it.

1.4. Research Findings

The main objective of the study was to put forward a unified new way to prove causality between the explanatory and the target variables. I managed to prove causality through one of the methods that I applied, which was the lagged ETR model. I had hoped that the disparities among the indicators would be sufficiently significant to enable me to confidently draw inferences regarding economic observations; however, the disparities between the scorecards were not sufficiently significant to enable such conclusions.

Furthermore, I also attempted to apply the same principle to the domains of neural network models, however, regrettably, the outcomes I had hoped for were not realized. The primary obstacle encountered in that approach was to identify a universal model that could be adapted similarly to any given explanatory variable, a task that I was unable to accomplish. During my inquiries, I observed that the examined explanatory variables have indeed enhanced the network models; however, they necessitated such fine-tuning that it would render the original methodology ineffective.

However, I managed to find validity in my hypotheses in the end, and I believe that my theory was validated. After rigorous testing, I managed to support my claims from a statistical stand point of view as well, and also from a logical one.

1.5. Hypotheses

In anchoring this research, several hypotheses were formulated to guide the inquiry and provide a theoretical framework for the analysis. These hypotheses are posited as tentative explanations for the observed relationships between gold prices and the selected economic variables, particularly inflation rates, and are subject to rigorous testing and validation through the course of this research. The primary hypotheses delineated are as follows:

H1: Any of the inflation data significantly enhances the models accuracy

H2: The HCPI index is significantly better than the other indexes

H3: The Argument for Causality

The various aspects of the research are governed by these hypotheses, which guide the creation of the research approach, the selection, and validation of models, and the interpretation of outcomes. The empirical validation or refutation of these hypotheses could potentially contribute to the practical understanding surrounding causality and the interactions between examined variables.

2. Theoretical Framework and Delimitation

In this section of the study, I would like to highlight the sources and material I used. I will discuss this from the perspective of what resources I had during my research and which resources I chose to use. I will also furnish a theoretical foundation on the subjects I utilized in conducting my research and arriving at my conclusions. Additionally, I will present arguments on the rationale behind my selection of these resources, data, and methodology, which I utilized to derive conclusions from during my research.

2.1. Theoretical Underpinning

This research focuses on the intersection of financial econometrics, economic theory, and machine learning methodologies. The pursuit of unraveling the dynamics governing gold prices necessitates a multidisciplinary approach that combines conventional economic theories with contemporary data-driven methodologies. Below are the key theoretical underpinnings that scaffold this research:

2.1.1 Economic Theories

Monetary Economics

The field of monetary economics serves as a fundamental theoretical foundation for analyzing gold price dynamics, as it focuses on the interaction between monetary variables and asset prices. The central theme of this discourse pertains to the recognition of gold as a distinct monetary commodity, particularly in the aftermath of the collapse of the Bretton Woods system. The inherent properties of gold, including its scarcity and universal acceptance, have historically rendered it a reliable store of value, thereby making it a pertinent hedge against inflationary pressures (Baur and Lucey, 2010). This conceptualization has been substantiated by empirical examinations, which highlight the continued relevance of gold in providing critical inflation forecasting information to monetary policy setters (Tkacz, 2007). Moreover, the gold price dynamics have been found to exhibit a level of instability over time, often attributed to unpredictable political attitudes, further underscoring the complex interaction between monetary economics and gold prices (Capie et al., 2005).

The clearly defined correlation between gold prices and monetary variables surpasses the simplistic notions of supply and demand, exhibiting a nuanced interaction that is significantly influenced by monetary policies and global economic turbulences. The underlying theoretical framework facilitates a systematic investigation into the diverse factors influencing gold prices, thereby laying the groundwork for the present investigation.

2.1.2. Financial Econometrics

2.1.2.1. Time Series Analysis

Time Series Analysis (TSA) is a key tool in financial econometrics. It provides a platform for scrutinizing sequential data points over specified intervals, enabling a thorough exploration of underlying patterns, trends, and cyclic behaviors inherent in gold prices (Ling et al. 2015). Furthermore, TSA is crucial in examining potential lag effects and temporal associations, thereby revealing a deeper appreciation of their intricately intertwined dynamics over time.

The use of techniques like autoregressive integrated moving average models and seasonal decomposition of time series (SDS) is employed to encapsulate temporal dependencies and cyclic fluctuations, providing invaluable insights into the temporal interaction between gold prices and economic variables.

2.1.2.2. Causal Inference

The objective of this inquiry is to establish causality between economic variables and gold prices, beyond the realms of correlations. Within the realm of financial econometrics, causal inference techniques such as Granger Causality and vector autoregressive models (VAR) are pivotal. Granger causality, in particular, facilitates a systematic examination into whether past values of one time series can be used to predict future ones, thereby hinting at possible causal connections (Chu et al. 2020). Using causal inference frameworks for time series analysis can help us check if something is related to something else. This helps us connect the theories with the evidence. (Moraffah et al.2022)

2.1.3. Machine Learning Methodologies

2.1.3.1. Regression Analysis

The application of regression analysis has emerged as a fundamental instrument for analyzing the intricate correlations between gold prices and a diverse range of economic variables. Establishing predictive models, regression analysis seeks to predict gold price fluctuations based on these financial variables, thereby facilitating an avenue for examining and quantifying their interconnections. Numerous studies have utilized various regression models to evaluate the impact of macroeconomic indicators on gold prices, highlighting the potential of regression analysis in delving deeper into the fundamental dynamics that govern gold price fluctuations. For instance, a study encapsulated in “How macroeconomic indicators influence gold price management” employed Ordinary Least Square regression analysis to discern the relationships and fluctuations between gold prices and macroeconomic indicators beginning from 1990 (A et al. 2021). Similarly, a diverse array of regression models, including multivariate regression and step-wise regression, have been employed to analyze the partial effects of economic variables on gold prices, further highlighting the versatility and efficacy of regression analysis in traversing the intricate realm of gold price dynamics. (Jiang 2020)

The fundamental premise of regression analysis in this particular context is to provide a structured and quantitative framework that facilitates a more profound comprehension of the numerous interactions between gold prices and economic variables. Through the application of regression models, this research aspires to unveil the nuanced determinants of gold prices, thereby contributing to the existing body of knowledge and practical discourse surrounding gold price management and economic policy formulation.

2.1.3.2. Neural Networks

Neural network methodologies have emerged as a robust avenue to explore the nuanced dynamics governing gold prices. The essence of neural networks lies in their ability to model intricate non-linear interactions, making them a potent instrument for navigating the intricate interaction between gold prices and economic variables.

Notably, one study has employed long short-term memory with principal component analysis to predict gold prices, leveraging a historical price series from 9/11/2016 to 9/10/2021. The results of this study demonstrate the effectiveness of neural networks in capturing the temporal dependencies inherent in gold price fluctuations, thereby enabling more precise forecasts (Zhang et al. 2022).

In a deep learning-based study, a multitude of neural network architectures were employed for forecasting gold prices, including convolutional neural networks (CNN), vector sequence output LSTM, bidirectional LSTM, and encoder-decoder LSTM. Furthermore, the results obtained by the vector sequence output LSTM model demonstrated a promising level of accuracy with a mean squared error (MSE) of 0.0021, which is about 0,046 MAE (Mohtasham Khani, Vahidnia and Abbasi, 2021).

A meta-analysis of 202 Scopus-indexed journal articles from 1993 to 2023 shows that the scholarly discourse about gold price forecasting is full of myriad neural network models. This analysis provided a comprehensive overview of the academic literature, highlighting the extensive use of neural network models in forecasting gold prices. (Kabra, 2023)

Furthermore, a distinct study advocated the utilization of neural networks within a dynamic framework to forecast gold price variations, further highlighting the versatility and efficacy of neural networks in tackling the intricate task of gold price forecasting(Parisi, Parisi and Díaz, 2008).

These studies show that using neural networks to analyze gold prices is more than just a theory. It has been empirically demonstrated through experiments that it can enhance our ability to predict gold prices and think about economics, thereby providing empirical evidence of the explanatory power of certain agents.

2.1.4. Model Selection and Validation

Predictive modeling is the process of using machine learning algorithms and statistical methods to predict different data, in the case of this study, the price of gold in USD. The success of these predictive tools rests on the careful selection and thorough evaluation of the models employed. Model selection requires a careful evaluation of a model's predictive accuracy, interpretability, and generalizability to ensure its suitability for the task at hand.

The validation phase is essential for ensuring the model's predictions are reliable and credible. The process involves a meticulous evaluation of the model's performance on unobserved or novel data to gauge its predictive proficiency and resilience to overfitting. Metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared are employed to quantify the model's predictive accuracy and general reliability.

2.1.4.1. Comparative Analysis

Engaging in a comparative analysis of diverse models is imperative for discerning the strengths, weaknesses, and distinctive contributions of each model in predicting gold prices. This discussion compares different models by looking at their accuracy, how well they work together, and how easy they are to understand. This kind of analysis shows how different modeling approaches work and helps to find the best model for the task, which helps us understand gold price dynamics more clearly.

The comparison almost always ends with the choice of a model or set of models that do better at forecasting gold prices. This is the basis for other analyses and applications. Through this process of choosing, testing, and comparing models, the research aims to make the predictive models more precise and reliable so that they can better understand and predict gold price dynamics.

This framework of predictive modeling and validation provides the foundation for the empirical analysis of gold prices and economic variables. The methodology it employs involves a planned, systematic approach to examining the intricate complexities of gold prices, thereby enhancing the academic debate and facilitating practical applications in the realm of financial econometrics and economic evaluation.

2.1.4.2. Model Selection and Validation

The science of predictive modeling beckons a meticulous paradigm of model selection and validation, especially when navigating the intricacies of gold price dynamics. The most important part of making a good prediction system is selecting the right machine learning model. You need to compare and evaluate different models to find the best one for predicting gold prices. An exceptional instance in this domain is the study, wherein six machine learning models were compared to identify the most efficient model for anticipating gold price movements (Jabeur, Mefteh-Wali, and Viviani, 2021).

The validation process is essential to make sure about the models' predictive ability. This involves a meticulous evaluation of the model's performance on unobserved data to guarantee its generalizability and robustness against overfitting. A study showcased a model employing a Random Forest regression algorithm, underscored by validation to ensure the model's predictive accuracy in forecasting future gold rates. (Gadhave, 2022)

The journey of predictive modeling and validation is a careful one that involves a combination of careful model selection, validation, and comparison. These procedures aid in enhancing the models' acuity and dependability, thereby enhancing our comprehension and anticipating of gold's price fluctuations, thereby significantly enriching the scholarly and practical discourse on gold's price forecasting and economic evaluation.

2.1.5. Empirical Validation Regarding the Price of Gold

The theoretical constructs delineated within the research are subjected to empirical validation using a rich dataset that spans 86 currency pairs and indices from 2007 to the cut-off dates set by the data available but after 2023.02. This validation aims to bridge the gap between theoretical assumptions and real-world dynamics, thereby providing a foundation for the investigation of gold price dynamics and their causal connections with various economic variables.

Many scholarly endeavors support the need and methodology of empirical validation in understanding the dynamics governing gold prices. A study done over a period of time from 1990 to 2001 showed that gold prices were mainly influenced by the level of central banks' sales of gold reserves and stock market activities. This shows the important role of empirical validation in assuring that gold prices were relatively stable Elfakhani, Baalbaki and Rizk, 2009).

Moreover, the complicated economic processes that drive gold price formation, such as inflation rate, interest rate, and exchange rate, need to be tested to show that the theories are true (Lili and Chengmei, 2013). Strong evidence of causality across a broad range of quantiles is revealed by the empirical findings from a study investigating the impact of economic policy uncertainty on gold prices (Balcilar, Gupta and Pierdzioch, 2016).

Studies strongly suggest that the link between gold & silver prices and financial and macroeconomic variables are influential in explaining why gold prices move, such as the dollar index, federal funds rate, and Consumer Price Index (CPI) (Pradhan et al., 2020).

2.2. Scope and Boundaries of Research

2.2.1 Temporal Scope

This research has a certain scope and boundaries that are important for guiding the inquiry and making sure the research is focused on achieving the research goals. Through the key aspects of my inquiry and the borders set by the available data and literature, I kept myself during the research to these scopes and boundaries :

The chronological scope of this investigation, encompassing financial data from 2007 to the present, is meticulously outlined to capture the numerous economic cycles and their consequent impact on gold prices. I chose the 2007 cut-off date since I tried to use as many potentially relevant data as possible and this was the common starting point for most.

Another issue I encountered during my inquiry was that the frequency of the data was daily, rather than having one every minute or even every second. Unfortunately, I have been unable to locate a suitable source for such a kind of data, and it would not have been feasible to appropriately handle such a substantial volume.

2.2.2 Geographical Scope

Because gold is a global financial asset, its price dynamics are subject to a multitude of factors across different geographical locations. Although it's officially valued in the US dollar, there are numerous potential connections and associations between various currencies, so I included all the currencies I could find in the explanation variables.

Gold trading takes place in different geographical markets, each of which has its way of forming prices. For instance, research has demonstrated that the different geographical markets of gold trading have a significant impact on the formation of its price, highlighting the significance of considering the geographical scope when analyzing the dynamics of gold price dynamics (O'Connor et al., 2015). To model these phenomena, I used all the national indexes I could find and brought them to the same time frame and cleaned them of any complications that were caused by different time zones.

Gold is connected to financial and macroeconomic variables, so it plays various roles in the global economy. The monetary value of gold is often sought after by central banks as part of their international reserves, revealing a complex interplay of geographical, financial, and macroeconomic factors that influence gold prices (Qian, Ralescu, and Zhang, 2019). As it is not feasible to accurately simulate the process by which central banks acquire and dispose of their gold reserves, nor would it be in a format that is usable for the study will be confined to the internationally traded investment and industrial commodities, designated as "Futures" from a global economic standpoint.

The adverse effects of geopolitical uncertainties on gold prices have been demonstrated by empirical investigation, revealing the vulnerability of gold prices to political developments and uncertainties across diverse geographical areas(Mitsas, Golitsis and Khudoykulov, 2022), (Chiang, 2022). The significance of the geographical scope is further emphasized by the correlation between gold returns and changes in economic policy uncertainty. Different regions with varying economic policies and geopolitical climates may exhibit distinct gold price dynamics, thus necessitating a geographical scope in the analysis. (Shaik et al., 2023).

2.2.3. Data Scope

The temporal and geographical dimensions have determined the scope of the data. Throughout the study I tried to gather as many relevant data as possible, and I curbed their number via different algorithms. The election of the final dataset is driven by the objective of capturing a rich tapestry of economic interactions and their manifestations in gold price movements. However, the data necessary for a complete model is incomplete on purpose, for the main focus of the research is finding causality through the means of machine learning. I therefore collected five different types of inflation indexes, which I will use for my experiments according to the methodology I have set up. The strong connection between gold and inflation has been shown in numerous studies – even some that I already cited – (Musah and Ibrahim, 2015).

The data needed to be large enough to show how economic conditions changed and how they affected gold prices. The range of economic scenarios makes it a good place to do a detailed analysis of gold price dynamics.

2.2.4. Methodological Scope

The underlying concept behind the methodology was to discover causality through machine learning, thereby generating a quasi-empirical setting for measurements. This involves constructing a machine-learning model, testing the subjects of the test, in this instance, the various kinds of inflation indices, and determining which variation yields the best results.

To achieve this, I tested all the applicable regression models I could find one by one and two separate tests. The Extra Trees Regressor (ETR) model came out as a winner.

The Extra Trees Regressor model is an ensemble learning method that operates by constructing a multitude of randomized decision trees at training time and output the mean prediction of the individual trees for regression problems (scikit-learn, n.d.). I will discuss further the details of these models in a later part of my study.

The efficacy of tree-based ensemble models, including the Extra Trees model, has been already demonstrated in predicting stock price movements, highlighting their potential for unraveling intricate financial dynamics (Ampomah, Qin and Nyame, 2020). Furthermore, the Extra Trees Regressor model has been effective in predicting stock market prices, further proving its capability in handling financial data (Stock Market Prices Prediction using Random Forest and Extra TreesRegression, 2019).

Neural networks are also a good way to solve problems in financial research that traditional methods can't handle because of mistakes in financial data. They've been around for about two decades to improve precision in financial data analysis, making them a solid machine learning framework for this study (Riyazahmed, 2021). The fact that I'm using both machine learning schools of methodologies is for experimental purposes, as the target of this research is to find ways to prove causality as easily as possible.

In addition to these machine learning models, traditional financial econometric methods are also integral to the investigated methodological framework. These methods provide a foundational structure upon which the machine learning models can be effectively applied, ensuring a comprehensive approach to analyzing the gold price dynamics. I will further discuss these additional frameworks in the relevant parts of my paper.

Through this methodological scope, I attempt to underpin the analytical rigor of the research as well as demonstrate a multi-faceted approach to investigating data phenomena. By embracing both modern machine learning models and traditional financial econometric methods, I aim to conduct balanced research regarding the methodology I apply.

As for the technical aspect of creating and running the necessary code, I used Jupyter Notebook, with the Python 3 language. The thought behind this decision was made from a practical aspect, as I used this environment in the past for similar tasks.

2.2.5. Economic Variables

The study spotlights inflation rates, examining their correlations and potential causality with gold prices, among other economic variables. I have already gone into details regarding the explanatory variables that form the basis of my research, so in this section, I will focus on the pivot of the whole inquiry, the different kinds of inflation indexes.

2.2.5.1. Headline Consumer Price Index (HCPI)

The Headline Consumer Price Index (HCPI) is an important economic indicator that shows how much consumer goods and services have changed in price. In most of the academic literature and in the economics jargon it is simply known as CPI, however as for my source used it as HCPI, I’m going to continue to refer it to it in this format (World Bank, 2023.).

The Headline Consumer Price Index (HCPI) is an essential tool in economic research, assessing the average variation in prices paid by urban dwellers for a diverse assortment of merchandise and services over time. It is widely utilized to monitor inflation, which refers to a broad rise in the prices of goods and services across an economy, eroding purchasing power for both consumers and businesses. (Bryan and Cecchetti, 1993)

Statistical agencies measure inflation by first determining the current value of a “basket” of various goods and services consumed by households, referred to as a price index. The calculation of the rate of inflation, or the percentage change, over a given period is accomplished by comparing the value of this index across various periods, such as month-to-month or year-to-year. The U.S. Bureau of Labor Statistics publishes the HCPI, which measures the out-of-pocket costs of items bought by urban consumers. It is reported regionally and nationally (Levine, 2023).

In the context of higher education, inflation influences tuition fees. For example, the cost of college education has almost tripled between 1979-80 and 2020-21 at four-year public and private institutions, even after accounting for inflation (Mckinsey & Company, 2022).

To conclude, HCPI is a key indicator of inflation, indicating the average price change of a basket of goods and services. Understanding it is essential for gauging economic health and making informed policy decisions.

2.2.5.2. Energy Price Index (ECPI - EPI)

This index shows how much energy costs change in an economy. It shows how prices change for different types of energy. ECPI isn't directly connected with gold prices by any academic papers I found, however, the broader narrative of energy prices holds significance in the economic realm, potentially impacting gold prices indirectly. For example, a rise in energy prices caused the energy index to go up. Energy prices went up 29.3% for the year, which was the biggest increase in food and energy prices in 13 years (Bureau of Labor Statistics, 2023)

Given the significance of energy prices in generating inflation and potentially exerting an impact on gold prices, the ECPI could potentially serve as a pivotal variable in this investigation. The in-depth examination of the energy cost proportional index (ECPI) with gold prices could yield insightful insights into the interaction between energy price swings and gold price fluctuations, thereby enhancing a greater comprehension of the economic variables investigated in this investigation.

2.2.5.3. Food Price Index (FCPI)

The Food Price Index (FCPI) is an economic indicator that shows how prices of food commodities change over time. This index is important for understanding how prices affect the economy, including food prices and other economic factors. I found several sources on how influential FCPI can be on the price of other goods. For instance, a study wanted to find out how food prices, energy, uncertainty about economic policies, and the value of the dollar in the United States are connected, which can directly or indirectly impact the price of gold. This study adds a lot to the knowledge about how food prices are calculated, and it explains how food prices change over time.(Kirikkaleli and Darbaz, 2022). Another report from the Congressional Research Service shows how economic policies impact how much food costs go up, which, I believe, is further proof of why the FCPI index can be relevant for my research(CRS INSIGHT Prepared for Members and Committees of Congress, n.d.). Another study has revealed the average food price index and the real exchange rate figures, highlighting the significance of comprehending the macroeconomic determinants of food price fluctuations (Samal, Ummalla and Goyari, 2022). Although these studies concentrate mainly on the US market, I believe them to be relevant, since the price of gold is measured in USD, and thus it is important to reflect upon such economic phenomenon.

From a more global standpoint of view, this academic paper discussed the use of the International Monetary Fund's monthly national consumer price index (CPI), and food and non-alcoholic beverage index (FCPI), for 133 countries from January 2017 to November 2020. This paper underscores the global perspective of food price indexing and its relevance in economic analysis (Wallingford and Masters, 2021).

These articles point to the possible interdependence between food prices and other economic variables, which can be crucial in examining broader economic phenomena. I believe it's also pertinent to further investigate its connection with gold prices.

2.2.5.4. Official Core Consumer Price Index(CCPI)

The Official Core Consumer Price Index (CCPI) is an essential economic indicator that shows how much people pay for a group of things they buy over time. This index doesn't determine the core inflation rate, as it excludes some erratic costs, like food and energy. This makes CCPI a steadier, less fluctuating, and thus a more measurable alternative to CPI, while it carries more or less the same information (www.chicagofed.org, n.d.).

A study examined the potential of the Consumer Price Index (CPI) to predict gold price returns, uncovering weak evidence of in-sample predictability and evidence of out-of-sample predictability. These findings show that the CPI could be a good predictor of gold prices over a longer timeframe (Sharma, 2016). Many economists and market analysts believe that the gold spot price is significantly influenced by expected inflation. It is often said that changes in inflation expectations are the main reason why gold prices change (Blose, 2010). There exists a historically weak linear relationship between gold returns and CPI changes. Since 1971, only about 16% of the variation in gold prices has been explained by changes in CPI inflation (Reuters, 2021).

2.2.5.5. The Producer Price Index (PPI)

This indicator is about how much money people in a country make by selling their products. The index provides a critical measure of inflation from the perspective of producers instead of consumers. Numerous instances have been documented wherein movements in the Personal Price Index were correlated with fluctuations in gold prices. For instance, during the month of December, when the Producer Price Index in the United States increased by 0.2%, following a rise of 0.8% in November, it was reported that gold prices experienced a decline, as the data was less than the anticipated increase of 0.4% (Kitco News, 2022). In a different scenario, gold prices remained above $1,800 despite the rise in the U.S. Personal Income Index by 7.4%, which was more pronounced than anticipated (Kitco News, 2022a). Moreover, in April, when the U.S. PPI rose by 0.5% following March's rise of 1.6%, gold prices continued to struggle as the data was in line with expectations, but annual inflation came in hotter than expected, rising to 11% (Kitco News, 2022b). Although the sources that tie PPI to the price of gold are mostly pragmatic rather than academic, they do show that the prices paid by gold producers are affected by the overall economic situation, which could affect the price of gold directly or through other agents.

2.2.6. Modeling Boundaries

The modeling efforts are limited by the availability of data, computational resources, and the assumptions in the models and statistical methods chosen. The exploration into the dynamics of gold prices via computational models requires a framework that can withstand the intricacies of financial data. Several limitations, however, pose a challenge to this simulation endeavor.

2.2.7. Data Availability

The amount of information available has a significant impact on the range and complexity of simulation. After considering all my options, I decided to use Yahoo Finance as a source, since it has the most relevant data for my research with the longest available time frame.

2.2.8. Computational Resources

Complex modeling approaches often require a lot of computing power. Frequently, adding a few extra steps to any model can exponentially increase the necessary computational power. For example, transient simulations of dynamic systems using physics-based scientific computing tools are often limited by the availability of computational resources and power. Some studies delve into finding ways to solve a difficult pharmaceutical problem using a combination of expensive computer programs, time-series forecasting, and machine learning. This reduced the amount of computer work needed, but still made the model accurate and efficient (Kumar et al., 2020). Although these aspirations are obviously very useful and necessary, they are not strongly tied to my research subject; thus I will not devote resources to optimize my computational resources. My rule of thumb during my research was that if I can run the model on my laptop, I deem it “generally accessible”, seeing how creating methods that only run on supercomputers would be beside the point.

2.2.9. Model Assumptions

The assumptions behind the models are important to clarify because they set the rules for how the models work. It's often necessary to simplify assumptions and use computationally affordable representations to describe or predict the overall state of a complex system (as discussed in the previous passage). To apply these aspects to my research, I only used models that are largely available and tested in the general discourse of the machine learning corpus, and I neglected the very specific models found in the deeper corners of the professional literature. This scenario shows some trade-off between accuracy and computational feasibility, but the more significant standpoint was from the perspective of the research material, as I tried to keep the circumstantial factors as simple as possible.

2.2.10. Analytical Boundaries

The investigation in this study is confined to the variables and models chosen, recognizing that alternative models or variables could potentially yield different insights. The selection of variables is recognized as an essential step in inductive modeling, with the aim of identifying the group of variables most likely to have true associations with the outcomes of interest. Since I attempted to base the subject of my research on empirical observations, I decided on the Mean Absolute Error to measure how a given model was performing I aimed to further my findings by confirming with already established tests, like the Granger Causality test (I will further discuss the mechanics of this test in a later part of the paper.

2.2.11. Practical Implications

Through the research, I attempt to provide practical insights and a methodological framework for a way to potentially prove causality; however, the applicability may be contingent on the external validation of the models and findings in real-world scenarios. The endeavor to dissect the dynamics of gold prices through a lens of various economic variables is not merely an academic exercise but seeks to transcend the theoretical realm into practical applications. This research can help with investment plans and economic policies by showing how things affect each other and what potential causality they possess. A meticulous external validation is often required for the bridge from theory to practice.

2.2.12. Ethical Boundaries

Throughout selecting the research material and data, for my analysis, I did my best to use generally trusted sources. For the data source, as a lack of better means, I searched on forums and used my judgement to decide on Yahoo Finance. As for the theoretical background, my rule of thumb was to use articles that have DOI codes, proving that they have been published in a respectable article. To show how economic laws work in real life, I used simpler examples. This is because practical examples are less likely to be published as scientific articles. Nevertheless, I used sources deemed trustworthy by the citation manager program I used.

I would also like to discuss the AI tools I used during my research since current state of IT development, especially in the field of AI, is providing opportunities that will no doubt be an important part of the scientific discourse in the future just as they are in the present, albeit the boundaries in the usage of modern AI has not yet been set into stone. Because we are in a transitional era, I would like to emphasize that, I used the aid of ChatGPT and similar AI tools to help with the citations, create some of the code, to double-check my algorithms and to promote the quality of the text structure.

3. Research Methodology and Methods

This research focuses on a development method that can be used to study the relationship between any given explanatory variable and the target variable. The methodology is engineered to foster a systematic inquiry, ensuring the integrity and validity of the findings. The premise of this inquiry is that it is possible to establish causality through the appropriate AI and machine learning techniques, thereby establishing a solid foundation for the verification of causality. Based on the theoretical framework that I have set, in this chapter I will delve into research itself by presenting the methodology I used and the theoretical background, supporting it.

3.1. Development Methodology

The strategy for constructing my thesis adhered to the customary approach of examining a phenomenon that piqued my curiosity for various reasons. I started my study by finding all possible literature that exists on the subject, and strategically adding more academic literature to that corpus, to get the most complete picture I can. I then started testing the different methods of what I should use to test my theory. After the lions share of the work, which was meticulous coding and testing for me to have a result that I can measure, I started testing my hypothesizes so that I can come to a conclusion regarding my theories and possibly suggest some real life applications for it. In the following paragraphs, I will discuss my work in detail.

3.2. Literature Review

A lot of research was done to understand how gold prices change, economic factors, and using machine learning in financial econometrics. This was the basis for figuring out where there was a lack of knowledge and coming up with research ideas. It is worth mentioning, that although there is some literature on the field that I'm studying, it is very little, and it can be challenging to get a hold on it. The study of how gold price changes are connected to economic variables is a well-known area of inquiry in financial econometrics. The emergence of AI-based approaches marks a fresh horizon in this scholarly discourse. The quest to uncover causal connections, rather than merely arithmetic connections, between gold prices and economic variables, particularly inflation indices, is the distinctive focus of this investigation.

The traditional approach to investigating gold price dynamics primarily hinges on financial econometric models. Numerous studies have explored the causal relationships between gold prices and various economic variables using traditional econometric techniques, that I aimed to draw a parallel to through new ML-based methods. On the contrary, the application of machine learning in this realm is relatively young. Some scholars have begun to leverage machine learning models to better understand the underlying mechanisms of causality. An example of this is the study by (Zhao and Liu, 2023); The subject of this paper focuses on a Python package called Causal ML, which is intended to facilitate causal inference by employing machine learning techniques. In this situation, causal inference means knowing how something happened that caused something else to happen, instead of just noticing it. The package includes tools for modeling uplift and making inferences about causality using machine learning based on recent research. Its goal is to provide a common interface for estimating the conditional average treatment effects (CATE) or individual treatment effects (ITE) based on experimental or observational evidence. These types of analyses, in my opinion, are crucial for comprehending causal relationships in data, which extends beyond the typical predictive modeling performed in machine learning.

The paper "Causal Machine Learning: A Survey and Open Problems" also focuses on a domain referred to as Causal Machine Learning (CausalML), which represents an umbrella term for machine learning methods that are structured around a framework known as a Structural Causal Model (SCM). Using this approach, it is possible to analyze the effects of interventions on a system, as well as explore what would have happened under different circumstances (Kaddour et al., 2022), thus making an effort towards proving causality. Another paper talks about how to make connections between things based on data. They use assumptions, study plans, and estimation strategies while also mentioning the challenges posed by unobserved confounders in many settings (Wu et al., 2022). Studies like this certainly add to my relevant research corpus, but at the same time, it is difficult to draw exact parallels to the subject that I'm investigating.

3.3. Data Collection and Preprocessing

As I previously stated, the source for all my data was Yahoo Finance. I wrote a function called **fetch_all_instruments** (all the code for the functions I will mention will be found in the Appendix of this paper). This function takes a list as an input, and downloads all the data from YFinance. I set the starting date later to 2007-08-30 later, as this was a common starting date for a lot of data from this source. I used the price each instrument had at the end of the day, otherwise known as the closing price.

All the 86 explanatory variables were handpicked from this source. Some of them weren't chosen because of their origin was deemed unimportant based on the academic literature I examined, or they just didn't have enough data entries to go by.

I gathered three separate kinds of data from this source; Currency pairs, indexes, and futures (in this case a future means general goods, like wheat, or gold for example). The list of these items will be included in the appendix.

The aim was to create one big data frame with all the variables that I gathered, which proposes its challenges. First and foremost, I decided that the index of the data frame should be the date, but because of different time zones, and given the general colorful nature of all the data that I will need to use, the date format has to be unified.

The second challenge that was proposed by the raw data was that because of the holidays and the different external influences, many data had gaps in them. These usually took only one or two days, but I do had to drop a few of these variables because it had too many gaps in them, reducing their explanatory power.

To fill these gaps, I used an interpolation method (Hintikka and Halonen, 1999). Basically, this is a method to fill up the missing rows with the average of the two data entry that we have. If there is more than one consecutive gap in the data, it fills it up by adding equal fractions to each row every time.

$$\text{Linear Interpolation}(y) = y_1 + (x - x_1)\frac{(y_2 - y_1)}{(x_2 - x_1)}$$

1. illustration, formula for linear regression, source: https://www.cuemath.com/linear-interpolation-formula/

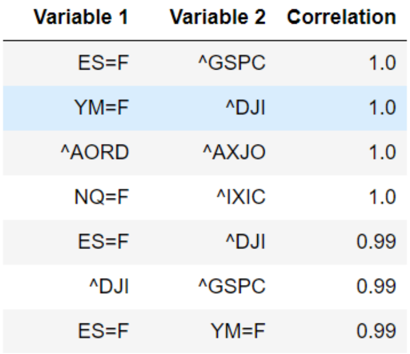
After these steps the final shape of my data frame was (5208, 87), which means it has 5208 lines of data and 87 columns, representing the 86 explanatory variables plus the target variable, the price of gold.

Preparing the inflation data was a bit simpler. I used the data from a World Bank Study (Ha, Kose and Ohnsorge, 2021) named A Cross-Country Database of Inflation, which was updated in 2023. Here if found the monthly number of each kind of inflation type that I have previously established as the subjects of the research. The only challenge proposed here was that I had to make my original data shorter because the data from this source only spanned until 2023.03. Furthermore, the Producer Price index only spans until 2022.08, so rather than making my original data frame even shorter, I used a linear regression model to predict the list 7 rows of this column, making a 188 unit long row 195 long. Because this data was on a monthly basis, I used an algorithm called **fill_daily_data** to create data for every day of each month, through the means of linear interpolation. It is critical to fill these gaps, so that the machine learning methods can run without getting an error. After cleaning and preparing all of my data frames, I was ready to begin the next phase of the research.

3.4. Correlation Analysis

In preparation of the data, it is critical to perform some tests, to make sure that the model won't have any kinds of distortions because of certain impacts of tangled information. In artificial intelligence applications, the correlation analysis technique is especially relevant. This helps us understand how data parts are connected and how they affect the thing we want to predict. This means that in preparation of the research, I must make sure to do away with the effects. Some of my data will surely have some correlation among itself, providing the same kind of explanatory force to the model more than twice, yielding inaccurate results (Daoud, 2017). Working from this, the current task should be to filter out this phenomenon.

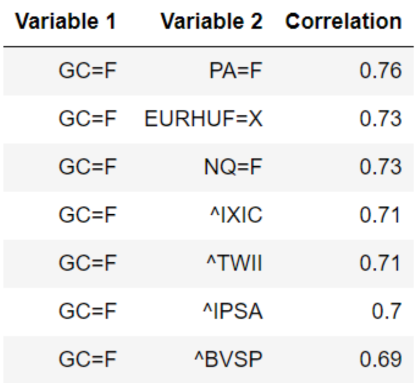
To combat this effect, I wrote an algorithm called "correlation". I run it through the whole data frame, revealing that there is, in fact, significant cross correlation among the explanatory variables. To illustrate this, I will paste a sample from this table—as I don't see the point in pasting the whole lengthy table into the paper:



Variable 1	Variable 2	Correlation
ES=F	^GSPC	1.0
YM=F	^DJI	1.0
^AORD	^AXJO	1.0
NQ=F	^IXIC	1.0
ES=F	^DJI	0.99
^DJI	^GSPC	0.99
ES=F	YM=F	0.99

2. illustration, correlation among all variables, a sample, source: own findings

To put this information into a relevant context from the perspective of the research, I will also need to investigate the correlation of the explanatory variables to the target variable.



Variable 1	Variable 2	Correlation
GC=F	PA=F	0.76
GC=F	EURHUF=X	0.73
GC=F	NQ=F	0.73
GC=F	^IXIC	0.71
GC=F	^TWII	0.71
GC=F	^IPSA	0.7
GC=F	^BVSP	0.69

3. illustration, correlation among explanatory variables and the target variable, a sample, source: own findings

The most effective way to rid of the cross correlation is to keep all the variables that correlate most with the target variable, but filter out the ones that possess a cross correlation that I deemed too high.

To overcome this technical challenge, I wrote the **select_variables_correlation** function, that I intend to use as a feature engineering tool. The function, performs variable selection based on the correlation between predictor variables in df_X and the target variable in df_Y, as well as the correlation between the predictor variables themselves.

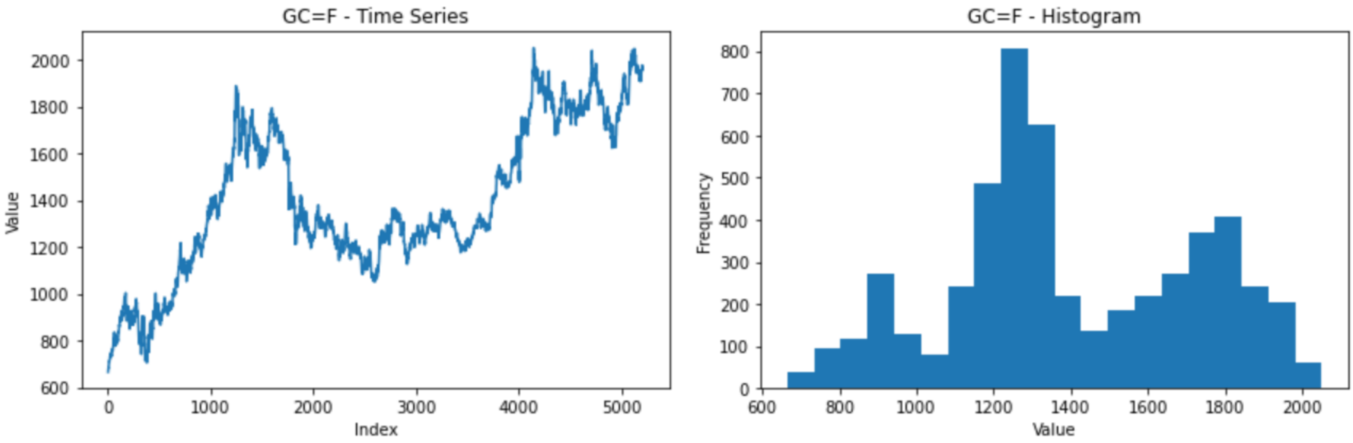
As feature engineering is a critical aspect of my research and of machine learning in general, please allow me to address the methodology behind this function briefly;

First, the function computes the Pearson correlation (Schober, Boer and Schwarte, 2018) between each column in df_X and df_Y. It sorts the things in df_X by how closely they are related to df_Y. Then initializes an empty list, selected_vars, to store the selected predictor variables. For each variable in df_X, it checks if the variable is not already in the selected_vars list, its correlation with all other already-selected variables is below the threshold correlation_level and if both conditions are met, the variable is added to the selected_vars list. The function then returns the list of selected variables and their respective correlations with the target variable. This function picks variables with a strong connection to the target, but it also checks to see if the selected predictor variables have a strong correlation with the target.

3.5. Plotting the Variables

In data analysis and machine learning, it is imperative to plot variables for several reasons. Visualizations help explain how data is spread out and show where things are connected, whether they are linear or not. These graphic representations show that many statistical methods are based on important assumptions, making them easier to understand and make clearer the findings. Furthermore, they serve as a highly effective instrument for identifying errors or inconsistencies within datasets, thereby guaranteeing precise analytical outcomes.

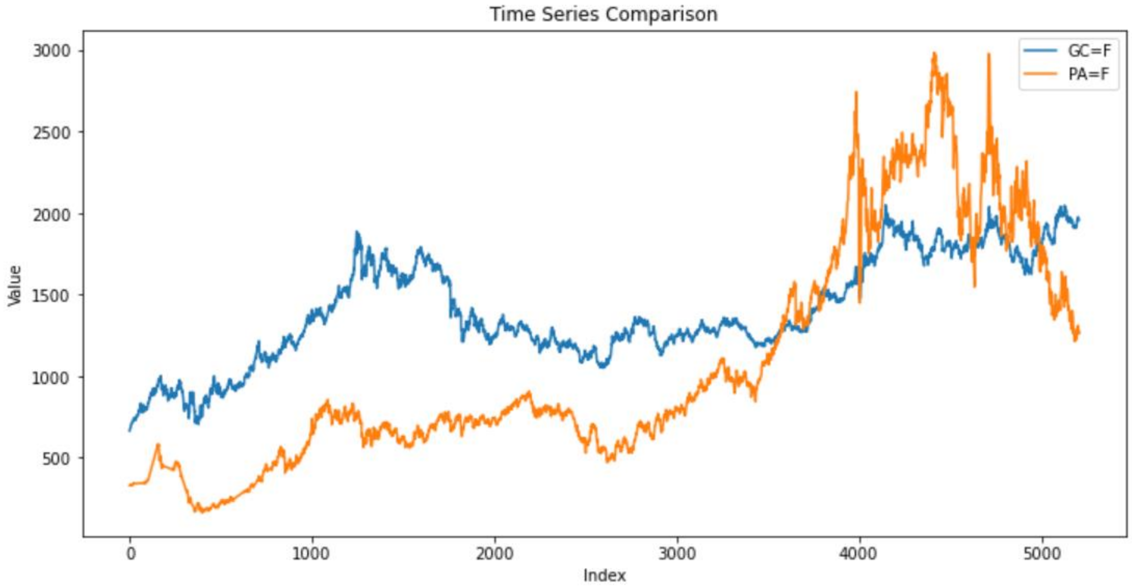
For this stage of the inquiry, I used the **plot_side_by_side** function that I wrote. This is a relatively simple algorithm that allows us to grasp the data from a visual perspective. It represents the data from two directions, one is the well-known plot that represents the values over time, and the other one is a frequency chart, displaying how many times each value is represented in the data, providing us a general idea about the distribution of the data.



4. Illustration, price on a time series and frequency distribution of gold prices, source: own findings

I have plotted all the data similarly, but to be page-efficient, I will only disclose the visual representation of the price of gold. This data provides us an easy-to-understand way to get a grasp of the scale of the data that I'm investigating. If any sort of data would be much different during my inquiries than the dimensions we've seen here, there should be ample suspicion that there is some technical error in the code yielding false results.

With this function, I'm also able to plot other variables, for example, here is the price of Palladium, which, according to the correlation table is the highest correlating item with the price of gold from among all the explanatory variables. In a visual representation, for the human spectator, it is much simpler to get a quick glance at the data, to confirm these findings in a "rule of thumb" manner.



5. Illustration, The price of gold vs. the price of palladium, source: own findings

3.6. Plotting Normalized Data

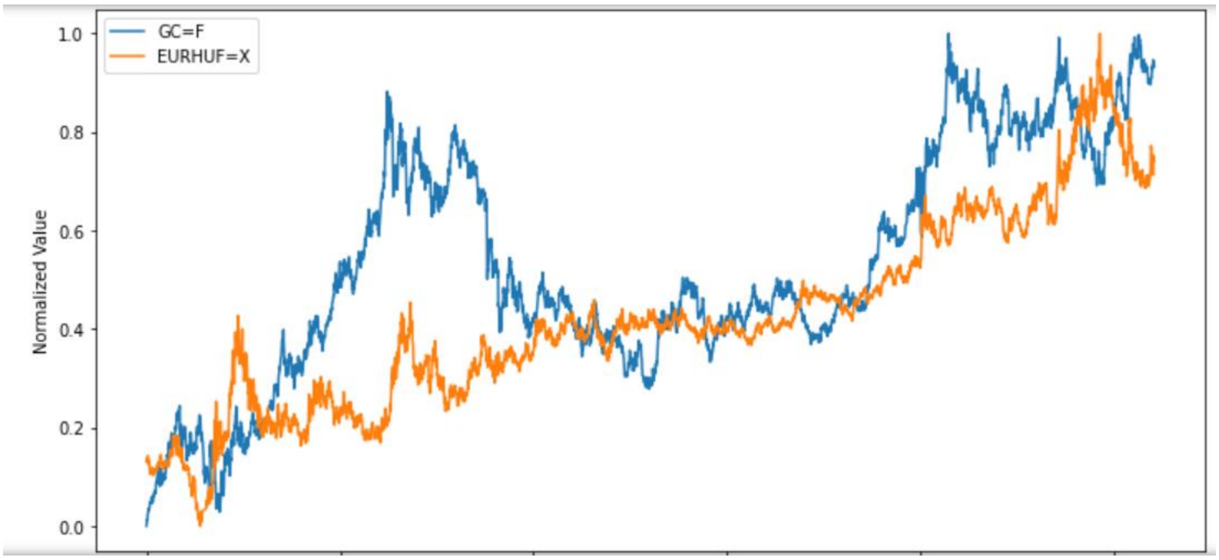
With explanatory variables, it that have the relatively same value—for example, like with palladium—it is easy to plot the variables and check if they are, in fact, correlating, but when it comes to data that has values in an entirely different dimension as the target variable, then the data has to be normalized on order for it to be visually compared. In data analysis and machine learning, normalization is a regular preprocessing step for scaling features to a similar range, not just for visual purposes. It ensures that all attributes play a fair role in the final result of techniques that rely on distances or gradients, such as clustering techniques or gradient descent in neural networks. The effectiveness of data normalization helps to minimize systematic variability in datasets. The variety of data types and their inherent variability have led to the development of multiple methods for normalizing data. This kind of variety is well represented in my explanatory variables as well.

For this purpose, I created the **plot_target_vs_others_normalized** function, that is based on the regular plot_target function, with the obvious difference that this function also normalizes the data. I used Min-Max Scaling, which is a common scaling method, where data is scaled to a fixed range, usually [0, 1]. The formula is:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

6. Formula, The formula for normalizing data, source:https://lifewithdata.com/2022/03/08/rescale-a-feature-with-minmaxscaler-in-sklearn/

I already defined a notable positive correlation of 0.73 between gold prices (GC=F) and the EURHUF exchange rate (EURHUF=X). This correlation suggests that as gold prices rise, the EURHUF tends to move in the same direction, and vice versa. While this correlation is intriguing, it's important to emphasize that correlation does not necessarily imply causation, making this plot an excellent example of why I'm conducting this study. As far as my academic background research went, I couldn't find any sort of literature that would provide grounds on why would the price of the Hungarian Forint would influence the price of gold or vice versa. This suggests that the correlation between the two variables is either completely accidental, or that they are effected by the same influences. Many external factors can influence both gold prices and exchange rates. This insight can be valuable for investors and analysts in making informed financial decisions, but it should be considered within the broader context of global economic and geopolitical factors because in and of itself this correlation doesn't mean that we should draw conclusions from it.



7. Illustration, The price of gold vs. the EURHUF, normalized, source: own findings

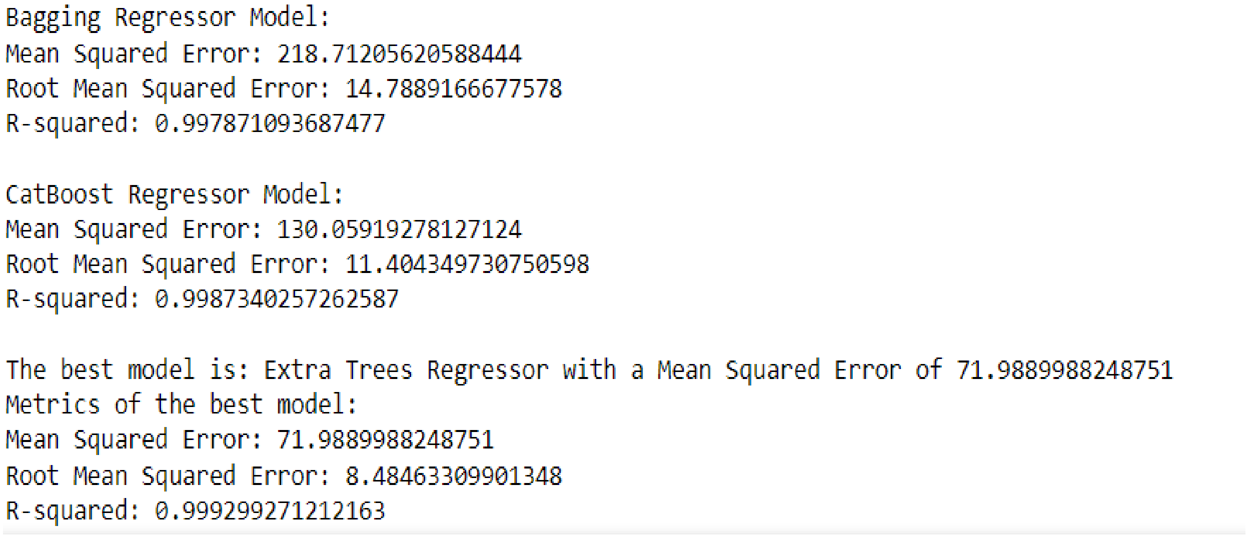
This kind of anomaly provides an interesting look into the complex entanglements in the field of world economy. To me, this high correlation with no logical background between these two variables provides more ground on why and how my potential findings can be used in the fields of economics.

3.7. Regression Model Testing

Creating an environment where I can empirically test the causal effects of the variables requires choosing a model out of the many available methods. For this reason, I wrote a function, called **find_best_model_regression**, that I have used outside this research as well. The reason I'm using such a casual manner regarding the selection of the model is that I want the final causality writing method as universal as possible. I must point out, that potentially there might be other models that would perform better in this exact scenario, but if we would ceteris paribus change up some variables, they might potentially do much worse. Keeping the method's general usability in mind, I came to the conclusion that the best way to approach this challenge would be to minimize the absolute minimal error in every single way. I have to address the fact that this path doesn't necessarily yield a global minimum for all the potentially MAE scores, as I haven't found any relevant literature, or evidence, that would support undoubtedly that this is the best approach, but testing and concluding every single model to a conclusion would exponentially increase the research time. For this reason, I will choose the models and methods that yield the lowest MAE at each step.

The function works in an uniformized way, where I only need to input the target variable and the explanatory variables in a format that has my data already split into a training and a testing set.

As the function traverses through each model, it trains the model on the designated training dataset and subsequently predicts the outcomes on the test set. Armed with these predictions, it calculates pivotal evaluation metrics such as the mean squared error (MSE), root mean squared error (RMSE), and the coefficient of determination (R^2). These metrics, especially MSE, become instrumental in the function's primary quest: identifying the best model. By maintaining a keen eye on the model yielding the lowest MSE, the function eventually heralds this model as the most optimal for the provided dataset. The difference between MAE and MSE is that MSE is the Mean Squared Error, while MAE is the Mean Absolute Error, but they do measure the same phenomenon. At the very end, the function provides the best model, with the lowest MSE after testing every single model type.



```
Bagging Regressor Model:
Mean Squared Error: 218.71205620588444
Root Mean Squared Error: 14.7889166677578
R-squared: 0.997871093687477

CatBoost Regressor Model:
Mean Squared Error: 130.05919278127124
Root Mean Squared Error: 11.404349730750598
R-squared: 0.9987340257262587

The best model is: Extra Trees Regressor with a Mean Squared Error of 71.9889988248751
Metrics of the best model:
Mean Squared Error: 71.9889988248751
Root Mean Squared Error: 8.48463309901348
R-squared: 0.999299271212163
```

8. Illustration, The output of the find_best_model_regression function, sample, source: own findings

The fundamental goal of regression models is to unravel and predict outcomes rooted in the interactions between independent and dependent variables (Seldon, 2021). The most classic example is the linear regression, which, as the name subtly suggests, focuses on capturing the linear interplay between the predictor and the response variable (Rosidi, 2022). The ubiquity of regression is hard to overlook. It underlies myriad applications, from the profound scientific discoveries that shape our understanding of the global economy. Because of this, there are many kinds of regressions tailored, to many needs and situations. To have a more complete picture regarding the potential models and to provide some theoretical background, I will touch up on all the models briefly and go into detail on the one that I chose in the end.

Another crucial aspect of the regression models is the fact that they have a way to bypass any kind of randomness, by setting their random state to a number (I usually used 1). This is important because a lot of these models have a random element, but by making sure that it is always random in the same predictable way allows us to recreate the same experiments to double-check the results and to change the components of the experiments ceteris paribus.

Linear Regression

Linear Regression is a foundational algorithm in statistics and machine learning designed to model the relationship between a dependent variable and one or more independent variables. By establishing a linear relationship represented as $y = \beta 0 + \beta 1x + \varepsilon$ it predicts the dependent variable based on the value of the independent variable. In this equation, y is the predicted value, x represents the independent variable, $\beta0$ is the y-intercept, $\beta1$ signifies the slope of the regression line, and ϵ is the error term. For instance, in a study between income (independent) and happiness (dependent), a value of 0.5 suggests that for every unit increase in income, happiness rises by 0.5 units. However, caution is advised when using linear regression for predictions outside observed data ranges, as extrapolation can yield inaccuracies. The method's applicability hinges on the assumption of a genuine linear relationship between involved variables. (www.britannica.com, n.d.)

Ridge Regression

Ridge Regression is a specialized technique within linear regression, aimed at addressing multicollinearity issues—where independent variables are highly correlated. By employing L2 regularization, Ridge Regression introduces a penalty based on the magnitude of coefficients, effectively reducing their variance. This process can lead to a slight increase in bias, but often results in models with lower mean squared error (MSE) and improved performance on new data. The regularization parameter, often denoted as λ, is pivotal in striking the right balance between bias and variance. The method's origin can be traced back to seminal works by Arthur Hoerl and Robert Kennard in 1970 and has since been widely adopted in fields ranging from econometrics to engineering (Zach, 2020).

Lasso Regression

Lasso Regression is a type of regression that tries to make models simpler by putting a penalty on their coefficients. The objective is to minimize this combined error and penalty term. By adding an L1 penalty term to the linear regression, Lasso encourages sparsity, causing certain coefficients to become precisely zero. This makes Lasso useful for feature selection, especially in high-dimensional datasets, as irrelevant or redundant predictors are automatically discarded . The formula for this method is LassoMSE(y, y_pred) = MSE(y, y_pred) + α ||θ||_1 where MSE(y, y_pred) denotes the mean squared error, α is a regularization parameter, and ||θ||_1 represents the L1 norm, essentially the sum of absolute values of coefficients (Machine Learning Compass, 2021).

ElasticNet Regression

It is a regularized linear regression technique that combines the penalties of two popular methods, Lasso and Ridge regression, thus trying to create a model that is up to the standards of both methods, aiming to capitalize on the strengths and mitigate the shortcomings. This model is technically the middle ground on the last two models. Lasso regression employs the L1 penalty, which can force some coefficients to be exactly zero, facilitating feature selection. On the other hand, Ridge regression uses the L2 penalty, which shrinks the coefficients but doesn't set any to zero. ElasticNet incorporates both penalties, allowing it to inherit the feature selection properties of Lasso while retaining Ridge's coefficient shrinking ability.

Mathematically, ElasticNet's objective function is defined as:

ElasticNetLoss(y, y_pred) = MSE(y, y_pred) + α * l1_penalty + (1 - α) * l2_penalty

MSE(y, y_pred) is the mean squared error.

α is a hyperparameter in the range [0, 1] that determines the mix of the penalties. An α of 1 leans entirely towards Lasso, and an α of 0 leans entirely towards Ridge.

l1_penalty and l2_penalty represent the L1 and L2 regularization penalties, respectively1 (Brownlee, 2020).

ElasticNet has a unique ability to manage multicollinearity. When predictors are highly correlated, Lasso might randomly pick one of them as significant, while Ridge might include all of them. ElasticNet can strike a balance, potentially selecting a group of correlated predictors together. (Corporate Finance Institute, n.d.)

Bayesian Ridge Regression Model

Bayesian Ridge Regression integrates the principles of Ridge Regression with Bayesian modeling. This method of predicting future outcomes uses probabilistic ideas called Ridge Regression, which is similar to Bayesian thinking. Ridge regression can be thought of as the Maximum A Posteriori estimation under Gaussian priors for model parameters when viewed through the lens of Bayesian Inference (Rothman, 2020). The central concept in Bayesian Ridge Regression is the infusion of prior beliefs about parameters, which are then updated with observed data to form posterior beliefs. Specifically, it employs Bayesian techniques to estimate a probability distribution for the model's coefficients, rather than determining a single best estimate of these coefficients. In practical terms, Bayesian Ridge Regression often results in similar coefficient estimates as traditional Ridge Regression. However, Bayesian methods provide a full distribution (posterior) over the model parameters, which can be advantageous in understanding the uncertainty associated with these parameter estimates. One notable aspect is the incorporation of prior knowledge (prior distributions) about the parameters. When new data becomes available, this prior is combined with the likelihood of observing the new data given the parameters, leading to an updated belief or posterior distribution of the parameters. Furthermore, Bayesian Ridge Regression inherently manages the regularization hyper-parameter tuning, providing a balance between fitting the data and maintaining a model that isn't overly complex. This balance is crucial in preventing overfitting, especially when working with high-dimensional datasets (Rashid, Altaf and Aslam, 2020).

In essence, Bayesian Ridge Regression offers a nuanced and probabilistic perspective on Ridge Regression, making it a valuable tool when uncertainties about model parameters need to be quantified (Shu Yu Tew, Boley and Schmidt, 2023).

Random Forest Regressor Model

Random Forest Regressor is an ensemble learning method that builds upon decision trees, aggregating their results to produce a more generalized and accurate model. It is particularly adept at handling large datasets with high dimensionality, capable of addressing thousands of input variables without variable deletion (Couronné, Probst and Boulesteix, 2018). An RFR model operates by merging multiple uncorrelated trees. It utilizes both boosting and aggregating techniques, ensuring generalizations and optimized accuracy without falling into overfitting or underfitting pitfalls (El Mrabet et al., 2022). In terms of a formula, Random Forest doesn't have a single formula like linear regression does, given its complexity and ensemble nature.

Gradient Boosting Regressor

The Gradient Boosting Regressor stands out as a noteworthy ensemble machine learning technique, enhancing its predictions by systematically reducing errors. The origin of the technique can be traced back to the realization that boosting techniques can be interpreted as optimizations over specific loss functions. This novel perspective resulted in an algorithm that is sufficiently adaptable to accommodate diverse loss functions and coexist harmoniously with a diverse range of weak learner models, although decision trees remain the preferred choice.

The Gradient Boosting Regressor dives into its fundamental mechanisms with a basic model, often derived from the inherent patterns in the data. As the algorithm goes along, it focuses on the residuals, or differences between the actual values and the predictions of the previous model. Subsequently, a fresh model is trained on these discrepancies. Gradient Boosting zeroes in on the collective error of the preceding ensemble, unlike its predecessor, Adaboost, which adjusts weights for each prediction. The objective is to minimize this error by incorporating a new model that synchronizes with the negative gradient of the loss function. This methodical approach is maintained for a specified number of iterations or until the discrepancies diminish to a satisfactory level. One distinctive feature of this technique is the weighting of each model that is introduced into the ensemble, thereby ensuring that these weights are precisely adjusted to further reduce the loss function. An invaluable asset in the realm of machine learning is the algorithms' compatibility with any differentiable loss function and its seamless integration with various weak learner models. (Brown, 2018)

XBoost Regressor

XGBoost, or eXtreme Gradient Boosting, is an ensemble learning algorithm that builds upon the gradient boosting framework. It is an efficient and scalable method, often hailed for its performance and speed. While it has been widely adopted due to its many benefits, it's important to note that its classification outcomes in imbalanced datasets might not always be optimal (Zhang, Jia and Shang, 2022).

It operates on decision trees, models that construct a graph examining input under various "if" conditions. The path of "if" conditions in the tree affects the eventual prediction. XGBoost progressively adds more conditions to the decision tree to enhance the model. The algorithm considers the leaves of the current decision tree and evaluates if turning a leaf into a new "if" condition would be beneficial to the model. This benefit is determined using the gradient of the loss function. The loss function consists of a scoring function that measures algorithm performance (source:https://www.springboard.com/blog/data-science/xgboost-explainer/)

In essence, XGBoost is a powerful gradient boosting algorithm that offers scalability and efficiency. It's versatility allows it to be applied across various domains, from technical to medical, and it continues to be a preferred choice among machine learning practitioners for its proven track record in performance.

LightGBM Regressor

LightGBM, short for Light Gradient Boosting Machine, is an advanced gradient boosting algorithm engineered for scalability and efficiency. At its core, LightGBM is an ensemble of decision trees, primarily applied for classification and regression predictions. The gradient boosting (GB) framework, of which LightGBM is a variant, is another ensemble model. The principle behind GB involves the aggregation of multiple weak learners to establish a robust model. As a tree-based ensemble algorithm, GB excels in interpreting categorical features. The primary distinction between GB and RF is in their tree construction approach: RF independently constructs trees and averages the results for the final prediction, whereas GB sequentially builds trees through boosting iterations. During each iteration, the current tree is formulated based on the prior one, where the disparity between the predicted and true values is gauged and set as the prediction objective for the current tree. This difference diminishes progressively after numerous iterations, and the results from all trees are cumulated for the final prediction. LightGBM was developed by Microsoft, and has showcased its prowess in managing vast structured datasets, achieving high training speeds. (Yan et al., 2021)

Support Vector Regressor

Support Vector Regression (SVR) is an advanced machine learning technique that builds upon the principles of Support Vector Machines. This model aims to establish a regression function that correlates input predictor variables with observed response values. SVR is very good at balancing how complicated a model is and how likely it is to make mistakes. It excels in handling large-scale data, especially when it comes to handling complex ones. At its core, SVR introduces a region, commonly termed as a "tube", around the function it seeks to optimize. The primary objective is to identify the tube that demonstrates the highest degree of accuracy in resembling the continuous-valued function, while simultaneously minimizing the prediction error, which refers to the discrepancy between the predicted and actual class labels. SVMs, which are based on Vapnik-Chervonenkis (VC) theory, can be used to generalize on data that hasn't been seen before (Zhang, 2017). SVR, in essence, leverages the principles of SVMs but adapts them for regression problems. It is renowned for its ability to balance model intricacy against prediction error, and is particularly adept at handling high-dimensional datasets (scikit learn, 2018), (Sharp 💻, 2020).

K-Nearest Regressor

The k-nearest neighbor (KNN) algorithm is a supervised machine learning algorithm predominantly utilized for both classification and regression tasks. At its core, KNN predicts the classification or regression of unlabeled data by considering the features and labels of the training data. It classifies datasets using a training model similar to the testing query, considering the k nearest training data points (neighbors) that are closest to the query it evaluates. The algorithm then employs a majority voting rule basically to finalize the classification (Uddin et al., 2022). One of the KNN algorithm's primary strengths is its adaptability and flexibility to handle various data types and adjust to irregular feature spaces.

A common challenge with traditional KNN, especially when dealing with heterogeneous data, is that its performance can be hindered by the often arbitrary choice of similarity metric. To address this, regression methods can be used to derive a similarity metric as a weighted combination of base similarity measures. This refined metric better identifies neighbors likely to belong to the same class as the target data. Additionally, a novel voting scheme can generate confidence scores that estimate prediction accuracy. This approach can be extended to multi-way classification scenarios.

When examining regression-based KNN in the context of diverse data, it becomes apparent that combining diverse data sources is crucial for predictive modeling. There are multiple prototypes for a class, which makes the flexibility of KNN particularly advantageous for classification problems. The classical KNN method starts by designing a set of numerical features to describe each data point and selecting a metric (like the Euclidean distance) to gauge the similarity of data points based on all features. The phenomena that I'm investigating is of course not a classification problem, but there are regression libraries created based on the KNN methodology. For a target data point, the method identifies its k-closest points in the training samples based on this similarity metric and classifies it via a majority vote of its neighbors. This approach's main drawback is its sensitivity to the chosen similarity metric, particularly when handling heterogeneous datasets. A modification to this paradigm involves designing several base similarity measures between data pairs, one for each data source, and estimating the likelihood of a pair being in the same class given these base similarities. This likelihood estimation in this case is viewed as a classical regression problem, with the resulting function serving as the global similarity measure (Yao and Ruzzo, 2006).

Multi Layer Perceptron

This regressor model has touched on the subject of neural networks that I will detail in a later part of my paper, but as in this case, I used it in the context of regression I will inspect it in some detail. Non-linear methods for functional data analysis have evolved, such as neural network models, which include multilayer perceptrons. In the realm of functional regression, one approach combines the efficiency of dimension reduction techniques using smoothing penalization with the adaptability of a neural network. This union harnesses the neural network's potential to offer highly non-linear solutions, especially when the number of predictors is too vast for traditional non-parametric methods like kernel smoothing.

The Multi-layer Perceptron (MLP) Regressor is an extension of the basic perceptron that introduces hidden layers between input and output layers. Since functional variables are frequently encountered in practical applications, the MLP regressor aims to utilize functional regression. For instance, predicting the fat content of meat pieces from their near-infrared absorbance spectrum exemplifies a regression problem where the regressors are curves (ar5iv, n.d.). In essence, MLP Regressor excels in capturing intricate data relationships, making it adept at modeling complex patterns, especially in functional data. It incorporates the strengths of neural networks and dimension reduction techniques, for regression tasks (Scikit-learn.org, 2010).

AdaBoost Regressor

AdaBoost, an acronym for Adaptive Boosting, stands as one of the pioneering techniques in the realm of boosting algorithms. It's foundational to later algorithmic advancements like gradient boosting and XGBoost. The essence of boosting is to amalgamate multiple weaker models to craft a singular, more effective model. This assembly is not achieved in parallel but sequentially, with each iteration emphasizing the discrepancies of its predecessor.

In the context of regression, AdaBoost operates in a systematic manner. Initially, it picks a random subset of the training data. As the model progresses, the data for the upcoming training is chosen based on the accuracy of the previous prediction. AdaBoost is judicious in its approach; it increases the weightage of the observations that were inaccurately classified. This tactic ensures a higher likelihood of correct classification in the succeeding iterations. Post each iteration, the trained classifier's accuracy determines its weight. The more accurate a classifier, the heavier it weighs in the ensemble. This iterative process either continues until the training data is impeccably fit without any misclassifications or until a pre-defined threshold of estimators is achieved. AdaBoost's training mechanism lies in its preference for weak classifiers, with decision stumps being a popular choice. Decision stumps are essentially decision trees restricted to just a single level. The formula that captures the essence of the misclassification rate during training is:

$\text{Error} = (\Sigma\ w(i) * \text{terror}(i)) / \Sigma\ w$

Here, w signifies the weight for the specific sample i, and terror represents the prediction error for the said sample, taking the value of 1 for misclassifications and 0 for correct classifications. AdaBoost's effectiveness resides in its dynamic adaptability. By focusing on errors made during early iterations and adjusting weights accordingly, it excels at addressing intricate scenarios within datasets, making it a formidable tool in regression tasks (scikit-learn.org, n.d.) (Prabhakaran, 2023).

Extra Trees Regressor

As I mentioned before, this model had the lowest error rate of all the models that I have tested, so I will go a bit more into details on how this model works.

The Extra Trees Regressor (ETR) model is an ensemble learning technique that builds upon the foundation of the decision tree algorithms, with a unique construction methodology that distinguishes it from other models such as Random Forests (RF). It is designed to improve prediction accuracy and control over-fitting by fitting multiple randomized decision trees on various subsamples of the dataset and using averaging. It is considered a developed approach derived from the RF model (Geurts, Ernst and Wehenkel, 2006).

The ETR model constructs a multitude of unpruned decision trees, or regression trees, which are developed using the conventional top-down technique (Rokach, 2016). The core of its methodology lies in the creation of these trees, which are cultivated using the entire learning samples rather than a bootstrapped sample, aiming to minimize bias. This is a stark contrast to the RF model, which utilizes a bootstrapping process to create a set of decision trees, each growing from a random training dataset sample. The RF model then employs bagging, creating an ensemble of decision trees combined and averaged to form a "forest" (Oshiro, Perez and Baranauskas, 2012).

The splitting process in ETR is critical and is influenced by two parameters:

K, referring to the number of features randomly picked up in the node, and n_{min}, indicating the minimum sample size expected to separate nodes. These parameters play a vital role in improving the precision of the model and reducing the likelihood of overfitting. The model's hyperparameters, such as the number of trees in the forest (n_estimators), the maximum depth of the tree (max_depth), the minimum number of samples required to split an internal node (min_samples_split), and the number of features to consider when looking for the best split (max_features), are adjustable and critical for the model's performance (scikit-learn, n.d.).

The ETR's approach to node splitting is also distinctive. A node is split if this action induces a decrease in the impurity that is greater than or equal to a defined value (min_impurity_decrease) . The equation representing the weighted impurity decrease is:

N_t / N * (impurity - N_t_R / N_t * right_impurity)

This quantification of impurity decrease is instrumental in determining the quality of the split and is fundamental to the ETR's algorithmic process. The performance of ETR models is typically evaluated using various statistical metrics, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2). These metrics serve to compare the predicted output with the actual values, providing a quantitative measure of the model's accuracy and efficiency. In practice, the ETR model has found applications in diverse sectors, including hydraulic prediction models. Due to its fast learning and universal approximation capacity, it is able to map relationships between variables effectively and efficiently. The practical implementation of the ETR model in Python, for instance, can be facilitated through the sklearn.ensemble.ExtraTreesRegressor class provided by the scikit-learn library, which encapsulates the model's functionality with user-defined parameters.

In summary, the Extra Trees Regressor model is a powerful machine learning algorithm that leverages the strengths of ensemble learning and randomized decision trees to provide robust predictive capabilities. Its randomization of features and samples, coupled with the averaging of unpruned trees, helps in reducing variance and bias, making it a valuable tool for regression tasks where prediction accuracy is paramount (scikit-learn, n.d.).

Stochastic Gradient Descent Regressor

Essentially, Stochastic Gradient Descent (SGD) is an iterative technique that is specifically designed to enhance objective functions with certain smoothness attributes, such as being differentiable or sub-differentiable. SGD approximates the gradient with a random subset of the data, unlike traditional gradient descent, which uses the entire dataset to calculate the gradient. This stochastic estimation of the gradient is a good technique for solving high-dimensional optimization problems. By using the actual gradient instead of its estimate, SGD can iterate faster, but its convergence rate is a little slower. (DeepAI, 2019)

At its heart, SGD aims to identify the global minimum by adjusting the machine learning model after every training instance. Instead of reducing the error over the complete dataset, SGD minimizes this error by estimating the gradient for a randomly chosen batch, which can even be a single training sample. This random selection is typically achieved by shuffling the dataset and sequentially processing through batches. However, SGD's path to the global minimum isn't always straightforward. It might stumble upon local minima, where any movement direction results in an increase. In such scenarios, the algorithm might decide to move out of the local minimum, potentially increasing the model's error, to continue its journey towards the global minimum. In summary, while SGD presents a computationally efficient optimization route, its convergence to a global minimum isn't always guaranteed. (scikit-learn, n.d.)

Bagging Regressor

The Bagging Regressor model is also an ensemble learning method that has emerged from a classification model. A prominent ensemble learning algorithm is the Bagging Regressor. Bagging, an abbreviation for Bootstrap Aggregating, is a meta-estimator that assembles base regressors on random subsets of the primary dataset, subsequently amalgamating their individual predictions to yield a conclusive prediction. It works by randomly generating subsets, with replacements, from the original dataset, and applying a classifier (for classification tasks) or a regression (for regression tasks) to every subset (Biswal, 2021).

Bootstrapping is the process of creating random samples from a dataset with replacement, enabling the estimation of a population parameter. The Bagging methodology involves extracting multiple bootstrap samples from the primary training dataset and constructing a decision tree for each sample. Predictions from these trees are then integrated into one model. For regression trees, predictions are averaged, while for classification trees, the most frequent prediction is selected. In models with little variation, bagging might make the model less effective. Furthermore, it's crucial to acknowledge that while Bagging bolsters predictive accuracy, it compromises on the interpretability inherent to individual decision trees (scikit-learn.org, n.d.).

CatBoost Regressor

CatBoost, an acronym for Categorical Boosting, is a gradient boosting algorithm explicitly designed to handle datasets with categorical features. Developed by the Russian company Yandex, CatBoost has garnered attention for its efficacy in both classification and regression tasks (Thiesen, 2021).

3.8. Forward Feature Selection

It is a critical step of building every machine learning model, to select the explanatory variables in a manner that yields the most precise final model. As for the universally applicable aspect of my research, I decided to use the Forward Feature Selection technique. The basis of this method is to create a linear regression model with every one of the explanatory features that we have selected. These models will have relatively high mean absolute errors, of course, so the lowest one is chosen. Subsequently, we start the process again, with one fixed variable, and create regression models with two explanatory variables (in this exact case after I found the first explanatory variable, I had to make 85 more, since I had 86 explanatory variables altogether). The process continues until the MAE score improves under a certain amount (in my method, I decided to run the algorithm until the model improved by 0,01). I must note it that this method is unorthodox in most cases and most scenarios for what we build machine learning models in practice have their own kind of methodology, however, the purpose of my research is not for me to create a model that can predict the price of gold in a precise manner, but a unified way I can possibly prove the presence of causality. I found that approaching my research from this pragmatic standpoint of view is the most optimal, as it fits into the empirical testing methods that I'm doing my research about, and also because the most effective way to unify this method in these early stages is to find the local minimum of errors in all cases (Macedo et al., 2019) (Venkatesh and Anuradha, 2019).

For these purposes, I wrote the **forward_feature_selection** algorithm. The algorithm is implemented iteratively, starting with an empty set of features and testing the performance of the model with each individual feature. Once the feature that maximizes the model's performance is identified, it's added to the set. This process continues, adding one feature at a time until no more features improve the model's performance. The algorithm, in general, aims to automatically identify attributes in data that are most pertinent to the predictive modeling task at hand. The primary goal is to select a subset of these features that optimally represent the underlying data and contribute to model accuracy. This process can significantly enhance a model's efficiency by eliminating irrelevant, redundant, or unneeded attributes. Wrapper methods, a category of feature selection techniques, view the feature selection process as a search problem (Analytics Vidhya, 2020). In this context, different feature combinations are generated, evaluated, and compared. Forward feature selection is a heuristic utilized within wrapper methods, which involves sequential passes to add or remove features based on their contribution to model accuracy.

3.8.1. Creating The Final Data Frames

After choosing the explanatory variables via the Forward Feature Selection, I will use the “correlation” function, to make sure that the explanatory variables won’t distort the data because of cross-correlation. I will also introduce the different kinds of inflation data among the explanatory variables, and I will repeat the feature engineering and the correlation filtering, to make sure that there is no real discrepancy among them.

I will write here the name of the columns I used for the data frames based on my findings. However, I will not delve into further economic background research on the relating economic factors, as the main purpose of the research is to find a unified method to possibly present causality. Doing any sort of theoretical background research at this point would interfere with the goal of automatization, and thus I will only lean on the data at hand. The final features for the data frames were as follows:

df_f (this data frame is there as a control data, it does not have any kind of inflation in it.: [['PA=F', 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', '^MERV', 'CT=F', 'EURCAD=X', '^KLSE', 'LBS=F', 'RB=F', 'CNY=X', 'EURGBP=X', 'ZL=F', '^VIX']]

df_HCPI: [['PA=F', 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', 'CT=F', 'EURCAD=X', '^KLSE', 'THB=X', 'SB=F', 'ZT=F', '000001.SS', 'ZS=F', 'USDJPY=X', 'HCPI']]

df_ECPI: [['PA=F', 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', 'CT=F','EURCAD=X', '^KLSE', 'CNY=X', 'USDJPY=X', 'EURGBP=X', 'ECPI', '^VIX']]

df_FCPI: [['PA=F', 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', 'CT=F', 'THB=X', '^KLSE', 'USDJPY=X', 'KC=F', 'ZT=F', 'HO=F', 'SGD=X','NZDUSD=X', 'FCPI']]

df_CCPI: [['PA=F', 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', 'CT=F', 'EURCAD=X', '^KLSE', 'THB=X', 'CC=F', 'HO=F', 'SGD=X', 'USDJPY=X', 'CNY=X', 'EURJPY=X', 'LE=F', 'KC=F','CCPI']]

df_PPI: [['PA=F' 'SI=F', 'ZN=F', 'ZR=F', '^IPSA', 'CT=F', 'EURCAD=X', '^KLSE', 'ZL=F', 'GBPJPY=X', 'RB=F','PPI']]

(The interpretation of these abbreviations can be found in the appendix)

3.9. Implementing the Regression Model

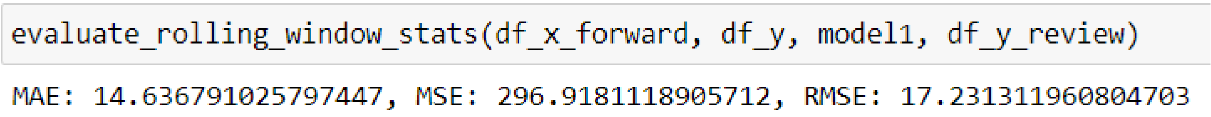
My theory is that to prove empirical utility, I not only need to represent a model in a machine learning way, but I also need to use it for forecasting purposes, at least in a simulated environment. I am making this argument because the aim of my research is to exceed the province of purely mathematical reasoning, and introduce pragmatical measuring in the most efficient way. The problem with the regression models in this particular case is that they are generally not made for forecasting. Regression is mostly used for determining the value of a target variable derived from the explanatory variables, not to figure out how much the price of gold will be tomorrow. To overcome this, I'm going to use one of the established methods to use regression for forecasting. There are more methods for achieving forecasting with regression, so I will test the most relevant of these and find the one with the lowest MAE score.

3.9.1. Rolling Windows

Rolling windows, in the context of time series analysis, serve as a technique to generate features by calculating statistical values based on past data points. Given a time series, the idea is to use a 'window' of fixed size and slide it across the data. For each position of the window, certain statistics like the mean, sum, min, or max value of the data points within the window are computed, creating a new feature. This method effectively transforms a single time series into multiple ones, each ending one (or n) time step later than the previous (www.mathworks.com, n.d.). When examining a time series, values are often influenced by their immediate predecessors. For instance, if there's significant traffic in a location at one time, it's likely that there will still be traffic a minute later. This kind of relationship is called autocorrelation.

After choosing a window size, I take the average of the values within that window and use it as a feature. The basic application allows us to consider other statistics, such as the sum, min, and max of the windowed values. To make recent data more important in a time series, you can use a weighted average. This means that the more recent data gets more weight.

By calculating various statistics over a fixed-size window and sliding them across the data, one can capture the underlying patterns and relationships inherent in the series. This will enhance the forecasting model's accuracy and reliability.



```
evaluate_rolling_window_stats(df_x_forward, df_y, model1, df_y_review)
```

```
MAE: 14.636791025797447, MSE: 296.9181118905712, RMSE: 17.231311960804703
```

8. Illustration: The Results for The Rolling Window Method, source: Own findings

3.9.2. Recursive Forecasting

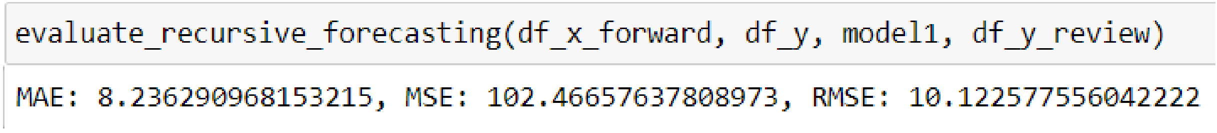
I will delve into this model a bit more in detail, as it had the best scores and I moved forward with this one. Tha main idea is to forecast one day at a time, then include the forecasted value as part of the input data for forecasting the next day, and so on for 30 days.

Recursive forecasting, at its essence, is a multi-step forecasting process where each subsequent prediction hinges upon its predecessor. The model iterates, making predictions in batches, until all desired future timestamps have been forecasted. This method's essence lies in its iterative nature, where each new prediction helps inform and refine the next (Cerliani, 2022).

To test this method I created the **create_lagged_features** Function which creates a set of lagged features for a given DataFrame (df_x). For a specified number of lags, the function shifts the original data, producing a set of new columns that represent past observations. The **recursive_forecasting** function serves as the hearth of the recursive forecasting approach, this function first trains the model on the initial training set. Following this, it iteratively predicts one step ahead, appending each prediction to the training set, and retraining the model for subsequent forecasts. Afterward I evaluate the results with the **evaluate_recursive_forecasting** function, which is basically the same for all the other models.

A distinct aspect of recursive forecasting is the generation of lagged features from the target series, followed by fitting a machine learning model to these features. As the model ventures to forecast further into the future, it uses previous predictions to create new lagged features, ensuring that the forecasting process remains informed by the most recent data points . (Staff, 2019)

In summary, recursive forecasting offers an intuitive yet powerful methodology for time series analysis, capitalizing on historical data to make informed future predictions. By iteratively using past predictions to refine future forecasts, this approach ensures that models remain attuned to the ever-evolving patterns in time series data.

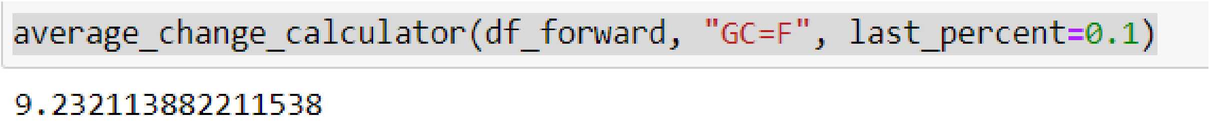


```
evaluate_recursive_forecasting(df_x_forward, df_y, model1, df_y_review)
```

```
MAE: 8.236290968153215, MSE: 102.46657637808973, RMSE: 10.122577556042222
```

9. Illustration: The Results for The Recursive Forecast Method, source: Own findings

This method yielded the best result of a mean absolute error of 8.2363. As I measured the average movement of the gold prices which was 9.2321 in the last 10% (see the code snippet below for reference), we could draw the conclusion that the model is starting to get to the state where it could be potentially useful, although much greater models than this could be built.



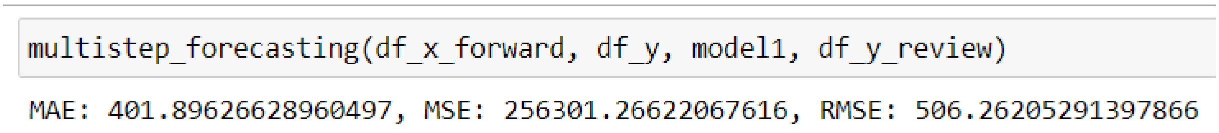
```
average_change_calculator(df_forward, "GC=F", last_percent=0.1)
```

```
9.232113882211538
```

10. Illustration: The average daily change of the price of gold in the last 10% of the data, source: Own findings

3.9.3. Multi-step Forecasting

Since the results it yielded appear to be due to some technical error, we will not dwell on this method for an extended period of time. In the field of time series forecasting, multi-step forecasting stands as an essential approach wherein the aim is to predict multiple future values simultaneously. The defining characteristic of this method is the modification of the target variable to encapsulate a vector of future values, effectively training the model to make predictions for several time steps ahead in one go.



```
multistep_forecasting(df_x_forward, df_y, model1, df_y_review)
```

```
MAE: 401.89626628960497, MSE: 256301.26622067616, RMSE: 506.26205291397866
```

11. Illustration: The Results for the Multi Step Forecasting Method, source: Own findings

As anyone can see from the results, this MAE is so outrageously high, that even if there is some technical error (which I haven't managed to find after numerous attempts to search forums, debuggers and other methods) this is probably not the best route to create a universally working method.

3.10. Forecasting

To summarize the current state of the research; I chose my explanatory variables, with forward selection, I filtered them for cross correlation, I chose a regressor method, and now we have a way to use this regressor method for forecasting purposes. Based on all the previous work, I wrote the function that I shall use to make forecasting with, and then I will proceed to the final testing ground of which one of the inflation indexes provides the best result. Afterwards, I will explain how the code works, which can be found in the appendix in more detail.

The function **forecast_with_etr_daily_check** is designed for forecasting using the Extra Trees Regressor (ETR) model. It splits a given dataset into training and testing sets, trains the model on the training data, and then makes successive daily forecasts. The function also provides daily metrics for the forecasted values, including the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Function Parameters:

df_input: The input DataFrame containing the features and target variable.

target_variable: The name of the column in the DataFrame that represents the target variable we're forecasting (in this case it is the GC=F column).

test size: The fraction of data to be used as the test set. Default is 30%.

forecast_days: The number of days for which we want to generate forecasts. Default is 5 days.

model: An optional machine learning model. If not provided, the function will use a fresh instance of the Extra Trees Regressor.

Code Breakdown:

Initializing the model: A fresh instance of the Extra Trees Regressor is created using default parameters, ensuring that the model starts anew for each data input.

Data Preparation: A deep copy of the input DataFrame is made to prevent accidental modifications to the original data. Features and the target variable are separated into X and y data frames, where y is the target variable. The data is split into training and testing sets based on the provided test_size.

Forecast Initialization: An empty list forecasts is initialized to store forecasted values. A dictionary metrics is initialized to store MAE, MSE, and RMSE values for each forecast day.

Iterative Forecasting:

For each day in the forecast_days, the function does the following:

a. Fits the model using the training data up to that day.

b. Makes a prediction for the next day using the most recent data.

c. If the current day is within the range of the test set, the forecasted value is compared with the actual value, and error metrics (MAE, MSE, RMSE) are computed and stored.

d. The last_known_data is updated for the next forecast.

Metrics Reporting:

The function prints the daily forecast metrics in a tabular format.

Average values for MAE, MSE, and RMSE are computed and displayed.

Return Values:

The trained model, forecasts, and metrics are returned.

To summarize, The **forecast_with_etr_daily_check** method is a way to use the Extra Trees Regressor technique to forecast. This approach ensures that it remains abreast of the most recent information by consistently acquiring knowledge and making forecasts. It also provides daily performance indicators, allowing users to gauge daily forecast accuracy. This approach presents a cost-effective approach for time-series forecasting in fields where daily insights and evaluations are imperative.

3.11. Neural Network Creator

The other method, next to regression, is the neural network models. They have been used for forecasting purposes for quite some time now, and there are interesting developments in this field every day (M. Al-Maqaleh, A. Al-Mansoub and N. Al-Badani, 2016). There are countless academic papers on Neural Networks and what they are potentially capable of, and looking over just half of the relevant academic literature would easily make a 100-page-long paper in and of itself (Kristjanpoller and Minutolo, 2015). For this reason, I will concentrate the most pertaining findings in this field. I would like to refer to the 3.1. paragraph of this paper, where I made a remark on how randomness can be bypassed with the regression models, making them ideal for experimental purposes. Neural Networks, on the other hand, are somewhat based on the notion of randomness, thus making them ideal for finding patterns that they can build upon

(Scardapane and Wang, 2017). This makes them suboptimal for my research purposes, nevertheless, I aimed to try to create a universally working way for the purposes of proving causality through the results of neural network models, for that would make an excellent argument for the empirical aspect of my research.

For this purpose, I created the **neuralnetworkcreator_daily** and the **run_multiple_times** function. The main thought is to test every kind of relevant neural network and use the one with the lowest absolute error. The provided code leverages the architectures of RNN, LSTM, GRU, CNN-LSTM, ANN, and BPNN.

Recurrent Neural Networks (RNNs): RNNs are designed to process sequential data by considering the previous state of the sequence. This "memory" of prior states enables them to handle temporal dynamic behavior. A key characteristic is the cyclic connections, which allow RNNs to generalize to sequences of varying lengths (Donges, 2019).

Long Short-Term Memory (LSTM): A specialized form of RNN, LSTM is distinguished by its feedback connections and ability to process entire sequences of data. With its foundation based on short-term memory processes, it seeks to create enduring, long-term memories (Staudemeyer, Ralf C and Morris, 2019).

Gated Recurrent Units (GRUs): Another variant of RNN, GRUs streamline the gating mechanism seen in LSTMs but without an output gate. This design solves the problem of vanishing gradients in regular RNNs. The GRU captures information from past time steps relevant to the current step (Analytics Vidhya, 2022).

CNN-LSTM: An amalgamation of Convolutional Neural Networks (CNNs) and LSTM, this architecture utilizes CNN layers for input data feature extraction coupled with LSTMs for sequence prediction, particularly for visual time series prediction tasks (Agga et al., 2022).

Artificial Neural Networks (ANNs): As the foundational neural network model, ANNs draw inspiration from the human brain, replicating how biological neurons signal to one another. It has a long history dating back to 1958 by F Rosenblatt (Rosenblatt, 1958). Although this is an umbrella term for all Artificial Neural Networks, Keras has a model called ANN based on perceptrons. Since I couldn’t find the exact documentation of this model, I will proceed with caution, but I don’t see the harm at this point to using an algorithm that I don’t know the exact working mechanism of. If this model turns out to be by far the most efficient model, we will deal more in depth with how this package works exactly.

Backpropagation Neural Networks (BPNNs): Backpropagation is a supervised learning algorithm for ANNs that uses gradient descent. The essence of neural net training, it adjusts weights by calculating the gradient of the error function (Wang, Zeng and Chen, 2015).

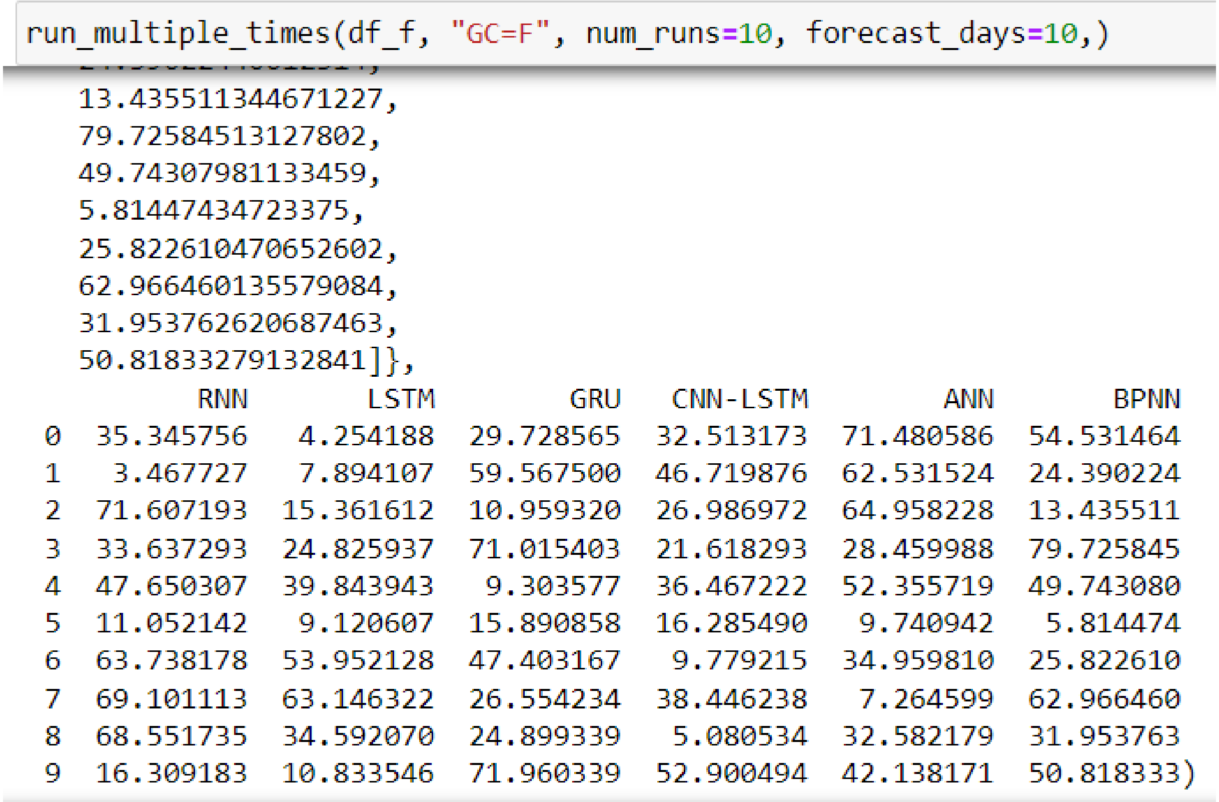
The primary function, **neuralnetworkcreator_daily**, initiates by preparing the dataset for training and testing. It checks for NaN and infinite values, then segments the data into features (X) and targets (y). Data normalization is achieved through the MinMaxScaler, ensuring values lie between 0 and 1. The dataset is then divided into training and testing subsets based on the forecast days specified.

The actual neural network model is initialized as Sequential, a linear stack of layers. Depending on the chosen model type, appropriate layers are added. Once the model is compiled using the 'adam' optimizer and mean squared error as the loss function, it is trained using the training dataset. After training, the model forecasts for the specified days, comparing the predicted and actual values. Mean Absolute Error (MAE) and Mean Squared Error (MSE) are computed for each forecast day, providing error metrics for model performance evaluation.

The secondary function, **run_multiple_times**, runs the primary function multiple times for each specified model type. It returns the Mean Absolute Errors for each run, giving a holistic view of the model's performance across multiple iterations.

In conclusion, the provided code is a representation of various neural network architectures tailored for forecasting. Its modular design enables easy integration and testing of different neural network models, providing a robust framework for predictive analytics.

The result of this analysis is as follows:



```
run_multiple_times(df_f, "GC=F", num_runs=10, forecast_days=10,)
```

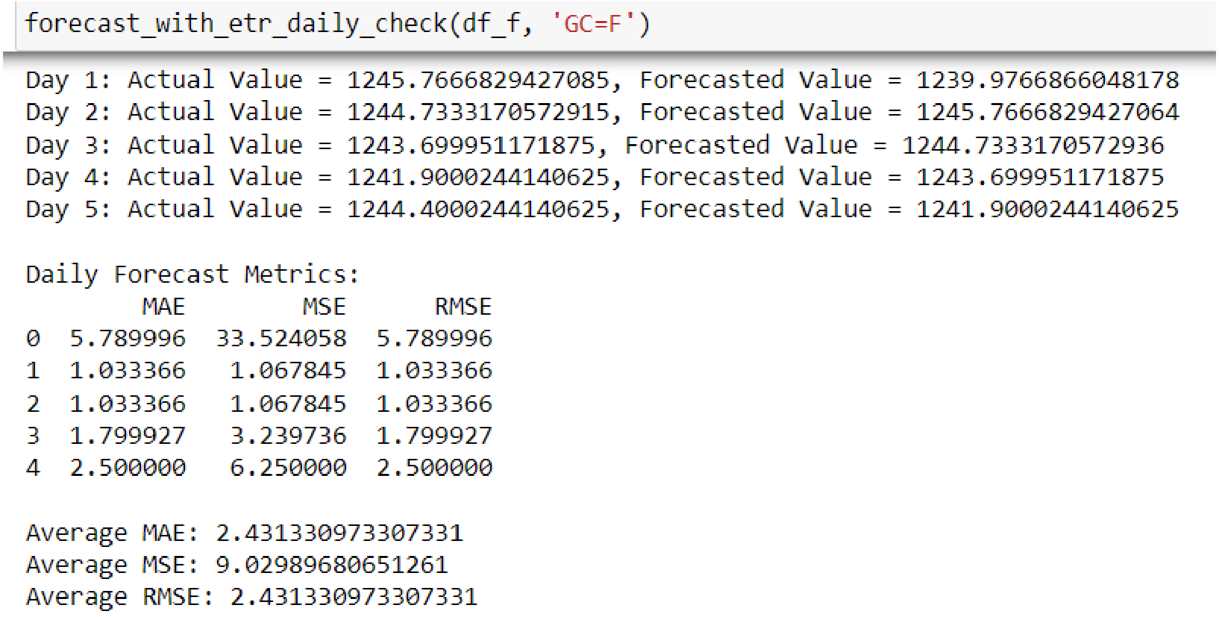
```
   13.435511344671227,
   79.72584513127802,
   49.74307981133459,
   5.81447434723375,
   25.822610470652602,
   62.966460135579084,
   31.953762620687463,
   50.81833279132841]},
           RNN       LSTM        GRU   CNN-LSTM        ANN       BPNN
 0  35.345756   4.254188  29.728565  32.513173  71.480586  54.531464
 1   3.467727   7.894107  59.567500  46.719876  62.531524  24.390224
 2  71.607193  15.361612  10.959320  26.986972  64.958228  13.435511
 3  33.637293  24.825937  71.015403  21.618293  28.459988  79.725845
 4  47.650307  39.843943   9.303577  36.467222  52.355719  49.743080
 5  11.052142   9.120607  15.890858  16.285490   9.740942   5.814474
 6  63.738178  53.952128  47.403167   9.779215  34.959810  25.822610
 7  69.101113  63.146322  26.554234  38.446238   7.264599  62.966460
 8  68.551735  34.592070  24.899339   5.080534  32.582179  31.953763
 9  16.309183  10.833546  71.960339  52.900494  42.138171  50.818333)
```

12. Illustration: The result of the Neural Network Creator, source: Own Findings

As we can see from the results, the main problem with these models lies in the highly fluctuating results. After running the test 10 times for 10 days for all the models in question, I have determined that this sort of fluctuation renders this approach unusable for my research purposes. The MAE score, for example, for one model, was 3.4677 for one RNN model, and subsequently, with the same data, it created a model with the score of 71.6072, which is absolutely preposterous. I managed to create models for the LSTM model that had lower MAE score than 0.4, which, I believe, is quite a good result, but once I started to use the same model with other interchangeable information that can potentially enhance the model, like currency pairs, the results were similarly unusable. This problem stems from the nature of neural networks, and the fact that they are created to solve specific challenges by rigorously tested and fine-tuned parameters, as opposed to my research goals, where I aim to find a universal methodology. I did manage to achieve slight progress with finding a model that had consecutively better results, nevertheless my efforts are not ready to be measurably provable to be able to complete their task at hand. Thus, I have to conclude, as much as it pains me, that this part of the research hasn't yielded the expected results, and it will require further study in the future to find a plausible solution.

4. Presenting Results

In this chapter, I will delve into my results and I will analyze my findings. I have examined the impact of introducing the five different kinds of inflation in to the data frame containing my previously established explanatory variables, that I have filtered for cross validation. To quickly recap, the **forecast_with_etr_daily_check** uses the Extra Trees Regressor model, with Recursive Forecasting. The results are as follows for the first data frame, **df_f**:



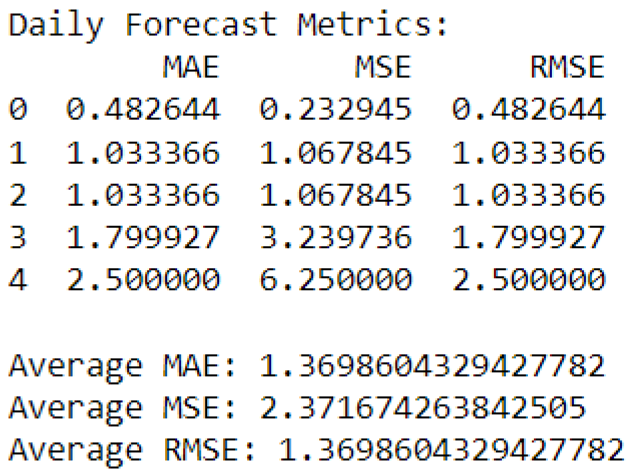
```
forecast_with_etr_daily_check(df_f, 'GC=F')
```

```
Day 1: Actual Value = 1245.7666829427085, Forecasted Value = 1239.9766866048178
Day 2: Actual Value = 1244.7333170572915, Forecasted Value = 1245.7666829427064
Day 3: Actual Value = 1243.699951171875, Forecasted Value = 1244.7333170572936
Day 4: Actual Value = 1241.9000244140625, Forecasted Value = 1243.699951171875
Day 5: Actual Value = 1244.4000244140625, Forecasted Value = 1241.9000244140625

Daily Forecast Metrics:
        MAE        MSE      RMSE
0  5.789996  33.524058  5.789996
1  1.033366   1.067845  1.033366
2  1.033366   1.067845  1.033366
3  1.799927   3.239736  1.799927
4  2.500000   6.250000  2.500000

Average MAE: 2.431330973307331
Average MSE: 9.02989680651261
Average RMSE: 2.431330973307331
```

13. Illustration: The results for df_f, source: own findings

The first data frame is here as a control group, it doesn’t have any kind of inflation data. For the exact variables of the data frame, I would like to refer to the 3.8.1. paragraph. From the output we can see, that the MAE score of this model was 2.4313, meaning that on average, the predicted value is off by this amount.

The next table is **df_f_HCPI**, which is the table that I have added the HCPI kind of inflation, in order to measure how much it improves the models' accuracy:

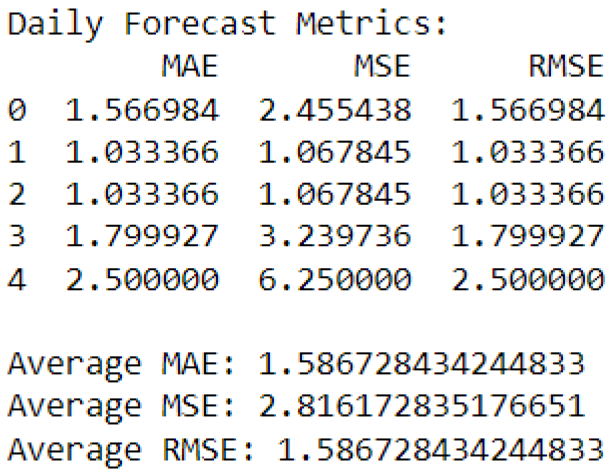


```
Daily Forecast Metrics:
        MAE       MSE      RMSE
0  0.482644  0.232945  0.482644
1  1.033366  1.067845  1.033366
2  1.033366  1.067845  1.033366
3  1.799927  3.239736  1.799927
4  2.500000  6.250000  2.500000

Average MAE: 1.3698604329427782
Average MSE: 2.371674263842505
Average RMSE: 1.3698604329427782
```

14. Illustration: The results for df_f_HCPI, source: own findings

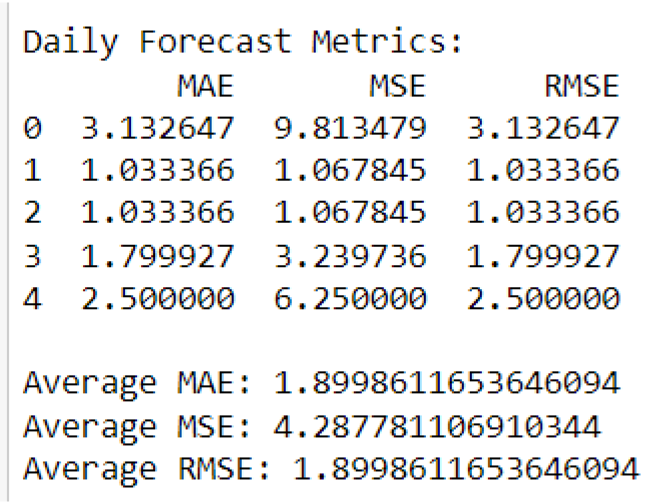
The results after introducing the HCPI among the variables gave quite the boost to the model, rendering the MAE score to just 1,3699. which is a decrease of 1,0614. This means that introducing the HCPI ceteris paribus to the model decreased the error by more than one, thus creating a much more reliable way to predict the price of gold, decreasing the amount of the average error by 43,66%. This has been tested on real life data in a real situation, simulating like I was predicting the price of gold every day, and thus, creating a testing environment, in an automatized way.

The rest of the data frames had the MAE scores as follows:



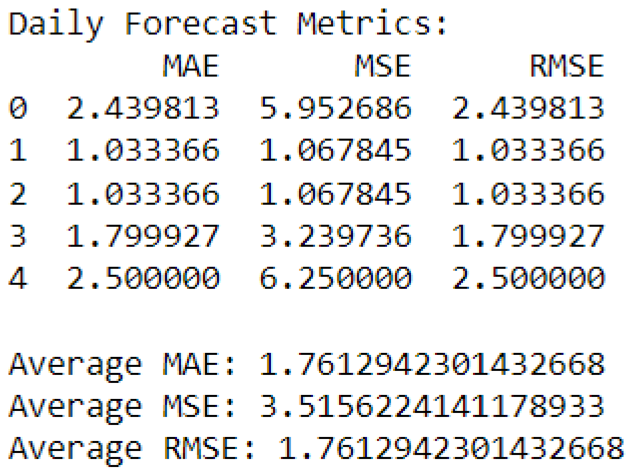
```
Daily Forecast Metrics:
        MAE       MSE      RMSE
0  1.566984  2.455438  1.566984
1  1.033366  1.067845  1.033366
2  1.033366  1.067845  1.033366
3  1.799927  3.239736  1.799927
4  2.500000  6.250000  2.500000

Average MAE: 1.586728434244833
Average MSE: 2.816172835176651
Average RMSE: 1.586728434244833
```

15. Illustration: The results for **df_f_ECPI**, source: own findings



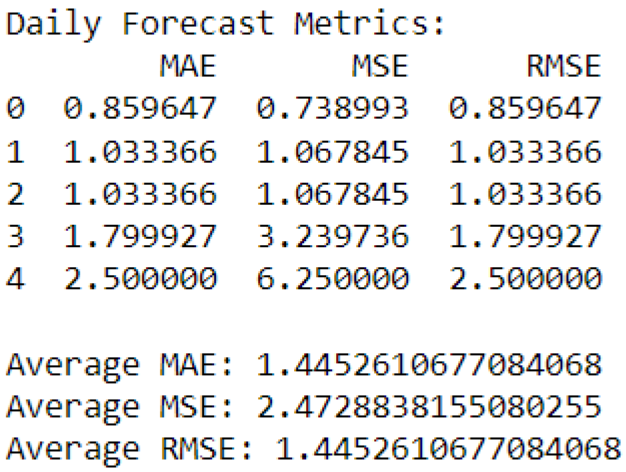
```
Daily Forecast Metrics:
        MAE       MSE      RMSE
0  3.132647  9.813479  3.132647
1  1.033366  1.067845  1.033366
2  1.033366  1.067845  1.033366
3  1.799927  3.239736  1.799927
4  2.500000  6.250000  2.500000

Average MAE: 1.8998611653646094
Average MSE: 4.287781106910344
Average RMSE: 1.8998611653646094
```

16. Illustration: The results for **df_f_FCPI**, source: own findings



```
Daily Forecast Metrics:
        MAE       MSE      RMSE
0  2.439813  5.952686  2.439813
1  1.033366  1.067845  1.033366
2  1.033366  1.067845  1.033366
3  1.799927  3.239736  1.799927
4  2.500000  6.250000  2.500000

Average MAE: 1.7612942301432668
Average MSE: 3.515622414117893
Average RMSE: 1.7612942301432668
```

17. Illustration: The results for **df_f_CCPI**, source: own findings



```
Daily Forecast Metrics:
        MAE       MSE      RMSE
0  0.859647  0.738993  0.859647
1  1.033366  1.067845  1.033366
2  1.033366  1.067845  1.033366
3  1.799927  3.239736  1.799927
4  2.500000  6.250000  2.500000

Average MAE: 1.4452610677084068
Average MSE: 2.4728838155080255
Average RMSE: 1.4452610677084068
```

18. Illustration: The results for **df_f_PPI**, source: own findings

To summarize, the efficiency of the different kinds of inflation:

The name of the inflation index	The respective MAE score
No inflation index	2,4313
HCPI - Headline Consumer Price Index	1,3699
ECPI - Energy Price Index	1,5867
FCPI - Food Price Index	1,8999
CCPI - Core Consumer Price Index	1,7613
PPI - Producer Price Index	1,4453

1.table each inflation index and their respective MAE score they produced in the testing environment, source: own findings

The table shows that introducing every kind of inflation ceteris paribus (all of them in different data frames) is promoting the accuracy of the model, however, some more than others. The HCPI yielded the best model which makes sense from an economic stand point of view (I already delved into the details in the 2.2.5. paragraph, but I would like to quickly reiterate): Gold prices have been historically intertwined with inflation, often serving as a hedge against rising prices. The Consumer Price Index (CPI), representing average changes in prices of goods and services, has been studied for its potential to predict gold price returns. A study titled "Gold and inflation(s) – A time-varying relationship" emphasizes the dynamic nature of the relationship between gold and inflation, suggesting that this relationship can change over different periods. Additionally, research has shown that inflation, as depicted by the CPI, can indeed predict gold price returns, reinforcing the belief in a tight-knit relationship between gold prices and inflation (Lucey, Sharma and Vigne, 2017). Finally, the concept of the "golden constant" in "Gold, the Golden Constant, and Déjà Vu" (Erb, Harvey and Viskanta, 2020) proposes that gold's long-term real return is zero, indicating its consistent value retention over time, especially in the context of inflation-adjusted prices. In essence, the CPI has demonstrated significance in predicting gold prices in the academic literature.

The PPI had much fewer sources than the HCPI (CPI in most literature), which led me to believe that it is a much weaker link in the general discourse. The Producer Price Index (PPI) serves as an index of prices at the producer level, rather than at the consumer level. Although one might initially deem it irrelevant, given that the Federal Reserve (Fed) emphasizes price stability for consumers, the U.S. central bank and investors closely monitor the PPI. This attention stems from PPI's ability to predict the Consumer Price Index (CPI), which is released shortly after the PPI. Historically, the relationship between PPI and gold has been more pronounced than that between CPI and gold. This correlation can be attributed to the PPI's status as a leading indicator of CPI, as well as its sensitivity to the business cycle since it tracks price changes at the wholesale level. Over time, the PPI has shown a strong inverse relationship with gold prices. For instance, during periods when the inflation rate of producer prices surged, such as in the 1970s and 2000s, or dipped, like in the 1980s and 1990s, gold prices responded in the opposite direction. Its interesting that important PPI announcements have been found to affect the price of gold. The main conclusion is that, while gold serves as a hedge against inflation, its effectiveness depends on the strength and acceleration of the inflation rate. Given the PPI's sensitivity to the business cycle and its early release schedule, it emerges as a strong predictor of gold prices, especially for investors leveraging short-term gold trading opportunities. (Sieroń, 2015)

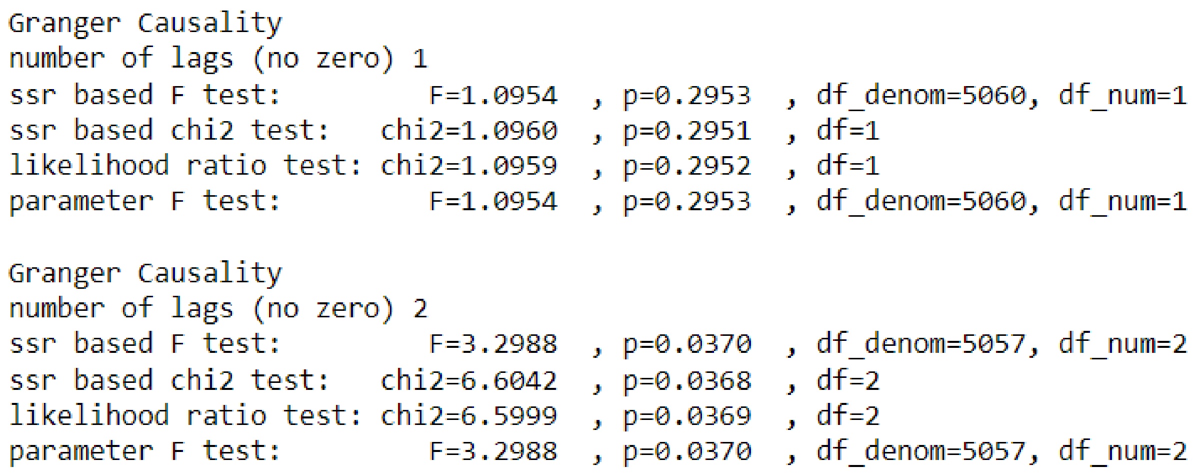
4.1. Granger Causality Test

In this paragraph, I will present the results of the Granger Causality test that I have performed to further support my previous findings.

Granger Causality is a statistical method that is used to see if one time series can predict another one. Rooted in the principle that if a variable X "Granger-causes" another variable Y, past values of X will furnish information that enhances the prediction of Y, surpassing the information exclusively contained in past values of Y. Originating over fifty years ago, Granger Causality has earned its reputation as an indispensable instrument for analyzing time series data in diverse sectors like economics, finance, genomics, and neuroscience. Yet, the legitimacy of this method in drawing genuine causal relationships among time series is an ongoing debate. Although its foundational definition was encompassing, computational constraints predominantly limited its applicability to simplistic bivariate vector auto-regressive processes or pairwise correlations between variables.

In multifaceted scenarios with more than two variables, for instance, X, Y, and Z, the testing methodology becomes intricate. While preliminary tests might probe if X Granger-causes Y by assessing certain hypotheses, this approach may overlook the broader causal landscape, necessitating the contemplation of the potential indirect causality of X on Y through its influence on Z.

Conventionally, Granger Causality is anchored in several assumptions, such as a linear vector autoregressive (VAR) model. However, the landscape of modern time series often challenges these assumptions, introducing non-linear dynamics and irregular sampling patterns. Some of these presuppositions encompass the demand for continuous-valued series, linearity, discrete-time sampling intervals, a predefined lag, and the mandate that all pertinent variables are meticulously observed, ensuring the absence of unmeasured confounders. While it stands as a very useful predictive tool, it is paramount to acknowledge that Granger Causality doesn't unequivocally signify genuine causation (Shojaie and Fox, 2021)(Eric, 2021). The golden standard for that is, as I mentioned before, the meticulous testing.



```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=1.0954  , p=0.2953  , df_denom=5060, df_num=1
ssr based chi2 test:   chi2=1.0960  , p=0.2951  , df=1
likelihood ratio test: chi2=1.0959  , p=0.2952  , df=1
parameter F test:         F=1.0954  , p=0.2953  , df_denom=5060, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=3.2988  , p=0.0370  , df_denom=5057, df_num=2
ssr based chi2 test:   chi2=6.6042  , p=0.0368  , df=2
likelihood ratio test: chi2=6.5999  , p=0.0369  , df=2
parameter F test:         F=3.2988  , p=0.0370  , df_denom=5057, df_num=2
```

19. Illustration: The results for the Granger Causality Test between GC=F and HCPI, source: own findings

These results mean that:

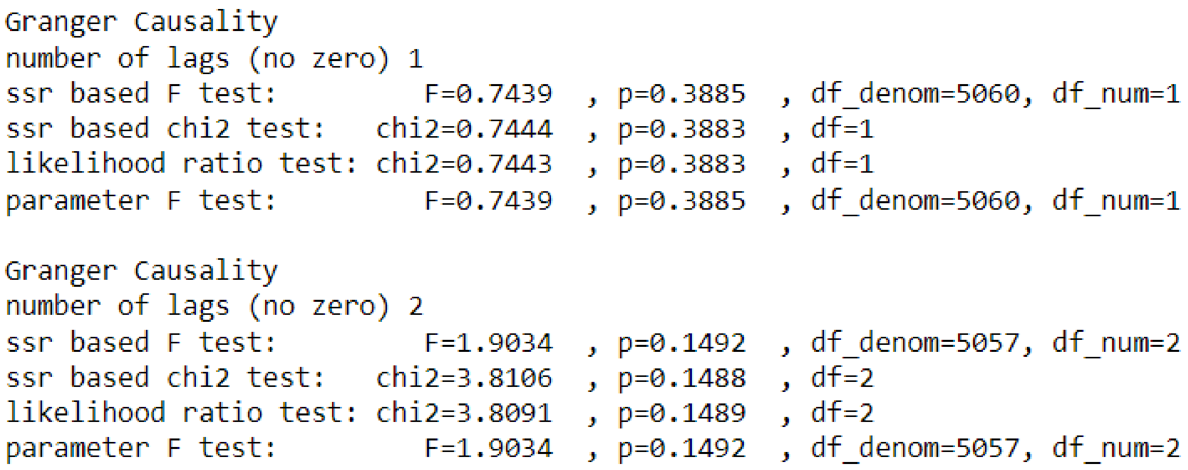
For 1 lag. all the tests (ssr based F test, ssr based chi2 test, likelihood ratio test, and parameter F test) have a p-value around 0.295. The p-values are greater than the typical significance level of 0.05, suggesting that with 1 lag, one time series does not Granger-cause the other.

For 2 lags, All the tests have a p-value around 0.037. The p-values are less than 0.05, indicating that with 2 lags, one time series does Granger-cause the other at the 5% significance level.

In summary, when considering 1 lag, there isn't sufficient evidence to claim that one time series Granger-causes the other. However, with 2 lags, there is evidence to suggest a Granger-causal relationship at the 5% significance level.

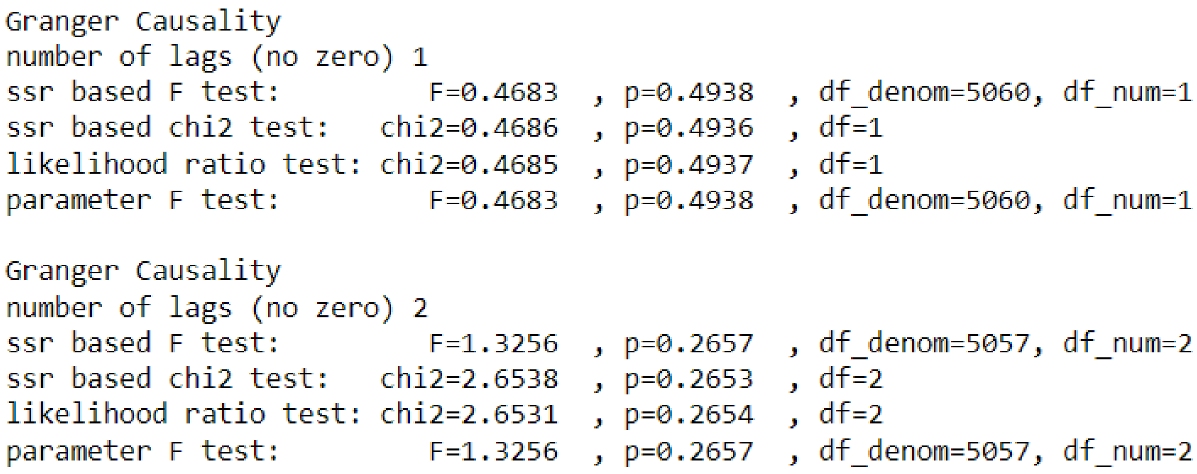
"Lags" in time series data refer to previous data points. Imagine you're looking at daily temperatures. Today's temperature is the current data point. If you go back one day, that's a "1-day lag." If you go back two days, that's a "2-day lag," and so on. In simple terms, lags help us see how past values might influence the present or future values. This means, that if the correlation was absolutely unequivocal, it would show up on the first lag, but since it only did on the second, the probability of the causality is lower, and the strength of it is weaker.

In order to show causality with the Granger test, we need a P value lower than 0,05. I will demonstrate the tests for the rest of the data frames here, as none of them show a strong sign of causality, not even at lag 2. The closest one is CCPI, with 0,0868, which is still not sufficient to pass the test:



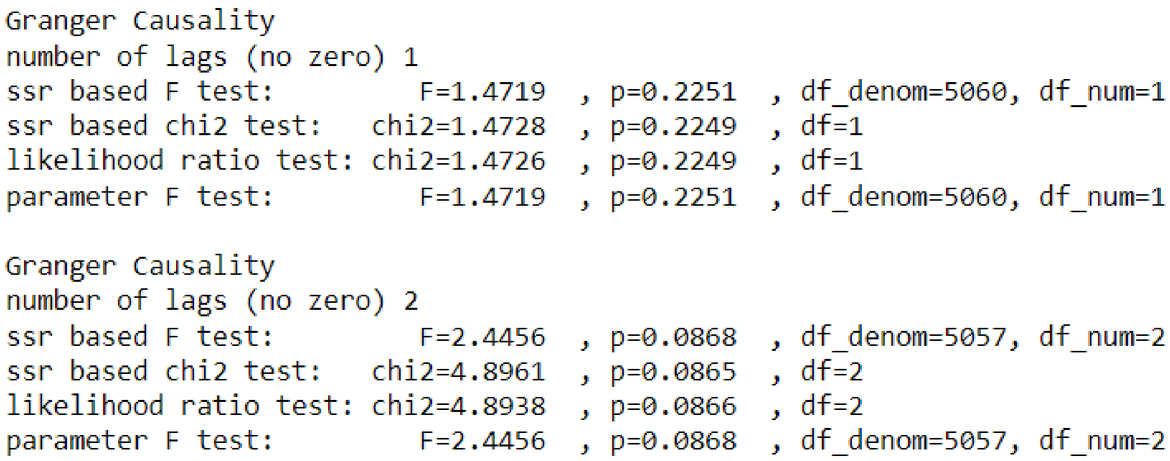
```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=0.7439  , p=0.3885  , df_denom=5060, df_num=1
ssr based chi2 test:   chi2=0.7444  , p=0.3883  , df=1
likelihood ratio test: chi2=0.7443  , p=0.3883  , df=1
parameter F test:         F=0.7439  , p=0.3885  , df_denom=5060, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=1.9034  , p=0.1492  , df_denom=5057, df_num=2
ssr based chi2 test:   chi2=3.8106  , p=0.1488  , df=2
likelihood ratio test: chi2=3.8091  , p=0.1489  , df=2
parameter F test:         F=1.9034  , p=0.1492  , df_denom=5057, df_num=2
```

20. Illustration: The results for the Granger Causality Test between GC=F and ECPI, source: own findings



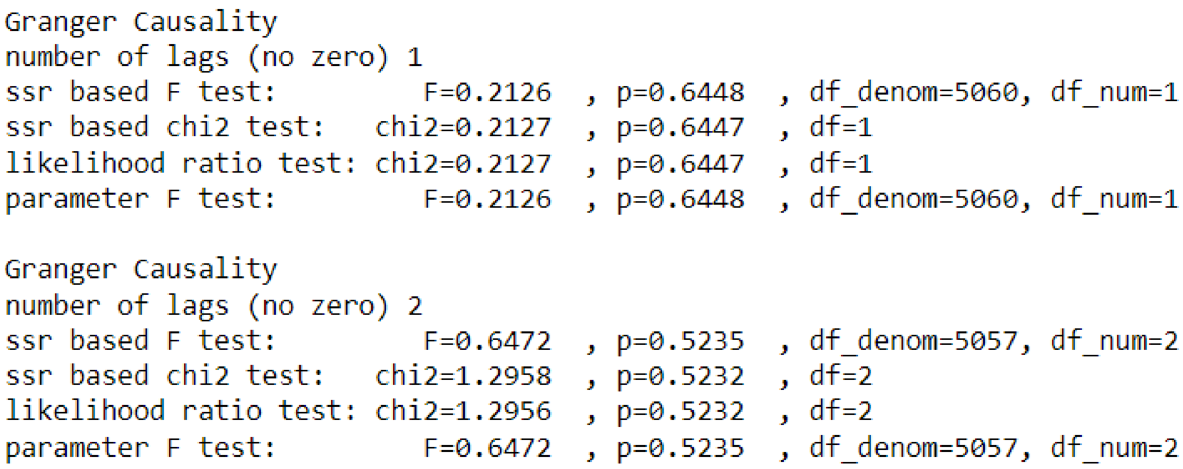
```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=0.4683  , p=0.4938  , df_denom=5060, df_num=1
ssr based chi2 test:   chi2=0.4686  , p=0.4936  , df=1
likelihood ratio test: chi2=0.4685  , p=0.4937  , df=1
parameter F test:         F=0.4683  , p=0.4938  , df_denom=5060, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=1.3256  , p=0.2657  , df_denom=5057, df_num=2
ssr based chi2 test:   chi2=2.6538  , p=0.2653  , df=2
likelihood ratio test: chi2=2.6531  , p=0.2654  , df=2
parameter F test:         F=1.3256  , p=0.2657  , df_denom=5057, df_num=2
```

21. Illustration: The results for the Granger Causality Test between GC=F and FCPI, source: own findings



```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=1.4719  , p=0.2251  , df_denom=5060, df_num=1
ssr based chi2 test:   chi2=1.4728  , p=0.2249  , df=1
likelihood ratio test: chi2=1.4726  , p=0.2249  , df=1
parameter F test:         F=1.4719  , p=0.2251  , df_denom=5060, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=2.4456  , p=0.0868  , df_denom=5057, df_num=2
ssr based chi2 test:   chi2=4.8961  , p=0.0865  , df=2
likelihood ratio test: chi2=4.8938  , p=0.0866  , df=2
parameter F test:         F=2.4456  , p=0.0868  , df_denom=5057, df_num=2
```

22. Illustration: The results for the Granger Causality Test between GC=F and CCPI, source: own findings



```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=0.2126  , p=0.6448  , df_denom=5060, df_num=1
ssr based chi2 test:   chi2=0.2127  , p=0.6447  , df=1
likelihood ratio test: chi2=0.2127  , p=0.6447  , df=1
parameter F test:         F=0.2126  , p=0.6448  , df_denom=5060, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=0.6472  , p=0.5235  , df_denom=5057, df_num=2
ssr based chi2 test:   chi2=1.2958  , p=0.5232  , df=2
likelihood ratio test: chi2=1.2956  , p=0.5232  , df=2
parameter F test:         F=0.6472  , p=0.5235  , df_denom=5057, df_num=2
```

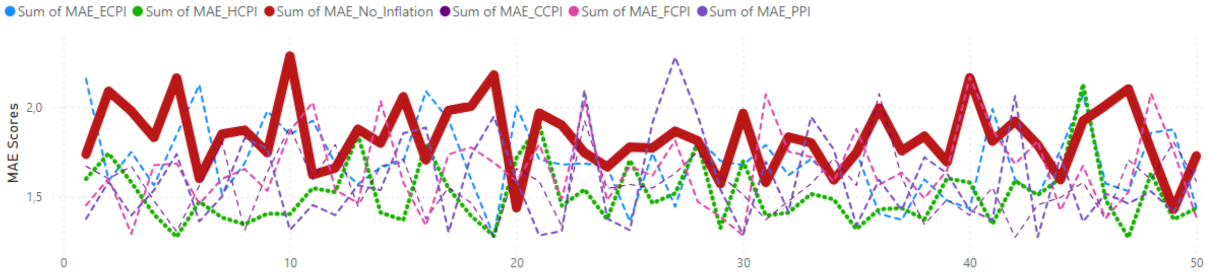
23. Illustration: The results for the Granger Causality Test between GC=F and PPI, source: own findings

It is a good indicator that the model that had the best results also showed the signs of causality according to the test. An essential distinction between this kind of Granger Causality test and the method that I'm researching is that this test was only performed between the price of gold and the given explanatory variable. The method I'm researching, on the other hand, delves into the fields of empirical testing (to some extent) and it examines the relationship between the explanatory variable and the target variable in a more complex environment, with other, potentially influential agents.

5. Hypothesis

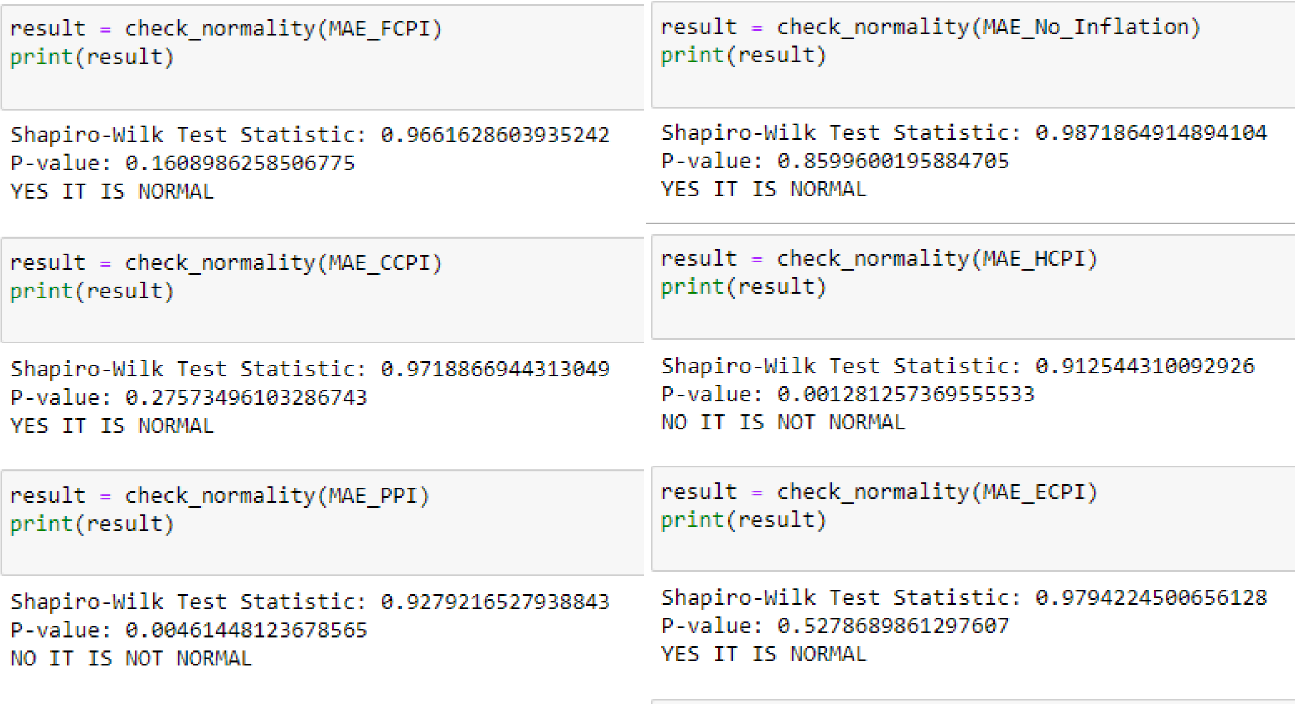
The purpose of the hypothesis is to prove based on numerical evidence that the change between the different MAE scores yielded by the inflation indexes has made a significant improvement in the model. The challenge here is that the research was aimed at creating a method that has no random element, thus making it measurable. Because of this, I basically have 6 numbers to go by (that can be found in the table in the last chapter) which is not nearly enough to do any sort of statistical experiments. There are two alternatives to overcome this situation: Bootstrapping (ar5iv, n.d.) and Hedge's g (Statology, 2021) (which is the low sample alternative to Cohen's d (Statistics How To, n.d.), but I still have my doubts about it being representative in this case). Another approach would be to take out the random state from the algorithm and run it 50 times, yielding slightly different MAE scores, which I can experiment on.

Considering all my options I wrote the **run_multiple_times_etr** function, that runs the **forecast_with_etr_daily_check** function 50 times. I put into lists the respective MAE scores that this method yielded: [MAE_No_Inflation], [MAE_HCPI],[MAE_ECPI],[MAE_FCPI],[MAE_CCPI],[MAE_PPI].



24. Illustration: The Visual Representation of the MAE Scores. RED: MAE without any inflation, GREEN: HCIP, source: Own findings

I also wrote the **check_normality** function, to make sure that my data has a normal distribution:



```
result = check_normality(MAE_FCPI)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.9661628603935242
P-value: 0.1608986258506775
YES IT IS NORMAL
```

```
result = check_normality(MAE_CCPI)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.9718866944313049
P-value: 0.27573496103286743
YES IT IS NORMAL
```

```
result = check_normality(MAE_PPI)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.9279216527938843
P-value: 0.00461448123678565
NO IT IS NOT NORMAL
```

```
result = check_normality(MAE_No_Inflation)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.9871864914894104
P-value: 0.8599600195884705
YES IT IS NORMAL
```

```
result = check_normality(MAE_HCPI)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.912544310092926
P-value: 0.001281257369555533
NO IT IS NOT NORMAL
```

```
result = check_normality(MAE_ECPI)
print(result)
```

```
Shapiro-Wilk Test Statistic: 0.9794224500656128
P-value: 0.5278689861297607
YES IT IS NORMAL
```

25,26. and Illustration: The results for the Shapiro_Wilk Test of the different MAE scores, source: own findings

The problem is that one of the conditions to perform the two paired T test is to have a normal distribution. This data will not be sufficient to test my hypotheses, with a paired-sample T-test (spssabc.hu, n.d.) (ResearchGate, n.d.). The alternative test for a not-normal distribution is the Paired Wilcoxon test (spssabc.hu, n.d.).

For this reason, I wrote the **paired_wilcoxon_test** function, that takes in the two lists in question and calculates the P value for them, in order to test my hypotheses.

5.1. H1 Hypothesis: Any of the inflation data significantly enhances the models accuracy

H0 (Null Hypothesis): There is no difference in the MAE scores between the model with HCPI and the model without any inflation index.

H1 (Alternative Hypothesis): There is a difference in the MAE scores between the model with HCPI and the model without any inflation index.

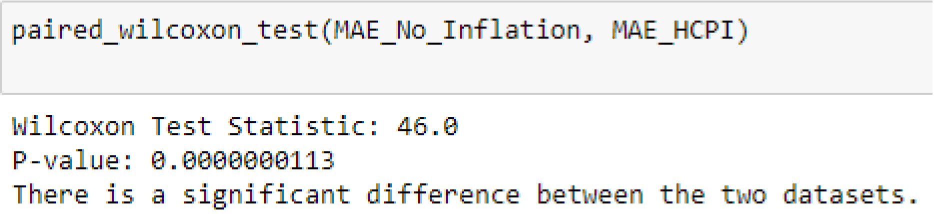
Test Used: Wilcoxon signed-rank test

Test Statistic Value: 46.0

P-value: 0.0000000113

Conclusion:

The p-value obtained (0.0000000113) is significantly less than the common alpha level of 0.05. This indicates that we reject the null hypothesis (H0) in favor of the alternative hypothesis (H1). Thus, there is a statistically significant difference in the MAE scores between the model with HCPI and the model without any inflation index.

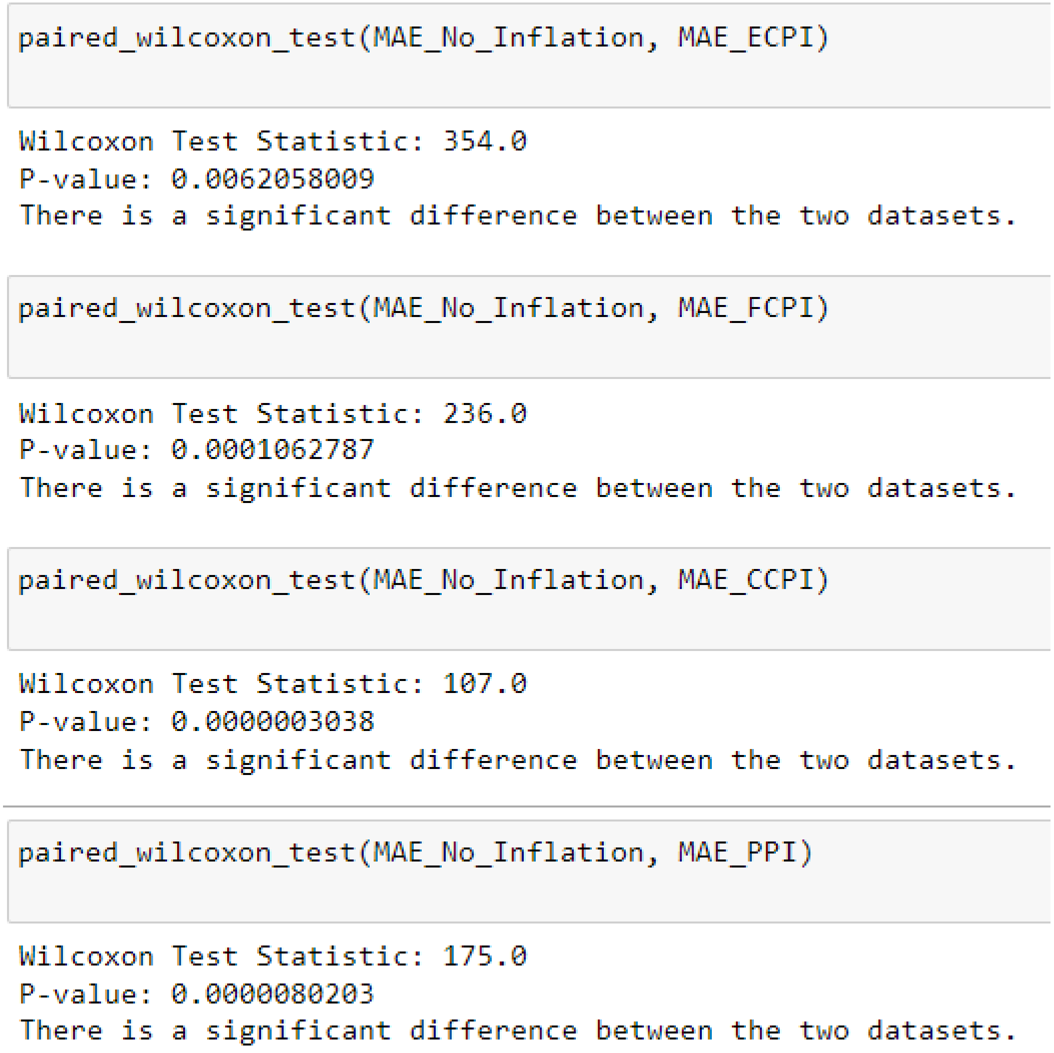


```
paired_wilcoxon_test(MAE_No_Inflation, MAE_HCPI)
```

```
Wilcoxon Test Statistic: 46.0
P-value: 0.0000000113
There is a significant difference between the two datasets.
```

27. Illustration, The code snippet from the Wilcoxon Test, source: Own findings

To make the most efficient use of this paper, I will not write the H0 and H1 structure for each one, but the results were the same, I will paste in the results of the outputs:



```
paired_wilcoxon_test(MAE_No_Inflation, MAE_ECPI)
```

```
Wilcoxon Test Statistic: 354.0
P-value: 0.0062058009
There is a significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_No_Inflation, MAE_FCPI)
```

```
Wilcoxon Test Statistic: 236.0
P-value: 0.0001062787
There is a significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_No_Inflation, MAE_CCPI)
```

```
Wilcoxon Test Statistic: 107.0
P-value: 0.0000003038
There is a significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_No_Inflation, MAE_PPI)
```

```
Wilcoxon Test Statistic: 175.0
P-value: 0.0000080203
There is a significant difference between the two datasets.
```

28. Illustration, the result of the Wilcoxon test for the other indexes, source: Own findings

Although there are some differences in the size of the P-value, which could make for some interesting conclusions, from the perspective of this research it is irrelevant how much they influence each other because I am attempting to establish a possible ubiety of causality.

The validity of this hypothesis is especially important because one of the main objectives of my research is to determine which kind of inflation affects the price of gold the most. If there is a way to support properly the presumption that one is significantly better, we might be able to reverse engineer this train of thought to gain theoretical knowledge. The fact that every inflation form of my choosing impacts the target variable is required for this purpose because if any of the inflation types wouldn’t have a significant effect on the price of gold, then they would have no relevancy at all for this line of inquiry.

5.2. H2 Hypothesis: The HCPI index is significantly better than the other indexes

To investigate this claim, I will go through every set of MAE scores that the indexes yielded one by one, and I will test them against the HCPI indexes MAE scores:

H0 (Null Hypothesis): There is no difference in the MAE scores between the model with HCPI and the model with ECPI.

H1 (Alternative Hypothesis): There is a difference in the MAE scores between the model with HCPI and the model with ECPI.

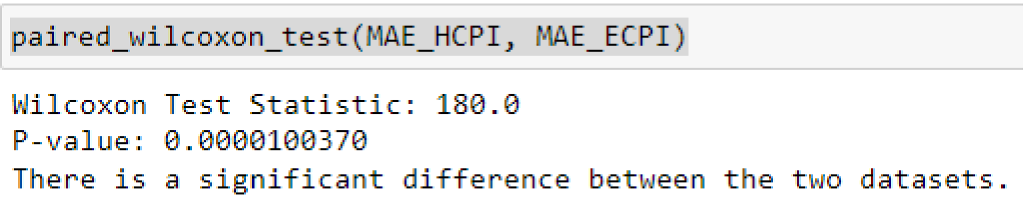
Test Used: Wilcoxon signed-rank test

Test Statistic Value: 180.0

P-value: 0.0000100370

Conclusion:

The p-value obtained (0.0000100370) is significantly less than the common alpha level of 0.05. This indicates that we reject the null hypothesis (H0) in favor of the alternative hypothesis (H1). Thus, there is a statistically significant difference in the MAE scores between the model with HCPI and the model with ECPI. In the context of your analysis, this suggests that the model's performance, as measured by MAE, is significantly impacted by whether HCPI or ECPI is incorporated as the inflation index.



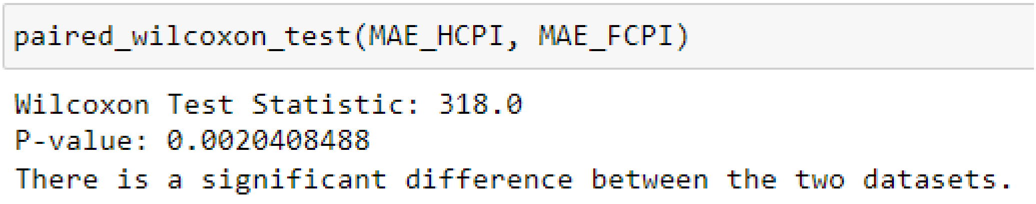
```
paired_wilcoxon_test(MAE_HCPI, MAE_ECPI)
```

```
Wilcoxon Test Statistic: 180.0
P-value: 0.0000100370
There is a significant difference between the two datasets.
```

29. Illustration, The result of the Wilcoxon test between MAE_HCPI and MAE_ECPI, source: Own Findings

As the ECPI yielded a less accurate model, then the HCPI, but still, a better model than the one without any inflation data, it makes sense, that the alpha value for this test is much larger than the one between the HCPI and the No_Inflation data. However, the result clearly shows that introducing the HCPI index into the model yields a significantly better result than introducing the ECPI index.

Testing the introduction of the HCPI against the FCPI yielded similar results:

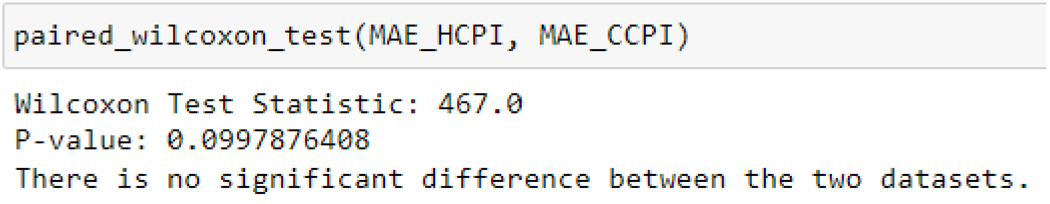


```
paired_wilcoxon_test(MAE_HCPI, MAE_FCPI)

Wilcoxon Test Statistic: 318.0
P-value: 0.0020408488
There is a significant difference between the two datasets.
```

30. Illustration, The result of the Wilcoxon test between MAE_HCPI and MAE_FCPI, source: Own Findings

The test showed that introducing the HCPI yields significantly better results than the FCPI.

The next investigation is for the CCPI index:

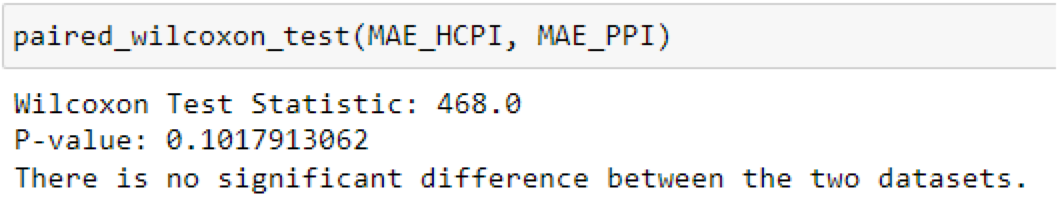


```
paired_wilcoxon_test(MAE_HCPI, MAE_CCPI)

Wilcoxon Test Statistic: 467.0
P-value: 0.0997876408
There is no significant difference between the two datasets.
```

31. Illustration, The result of the Wilcoxon test between MAE_HCPI and MAE_CCPI, source: Own Findings

The test showed that introducing the HCPI into the model instead of the CCPI did not show a significant increase in the results of the model in a 95% confidentiality rate. From a scientific perspective, this is great news because this means that this method can actually distinct between explanatory variables that influence the target variable significantly, and between the ones that do not. If the HCPI index, for example would have been significantly better than each one of the other variables, there would have been a chance, that the significance test is not sensitive enough for this test, or there is a methodical or technical error somewhere, that makes it impossible to detect the nuances of the explanatory variables. Fortunately, this test proved, that although the HCPI index is better for the purposes of explaining the price of gold, it is not significantly better than the CCPI index.

The result of the test was similar in the case of the PPI index:



```
paired_wilcoxon_test(MAE_HCPI, MAE_PPI)

Wilcoxon Test Statistic: 468.0
P-value: 0.1017913062
There is no significant difference between the two datasets.
```

32. Illustration, The result of the Wilcoxon test between MAE_HCPI and MAE_PPI, source: Own Findings

The result of the test means that although introducing the HCPI into the model is better than introducing the PPI, but not significantly.

In conclusion, we rejected the H0 hypothesis in the case of ECPI and FCPI and accepted H1, but we accepted H0 in the case of CCPI and PPI and rejected H1. This means that the HCPI is significantly better at explaining the price of gold than ECPI and FCPI, but not significantly better than CCPI and PPI.

These findings could transfer to the academic field of economy. Based on these findings, we could conclude (as an example) that the price index of food has a much less effect on the price of gold than all the other price indexes, as the Consumer Price Index is significantly better than the Food Price Index.

I performed the tests on the other indexes as well, to know how they affect each other:

	No Inflation	HCPI	ECPI	FCPI	CCPI	PPI
HCPI	🟩	⬛	🟩	🟩	🟨	🟨
ECPI	🟩	🟧	⬛	🟨	🟩	🟧
FCPI	🟩	🟧	🟧	⬛	🟧	🟧
CCPI	🟩	🟧	🟩	🟧	⬛	🟧
PPI	🟩	🟧	🟩	🟨	🟨	⬛

2. Table, The relationship of the indexes to the price of gold, and if they influence it significantly better than the other one, source: Own Findings

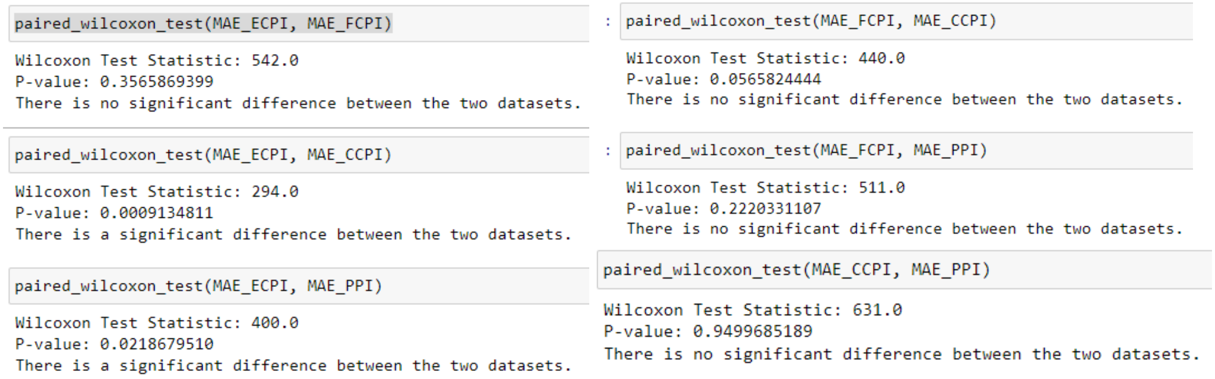
Green: The index on the left yields significantly better results than the index on top.

Yellow: The index on the left yields better results than the one on top, but not significantly.

Orange: The index on the left yields worse results than the one on top

Black: Not interpretable combinations.

The test for these indexes are as follows:



```
paired_wilcoxon_test(MAE_ECPI, MAE_FCPI)

Wilcoxon Test Statistic: 542.0
P-value: 0.3565869399
There is no significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_ECPI, MAE_CCPI)

Wilcoxon Test Statistic: 294.0
P-value: 0.0009134811
There is a significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_ECPI, MAE_PPI)

Wilcoxon Test Statistic: 400.0
P-value: 0.0218679510
There is a significant difference between the two datasets.
```

```
: paired_wilcoxon_test(MAE_FCPI, MAE_CCPI)

Wilcoxon Test Statistic: 440.0
P-value: 0.0565824444
There is no significant difference between the two datasets.
```

```
: paired_wilcoxon_test(MAE_FCPI, MAE_PPI)

Wilcoxon Test Statistic: 511.0
P-value: 0.2220331107
There is no significant difference between the two datasets.
```

```
paired_wilcoxon_test(MAE_CCPI, MAE_PPI)

Wilcoxon Test Statistic: 631.0
P-value: 0.9499685189
There is no significant difference between the two datasets.
```

34. Illustration, The result of the Wilcoxon test between MAE_ECPI, MAE_FCPI and MAE_PPI, source: Own Findings

5.3. H3 The Argument for Causality

The objective of this paper was to find a way to display causality through the instruments of machine learning. I will examine this main hypothesis in the context of all my research and my findings. The Oxford Dictionary defines causality this way: "*the relationship between something that happens and the reason for it happening"* (www.oxfordlearnersdictionaries.com, n.d.). This means that in order to prove the presence of causality, we need to observe an event, measure the impact of it (or make it measurable), and exclude the possibility that anything else could have caused it.

During my research, I created an environment for an event to occur, which was creating a model in a uniformed, repeatable manner. The event here was explaining the price of gold with different variables, and the impact was the way of perfecting the model through changing its explanatory variables in a controlled manner. I measured this impact, yet again in a uniformed way, through the mean absolute error (MAE).

Through these methods, I introduced the different kinds of index variables into this controlled environment. Because the introduction of these variables were ceteris paribus, I excluded the possibility that anything else has caused the effect, that it had on the outcome of the precision of the model. I have also supported statistically that there is a mathematically significant impact, after making sure that the conditions of the tests were present.

Based on all this information, I conclude that my research about finding a method that can prove the presence of causality in a mathematical sense is a success, and through the logical application of the steps that I have taken leaves small doubt that the phenomenon I observed and showed is anything else, but causality.

5.4. Future Research and Applications

In the future, I plan to delve further into the subject so that I can to optimize the procedure by experimenting with different ways to demonstrate causality, and I also wish to find a computationally less expensive way, that can still be unified. I would also like to write my method as a package so that it will be more accessible, and I will create a way to be measurable on a more precise scale (for example an index between 0 and 1).

Regarding pragmatical applications, after my method is rigorously tested and peer reviewed, I believe it can save a lot of time and effort for plenty of researchers, as it can somewhat replace the golden standard of proving causality, the observational research. Technically, this method is observational research, based on data that we already possess, possibly providing new insights into causal relations. The general uses can include economic studies, but with sufficient data, this method can be used in many other fields of science. After all, one of the basis of all science is to understand what causes what, and once we find out the link between the phenomena, we will have a much easier time finding out why.

6. Summary

Throughout this paper, I aimed to find an alternative way to prove the presence of causality with machine learning methods. My thought process was that if we can create empirical advances through machine learning methods, like determining or forecasting a variable through agents, then surely we can argue for a causal connection between them. I chose the forecasting the price of gold, more precisely, the question "Which inflation index explains the movement of the price of gold the best" for my research because I thought it was a good and complex example with numerous resources on the subject. I worked in a manner where the goal was to be universal and reproducible; thus the objective was not merely to create a model that is actually good at predicting the price of gold, but rather the creation of an environment that allows us to make measurements. I cast a large net in the explanatory variables and selected 86 of potentially relevant economic indicators. Likewise, I created machine learning models through rigorous testing, and introduced the different kinds of inflation indexes to it one by one. This allowed me, to measure which one of the inflation indexes had the best effect on the model, and by extension, which index explained the movement of the price of gold the most. I made

50 measurements of the mean average error of how each inflation index impacted the forecasting power of the model, and I used the Wilcoxon test to show which one of the indexes improved the model significantly. I used the original data frame as a control model, and after showing that all the models improved the model significantly, I examined their relationship towards each other. My first hypothesis (which included five sub-experiments) that all indexes improve the model significantly was proven. My second hypothesis was about the relationships of the indexes among themselves, with ten sub-experiments or hypotheses testings. My third hypothesis, sort of the summary of the goal of the whole study, was not a measurable experiment, but was rather a logical summary and conclusion to all my other hypotheses, that I believe I managed to find causal links between the examined agents. All in all, we can declare that this evaluation suggests a high likelihood of causality between the examined types of inflation and the price of gold, and we can affirm that this method of causality analysis holds great promise.

7. Bibliography

A, L., Qin, Q., Kamran, H.W., Sawangchai, A., Wisetsri, W. and Raza, M. (2021). How macroeconomic indicators influence gold price management. *Business Process Management Journal*, 27(7), pp.2075–2087. doi:https://doi.org/10.1108/bpmj-12-2020-0579.

Agga, A., Abbou, A., Labbadi, M., Houm, Y.E. and Ou Ali, I.H. (2022). CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electric Power Systems Research*, 208, p.107908. doi:https://doi.org/10.1016/j.epsr.2022.107908.

Ahmed, I. and Ishtiaq, S. (2021). Reliability and validity: Importance in Medical Research. *The Journal of the Pakistan Medical Association*, [online] 71(10). doi:https://doi.org/10.47391/JPMA.06-861.

Almeida-Filho, A.T. de, de Lima Silva, D.F. and Ferreira, L. (2020). Financial modelling with multiple criteria decision making: A systematic literature review. *Journal of the Operational Research Society*, 72(10), pp.1–19. doi:https://doi.org/10.1080/01605682.2020.1772021.

Ampomah, E.K., Qin, Z. and Nyame, G. (2020). Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information*, 11(6), p.332. doi:https://doi.org/10.3390/info11060332.

Analytics Vidhya. (2020). *Feature Selection using Wrapper Method - Python Implementation*. [online] Available at: https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/.

Analytics Vidhya. (2022). *Tutorial on RNN | LSTM |GRU with Implementation*. [online] Available at: https://www.analyticsvidhya.com/blog/2022/01/tutorial-on-rnn-lstm-gru-with-implementation/.

Anandasayanan, S., Thevananth, J. and Mathuranthy, A. (2019). The Relationship Between Inflation and Gold Price: Evidence From Sri Lanka. *International Journal of Accounting and Financial Reporting*, 9(4), p.322. doi:https://doi.org/10.5296/ijafr.v9i4.15750.

ar5iv. (n.d.). *Deterministic bootstrapping for a class of bootstrap methods*. [online] Available at: https://ar5iv.labs.arxiv.org/html/1903.10816v2#:~:text= [Accessed 6 Nov. 2023].

ar5iv. (n.d.). *Multi-layer perceptron with functional inputs: an inverse regression approach*. [online] Available at: https://ar5iv.labs.arxiv.org/html/0705.0211 [Accessed 4 Nov. 2023].

Balcilar, M., Gupta, R. and Pierdzioch, C. (2016). Does uncertainty move the gold price? New evidence from a nonparametric causality-in-quantiles test. *Resources Policy*, 49, pp.74–80. doi:https://doi.org/10.1016/j.resourpol.2016.04.004.

Baur, D.G. and Lucey, B.M. (2009). Is Gold a Hedge or a Safe Haven? an Analysis of Stocks, Bonds and Gold. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.952289.

Beretta, E. and Peluso, S. (2021). Gold and bubbles: an impossible binomial? A review of historical and current evidence. *Applied Economics Letters*, 29(3), pp.272–276. doi:https://doi.org/10.1080/13504851.2021.1897070.

Biswal, A. (2021). *What is Bagging in Machine Learning And How to Perform Bagging*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning.

Blose, L.E. (2010). Gold prices, cost of carry, and expected inflation. *Journal of Economics and Business*, 62(1), pp.35–47. doi:https://doi.org/10.1016/j.jeconbus.2009.07.001.

Brownlee, J. (2020). *How to Develop Elastic Net Regression Models in Python*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/elastic-net-regression-in-python/.

Bryan, M. and Cecchetti, S. (1993). The Consumer Price Index as a Measure of Inflation. doi:https://doi.org/10.3386/w4505.

Bureau of Labor Statistics (2023). *U.S. Bureau of Labor Statistics*. [online] Bls.gov. Available at: https://www.bls.gov.

Cai, J., Luo, J., Wang, S. and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, [online] 300, pp.70–79. doi:https://doi.org/10.1016/j.neucom.2017.11.077.

Capie, F., Mills, T.C. and Wood, G. (2005). Gold as a hedge against the dollar. *Journal of International Financial Markets, Institutions and Money*, [online] 15(4), pp.343–352. Available at:

https://econpapers.repec.org/article/eeeintfin/v_3a15_3ay_3a2005_3ai_3a4_3ap_3a343-352.htm [Accessed 30 Oct. 2023].

Cerliani, M. (2022). *How to Improve Recursive Time Series Forecasting*. [online] Medium. Available at: https://towardsdatascience.com/how-to-improve-recursive-time-series-forecasting-ff5b90a98eeb.

Chai, J., Zhao, C., Hu, Y. and Zhang, Z.G. (2021). Structural analysis and forecast of gold price returns. *Journal of Management Science and Engineering*, 6(2), pp.135–145. doi:https://doi.org/10.1016/j.jmse.2021.02.011.

Chiang, T.C. (2022). The effects of economic uncertainty, geopolitical risk and pandemic upheaval on gold prices. *Resources Policy*, 76, p.102546. doi:https://doi.org/10.1016/j.resourpol.2021.102546.

Chu, Y., Wang, X., Ma, J., Jia, K., Zhou, J. and Yang, H. (2020). Inductive Granger Causal Modeling for Multivariate Time Series. *arXiv (Cornell University)*. doi:https://doi.org/10.1109/icdm50108.2020.00111.

Corporate Finance Institute. (n.d.). *Elastic Net*. [online] Available at: https://corporatefinanceinstitute.com/resources/data-science/elastic-net/.

Corporate Finance Institute. (n.d.). *Elastic Net*. [online] Available at: https://corporatefinanceinstitute.com/resources/data-science/elastic-net/.

Couronné, R., Probst, P. and Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, [online] 19(1). doi:https://doi.org/10.1186/s12859-018-2264-5.

CRS INSIGHT Prepared for Members and Committees of Congress. (n.d.). Available at: https://crsreports.congress.gov/product/pdf/IN/IN11945.

Cuemath. (n.d.). *Linear Interpolation Formula - Derivation, Formulas, Examples*. [online] Available at: https://www.cuemath.com/linear-interpolation-formula/.

Daoud, J.I. (2017). Multicollinearity and Regression Analysis. *Journal of Physics: Conference Series*, 949(1), p.012009. doi:https://doi.org/10.1088/1742-6596/949/1/012009.

Das, S., Nayak, J., Kamesh Rao, B., Vakula, K. and Ranjan Routray, A. (2021). Gold Price Forecasting Using Machine Learning Techniques: Review of a Decade. *Computational Intelligence in Pattern Recognition*, pp.679–695. doi:https://doi.org/10.1007/978-981-16-2543-5_58.

DeepAI. (2019). *Stochastic Gradient Descent*. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/stochastic-gradient-descent.

Dhir, R. (2021). *Is the Stock Market Efficient?* [online] Investopedia. Available at: https://www.investopedia.com/articles/basics/04/022004.asp.

Donges, N. (2019). *Recurrent neural networks 101: Understanding the basics of RNNs and LSTM*. [online] Built In. Available at: https://builtin.com/data-science/recurrent-neural-networks-and-lstm.

Economist, G. (2022). *Gold: Consumer Price Index As An Indicator Of Demand | Seeking Alpha*. [online] seekingalpha.com. Available at: https://seekingalpha.com/article/4544035-assessing-the-gold-consumer-price-index [Accessed 31 Oct. 2023].

Eker, S., Rovenskaya, E., Obersteiner, M. and Langan, S. (2018). Practice and perspectives in the validation of resource management models. *Nature Communications*, 9(1). doi:https://doi.org/10.1038/s41467-018-07811-9.

El Mrabet, Z., Sugunaraj, N., Ranganathan, P. and Abhyankar, S. (2022). Random Forest Regressor-Based Approach for Detecting Fault Location and Duration in Power Systems. *Sensors*, 22(2), p.458. doi:https://doi.org/10.3390/s22020458.

Elfakhani, S., Baalbaki, I.B. and Rizk, H. (2009). Gold price determinants: empirical analysis and implications. *J. for International Business and Entrepreneurship Development*, 4(3), p.161. doi:https://doi.org/10.1504/jibed.2009.029010.

Erb, C., Harvey, C.R. and Viskanta, T. (2020). Gold, the Golden Constant, and Déjà Vu. *Financial Analysts Journal*, 76(4), pp.134–142. doi:https://doi.org/10.1080/0015198x.2020.1817698.

Eric (2021). *Introduction to Granger Causality*. [online] Aptech. Available at: https://www.aptech.com/blog/introduction-to-granger-causality/.

Farhat, S. and Ghalayini, L. (2020). Modeling and Forecasting Gold Prices. doi:https://doi.org/10.21203/rs.3.rs-23825/v1.

feature-engine.trainindata.com. (n.d.). *LagFeatures — 1.6.2*. [online] Available at: https://feature-engine.trainindata.com/en/latest/user_guide/timeseries/forecasting/LagFeatures.html#:~:text=Lag%20features%20are%20commonly%20used [Accessed 4 Nov. 2023].

Gadhave, A. (2022). Gold Price Prediction using Machine Learning. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 06(05). doi:https://doi.org/10.55041/ijsrem15027.

Geurts, P., Ernst, D. and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, [online] 63(1), pp.3–42. doi:https://doi.org/10.1007/s10994-006-6226-1.

Ha, J., Kose, M.A. and Ohnsorge, F. (2021). One-Stop Source: A Global Database of Inflation. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3881504.

Heinze, G., Wallisch, C. and Dunkler, D. (2018). Variable selection - A review and recommendations for the practicing statistician. *Biometrical Journal*, [online] 60(3), pp.431–449. doi:https://doi.org/10.1002/bimj.201700067.

Hintikka, J. and Halonen, I. (1999). Interpolation as Explanation. *Philosophy of Science*, [online] 66(S3), pp.S414–S423. doi:https://doi.org/10.1086/392742.

https://www.facebook.com/jason.brownlee.39 (2018). *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

Huang, W., Ye, M., Shi, Z. and Du, B. (2023). Generalizable Heterogeneous Federated Cross-Correlation and Instance Similarity Learning. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2309.16286.

Ishibashi, K., Iwasaki, T., Otomasa, S. and Yada, K. (2016). Model Selection for Financial Statement Analysis: Variable Selection with Data Mining Technique. *Procedia Computer Science*, 96, pp.1681–1690. doi:https://doi.org/10.1016/j.procs.2016.08.216.

Jabeur, S.B., Mefteh-Wali, S. and Viviani, J.-L. (2021). Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Annals of Operations Research*. doi:https://doi.org/10.1007/s10479-021-04187-w.

Jiang, T. (2020). Economic Variables Affecting the Gold Price : The Partial Effects of Linear Models. *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*. doi:https://doi.org/10.1109/icemme51517.2020.00112.

Kabra, G. (2023). *Bibliometric Analysis of Forecasting Gold Prices using Neural Network Models: A Review and Future Research Agenda*. [online] Social Science Research Network. Available at: https://ssrn.com/abstract=4518171 [Accessed 30 Oct. 2023].

Kaddour, J., Lynch, A., Liu, Q., Kusner, M.J. and Silva, R. (2022). Causal Machine Learning: A Survey and Open Problems. *arXiv:2206.15475 [cs, stat]*. [online] Available at: https://arxiv.org/abs/2206.15475.

Kirikkaleli, D. and Darbaz, I. (2022). New insights into an old issue: modelling the U.S. food prices. *Letters in Spatial and Resource Sciences*, 15(3), pp.675–689. doi:https://doi.org/10.1007/s12076-022-00319-3.

Kitco News. (2022a). *Gold prices continue to struggle, holding support above $1,700 following mixed U.S. PPI data*. [online] Available at: https://www.kitco.com/news/2022-09-14/Gold-prices-continue-to-struggle-holding-support-above-1-700-following-mixed-U-S-PPI-data.html [Accessed 31 Oct. 2023].

Kitco News. (2022b). *Gold prices holding above $1,800 as US PPI rises 7.4% in November, coming in hotter than expected*. [online] Available at: https://www.kitco.com/news/2022-12-09/Gold-prices-holding-above-1-800-as-US-PPI-rises-7-4-in-November-coming-in-hotter-than-expected.html [Accessed 31 Oct. 2023].

Kitco News. (2022c). *Gold prices struggle as U.S. PPI rises 0.2% in December*. [online] Available at: https://www.kitco.com/news/2022-01-13/Gold-prices-struggle-as-U-S-PPI-rises-0-2-in-December.html [Accessed 31 Oct. 2023].

Kristjanpoller, W. and Minutolo, M.C. (2015). Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert Systems with Applications*, 42(20), pp.7245–7251. doi:https://doi.org/10.1016/j.eswa.2015.04.058.

Kumar, P., Sinha, K., Nere, N.K., Shin, Y., Ho, R., Mlinar, L.B. and Sheikh, A.Y. (2020). A machine learning framework for computationally expensive transient models. *Scientific Reports*, 10(1). doi:https://doi.org/10.1038/s41598-020-67546-w.

leon.bottou.org. (n.d.). *projects:sgd [leon.bottou.org]*. [online] Available at: https://leon.bottou.org/projects/sgd.

Levine, P. (2023). *Inflation affects the price of everything—including a college education*. [online] Brookings. Available at: https://www.brookings.edu/articles/inflation-affects-the-price-of-everything-including-a-college-education/.

Lili, L. and Chengmei, D. (2013). Research of the Influence of Macro-Economic Factors on the Price of Gold. *Procedia Computer Science*, 17, pp.737–743. doi:https://doi.org/10.1016/j.procs.2013.05.095.

Lima, E., Davies, P., Kaler, J., Lovatt, F. and Green, M. (2020). Variable selection for inferential models with relatively high-dimensional data: Between method heterogeneity and covariate stability as adjuncts to robust selection. *Scientific Reports*, 10(1). doi:https://doi.org/10.1038/s41598-020-64829-0.

Ling, S., McAleer, M. and Tong, H. (2015). Frontiers in Time Series and Financial Econometrics: An overview. *Journal of Econometrics*, 189(2), pp.245–250. doi:https://doi.org/10.1016/j.jeconom.2015.03.019.

Lucey, B.M., Sharma, S.S. and Vigne, S.A. (2017). Gold and inflation(s) – A time-varying relationship. *Economic Modelling*, 67, pp.88–101. doi:https://doi.org/10.1016/j.econmod.2016.10.008.

M. Al-Maqaleh, B., A. Al-Mansoub, A. and N. Al-Badani, F. (2016). Forecasting using Artificial Neural Network and Statistics Models. *International Journal of Education and Management Engineering*, 6(3), pp.20–32. doi:https://doi.org/10.5815/ijeme.2016.03.03.

Macedo, F., Rosário Oliveira, M., Pacheco, A. and Valadas, R. (2019). Theoretical foundations of forward feature selection methods based on mutual information. *Neurocomputing*, 325, pp.67–89. doi:https://doi.org/10.1016/j.neucom.2018.09.077.

Machine Learning Compass. (2021). *Lasso Regression Explained, Step by Step*. [online] Available at: https://machinelearningcompass.com/machine_learning_models/lasso_regression/.

Mckinsey & Company (2022). *What is inflation: The causes and impact*. [online] www.mckinsey.com. Available at: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-inflation.

Mingers, J. and Meyer, M. (2017). Normalizing Google Scholar data for use in research evaluation. *Scientometrics*, 112(2), pp.1111–1121. doi:https://doi.org/10.1007/s11192-017-2415-x.

Mitsas, S., Golitsis, P. and Khudoykulov, K. (2022). Investigating the impact of geopolitical risks on the commodity futures. *Cogent Economics & Finance*, 10(1). doi:https://doi.org/10.1080/23322039.2022.2049477.

Mohtasham Khani, M., Vahidnia, S. and Abbasi, A. (2021). A Deep Learning-Based Method for Forecasting Gold Price with Respect to Pandemics. *SN Computer Science*, 2(4). doi:https://doi.org/10.1007/s42979-021-00724-3.

Moraffah, R., Sheth, P., Karami, M., Bhattacharya, A., Wang, Q., Tahir, A., Raglin, A. and Liu, H. (2021). Causal inference for time series analysis: problems, methods and evaluation. *Knowledge and Information Systems*, 63(12), pp.3041–3085. doi:https://doi.org/10.1007/s10115-021-01621-0.

Moussa, W., Mgadmi, N., Regaieg, R. and Bejaoui, A. (2020). The relationship between gold price and the American financial market. *International Journal of Finance & Economics*, 26(4), pp.6149–6155. doi:https://doi.org/10.1002/ijfe.2113.

Musah, A. and Ibrahim, M. (2015). Can gold investment provide a hedge against inflation and exchange rate? Evidence from a former Gold Coast. *International Journal of Economics and Business Research*, 10(2), p.148. doi:https://doi.org/10.1504/ijebr.2015.070981.

O’Connor, F.A., Lucey, B.M., Batten, J.A. and Baur, D.G. (2015). The financial economics of gold — A survey. *International Review of Financial Analysis*, 41(1), pp.186–205. doi:https://doi.org/10.1016/j.irfa.2015.07.005.

Oshiro, T.M., Perez, P.S. and Baranauskas, J.A. (2012). How Many Trees in a Random Forest? *Machine Learning and Data Mining in Pattern Recognition*, 7376, pp.154–168. doi:https://doi.org/10.1007/978-3-642-31537-4_13.

Oza, A. (2023). Reproducibility trial: 246 biologists get different results from same data sets. *Nature*. doi:https://doi.org/10.1038/d41586-023-03177-1.

Parisi, A., Parisi, F. and Díaz, D. (2008). Forecasting gold price changes: Rolling and recursive neural network models. *Journal of Multinational Financial Management*, 18(5), pp.477–487. doi:https://doi.org/10.1016/j.mulfin.2007.12.002.

Patino, C.M. and Ferreira, J.C. (2018). Internal and external validity: Can you apply research study results to your patients? *Jornal Brasileiro de Pneumologia*, [online] 44(3), pp.183–183. doi:https://doi.org/10.1590/s1806-37562018000000164.

Prabhakaran, S. (2023). *AdaBoost - An Introduction to AdaBoost*. [online] Machine Learning Plus. Available at: https://www.machinelearningplus.com/machine-learning/introduction-to-adaboost/.

Pradhan, A.K., Mishra, B.R., Tiwari, A.K. and Hammoudeh, S. (2020). Macroeconomic factors and frequency domain causality between Gold and Silver returns in India. *Resources Policy*, 68, p.101744. doi:https://doi.org/10.1016/j.resourpol.2020.101744.

Qian, Y., Ralescu, D.A. and Zhang, B. (2019). The analysis of factors affecting global gold price. *Resources Policy*, 64(0301-4207), p.101478. doi:https://doi.org/10.1016/j.resourpol.2019.101478.

Rashid, F., Altaf, S. and Aslam, M. (2020). Bayesian estimation of the biasing parameter for ridge regression: A novel approach. *Communications in Statistics - Simulation and Computation*, 51(12), pp.7215–7225. doi:https://doi.org/10.1080/03610918.2020.1827266.

ResearchGate. (n.d.). *(PDF) Paired Samples t-test*. [online] Available at: https://www.researchgate.net/publication/274635625_Paired_Samples_t-test.

Reuters. (2021). *Beyond CPI: Gold as a strategic inflation hedge*. [online] Available at: https://www.reuters.com/plus/beyond-cpi-gold-as-a-strategic-inflation-hedge.

Riyazahmed, D.K. (2021). *Neural Networks in Finance: A Descriptive Systematic Review*. [online] Social Science Research Network. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3812945 [Accessed 31 Oct. 2023].

Rokach, L. (2016). Decision forest: Twenty years of research. *Information Fusion*, 27, pp.111–125. doi:https://doi.org/10.1016/j.inffus.2015.06.005.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), pp.386–408. doi:https://doi.org/10.1037/h0042519.

Rosidi, N. (2022). *Overview of Machine Learning Algorithms: Regression*. [online] Medium. Available at: https://towardsdatascience.com/overview-of-machine-learning-algorithms-regression-e0f5510e84c.

Rosner, B., Glynn, R.J. and Lee, M.-L.T. (2006). The Wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, [online] 62(1), pp.185–92. doi:https://doi.org/10.1111/j.1541-0420.2005.00389.x.

Rothman, A. (2020). *The Bayesian Paradigm & Ridge Regression*. [online] Medium. Available at: https://towardsdatascience.com/the-bayesian-paradigm-ridge-regression-418af128ae8c.

Samal, A., Ummalla, M. and Goyari, P. (2022). The impact of macroeconomic factors on food price inflation: an evidence from India. *Future Business Journal*, 8(1). doi:https://doi.org/10.1186/s43093-022-00127-7.

Scardapane, S. and Wang, D. (2017). Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2), p.e1200. doi:https://doi.org/10.1002/widm.1200.

Schober, P., Boer, C. and Schwarte, L.A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, [online] 126(5), pp.1763–1768. doi:https://doi.org/10.1213/ane.0000000000002864.

scikit learn (2018). *1.4. Support Vector Machines — scikit-learn 0.20.3 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/svm.html.

scikit-learn. (n.d.). *1.5. Stochastic Gradient Descent*. [online] Available at: https://scikit-learn.org/stable/modules/sgd.html#regression [Accessed 4 Nov. 2023].

scikit-learn. (n.d.). *sklearn.ensemble.ExtraTreesRegressor*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html.

Scikit-learn.org. (2010). *sklearn.neural_network.MLPRegressor — scikit-learn 0.21.3 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html.

scikit-learn.org. (n.d.). *sklearn.ensemble.AdaBoostRegressor — scikit-learn 0.23.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html.

scikit-learn.org. (n.d.). *sklearn.ensemble.BaggingRegressor — scikit-learn 0.23.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html.

Seldon (2021). *Machine Learning Regression Explained*. [online] Seldon. Available at: https://www.seldon.io/machine-learning-regression-explained#:~:text=Machine%20learning%20regression%20models%20are%20mainly%20used%20in.

Shaik, M., Jamil, S.A., Hawaldar, I.T., Sahabuddin, M., Rabbani, M.R. and Atif, M. (2023). Impact of geo-political risk on stocks, oil, and gold returns during GFC, COVID-19, and Russian – Ukraine War. *Cogent Economics & Finance,* 11(1). doi:https://doi.org/10.1080/23322039.2023.2190213.

Sharma, S.S. (2016). Can consumer price index predict gold price returns? *Economic Modelling*, 55, pp.269–278. doi:https://doi.org/10.1016/j.econmod.2016.02.014.

Sharp 💻, T. (2020). *An Introduction to Support Vector Regression (SVR)*. [online] Medium. Available at: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2.

Shojaie, A. and Fox, E.B. (2021). Granger Causality: A Review and Recent Advances. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2105.02675.

Shu Yu Tew, Boley, M. and Schmidt, D.F. (2023). Bayes beats Cross Validation: Efficient and Accurate Ridge Regression via Expectation Maximization. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2310.18860.

Sieroń, A. (2015). *Do Inflation Indicators Drive The Prices Of Gold? | Seeking Alpha*. [online] seekingalpha.com. Available at: https://seekingalpha.com/article/3765476-inflation-indicators-drive-prices-of-gold [Accessed 5 Nov. 2023].

spssabc.hu. (n.d.). *Kétmintás T próba: típusai és elemzése | SPSSABC.HU*. [online] Available at: https://spssabc.hu/ketvaltozos-elemzes/ketmintas-t-proba-tipusai-es-elemzese/ [Accessed 6 Nov. 2023].

spssabc.hu. (n.d.). *Wilcoxon próba | SPSSABC.HU*. [online] Available at: https://spssabc.hu/ketvaltozos-elemzes/wilcoxon-proba/ [Accessed 6 Nov. 2023].

Staff, P.E. (2019). *Cross-Validation strategies for Time Series forecasting [Tutorial]*. [online] Packt Hub. Available at: https://hub.packtpub.com/cross-validation-strategies-for-time-series-forecasting-tutorial/.

Statistics By Jim. (2017). *Multicollinearity in Regression Analysis: Problems, Detection, and Solutions*. [online] Available at: https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/#:~:text=Multicollinearity%20occurs%20when%20independent%20variables.

Statistics How To. (n.d.). *Cohen’s D: Definition, Examples, Formulas*. [online] Available at: https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/cohens-d/.

Statology. (2021). *What is Hedges’ g? (Definition & Example)*. [online] Available at: https://www.statology.org/hedges-g/.

Staudemeyer, Ralf C and Morris, E.R. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.1909.09586.

Stock Market Prices Prediction using Random Forest and Extra Tree Regression. (2019). *International Journal of Recent Technology and Engineering*, 8(3), pp.1224–1228. doi:https://doi.org/10.35940/ijrte.c4314.098319.

Thiesen, S. (2021). *CatBoost regression in 6 minutes*. [online] Medium. Available at: https://towardsdatascience.com/catboost-regression-in-6-minutes-3487f3e5b329#:~:text=CatBoost%20is%20a%20relatively%20new [Accessed 4 Nov. 2023].

Thorn, J., Noble, S., Moore, T. and Hollingworth, W. (2013). Empirical evidence for the validity and reliability of resource-use measures based on patient recall: a systematic review. *Trials*, 14(S1). doi:https://doi.org/10.1186/1745-6215-14-s1-p55.

Tkacz, G. (2007). *Gold Prices and Inflation*. [online] Available at: https://www.bankofcanada.ca/wp-content/uploads/2010/02/wp07-35.pdf [Accessed 30 Oct. 2023].

Uddin, S., Haque, I., Lu, H., Moni, M.A. and Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, 12(1). doi:https://doi.org/10.1038/s41598-022-10358-x.

Valente, J.M. and Maldonado, S. (2020). SVR-FFS: A novel forward feature selection approach for high-frequency time series forecasting using support vector regression. *Expert Systems with Applications*, p.113729. doi:https://doi.org/10.1016/j.eswa.2020.113729.

Venkatesh, B. and Anuradha, J. (2019). A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, [online] 19(1), pp.3–26. doi:https://doi.org/10.2478/cait-2019-0001.

Wallingford, J. and Masters, W. (2021). Stringency of Movement Restrictions Linked With Higher Retail Food Prices but Not Overall Prices for All Consumer Goods During COVID-19 Pandemic. *Current Developments in Nutrition*, 5(Supplement_2), pp.254–254. doi:https://doi.org/10.1093/cdn/nzab029_055.

Wang, L., Zeng, Y. and Chen, T. (2015). Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Applications*, 42(2), pp.855–863. doi:https://doi.org/10.1016/j.eswa.2014.08.018.

World Bank. (n.d.). *A Global Database of Inflation*. [online] Available at: https://www.worldbank.org/en/research/brief/inflation-database.

Wu, A., Kuang, K., Xiong, R. and Wu, F. (2022). Instrumental Variables in Causal Inference and Machine Learning: A Survey. doi:https://doi.org/10.48550/arxiv.2212.05778.

www.britannica.com. (n.d.). *Linear regression | Definition, Formula, & Facts | Britannica*. [online] Available at: https://www.britannica.com/topic/linear-regression.

www.chicagofed.org. (n.d.). *What Drives Gold Prices? - Federal Reserve Bank of Chicago*. [online] Available at: https://www.chicagofed.org/publications/chicago-fed-letter/2021/464.

www.ibm.com. (n.d.). *What are Neural Networks? | IBM*. [online] Available at: https://www.ibm.com/topics/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as.

www.mathworks.com. (n.d.). *Rolling-Window Analysis of Time-Series Models - MATLAB & Simulink*. [online] Available at: https://www.mathworks.com/help/econ/rolling-window-estimation-of-state-space-models.html.

www.oxfordlearnersdictionaries.com. (n.d.). *causality noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com*. [online] Available at: https://www.oxfordlearnersdictionaries.com/definition/english/causality?q=causality.

Yan, J., Xu, Y., Cheng, Q., Jiang, S., Wang, Q., Xiao, Y., Ma, C., Yan, J. and Wang, X. (2021). LightGBM: accelerated genomically designed crop breeding through ensemble learning. *Genome Biology*, 22(1). doi:https://doi.org/10.1186/s13059-021-02492-y.

Yao, Z. and Ruzzo, W.L. (2006). A Regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*, 7(S1). doi:https://doi.org/10.1186/1471-2105-7-s1-s11.

Zach (2020a). *Introduction to Lasso Regression*. [online] Statology. Available at: https://www.statology.org/lasso-regression/.

Zach (2020b). *Introduction to Ridge Regression*. [online] Statology. Available at: https://www.statology.org/ridge-regression/.

Zhang, P., Jia, Y. and Shang, Y. (2022). Research and application of XGBoost in imbalanced data. *International Journal of Distributed Sensor Networks*, 18(6), p.155013292211069. doi:https://doi.org/10.1177/15501329221106935.

Zhang, X. (2017). Support Vector Machines. pp.1214–1220. doi:https://doi.org/10.1007/978-1-4899-7687-1_810.

Zhang, X., Zhang, L., Zhou, Q. and Jin, X. (2022). A Novel Bitcoin and Gold Prices Prediction Method Using an LSTM-P Neural Network Model. *Computational Intelligence and Neuroscience*, 2022, pp.1–12. doi:https://doi.org/10.1155/2022/1643413.

Zhao, Y. and Liu, Q. (2023). Causal ML: Python package for causal inference machine learning. *SoftwareX*, 21, p.101294. doi:https://doi.org/10.1016/j.softx.2022.101294.

8. Appendices

I. Codes

```
def fetch_all_instruments(instruments):
    import pandas as pd
    import yfinance as yf
    from datetime import datetime

    data_frames = []

    for instrument in instruments:
        try:
            end_date = datetime.today().strftime('%Y-%m-%d')
            data = yf.download(instrument, start="2007-08-30", end=end_date)
            data_frames.append(data["Close"].rename(instrument))
        except Exception as e:
            print(f"Error fetching data for {instrument}: {e}")

    if not data_frames:
        print("No data was fetched.")
        return None

    df_currencies = pd.concat(data_frames, axis=1, join="outer")
    return df_currencies

import pandas as pd

def fill_daily_data(df):
    daily_df = pd.DataFrame()

    start_date = pd.to_datetime(df.index[0])
    end_date = pd.to_datetime(df.index[-1])
    daily_dates = pd.date_range(start=start_date, end=end_date, freq='D')

    for col in df.columns:

        daily_series = pd.Series(index=daily_dates)

        for i in range(len(df.index) - 1):

            start_date = pd.to_datetime(df.index[i])
            end_date = pd.to_datetime(df.index[i + 1])
            days_in_month = (end_date - start_date).days

            value_increment = (df[col].iloc[i + 1] - df[col].iloc[i]) / days_in_month

            for j in range(days_in_month):
                daily_series[start_date + pd.Timedelta(days=j)] = df[col].iloc[i] + j * value_increment

        daily_series[end_date] = df[col].iloc[-1]

        daily_df[col] = daily_series

    return daily_df

daily_df = fill_daily_data(df)

def correlation(df, x_variable=None):
    import pandas as pd
    from IPython.core.display import HTML
    if x_variable:
        corr = df.corr()[x_variable].sort_values(ascending=False).drop(x_variable)
    else:
        corr = df.corr().stack().reset_index()
        corr.columns = ['var1', 'var2', 'corr']
        corr = corr[(corr['corr'] != 1.0) & (corr['corr'] != -1.0)]
        corr = corr[corr['var1'] < corr['var2']]
        corr.sort_values(by=['corr'], ascending=False, inplace=True)

    html_table = '<table><thead><tr><th>Variable 1</th><th>Variable 2</th><th>Correlation</th></tr></thead>'
    html_table += '<tbody>'

    if x_variable:
        for index, row in corr.iteritems():
            html_table += '<tr><td>{}</td><td>{}</td><td>{}</td></tr>'.format(x_variable, index, round(row, 2))
    else:
        for index, row in corr.iterrows():
            html_table += '<tr><td>{}</td><td>{}</td><td>{}</td></tr>'.format(row['var1'], row['var2'], round(row['corr'], 2))

    html_table += '</tbody></table>'

    return HTML('<div style="max-height:300px; overflow-y:auto;">{}</div>'.format(html_table))
def select_variables_correlation(df_X, df_Y, correlation_level=0.8):
    correlations = df_X.apply(lambda x: pearsonr(x, df_Y)[0]).sort_values(ascending=False) #mondjuk itt lehetne .abs()-ot használni, de nekem így logikus
    selected_vars = []
    for var in correlations.index:
        if var not in selected_vars and all(pearsonr(df_X[var], df_X[other_var])[0] < correlation_level for other_var in selected_vars):
            selected_vars.append(var)
    return selected_vars, correlations[selected_vars]


def plot_side_by_side(data, columns):
```

```
import pandas as pd

import matplotlib.pyplot as plt

fig, axs = plt.subplots(len(columns), 2, figsize=(12, len(columns) * 4))


for i, col in enumerate(columns):

    axs[i, 0].plot(data[col].values)

    axs[i, 0].set_title(f"{col} - Time Series")

    axs[i, 0].set_xlabel("Index")

    axs[i, 0].set_ylabel("Value")


    axs[i, 1].hist(data[col], bins=20)

    axs[i, 1].set_title(f"{col} - Histogram")

    axs[i, 1].set_xlabel("Value")

    axs[i, 1].set_ylabel("Frequency")


plt.tight_layout()


plt.figure(figsize=(12, 6))

for col in columns:

    plt.plot(data[col].values, label=col)

plt.title("Time Series Comparison")

plt.xlabel("Index")

plt.ylabel("Value")

plt.legend()

plt.show()
```

```
def plot_target_vs_others_normalized(data, target_variable):
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.preprocessing import MinMaxScaler
    target = data[target_variable]
    other_variables = data.drop(columns=target_variable).columns

    data_without_target = data.drop(columns=target_variable)

    scaler = MinMaxScaler()
    normalized_data = scaler.fit_transform(data_without_target)

    target_scaler = MinMaxScaler()
    normalized_target = target_scaler.fit_transform(target.values.reshape(-1, 1))

    plt.figure(figsize=(12, 6))
    plt.plot(range(len(normalized_target)), normalized_target, label=target_variable)
    plt.title(f"{target_variable} - Time Series (Normalized)")
    plt.xlabel("Data Point")
    plt.ylabel("Normalized Value")
    plt.legend()
    plt.show()

    for variable in other_variables:
        variable_index = data.columns.get_loc(variable)
        normalized_variable = normalized_data[:, variable_index]
        plt.figure(figsize=(12, 6))

        plt.plot(range(len(normalized_target)), normalized_target, label=target_variable)

        plt.plot(range(len(data)), normalized_variable, label=variable)

        plt.title(f"{target_variable} vs {variable} - Time Series (Normalized)")

        plt.xlabel("Data Point")

        plt.ylabel("Normalized Value")

        plt.legend()

        plt.show()

def find_best_model_regression(df_X_train, df_X_test, df_y_train, df_y_test):



    import numpy as np

    import pandas as pd

    from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet, BayesianRidge, SGDRegressor

    from sklearn.tree import DecisionTreeRegressor

    from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, AdaBoostRegressor, ExtraTreesRegressor, BaggingRegressor

    from sklearn.svm import SVR

    from sklearn.neighbors import KNeighborsRegressor

    from sklearn.neural_network import MLPRegressor

    from xgboost import XGBRegressor

    from lightgbm import LGBMRegressor

    from catboost import CatBoostRegressor

    from sklearn import metrics



    models = [
        ("Linear Regression", LinearRegression()),
        ("Ridge Regression", Ridge()),
        ("Lasso Regression", Lasso()),
        ("ElasticNet Regression", ElasticNet()),
        ("Bayesian Ridge Regression", BayesianRidge()),
        ("Decision Tree Regressor", DecisionTreeRegressor()),
        ("Random Forest Regressor", RandomForestRegressor()),
        ("Gradient Boosting Regressor", GradientBoostingRegressor()),
        ("XGBoost Regressor", XGBRegressor(eval_metric="rmse")),
        ("LightGBM Regressor", LGBMRegressor()),
        ("Support Vector Regressor", SVR()),
        ("K-Nearest Neighbors Regressor", KNeighborsRegressor()),
        ("Multi-layer Perceptron Regressor", MLPRegressor(max_iter=1000)),
        ("AdaBoost Regressor", AdaBoostRegressor()),
        ("Extra Trees Regressor", ExtraTreesRegressor()),
        ("Stochastic Gradient Descent Regressor", SGDRegressor()),
        ("Bagging Regressor", BaggingRegressor()),
        ("CatBoost Regressor", CatBoostRegressor(verbose=0, random_seed=42)),
    ]


    best_model = None
    best_mse = float("inf")
    best_model_instance = None


    for name, model in models:
        model.fit(df_X_train, df_y_train)
        y_pred = model.predict(df_X_test)

        print(name + " Model:")
        mse = metrics.mean_squared_error(df_y_test, y_pred)
        rmse = np.sqrt(mse)
        r2 = metrics.r2_score(df_y_test, y_pred)

        print("Mean Squared Error:", mse)
        print("Root Mean Squared Error:", rmse)
        print("R-squared:", r2)
        print()

        if mse < best_mse:
            best_mse = mse
            best_model = name
            best_model_instance = model

    print("The best model is:", best_model, "with a Mean Squared Error of", best_mse)

    # Print the metrics for the best model
    y_pred_best = best_model_instance.predict(df_X_test)
    best_rmse = np.sqrt(best_mse)
    best_r2 = metrics.r2_score(df_y_test, y_pred_best)
    print("Metrics of the best model:")
    print("Mean Squared Error:", best_mse)
    print("Root Mean Squared Error:", best_rmse)
```

```
    print("R-squared:", best_r2)

def forward_feature_selection(X, y, max_features=None, test_size=0.3, mae_threshold=1):
    remaining_features = list(X.columns)
    selected_features = []
    num_iterations = len(remaining_features)
    metrics_history = {}
    best_mae_global = float('inf')

    if max_features is not None:
        num_iterations = min(num_iterations, max_features)

    for i in range(1, num_iterations + 1):
        best_mae = float('inf')
        best_r2 = float('-inf')
        best_feature = None

        for feature in remaining_features:
            candidate_features = selected_features + [feature]
            X_subset = X[candidate_features]

            X_train, X_test, y_train, y_test = train_test_split(
                X_subset, y, test_size=test_size, random_state=1)

            model = LinearRegression()
            model.fit(X_train, y_train)
            y_pred = model.predict(X_test)

            mae = mean_absolute_error(y_test, y_pred)
            r2 = r2_score(y_test, y_pred)

            if mae < best_mae:
                best_mae = mae
                best_r2 = r2
                best_feature = feature

        if best_mae_global - best_mae < mae_threshold:
            print(f"Stopping: Improvement in MAE below {mae_threshold}")
            break

        best_mae_global = best_mae

        if best_feature is not None:
            remaining_features.remove(best_feature)
            selected_features.append(best_feature)
            metrics_history[i] = {'MAE': best_mae, 'R2': best_r2}

        print(f"Progress: Iteration {i}/{num_iterations}")

    return selected_features, metrics_history

def create_lagged_features(df_x, lags=1):
    df_x_lagged = df_x.copy()
    for i in range(1, lags + 1):
        lagged = df_x.shift(i)
        df_x_lagged = pd.concat([df_x_lagged, lagged.add_suffix(f'_lag{i}')], axis=1)
    df_x_lagged.dropna(inplace=True)
    return df_x_lagged


def evaluate_lagged_features(df_x, df_y, model, df_y_review, days_to_forecast=30):
    df_x_lagged = create_lagged_features(df_x, days_to_forecast)


    df_y = df_y.loc[df_x_lagged.index]


    X_train, X_review, y_train, y_review = train_test_split(df_x_lagged, df_y, test_size=days_to_forecast, shuffle=False)


    model.fit(X_train, y_train)


    predictions = model.predict(X_review)


    mae = mean_absolute_error(df_y_review, predictions)
    mse = mean_squared_error(df_y_review, predictions)
    rmse = math.sqrt(mse)
    print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')
def create_date_features(df_x):
    df_x_date_features = df_x.copy()
    df_x_date_features.index = pd.to_datetime(df_x_date_features.index)  # Convert index to datetime
    df_x_date_features['year'] = df_x_date_features.index.year
    df_x_date_features['month'] = df_x_date_features.index.month
    df_x_date_features['day'] = df_x_date_features.index.day
    df_x_date_features['dayofweek'] = df_x_date_features.index.dayofweek
    df_x_date_features['quarter'] = df_x_date_features.index.quarter
    return df_x_date_features


def evaluate_date_features(df_x, df_y, model, df_y_review, days_to_forecast=30):
    df_x_date_features = create_date_features(df_x)
    df_x_date_features.reset_index(inplace=True)
    df_y = df_y.reset_index()
    df_y['Date'] = pd.to_datetime(df_y['Date'])
    df_y.columns = ['Date', 'Price']
    merged_df = pd.merge(df_x_date_features, df_y, on='Date')
    X = merged_df.drop(['Date', 'Price'], axis=1)
    y = merged_df['Price']
    X_train, X_review, y_train, y_review = train_test_split(X, y, test_size=days_to_forecast, shuffle=False)
    model.fit(X_train, y_train)
    predictions = model.predict(X_review)
    mae = mean_absolute_error(df_y_review, predictions)
    mse = mean_squared_error(df_y_review, predictions)
    rmse = math.sqrt(mse)
    print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')


def neuralnetworkcreator_daily(df, target, model_type='GRU', forecast_days=5, num_epochs=50):
    results = {}
```

```
predicted_values = []

maes, mses = [], []


assert not df.isnull().any().any(), "DataFrame contains NaN values"

assert np.isfinite(df).all().all(), "DataFrame contains infinite values"


df.sort_index(inplace=True)


X = df.drop(target, axis=1)

y = df[target]


scaler_X = MinMaxScaler()

scaler_y = MinMaxScaler()


y = np.reshape(y.values, (-1, 1))


X_scaled = pd.DataFrame(scaler_X.fit_transform(X), columns=X.columns, index=X.index)

y_scaled = pd.DataFrame(scaler_y.fit_transform(y), columns=[target], index=X.index).iloc[:, 0]


X_train = X_scaled[:-forecast_days]

y_train = y_scaled[:-forecast_days]

X_test = X_scaled[-forecast_days:]

y_test = y_scaled[-forecast_days:]


test_dates = y_test.index
```

```
    X_train = np.reshape(X_train.values, (X_train.shape[0], 1, X_train.shape[1]))

    model = Sequential()

    if model_type == 'RNN':
        model.add(SimpleRNN(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    elif model_type == 'LSTM':
        model.add(LSTM(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    elif model_type == 'GRU':
        model.add(GRU(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    elif model_type == 'CNN-LSTM':
        model.add(Conv1D(filters=64, kernel_size=1, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
        model.add(LSTM(50, activation='relu'))
    elif model_type == 'ANN':
        model.add(Dense(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    elif model_type == 'BPNN':
        model.add(Dense(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    else:
        print("Invalid model type.")
        return None

    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')

    model.fit(X_train, y_train, epochs=num_epochs, batch_size=32, verbose=0)
```

```
print(f"Forecasting with {model_type} for the last {forecast_days} days...")

for i in range(forecast_days):
    x_input = np.reshape(X_test.iloc[i].values, (1, 1, X_test.shape[1]))
    y_pred_output = model.predict(x_input)
    if len(y_pred_output.shape) > 2:
        y_pred_scaled = y_pred_output.reshape(-1, 1)[0][0]
    else:
        y_pred_scaled = y_pred_output[0][0]

    y_pred = scaler_y.inverse_transform([[y_pred_scaled]])[0][0]
    y_actual = scaler_y.inverse_transform([[y_test.iloc[i]]])[0][0]

    print(f"Day {i}: Actual = {y_actual}, Forecast = {y_pred}, Difference = {y_actual - y_pred}")

    predicted_values.append(y_pred)
    maes.append(mean_absolute_error([y_actual], [y_pred]))
    mses.append(mean_squared_error([y_actual], [y_pred]))

results[model_type] = {'MAE': np.mean(maes), 'MSE': np.mean(mses)}
print(f"Results for {model_type}: {results[model_type]}")

forecast_df = pd.DataFrame({
    'Date': test_dates,
    'Actual': scaler_y.inverse_transform(np.reshape(y_test.values, (-1, 1)))[:, 0],
        'Forecast': predicted_values,
        'Difference': scaler_y.inverse_transform(np.reshape(y_test.values, (-1, 1)))[:, 0] - np.array(predicted_values)
    })

    return results, forecast_df

def run_multiple_times(df, target, num_runs=3, forecast_days=5, specific_model=None):
    if specific_model:
        model_types = [specific_model]
    else:
        model_types = ['RNN', 'LSTM', 'GRU', 'CNN-LSTM', 'ANN', 'BPNN']

    all_results = {model: [] for model in model_types}

    for model_type in model_types:
        print(f"Running {model_type} model {num_runs} times...")
        for i in range(num_runs):
            results, forecast_df = neuralnetworkcreator_daily(df, target, model_type=model_type, forecast_days=forecast_days)
            all_results[model_type].append(results[model_type]['MAE'])

        print("Forecasted Values:")
        print(forecast_df)

    results_df = pd.DataFrame(all_results)
    print("MAE Scores:")
    print(results_df)

    sum_mae = results_df.sum()
    sumsum_mae = sum_mae.sum()
    print("\nSum of MAE Scores:")
    print(sum_mae)
    print("\nSum of all MAE Scores:")
    print(sumsum_mae)

    return all_results, results_df

def average_change_calculator(df, target_variable, last_percent=1.0):
    """
    Calculate the average absolute change of a target variable in a Pandas DataFrame.

    Parameters:
        df (DataFrame): The input DataFrame.
        target_variable (str): The target variable's column name.
        last_percent (float): The percentage of the last portion of the data to consider,
                              value between 0 and 1. Default is 1.0 (consider all data).

    Returns:
        float: The average absolute change of the target variable.
    """
    if target_variable not in df.columns:
        return "The target variable does not exist in the DataFrame."
    if last_percent < 0 or last_percent > 1:
        return "The last_percent value should be between 0 and 1."

    start_idx = int(len(df) * (1 - last_percent))
    df_sliced = df.iloc[start_idx:]

    change = df_sliced[target_variable].diff().abs()
    average_change = change.mean()

    return average_change

def create_rolling_window_stats(df_x, window_size=5):
    rolling_means = df_x.rolling(window=window_size).mean().add_suffix('_mean')
    rolling_stds = df_x.rolling(window=window_size).std().add_suffix('_std')
    rolling_mins = df_x.rolling(window=window_size).min().add_suffix('_min')
    rolling_maxs = df_x.rolling(window=window_size).max().add_suffix('_max')

    rolling_stats = pd.concat([rolling_means, rolling_stds, rolling_mins, rolling_maxs], axis=1)
    rolling_stats.dropna(inplace=True)

    return rolling_stats


def evaluate_rolling_window_stats(df_x, df_y, model, df_y_review, days_to_forecast=30):
    df_x_rolling_stats = create_rolling_window_stats(df_x, days_to_forecast)
    df_y = df_y.loc[df_x_rolling_stats.index]

    X_train, X_review, y_train, y_review = train_test_split(df_x_rolling_stats, df_y, test_size=days_to_forecast, shuffle=False)

    model.fit(X_train, y_train)

    predictions = model.predict(X_review)

    mae = mean_absolute_error(df_y_review, predictions)
    mse = mean_squared_error(df_y_review, predictions)
    rmse = math.sqrt(mse)
    print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')

def neuralnetworkcreator_daily(df, target, model_type='GRU', forecast_days=5, num_epochs=50):
    results = {}
    predicted_values = []
    maes, mses = [], []

    assert not df.isnull().any().any(), "DataFrame contains NaN values"
    assert np.isfinite(df).all().all(), "DataFrame contains infinite values"

    df.sort_index(inplace=True)

    X = df.drop(target, axis=1)
    y = df[target]
```

```
scaler_X = MinMaxScaler()

scaler_y = MinMaxScaler()


y = np.reshape(y.values, (-1, 1))


X_scaled = pd.DataFrame(scaler_X.fit_transform(X), columns=X.columns, index=X.index)

y_scaled = pd.DataFrame(scaler_y.fit_transform(y), columns=[target], index=X.index).iloc[:, 0]


X_train = X_scaled[:-forecast_days]

y_train = y_scaled[:-forecast_days]

X_test = X_scaled[-forecast_days:]

y_test = y_scaled[-forecast_days:]


test_dates = y_test.index


X_train = np.reshape(X_train.values, (X_train.shape[0], 1, X_train.shape[1]))


model = Sequential()


if model_type == 'RNN':

    model.add(SimpleRNN(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))

elif model_type == 'LSTM':

    model.add(LSTM(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))

elif model_type == 'GRU':

    model.add(GRU(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
elif model_type == 'CNN-LSTM':
    model.add(Conv1D(filters=64, kernel_size=1, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    model.add(LSTM(50, activation='relu'))
elif model_type == 'ANN':
    model.add(Dense(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
elif model_type == 'BPNN':
    model.add(Dense(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
else:
    print("Invalid model type.")
    return None

model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

model.fit(X_train, y_train, epochs=num_epochs, batch_size=32, verbose=0)

print(f"Forecasting with {model_type} for the last {forecast_days} days...")

for i in range(forecast_days):
    x_input = np.reshape(X_test.iloc[i].values, (1, 1, X_test.shape[1]))
    y_pred_output = model.predict(x_input)
    if len(y_pred_output.shape) > 2:
        y_pred_scaled = y_pred_output.reshape(-1, 1)[0][0]
    else:
        y_pred_scaled = y_pred_output[0][0]
    y_pred = scaler_y.inverse_transform([[y_pred_scaled]])[0][0]
    y_actual = scaler_y.inverse_transform([[y_test.iloc[i]]])[0][0]

    print(f"Day {i}: Actual = {y_actual}, Forecast = {y_pred}, Difference = {y_actual - y_pred}")

    predicted_values.append(y_pred)
    maes.append(mean_absolute_error([y_actual], [y_pred]))
    mses.append(mean_squared_error([y_actual], [y_pred]))

results[model_type] = {'MAE': np.mean(maes), 'MSE': np.mean(mses)}
print(f"Results for {model_type}: {results[model_type]}")

forecast_df = pd.DataFrame({
    'Date': test_dates,
    'Actual': scaler_y.inverse_transform(np.reshape(y_test.values, (-1, 1)))[:, 0],
    'Forecast': predicted_values,
    'Difference': scaler_y.inverse_transform(np.reshape(y_test.values, (-1, 1)))[:, 0] - np.array(predicted_values)
})

return results, forecast_df

def run_multiple_times(df, target, num_runs=3, forecast_days=5, specific_model=None):
if specific_model:
    model_types = [specific_model]
else:
    model_types = ['RNN', 'LSTM', 'GRU', 'CNN-LSTM', 'ANN', 'BPNN']

all_results = {model: [] for model in model_types}

for model_type in model_types:
    print(f"Running {model_type} model {num_runs} times...")
    for i in range(num_runs):
        results, forecast_df = neuralnetworkcreator_daily(df, target, model_type=model_type, forecast_days=forecast_days)
        all_results[model_type].append(results[model_type]['MAE'])

    print("Forecasted Values:")
    print(forecast_df)

results_df = pd.DataFrame(all_results)
print("MAE Scores:")
print(results_df)

sum_mae = results_df.sum()
sumsum_mae = sum_mae.sum()
print("\nSum of MAE Scores:")
print(sum_mae)
print("\nSum of all MAE Scores:")
print(sumsum_mae)

return all_results, results_df
```

```
def run_multiple_times_etr(df_input, target_variable, test_size=0.3, forecast_days=5, model=None, num_runs=50):
    all_mae_scores = []


    for i in range(num_runs):
        print(f"Run {i + 1}/{num_runs}")
        _, _, metrics = forecast_with_etr_daily_check(df_input, target_variable, test_size, forecast_days, model)
        avg_mae = np.mean(metrics['MAE'])
        all_mae_scores.append(avg_mae)
        print(f"Average MAE for run {i + 1}: {avg_mae}\n")


    print("Collected MAE scores:")
    print(all_mae_scores)


    return all_mae_scores
def paired_ttest(mae_list1, mae_list2):
    """
    Perform a paired-sample t-test on two lists of MAE scores.


    Parameters:
    - mae_list1: List of MAE scores for the first model (e.g., No inflation or CCPI).
    - mae_list2: List of MAE scores for the second model (e.g., HCPI).


    Returns:
    - p-value from the t-test.
    """
    _, p_value = ttest_rel(mae_list1, mae_list2)
    formatted_p_value = "{:.15f}".format(p_value)

    return formatted_p_value

from scipy.stats import shapiro

def check_normality(data_list):
    # Perform Shapiro-Wilk test
    stat, p = shapiro(data_list)

    # Print the statistics
    print(f"Shapiro-Wilk Test Statistic: {stat}")
    print(f"P-value: {p}")

    # Check the p-value against the threshold
    if p > 0.05:
        return "YES IT IS NORMAL"
    else:
        return "NO IT IS NOT NORMAL"

from scipy.stats import wilcoxon

def paired_wilcoxon_test(data1, data2):
    stat, p = wilcoxon(data1, data2)
    # Format the p-value for full display
    formatted_p = "{:.10f}".format(p)


    print(f"Wilcoxon Test Statistic: {stat}")
    print(f"P-value: {formatted_p}")


    if p < 0.05:
        print("There is a significant difference between the two datasets.")
    else:
        print("There is no significant difference between the two datasets.")
```

II. Additional Charts and Tables

Forex (Foreign Exchange) Rates:¶

EURUSD=X: Euro to US Dollar exchange rate

USDJPY=X: US Dollar to Japanese Yen exchange rate

GBPUSD=X: British Pound to US Dollar exchange rate

AUDUSD=X: Australian Dollar to US Dollar exchange rate

NZDUSD=X: New Zealand Dollar to US Dollar exchange rate

EURJPY=X: Euro to Japanese Yen exchange rate

GBPJPY=X: British Pound to Japanese Yen exchange rate

EURGBP=X: Euro to British Pound exchange rate

EURCAD=X: Euro to Canadian Dollar exchange rate

EURSEK=X: Euro to Swedish Krona exchange rate

EURCHF=X: Euro to Swiss Franc exchange rate

EURHUF=X: Euro to Hungarian Forint exchange rate

CNY=X: Chinese Yuan exchange rate

HKD=X: Hong Kong Dollar exchange rate

SGD=X: Singapore Dollar exchange rate

INR=X: Indian Rupee exchange rate

MXN=X: Mexican Peso exchange rate

PHP=X: Philippine Peso exchange rate

IDR=X: Indonesian Rupiah exchange rate

THB=X: Thai Baht exchange rate

MYR=X: Malaysian Ringgit exchange rate

ZAR=X: South African Rand exchange rate

RUB=X: Russian Ruble exchange rate

Futures:

ES=F: E-mini S&P 500 Futures

YM=F: Mini-sized Dow Jones Industrial Average Futures

NQ=F: E-mini NASDAQ-100 Futures

ZB=F: 30 Year U.S. Treasury Bond Futures

ZN=F: 10 Year U.S. Treasury Note Futures

ZF=F: 5 Year U.S. Treasury Note Futures

ZT=F: 2 Year U.S. Treasury Note Futures

GC=F: Gold Futures

SI=F: Silver Futures

PL=F: Platinum Futures

HG=F: Copper Futures

PA=F: Palladium Futures

CL=F: Crude Oil Futures

HO=F: Heating Oil Futures

NG=F: Natural Gas Futures

RB=F: RBOB Gasoline Futures

BZ=F: Brent Crude Oil Futures

ZC=F: Corn Futures

ZO=F: Oats Futures

KE=F: Hard Red Spring Wheat Futures

ZR=F: Rough Rice Futures

ZM=F: Soybean Meal Futures

ZL=F: Soybean Oil Futures

ZS=F: Soybean Futures

GF=F: Feeder Cattle Futures

HE=F: Lean Hog Futures

LE=F: Live Cattle Futures

CC=F: Cocoa Futures

KC=F: Coffee Futures

CT=F: Cotton Futures

LBS=F: Random Length Lumber Futures

SB=F: Sugar Futures

Indices:

^GSPC: S&P 500 Index

^DJI: Dow Jones Industrial Average

^IXIC: NASDAQ Composite Index

^NYA: NYSE Composite Index

^XAX: NYSE American Composite Index

^RUT: Russell 2000 Index

^VIX: CBOE Volatility Index

^FTSE: FTSE 100 Index (UK)

^GDAXI: DAX Index (Germany)

^FCHI: CAC 40 Index (France)

^STOXX50E: EURO STOXX 50 Index (Europe)

^N100: Euronext 100 Index

^BFX: BEL 20 Index (Belgium)

^N225: Nikkei 225 (Japan)

^HSI: Hang Seng Index (Hong Kong)

000001.SS: SSE Composite Index (China, Shanghai Stock Exchange)

399001.SZ: Shenzhen Index (China, Shenzhen Stock Exchange)

^STI: STI Index (Singapore)

^AXJO: S&P/ASX 200 (Australia)

^AORD: ALL ORDINARIES (Australia)

^BSESN: S&P BSE SENSEX, an index of the Bombay Stock Exchange in India

^JKSE: The IDX Composite, an index of the Indonesia Stock Exchange

^KLSE: FTSE Bursa Malaysia KLCI, an index of the Kuala Lumpur Stock Exchange

^NZ50: S&P/NZX 50 INDEX GROSS, an index of the New Zealand Stock Exchange

^KS11: KOSPI Composite Index, an index of the Korea Exchange

^TWII: TSEC weighted index, an index of the Taiwan Stock Exchange

^GSPTSE: S&P/TSX Composite index, an index of the Toronto Stock Exchange

^BVSP: IBOVESPA, an index of the São Paulo Stock Exchange in Brazil

^MXX: IPC MEXICO, an index of the Mexican Stock Exchange

^IPSA - S&P/CLX IPSA - Santiago Delayed Price1

^MERV - MERVAL - Buenos Aires Delayed Price2

^TA125.TA - TA-125 Tel Aviv - Tel Aviv Delayed Price3

ABC Full List:

000001.SS: SSE Composite Index (China, Shanghai Stock Exchange)

399001.SZ: Shenzhen Index (China, Shenzhen Stock Exchange)

AORD: ALL ORDINARIES (Australia)

AUDUSD=X: Australian Dollar to US Dollar exchange rate

AXJO: S&P/ASX 200 (Australia)

BFX: BEL 20 Index (Belgium)

BSESN: S&P BSE SENSEX, an index of the Bombay Stock Exchange in India

BVSP: IBOVESPA, an index of the São Paulo Stock Exchange in Brazil

BZ=F: Brent Crude Oil Futures

CC=F: Cocoa Futures

CL=F: Crude Oil Futures

CNY=X: Chinese Yuan exchange rate

CT=F: Cotton Futures

DJI: Dow Jones Industrial Average

ES=F: E-mini S&P 500 Futures

EURCAD=X: Euro to Canadian Dollar exchange rate

EURCHF=X: Euro to Swiss Franc exchange rate

EURGBP=X: Euro to British Pound exchange rate

EURHUF=X: Euro to Hungarian Forint exchange rate

EURJPY=X: Euro to Japanese Yen exchange rate

EURSEK=X: Euro to Swedish Krona exchange rate

EURUSD=X: Euro to US Dollar exchange rate

FCHI: CAC 40 Index (France)

FTSE: FTSE 100 Index (UK)

GBPJPY=X: British Pound to Japanese Yen exchange rate

GBPUSD=X: British Pound to US Dollar exchange rate

GC=F: Gold Futures

GDAXI: DAX Index (Germany)

GF=F: Feeder Cattle Futures

GSPC: S&P 500 Index

GSPTSE: S&P/TSX Composite index, an index of the Toronto Stock Exchange

HE=F: Lean Hog Futures

HG=F: Copper Futures

HKD=X: Hong Kong Dollar exchange rate

HO=F: Heating Oil Futures

HSI: Hang Seng Index (Hong Kong)

IDR=X: Indonesian Rupiah exchange rate

INR=X: Indian Rupee exchange rate

IPSA - S&P/CLX IPSA - Santiago Delayed Price1

IXIC: NASDAQ Composite Index

JKSE: The IDX Composite, an index of the Indonesia Stock Exchange

KC=F: Coffee Futures

KE=F: Hard Red Spring Wheat Futures

KLSE: FTSE Bursa Malaysia KLCI, an index of the Kuala Lumpur Stock Exchange

KS11: KOSPI Composite Index, an index of the Korea Exchange

LBS=F: Random Length Lumber Futures

LE=F: Live Cattle Futures

MERV - MERVAL - Buenos Aires Delayed Price2

MXN=X: Mexican Peso exchange rate

MXX: IPC MEXICO, an index of the Mexican Stock Exchange

MYR=X: Malaysian Ringgit exchange rate

N100: Euronext 100 Index

N225: Nikkei 225 (Japan)

NG=F: Natural Gas Futures

NQ=F: E-mini NASDAQ-100 Futures

NZ50: S&P/NZX 50 INDEX GROSS, an index of the New Zealand Stock Exchange

NZDUSD=X: New Zealand Dollar to US Dollar exchange rate

NYA: NYSE Composite Index

PA=F: Palladium Futures

PHP=X: Philippine Peso exchange rate

PL=F: Platinum Futures

RB=F: RBOB Gasoline Futures

RUB=X: Russian Ruble exchange rate

RUT: Russell 2000 Index

SB=F: Sugar Futures

SGD=X: Singapore Dollar exchange rate

SI=F: Silver Futures

STI: STI Index (Singapore)

STOXX50E: EURO STOXX 50 Index (Europe)

TA125.TA - TA-125 Tel Aviv - Tel Aviv Delayed Price3

THB=X: Thai Baht exchange rate

TWII: TSEC weighted index, an index of the Taiwan Stock Exchange

USDJPY=X: US Dollar to Japanese Yen exchange rate

VIX: CBOE Volatility Index

XAX: NYSE American Composite Index

YM=F: Mini-sized Dow Jones Industrial Average Futures

ZAR=X: South African Rand exchange rate

ZB=F: 30 Year U.S. Treasury Bond Futures

ZC=F: Corn Futures

ZF=F: 5 Year U.S. Treasury Note Futures

ZL=F: Soybean Oil Futures

ZM=F: Soybean Meal Futures

ZN=F: 10 Year U.S. Treasury Note Futures

ZO=F: Oats Futures

ZR=F: Rough Rice Futures

ZT=F: 2 Year U.S. Treasury Note Futures

ZS=F: Soybean Future