ABSTRACT

This study explores the use of Mandarin text-derived news sentiment for forecasting Malaysia's GDP and its demand components: private investment, private consumption, imports, and exports. Utilizing 3,361 articles from See Hua Daily News spanning 2022-2023, the research applies machine learning techniques to assess the predictive power of news sentiment on economic indicators. The study found that the Mandarin-based news sentiment index significantly predicted the Business Condition Index (BCI) but not the Consumer Sentiment Index (CSI), aligning partially with previous research on English-language news sentiment. Unexpectedly, negative correlations were observed between the sentiment index and imports, exports, GDP, and private consumption, while a weak positive correlation was found with private investment. In forecasting macroeconomic variables, the LASSO model emerged as the most robust, effectively predicting GDP, private investment, and private consumption across one-, two-, and three-quarter horizons. These findings suggest that Mandarin news sentiment can be a valuable tool in economic forecasting, despite the study's limitations of using a single news source and a short time frame. The research highlights the potential of multilingual sentiment analysis in capturing diverse economic perspectives in Malaysia's multicultural context. Future work should focus on expanding data sources, extending the analysis period, and incorporating additional variables to enhance the robustness and accuracy of predictive models using news sentiment indices.

Keywords: Mandarin News Sentiment; Macroeconomic Forecasting; Machine Learning; Malaysian Macroeconomy; MIER Indices

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Definition
BCI	Business Conditions Index
CRISP-DM	Cross-industry Standard Process for Data Mining
CSI	Consumer Sentiments Index
DOSM	Department of Statistics Malaysia
DT	Decision Tree
DT	Decision Tree Regressor
ET	Extra Tree Regressor
ETS	Extra Tree
GDP	Gross Domestic Product
HFIs	High-Frequency Indicators
HTML	Hypertext Markup Language
JMZ	Sentiment dictionary by Jiang et al. (2019)
LIGHTGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MIER	Malaysian Institute of Economic Research
NLP	Natural Language Processing
OLS	Ordinary Least Squares
OLS-AR(1)	Ordinary Least Squares with Autoregressive Order 1
PCR	Principal Component Regression
RF	Random Forest
RMSE	Root Mean Squared Error
SVR	Support Vector Machine
VIF	Variance Inflation Factor
XGBoost	Extreme Gradient Boosting
YoY	Year-over-Year

CHAPTER 1

INTRODUCTION

The Gross Domestic Product (GDP) of a nation is an imperative measure of the country's economic vitality and the size of the country (Statistics Explained, 2023). The measure is often adopted by economists when evaluating the country's current economic state in that, an increasing GDP is indicative of an economic growth, while a decreasing GDP would be indicative of an economic decline instead. At the same time, considering the substantial influence of the sentiments and expectations of economic agents on the country's real economic activities, it is no surprise that central banks and policymakers place heavy emphasis on understanding these perceptions, in order to better prepare for changes in consumption levels, saving decisions, as well as the investment attitudes of the various economic agents. Traditionally, sentiments are measured through surveys, for instance, in the context of the Malaysian economy, the Business Conditions Index (BCI) and Consumer Sentiments Index (CSI) both serve the said purpose (Zulkefly Abdul Karim et al., 2022). These two indices are published by the Malaysian Institute of Economic Research (MIER), providing insights into the current and future economic prospects of the country. While the indices prove to be insightful, the implementation of the survey, which is laborious and time-consuming, presents considerable resistance, more so in terms of ensuring adequate frequency and precise representativeness of the country's population (Chong et al, 2021).

Such resistance increases more so during economic distress, such as those precipitated by the COVID-19 pandemic, where businesses that have either closed down or reduced operations may not respond to the surveys, thereby causing the data collection process to be incomplete. Such a scenario can lead to an inadequate representation of the public's sentiments, where the voices of those who are most vulnerable are more likely to be omitted, given their greater exposure to the economic downturn, eventually causing the sentiments to be lacking. Besides that, in volatile environments such as the likes brought upon by the pandemic, the need for prompt and agile approaches to sentiment analysis was made even more apparent, once again highlighting the relevance of considering alternative, and more robust methods to capture economic sentiments for uninterrupted insights (Committee for the Coordination of Statistical Activities [CCSA], 2020; Karim et al., 2022; Putra & Arini, 2020). Having said that, recent

technological development has unlocked new avenues for capturing the sentiments of economic players, one of which pertains to the increased availability of digital newspapers and annual reports, which economic agents grew to rely on for making informed decisions (Ardia et al., 2019; Buckman et al., 2020; Rambaccussing & Kwiatkowski, 2020; Bybee et al., 2021; Aprigliano et al., 2023). The other notable development pertains to advancements in computational linguistics and the improvement in computation capabilities. Collectively, these advancements have fostered an environment conducive for the analysis of vast volumes of textual data from digital media like newspapers and reports, essentially representing promising solutions for the extraction of the public's economic sentiments.

The Business Consumer Index (BCI) and Consumer Sentiment Index (CSI) are closely related to private consumption and private investment, as consumer confidence and business conditions directly influence these components (Juhro & Iyke, 2020; Ghosh, 2021; Khan & Upadhayaya, 2020; Qeqe & Sibanda, 2022). Optimism leads to increased spending and investment, while pessimism results in decreased economic activity. This strong interrelationship justifies the inclusion of private consumption and private investment in this study, as they are among the key indicators of economic health and drivers of GDP growth. Equally important to the analysis are imports and exports, as their integration provides a comprehensive understanding of the demand-side GDP. The dynamics of imports and exports critically reflect the country's trade balance, offering insights into its economic interactions with the global market. Including these components is crucial because they not only reflect the domestic economic landscape but also the country's global economic interactions. A trade surplus, where exports exceed imports, leads to currency appreciation, while a trade deficit, where imports exceed exports, results in currency depreciation (Lioudis, 2023). Understanding a country's trade activities provides valuable insights into its competitive position on the global stage, highlighting both its strengths and weaknesses. In short, the inclusion of private consumption, private investment, imports, and exports in this study allows for a holistic analysis of the Malaysian GDP.

Government spending being one of the components in the expenditure approach for calculating GDP will not be included as a focus here because of its dissimilar nature from the sentiment-driven market activities of the private sector, which will the primary focus of this study. That is, because the component of government spending is solely governed by fiscal policies and fiscal decisions, it is thereby less sensitive to the direct influence of the sentiments of both

consumers and businesses (The Economic Times, 2024). In essence, this distinct characteristic of the government spending component renders it less relevant when analysis is centered on evaluating the immediate impact of market sentiments on real economic activities within an economy, as is the case for this study. While the aim and objectives of this study will be discussed further in the following sections, it is worth noting that this study serves to be an extension of the work carried out by Chong et al. (2021), where several avenues for future research following the work by the authors will be addressed in this study. More in-depth discussion on the work done by Chong et al. (2021) will be addressed in *Section 2.4*. The subsequent subsections in the first chapter will cover the problem statements, project questions, aim and objectives, scope, and significance of the study. *CHAPTER 2* will provide a comprehensive literature review. *CHAPTER 3* will discuss the methodology employed in this study, followed by *CHAPTER 4*, which will document the actual implementation process. *CHAPTER 5* will present the study's results and analysis. Finally, *CHAPTER 6* will conclude the study with a discussion of the findings and overall conclusions.

1.1 Problem Statements

The nowcasting of macroeconomic components is crucial because key economic indicators are often released with a considerable delay (Aoki et al., 2023). For instance, GDP figures, which are important for the assessment of the economic health of one's country, are published once every quarter. While it is more than common to have long lags in the publication of key statistics in developing countries due to the shortage of infrastructure and resources for prompt data collection and processing (Ghosh & Ranjan, 2021), such lag is also prevalent in developed nations like the United States and the United Kingdom. In both the United States (Bureau of Economic Analysis [BEA], 2023.) and the United Kingdom (Office for National Statistics, n.d.), the formal estimates of their GDPs will only be published one month after a quarter ends. While the lag is considerably smaller in comparison to those of developing countries, it is nonetheless still a challenge for timely economic analysis and prompt decision-making. In Malaysia, GDP figures are released with a 90-day delay (OpenDOSM, 2024-a), once again underscoring the challenges of timely economic evaluation consequent to the publication lag that exists. Similarly, the official BCI and CSI figures as previously discussed in *Section 1.1* are only released at a quarterly basis (Malaysian Institute of Economic Research [MIER], 2023), suggesting the same challenges faced as with the Malaysian GDP estimates.

Even though the nowcasting of macroeconomic components has always been important due to the natural delay in the publication of key statistics, the COVID-19 pandemic as was previously mentioned in the above section, has evidently highlighted the critical importance of nowcasting across various decision-making levels of all economic stakeholders. The growing need for macroeconomic nowcasting has been particularly emphasized during the pandemic where economies all over the world were sent into an unprecedented stop as movement control orders within, and across borders were implemented in the attempt to curb the spread of the virus. It is then that governments and central banks realized the need for real-time or near-real-time data to formulate and implement effective policy responses to mitigate the economic turbulence consequent to the virus outbreak. While it is apparent how nowcasting gained its prominence in times of crisis, its contribution in ensuring continued, and proactive monitoring of the economy, as well as in decision-making during stable times must not be disregarded (Takahashi, 2022). In essence, the lag in the publication of important economic indicators coupled with the need for continuous pulse-checking of the economy's health are among the motivations driving this study.

Additionally, and as indicated in the works of Chong et al. (2021), even though there has been a significant body of research done on the capturing of sentiments from English-based news articles to predict macroeconomic outcomes, derivation of news sentiments from non-English-based languages still remained relatively underexplored. In the case of this study and given the multilingual and multiracial context of the Malaysian community, the incorporation of newspapers that are in Malay, Mandarin, and Tamil is thus crucial for a comprehensive sentiment analysis which could adequately reflect the national sentiment. While it would be ideal to have all three non-English languages coupled with the English language newspapers to be examined simultaneously, for a complete and comprehensive analysis of the sentiments of the Malaysian readers, constraints of the project in the form of time and resources do not allow for that. Hence, to accommodate for the constraints present, the focus of this study will be solely on news articles of the Mandarin language. Given the above discussions on the problems identified, these problems will then be directly translated to form the aim and objectives of this study.

Before that however, it is important to clarify the distinction between the concepts of "nowcasting" and that of "forecasting" in the context of this study. That is, as GDP data for the Malaysian economy is only available at a quarterly basis, this quarterly period would then serve

as the closest and shortest forecast interval for this project. That said, even though the term nowcasting is theoretically appropriate in this context as the term refers to the estimation of the present or very near-future economic conditions with the latest data as inputs for the estimation, this study would primarily adopt the term "forecasting' instead, throughout the study. The decision to do so is established on the fact that the quarterly interval of this study extends beyond the conventional nowcasting horizon of shorter periods, for instance, weeks or months within a quarter. Nevertheless, while the analysis of this study is framed as a forecasting task, it is worth noting that the core of this study still lies in illustrating the application of High-Frequency Indicators (HFIs) in the nowcasting of GDP, and its four other demand-side components previously mentioned in *Section 1.1,* namely private investment, private consumption, imports, and exports. This approach demonstrates the study's attempt to employ data that are generated at a high frequency – news sentiment, to provide insights into the economic conditions of the Malaysian economy, essentially unifying the fundamentals of nowcasting within a forecasting framework.

1.2 Project Questions

The questions that this study aims to address are outlined as follows:

- Does the news sentiment index which is computed in the study accurately reflect the BCI and CSI published by the MIER?
- How was the performance of the Mandarin text-derived news sentiment when compared to the English text-derived news sentiment in the work of Chong et al. (2021)?
- Which of the four demand-side components (private investment, private consumption, imports, and exports) of GDP exhibited a strong correlation with the newly constructed news sentiment index, and which of the components showed a weak correlation to the index?

1.3 Aim and Objectives of the Study

The aim of this study is to assess the role of text-derived news sentiment from Mandarin-based news articles, in the forecasting of the Malaysian GDP and its four demand-side components of the expenditure approach namely, private investment, private consumption, import and export, using machine learning techniques.

The objectives of this study are outlined as follows:

- To evaluate the ability of Mandarin-based news sentiment indices in predicting sentiments reflected by the BCI and CSI figures.
- To compare the performance of the Mandarin text-derived news sentiments to that of the English text-derived news sentiments reported in previous published work.
- To evaluate the correlation between the four demand-side components of GDP and the news sentiment index.

1.4 Scope of the Study

The boundaries of the study are divided into eight subsections namely, features of dataset, geographical region, domain of study, time frame, lexicon employed, selection of models, software and tools adopted, and finally, the evaluation metrics.

1.4.1 Features of Dataset

The dataset for deriving news sentiment will include only news articles from the Finance/Business (财经) section of See Hua Daily News (詩華日報) portal. Despite the identification of 7 other Mandarin news portals by Chow (2023)—namely Asia Times, China Press, Guangmin Daily, Kwong Wah Jit Poh, Nanyang Siang Pau, Overseas Chinese Daily News, and Sin Chew Jit Poh—these portals will not be included in this study due to significant data collection challenges. A detailed elaboration of these challenges will be provided in *Section 4.1.3*. While the preferred method is to select news portals with the highest readership to ensure better representation, only See Hua Daily News allows for data collection in this study. Therefore, the choice of news portal is based on data availability rather than readership due to the circumstances faced. Consequently, the number of news articles available for analysis is capped at 3361, significantly fewer than the initial target of 10,000 articles. Although it would have been ideal to match the number of articles used in the study by Chong et al. (2021) for fair and consistent comparisons, constraints on time and data availability limit this effort. For macroeconomic indicators, specifically GDP, private consumption, private investment, imports, and exports, data will be sourced from the OpenDOSM platform (an official platform of the Department of Statistics Malaysia (DOSM)). Survey-based sentiment indices of BCI and CSI will be obtained from the Malaysian Institute of Economic Research (MIER) through data purchase.

1.4.2 Geographical Region

This proposed study will focus exclusively on the Malaysian economy, examining it at the national level.

1.4.3 Domian of Study

This study will restrict its analysis exclusively to the subject area represented within the dataset, omitting any subjects that are not covered within this scope, thus focusing only on the topics covered by the articles in the dataset.

1.4.4 Time Frame

The sample period for this study is set from 2022 to 2023, during which news articles, macroeconomic indicators, and survey-based indices will be collected and analysed. Ideally, this study would adopt the same sampling period as Chong et al. (2021), which spans from the first quarter of 2006 to the second quarter of 2021. However, constraints on time and data availability, as previously mentioned, prevent this study from covering such an extensive timeframe.

1.4.5 Lexicon and Dictionaries Employed

The Mandarin financial sentiment lexicon developed by Jiang et al. (2019) will be adopted in this study for the derivation of Mandarin text-based news sentiment. The lexicon will be referred to as JMZ from here forward for simplicity's sake. Only one lexicon will be employed in this study as compared to the study of Chong et al. (2021) where three other English-based lexicons were used because the said lexicon of JMZ is deemed to be sufficient for the purpose of this study. This is due to the comprehensive development approach of the lexicon, which involved creating a newer and more detailed Chinese financial sentiment dictionary. That is, the dictionary was based on the Loughran and McDonald (2011) framework and further enhanced through manual screening, and the word2vec algorithm expansion. Additionally, the stop word dictionaries developed by Diaz & fseasy (2020) will be utilized in this study to filter out common and irrelevant words from the text data.

1.4.6 Selection of Models

The selection of models indicated in the table below is based on the literature reviewed in *CHAPTER 2*.

Models	Objective
Linear Regression applying JMZ Mandarin Lexicon	Forecasting BCI and CSI values using the news sentiment index computed.
Ordinary Least Squares with Autoregressive Order 1 [OLS-AR(1)]	A baseline model without incorporating the sentiment measure.
Ridge Regression	Forecasting GDP and its four demand-side components with the incorporation of the news sentiment index computed.
LASSO Regression	
Random Forest (RF)	
Extreme Gradient Boosting (XGBoost)	
Support Vector Machine (SVR)	

Table 1: Selection of Models in this Study

1.4.7 Software, Tools and Libraries Adopted

ParseHub (version 2.4.35) will be adopted in this study for scrapping news articles from the official website of the news portal chosen. ParseHub was chosen due to its intuitive, user-friendly interface, which does not require additional Python libraries such as BeautifulSoup or Selenium, thereby eliminating the need for writing complex scripts, and managing dependencies. Dependencies here refer to the various libraries and packages that Python scripts rely on to function correctly. Managing these dependencies often involves ensuring compatibility between different versions and resolving conflicts, which can be time-consuming and challenging. ParseHub's robust features, including point-and-click functionality and the ability to handle pagination and nested elements, ensure comprehensive data extraction without the need for extensive coding. Compared to Python-based scraping, which involves writing and debugging code and handling dependencies, ParseHub streamlines the process, allowing for faster setup and execution, making it a more efficient choice for this study. Additionally,

Visual Studio Code (version 1.91.1) will be used to execute the remaining parts of the project. The use of Visual Studio Code ensures a robust development environment with support for various extensions and tools that enhance coding efficiency and effectiveness. Its comprehensive debugging and integration capabilities make it ideal for the development and execution of the analytical and processing tasks required in this study. The Python libraries and their respective submodules that will be utilized in this study are detailed in the following tables.

Library	Functionality
pandas	Data manipulation and analysis
numpy	Numerical operations
statsmodels	Statistical models and tests
unicodedata	Unicode character database operations
jieba	Chinese text segmentation
re	Regular expression operations
json	JSON manipulation
matplotlib	Plotting and visualization
xgboost	XGBoost Regressor model
scikit-learn	Machine learning library for model evaluation, regression, feature scaling, hyperparameter tuning and creating pipelines.

Table 2: Primary Python Libraries Utilized in the Study

Library	Submodule	Functionality
sklearn	metrics	Evaluation metrics for machine learning models
sklearn	svm	Support Vector Regressor model
sklearn	ensemble	Random Forest Regressor model
sklearn	linear_model	Linear models (Linear Regression, LASSO, Ridge)
sklearn	preprocessing	Feature scaling
sklearn	model_selection	Hyperparameter tuning (GridSearchCV)
sklearn	pipeline	Creating machine learning pipelines
statsmodels	stats.outliers_influence	Calculating Variance Inflation Factor (VIF)

Table 3: Submodules of Python Libraries and Their Functionalities Utilized in the Study

1.4.8 Evaluation Metrics

The Pearson correlation coefficients as well as the statistical significance value of the news sentiment in the linear regression output will be used simultaneously to evaluate the ability of the news sentiment in nowcasting the two indices published by the MIER. For the evaluation of the forecasting ability of different machine learning models employed in this study, the ratio Root Mean Squared Error (RMSE) value is considered. It is defined as the non-baseline model's respective RMSE relative to that of the baseline model [OLS-AR(1) model]. A value less than "1" would be desirable as it implies that there is an improvement of the forecast in relation to the baseline model. In addition to the ratio RMSE, the ratio Mean Absolute Error (MAE) will also be used to evaluate the fitness of the model. The concept of the ratio MAE is similar to that of the ratio RMSE. Including both ratio RMSE and ratio MAE is relevant in this study because they provide complementary perspectives on model performance. Ratio RMSE is sensitive to larger errors due to the squaring of residuals, making it useful for identifying models with significant outliers. Ratio MAE, on the other hand, offers a straightforward interpretation of average error magnitude and is less influenced by outliers. Using both metrics ensures a comprehensive evaluation of model accuracy and robustness, ultimately leading to more reliable and nuanced insights into the forecasting ability of the models.

1.5 Significance of the Study

The contribution of this study manifests in two significant ways. Firstly, despite the growing literature on sentiment analysis using textual data, studies emphasizing on the use of vernacular languages in the said domain, particularly Mandarin in the context of this study, remains largely lacking as was underscored in *Section 1.1*. While exploration in this regard is sure to provide a more comprehensive understanding of the economic sentiment within the Malaysian community, exploration in this domain would also serve as a benchmark for other multiethnic and multilingual economies, essentially enriching the global discourse on economic sentiment analysis. Secondly, given the scarcity of studies done on the use of HFIs, like news sentiment in the forecasting of macroeconomic variables, especially in the case of Malaysia, this study would then serve to facilitate the augmentation of local academic resources in the said field. Besides that, as it is acknowledged that methodologies and conclusions derived from studies done on developed nations must not be indiscriminately applied to any other countries, which in this case refers to the Malaysian economy, due to the distinct economic, social and political landscape that exists, the contribution of this study once again becomes apparent for that reason.

In essence, this tailored approach to the refinement of forecasting abilities in the context of Malaysia, would significantly aid central bank and policymakers in devising more timely and precise policies, which in turn, could effectively promote economic growth and stability within the nation, making sure that policies are closely aligned with the unique dynamics of the Malaysian economy. On a separate note, it is worth pointing out that all two of the contributions discussed above directly address the research gaps identified in the LITERATURE REVIEW section (*CHAPTER 2*), which will be explored in greater detail in the following section.

CHAPTER 2

LITERATURE REVIEW

The caveat for this literature review section is that it does not intend to include every research paper that relates to the sentiment analysis of non-English languages. Instead, this literature review seeks to offer an overview of the general research conducted in this field, incorporating, to the best of the author's knowledge, the representative papers. Put simply, this section serves to lay the foundation for a better understanding of the studies' scopes, at the same time, highlight existing research gaps that this study intends to address. There are four parts to this literature review section. Firstly, it starts with a review of papers on the adoption of news sentiment for forecasting key macroeconomic indicators such as GDP growth, inflation, and unemployment. This is followed by a comprehensive evaluation of existing literature on the sentiment analysis of textual data in various vernacular languages, with a particular focus on economic and financial texts in Chinese. Thirdly, the review examines the application of machine learning models in sentiment analysis and their effectiveness in economic forecasting. Finally, the section concludes with an in-depth discussion on the work done by Chong et al. (2021), which is essential as this study aims to build upon their foundational work.

2.1 Forecasting Key Elements of Interest through Text-Derived News Sentiments

In recent years, the convergence of economic forecasting and sentiment analysis has gained sizable attention within the research community. In this subsection of the literature review, findings from several key studies on the integration of text-derived news sentiment into forecasting models for the prediction of macroeconomic indicators, namely GDP growth, inflation, and unemployment, will be explored in greater detail. Central to the discussion here is the identification of unexplored research areas that the following studies jointly emphasizes on, suggesting avenues for future work to be conducted.

Among the papers reviewed, a recurring theme observed is that there is a need for sentiment analysis models that are domain specific. For instance, both Ashwin et al. (2021) and Shapiro et al. (2022) in their work, pointed to the possible benefits of developing lexicons and sentiment-scoring models which are tailored to the economics domain, thereby suggesting that generic sentiment analysis tools might not sufficiently capture the subtleties of economic

discussions. This underscores a notable gap in existing literature where there is a need for more sophisticated and domain-specific sentiment analysis methodologies, so that sentiments from economic news articles could be more accurately interpreted and analysed. At the same time, several of the studies reviewed have also collectively underscored the significance of exploring how different big data measures, and textual data are influencing the accuracy of economic forecasts. In both their work, Ashwin et al. (2021) and Barbaglia et al. (2023) suggested that future research should include the comprehensive examination of a broader array of big data indicators and textual data that are beyond traditional news articles. It is also suggested that future research should include the sentiments that are expressed in multiple languages to ensure that the sentiments derived are representative and comprehensive. This represents a growing interest in the understanding of how diverse data types and sources can augment economic forecasts, more so during times of economic instability, like the COVID-19 pandemic and the Great Recession as pointed out by Ashwin et al. (2021) in their findings.

In the study of Kalamara et al. (2022) and Barbaglia et al. (2024) however, their studies emphasis on the need to examine the predictive ability of sentiment indicators across different economic landscapes, and different times of crises. That is, both groups of authors suggest that the utility of sentiment analysis may vary with the economic context of each country, thus emphasizing the importance of conducting comparative studies across the different economies. Once again, this presents itself as another research gap to be addressed, precisely by evaluating the generalizability and reliability of sentiment indicators in forecasting economic outcomes in different settings. The integration of sophisticated machine learning techniques with sentiment analysis for economic forecasting is yet another recurring subject. In their work, both Ashwin et al. (2021) and Kalamara et al. (2022) encourages future research to extend its exploration to more complex models which are beyond the scope of those conventional linear, and nonlinear models. While not explicitly mentioned, implications of their encouragement are that complex and more sophisticated machine learning techniques could potentially offer more nuanced insights into the forecasting ability of news sentiment, essentially presenting itself as yet another research gap to be addressed. Finally, and according to Thorsrud (2016), and Ashwin et al. (2021), the integration of sentiment analysis with other economic indicators could perhaps open doors to predictive models that are more robust. This points to a broader research gap where integration of sentiment indicators with traditional economic data are called to take place. Adopting such an approach could potentially yield a more holistic view of economic trends, as

the immediate nature of news sentiments are leveraged upon to complement the delays that are inherent to traditional economic data.

In essence, the literature discussed reveals a unified effort in understanding the role of sentiment analysis in the forecasting of economic variables, which at the same time, serves to lay the groundwork for future research. The identified gaps, precisely the need for a more domain-specific models and the exploration of diverse data sources, the incorporation of advanced machine learning techniques, and the suggested integration of sentiment analysis with traditional economic metrics, collectively represents a roadmap for enhancing the accuracy, reliability, and relevancy of sentiment-influence economic forecasts.

2.2 Multilingual Sentiment Analysis with Emphasis on Economic and Financial Texts in Chinese

In the growing field of sentiment analysis, a varied array of languages has been considered, each exhibiting its own set of challenges and opportunities. In this section of the literature review, the aim is to integrate the findings of the work done on sentiment analysis across different languages, ultimately directing focus to the exploration of the Chinese language, especially in relation to the financial and economic contexts. The discussion is structured to transition seamlessly across different linguistic studies, to illustrate the consistency in the overall theme, and the literature gaps that exist across the reviewed papers, essentially underscoring the novelty and significance of this proposed study.

Notable contributions in languages in terms of sentiment analysis includes those of the Algerian dialect, Dravidian languages (Tamil and Tulu), Roman Urdu, Hindu, Malay, and Emirati and Tunisian dialects, as will be demonstrated in the research works to be discussed hereon. In their work of studying and comparing different approaches for the sentiment analysis of textual data of the Algerian dialect, both Moudjari & Akli-Astouati (2020) have underscored the necessity for more resources and tools to be introduced and implemented for the Algerian dialect. Kogilavani Shanmugavadivel et al. (2022) and Hegde et al. (2023) on the other hand, explored the Dravidian languages, encouraging future work to address the unique challenges posed by code-mixing in the sentiment analysis domain. In the analysis of textual data in Roman Urdu (Rana et al. 2021), Hindi (Gupta et al., 2021), and Malay (Ong et al. 2020; Muhammad Fakhrur Razi Abu Bakar et al., 2020) languages, all authors, in their respective language-specific

domain, has rallied together to call for an expansion of boundaries towards the exploration of more sophisticated deep learning methods, and the development of more robust datasets. Additionally, in the studies into the Arabic dialects, as documented by Abir Masmoudi et al. (2021) and Al Shamsi & Sherief Abdallah (2022), both groups of authors have underscored the need for a more advanced Natural Language Processing (NLP) approach in sentiment analysis, coupled with a more diverse, and comprehensive dataset for the analysis.

As discussion transitions to the evaluation of sentiment analysis within the context of the Chinese language, the landscape reveals a wide array of research focusing on the different applications of textual data in Chinese, ranging from social media platforms such as Weibo to product reviews on e-commerce platforms. While the aim of the various research reviewed remained relatively similar in nature, different authors do have different emphasis on the research gaps to be addressed. Wang and Alfred (2020) together with Wei et al. (2022) for instance, has underscored the need to have advanced sentiment analysis models that are both culturally sensitive and capable of real-time analysis, more so for platforms like Weibo. Contrary to that, the work of Tan and Zhang (2008), Zhang et al. (2018), and Yang et al. (2020), places emphasis on the significance of sentiment classifiers that are domain independent, as well as the need for the refinement of sentiment classification beyond the binary categories. Despite the breadth of studies on the sentiment analysis of textual data of the Chinese language, a notable gap is apparent where exploration of the financial and economic aspects is largely lacking. While this gap was singularly addressed by Huang et al. (2020), who have considered the financial markets in the Chinese economy, hence the Chinese textual data, this singular approach does at the same time, underscores the stark scarcity of research in the financial and economics domains, emphasizing the pressing need for further exploration.

In summary, the above narrative of sentiment analysis research reveals that while there is indeed an extensive engagement of sentiment analysis across different languages, there is however, also a pronounced underrepresentation, in terms of the context of finance and economics, within the Chinese sentiment analysis literature. Given that, this gap that exists then pave the way for the current proposed study to take place, and to address these overlooked areas. That is, by narrowing down the focus to the financial and economic implications of sentiment analysis within the context of the Chinese language, this study does not only address the gap in the current literature landscape, but also open up doors for a deeper understanding of the intricate interplay between sentiment in the Chinese language, and the economic activity

within the Malaysian community. In other words, the relevance and importance of this study must not be seen as merely academic but should also be seen as having significant implications for policymakers to make informed decisions with the added findings of the study.

2.3 Review of Machine Learning Models for Sentiment Analysis

The accurate forecasting and nowcasting of macroeconomic variables are crucial for policymakers, economists, and financial institutions. Recent years have seen growing interest in applying machine learning techniques to this challenge, aiming to improve upon traditional econometric methods. This subsection of literature review examines key studies exploring the application of machine learning models to GDP forecasting and nowcasting across various economies, including both developed and emerging markets. Chu and Qureshi (2023) conducted a comprehensive study on forecasting U.S. GDP growth using various machine learning methods. Their findings indicate that density-based methods such as bagging, boosting, and neural networks outperform sparsity-based methods like LASSO for short-horizon forecasts. However, this performance difference diminishes for longer-horizon forecasts, highlighting the importance of considering forecast horizons when selecting models. In the context of emerging economies, Zhang et al. (2023) examined machine learning algorithms for nowcasting Chinese GDP. Their research is notable for comparing model performance during both normal economic conditions, and crisis periods. Ridge Regression consistently outperformed other methods, including dynamic factor models, underscoring the importance of regularization in handling high-dimensional datasets, and suggesting that simpler models can sometimes yield superior results, especially in capturing economic turning points.

Richardson et al. (2021) conducted a real-time assessment of machine learning algorithms for nowcasting New Zealand's GDP, using real-time vintages of GDP data, and a large set of predictor variables. They found that machine learning models, particularly Support Vector Machines and boosted trees, significantly outperformed traditional benchmarks such as autoregressive models. This research emphasizes the potential of machine learning in handling large datasets and capturing complex relationships between economic variables. Focusing on developing economies, Muchisha et al. (2021) evaluated various machine learning algorithms in nowcasting Indonesia's GDP growth. Random Forest performed best among individual models, while a combination of models using LASSO regression yielded the most accurate forecasts. This study highlights the potential benefits of model averaging, and the importance

of considering computational efficiency in model selection. Besides that, several studies have explored the use of novel data sources, and high-frequency indicators in GDP forecasting. Nakazawa (2022) investigated alternative data sources for Japan, incorporating high-frequency data from point-of-sale systems, and mobile phone location data. The results demonstrated that these novel data sources can significantly improve nowcasting accuracy, particularly in capturing sudden economic changes.

Dauphin et al. (2022) proposed a scalable approach to nowcasting GDP for European economies using a combination of dynamic factor models, machine learning techniques, and novel data sources. Their research found that machine learning models, particularly tree-based methods like Random Forest and XGBoost, often outperformed traditional approaches. However, model performance varied across countries and economic conditions, emphasizing the need for careful model selection. Agu et al. (2022) applied machine learning methods to predict GDP using macroeconomic indicators for Nigeria. They found that Principal Component Regression (PCR) achieved the highest predictive accuracy, followed closely by Ridge Regression. This study highlights the potential of dimensionality reduction techniques in handling high-dimensional macroeconomic datasets. Kant et al. (2022) compared various nowcasting models for the Netherlands economy, finding that Random Forest algorithms stood out in terms of performance. Their study emphasized the importance of feature selection and handling mixed-frequency data, which are common challenges in macroeconomic forecasting.

Martin (2019) evaluated machine learning techniques for nowcasting South African GDP. Their results showed that tree-based models, particularly Random Forest and Gradient Boosting, performed exceptionally well compared to traditional methods. This finding aligns with the broader trend observed across multiple studies, suggesting the robustness of tree-based methods in handling economic data. Jasni et al. (2022) applied machine learning algorithms to nowcast Malaysia's GDP, focusing on performance during normal and crisis times. Their study found that XGBoost and Random Forest models performed particularly well, especially in identifying turning points in the economy. This highlights the potential of machine learning methods in capturing non-linear relationships and adapting to changing economic conditions. Chong et al. (2021) explored the use of news sentiment for economic forecasting in Malaysia. They extracted sentiment from business and financial news articles using dictionary-based methods and evaluated its relationship with existing survey-based sentiment measures and macroeconomic growth outcomes. Their findings showed that news sentiment could nowcast

survey-based business sentiment measures and had reliable predictive ability for private investment growth within a two to three-quarter forecast horizon. This study underscores the potential of incorporating alternative data sources, such as text-based sentiment, into macroeconomic forecasting models.

Given the discussion thus far, a common theme across these studies is the superior performance of ensemble methods and model combinations. Many researchers found that averaging predictions from multiple models or using techniques like Random Forest and Gradient Boosting often yielded more accurate forecasts than individual models. This aligns with the broader machine learning literature, which suggests that ensemble methods can help reduce overfitting and improve generalization. However, the application of machine learning to GDP forecasting is not without challenges. Issues related to data quality and availability, particularly for emerging economies, are prevalent. Handling mixed-frequency data and accounting for publication lags remain significantly challenging as well. Additionally, while machine learning models often demonstrated superior predictive performance, many researchers noted the trade-off between model complexity and interpretability, a crucial consideration for policymakers who need to understand the drivers behind economic forecasts.

To summarize, based on the reviewed studies, several models consistently demonstrate superior performance across different economic contexts and datasets. Random Forest and Gradient Boosting Machine (including XGBoost) emerge as top performers, showing robust predictive power and the ability to handle complex, high-dimensional data. These tree-based ensemble methods appear particularly effective in capturing non-linear relationships and adapting to changing economic conditions. Regularized regression methods, especially Ridge Regression and LASSO, also show promising results, particularly in handling high-dimensional datasets and mitigating overfitting. Support Vector Machines demonstrated strong performance in some studies, while neural network models, including Long Short-Term Memory (LSTM) networks, show potential in capturing complex temporal dependencies, though their performance was less consistent across studies.

Given the insights derived from the literature review, this thesis will employ a comprehensive suite of machine learning models, including LASSO Regression, Ridge Regression, XGBoost, Random Forest, and Support Vector Machine (SVR). Although the dataset for this study is constrained to two variables and comprises only of 8 quarterly observations of averaged news

sentiment from a single Mandarin news portal, the implementation of these diverse models serves multiple purposes. Firstly, it provides a thorough exploration of various machine learning techniques in the context of limited data. Secondly, it establishes a methodological framework that can be expanded upon in future research. While some of these models may exceed the complexity required for the current dataset, their inclusion allows for a robust comparative analysis and sets a precedent for future studies with more extensive data. It is noteworthy that Long Short-Term Memory (LSTM) networks, despite their demonstrated efficacy in some studies, will not be incorporated due to the limited size of the available dataset, which is insufficient for the optimal performance of such deep learning models. Subsequent research endeavours could potentially investigate the application of LSTM and other advanced neural network architectures when more substantial datasets become accessible.

2.4 Chong et al. (2021) and News Sentiment Analysis for Macroeconomic Forecasting

While the previous subsection provided an overview of machine learning models for sentiment analysis in macroeconomic forecasting, it is crucial to examine in greater detail the work of Chong et al. (2021), which serves as a benchmark for this thesis project as was mentioned in *CHAPTER 1*. Their study is particularly significant due to the scarcity of published research on using machine learning to analyse Mandarin news sentiment and the limited exploration of news sentiment in forecasting macroeconomic variables other than GDPs, through machine learning techniques. This lack of extensive literature underscores the importance of the authors' work as a foundation for further research in this area, including this present thesis. Chong et al. (2021) focused on using news sentiment for economic forecasting in Malaysia, an approach that is highly relevant to the current project's objectives.

A systematic approach was employed by the authors in working towards their aim. The methodology began with the compilation of over 720 thousand daily news articles from a total of 16 English-based online news portals in Malaysia. These articles, covering the period between 2006 and June 2021, were taken primarily from the business and financial sections to ensure that they are applicable and relevant for their study. The authors utilized three different lexicons for sentiment scoring namely, the financial stability dictionary by Correa et al. (2017), the finance-oriented dictionary by Loughran & McDonald (2011), and a lexicon created by Shapiro et al. (2022) specifically for economic news articles. This multi-lexicon approach provides a robust foundation for sentiment analysis, accounting for different nuances in

financial and economic language. Following the collection of the news articles, the authors proceeded to perform text preprocessing on the articles. The process includes the deletion of punctuations, hyperlinks, Hypertext Markup Language (HTML) tags, any uncommon characters, and excess white spaces. Besides that, conventional stop words according to those listed by Nothman et al. (2019) were omitted from the articles too. Lastly, all wordings within each article were converted to lowercases to prepare them for the next step in the analysis. Stemming and lemmatisation were not performed during the cleaning process because the lexicons employed in their study have had stem words and their inflections included already. Once the text has been preprocessed, the authors then moved on to the nowcasting of the survey-based sentiment measures, represented by the BCI and CSI measures, using the news sentiment that they have since created. To prevent redundancies and repetition, the specifics of the computation of the sentiment indices will be discussed further under the methodology section below (*Section 3.3.1*).

In the last stage of the methodology, forecasting was conducted for the GDP and its four other components which were all on a year-on-year basis. Linear and non-linear machine learning models were employed during the forecasting activity, with the news sentiment acting as the predictor variable. The linear machine learning models consisted of OLS Regression, Ridge Regression, and Huber Regression, while the non-linear machine learning models consisted of LightGBM, RF, ET, Orthogonal Matching Pursuit, Gradient Boosting, DT, AdaBoost, and LSTM neural network model. For each model, except for the LSTM neural network model, there will be three variations corresponding to the three sentiment measures used in the forecasting exercise. For the LSTM model, the authors included 21 different combinations of the independent variables in the forecasting of the five target variables previously mentioned. Besides that, a baseline model without the incorporation of any sentiment measures was employed for evaluation purposes. This baseline model is referred to as the OLS-AR(1), and it is the same baseline model as the one mentioned in *Section 1.4.6*. Again, the details of how forecasting was conducted will be further discussed in the methodology section (*Section 3.4.2*) below.

Evaluation was divided in two separate parts, the first being the evaluation of the three news sentiments on their ability to nowcast the BCI and CSI measures, as well as their contemporaneous correlations with the five target variables mentioned. To check for their contemporaneous correlations, the Pearson correlation coefficients were considered. Besides

that, to infer about the nowcasting ability of the news sentiment, once again in relation to the BCI and CSI measures, the statistical significance of each of the news sentiment measures when linear regressions were carried out served as another measure of the nowcasting ability of the three sentiment measures. The second part of the evaluation, on the other hand, pertains to the forecasting ability of the news sentiment when different machine learning models are employed. That is, for all of the models employed, a ratio RMSE value will be computed for every single one of the models. The ratio RMSE value referred to here is the same as the one mentioned in *Section 1.4.8*.

Their findings revealed that news sentiment, especially those derived from business and financial news articles were found to correlate well with the BCI sentiment measures published by MIER. Besides, the monthly news sentiment was proven to be a leading indicator of investment activities, especially in terms of private investment, providing foresight into their developments up to two to three quarters in advance. Interestingly, they found that the predictive power of news sentiment for private investment persisted even during periods of macroeconomic stress, such as the COVID-19 pandemic, demonstrating the robustness of this approach in varying economic conditions. Nonetheless, the study also found that improvement in the forecasting of the rest of the components were very minimal.

Chong et al.'s work also addresses several challenges inherent in using news sentiment for economic forecasting. These include handling high-dimensional data, dealing with the time-varying nature of economic relationships, and balancing model complexity with interpretability. Their approach to these challenges, including their methods for feature selection and model evaluation, offers valuable guidance for similar research endeavours. Future work suggested by the study includes the exploration of text-derived news sentiment from non-English based (Malay, Chinese or Tamil) newspapers, for reason of better reflecting Malaysia's ethnically and linguistically diverse society. The authors' acknowledgment of these limitations provides a clear direction for the current thesis project, which aims to extend this line of research to Mandarin news sources. Besides that, Chong et al. (2021) suggested that news sentiment should extend beyond the business and financial domain. Lastly, as part of the study’s limitation, it is agreed among the authors that more granular dataset, coupled with the application of non-linear machine learning models could potentially refine the study's forecast precision. In short, and to sum up this subsection, the purpose of reviewing the work of Chong et al. (2021) in greater

detail was so that it could serve not only as a methodological guide but also as a point of comparison for the results of the current thesis.

CHAPTER 3

PROJECT METHODOLODY

This section addresses the methodology adopted for the proposed study, which is the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. There are six stages to the framework, starting with Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and lastly, Deployment (Tripathi et al., 2021). Beginning with the first stage of Business Understanding, researchers are called to clearly define the aims and objectives of their research, and to identify the specific requirements of their work. This knowledge would then be translated into data mining problems and would form the basis of the research plan to be executed. Following this, data will then be collected, and subsequently explored to uncover any underlying patterns or issues that are inherent to the dataset. This stage of Data Understanding is essential as it ensures that data quality and integrity will be upheld for further analysis to take place.

In Stage 3, the preliminary raw data will be preprocessed to ensure that it is ready for the Modelling stage. This preprocessing often includes the cleansing, selecting, and transforming of the data to address any issues which were identified previously at Stage 2. Once data has been preprocessed, the researchers would then proceed to implement the chosen modelling approaches, where optimizations through the tuning of the models' respective parameters would take place at the same time. Proceeding to the next stage of Evaluation, it involves the assessment of the models previously created, to ensure that they are fit in achieving the objectives identified at the beginning of the framework. Lastly, at Stage 6 of Deployment, it involves integrating the data mining solution into the organization's workflow, marking the completion of the CRISP-DM process. Considering that this study will not include the deployment of the finalized models created, the last stage of Deployment in the framework will hence be excluded from the discussion. Detailed discussions of each stage as they apply to this study are provided in the following subsections.

Before proceeding with the discussion below, it is important to first note the distinction between the intended project methodology detailed in *CHAPTER 3* and the actual implementation detailed in *CHAPTER 4*. The distinction between both chapters lies in their focus and content. *CHAPTER 3* provides a general framework for the project execution,

detailing the planned methodology and approach. It sets the stage for how the project is intended to proceed, including objectives, strategies for data collection, and the tools and techniques anticipated for use. In contrast, *CHAPTER 4* documents the real and exact scenarios encountered during the project execution. It captures the practical implementation of the methodology, including any deviations from the initial plan, challenges faced, and adjustments made in response to actual constraints and data availability. The chapter provides a detailed account of the actual processes and outcomes, offering insights into the practical aspects of executing the project methodology.

3.1 Business Understanding

The task of business understanding has been addressed in the sections above, precisely those under *CHAPTER 1*.

3.2 Data Understanding

In this stage, 10 thousand news articles published from the beginning of 2022 to the end of 2023, from the selection of the 8 news portals mentioned in *Section 1.5.1*, will be scraped, and gathered using ParseHub. The dataset will be explained in terms of its structure, content, and any initial observations about data quality observed during the preliminary exploration. Additionally, data on the five macroeconomic indicators—GDP, private investment, private consumption, imports, and exports, covering the same time period will be sourced from the official platform of the Department of Statistics Malaysia (OpenDOSM) and subjected to preliminary exploration. Since these five target variables are to be forecasted on a quarterly basis, the data for these variables will also be organized on a quarterly basis. Data on the Business Condition Index (BCI) and Consumer Sentiment Index (CSI) will be sourced from the Malaysian Institute of Economic Research (MIER) through data purchase, also at a quarterly frequency, and will undergo preliminary data exploration. Exploration of all datasets will be conducted in Visual Studio Code. During the exploration process, problematic entries will be identified and noted to facilitate necessary preprocessing steps.

3.3 Data Preparation

The data preparation stage is divided into two subsections, aiming to individually tackle the preprocessing required for the textual and numerical datasets.

3.3.1 Textual Dataset

The textual data preprocessing tasks begin with cleaning the data entries by removing missing entries and correcting or eliminating any wrongly formatted entries. Following this, segmentation will be performed on each article using the Visual Studio Code environment and the Jieba library to separate the text data into individual Mandarin characters. This segmented text will then be saved in a structured format for subsequent analysis. In the subsequent preprocessing stage, stop words will be filtered out in accordance with the list developed by Diaz and fseasy (2020). Simultaneously, punctuation marks, special characters, hyperlinks, HTML tags, and extra white spaces will be removed from the dataset. Once cleaned and segmented, sentiment analysis will be performed on the dataset using the JMZ lexicon to evaluate the sentiment of the articles. Besides that, since the JMZ lexicon includes the two major negators, namely 不 (bu) and 没 (mei) (Xiao & McEnery, 2008), and has categorized them within their respective sentiment classes (positive or negative), the task of handling negation will not be required.

To compute the news sentiment index for this study, each of the 10 thousand articles will first be organized according to their respective publication dates. The sentiment score for each article will then be computed, defined as the net number of positive words minus negative words relative to the total word count in each article. The formula for this computation is shown below, where *k* represents the individual articles. To normalize the sentiment scores to represent a net count per thousand words and transform them into an index format, the net number of positive words minus negative words, relative to the total word count, is multiplied by 1000. Values above 100 imply more positive sentiments, while values below 100 suggest otherwise. Given the variation in the number of articles across different news portals and time periods, all previously computed indices will be scaled by the total number of articles from the respective news portals and time periods. This process will create a news sentiment time series for each portal. Consequently, the number of individual sentiment indices created will correspond to the number of news portals scraped in this study. These individual indices will then be averaged across all the news portals, resulting in a single, aggregated news sentiment index at the chosen frequency (quarterly).

$$sentiment\ index_k = 100 + \frac{\Sigma Positive_k - \Sigma Negative_k}{Total\ Word\ Count_k} \times 1000$$

3.3.2 Numerical Macroeconomics Dataset

For the numerical macroeconomics datasets, preprocessing steps include normalization, imputation should there be any missing values, and the detection and treatment of outliers.

3.4 Modelling

The Modelling stage will be segmented into two different parts, starting with the nowcasting of the BCI and CSI values using the news sentiment index of the study, followed by the forecasting of the five macroeconomics indicators using machine learning models.

3.4.1 Nowcasting the BCI and CSI Values

In the nowcasting activity for the BCI and CSI values using the news sentiment index of this study, the aim is to investigate if the sentiment index on a quarterly basis, is an adequate indicator of the current quarter's BCI and CSI values. Estimates are done for each quarter as denoted in the equations below. Estimates for both equations will be done where *t* represents the quarter to be nowcasted, and $t-1$ represents the lag values of the respective BCI and CSI indices, with s_t representing the sentiment index. The nowcasting activity will be performed using the Linear Regression model.

$$BCI_t = \alpha + \beta\, BCI_{t-1} + n\, s_t + \varepsilon_t$$

$$CSI_t = \alpha + \beta\, CSI_{t-1} + n\, s_t + \varepsilon_t$$

3.4.2 Forecasting the Five Target Variables Using Machine Learning Models

For the forecasting of the five target variables namely, the quarter-on-quarter GDP, private investment, private consumption, imports, and exports, all machine learning models mentioned in *Section 1.4.6* will be employed. The forecasting process involves fitting the models within a defined training period using the sentiment index in a sequential manner. The rolling window approach is adopted in the process for the training of the models, enabling predictions of the target variables to be made for one, two, and three quarters into the future. In essence, the forecasting process serves to mimic real-world scenarios, where policymakers have access to historical economic data, (for instance y_{t-1}, y_{t-2},…) and news sentiment at that time x_t to forecast future economic outcome y_{t-1+h}, where *h* denotes the number of periods (in quarters) ahead of the current time *t*. That is, referring to *Figure 1* below, for each forecast, the model is trained on a fixed window of four most recent quarters. For instance, to predict for Q1 2023, the model

takes data from all four quarters of 2022. With each new forecast, the training window will move forward by one quarter, maintaining a fixed window size, generating forecasts sequentially, with the training set shifting accordingly to include the latest available data while excluding the oldest.

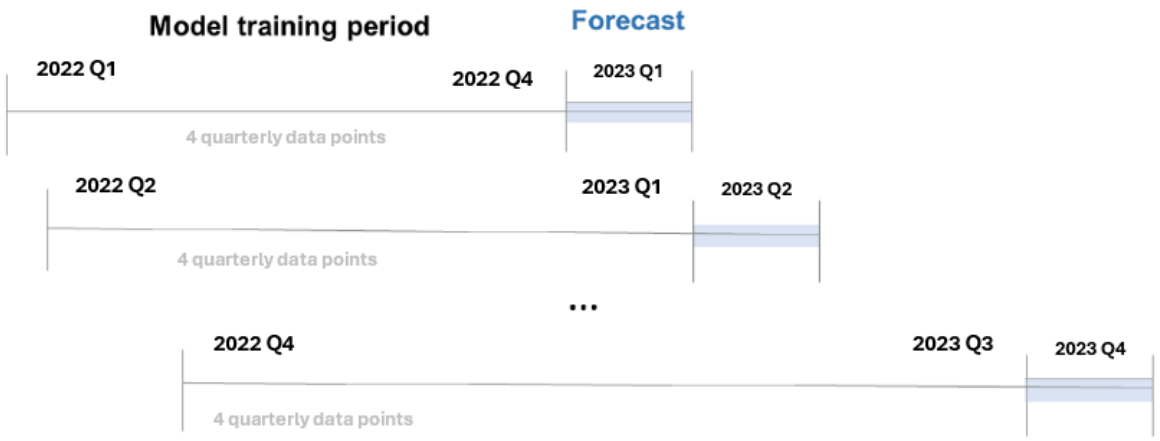


Figure 1: Rolling Window Forecasting Timeline with Quarterly Model Training Periods

3.5 Evaluation

Similarly to the Modelling stage, this stage of Evaluation will divide its discussion into two segments, corresponding to the two modelling approaches mentioned above.

3.5.1 Evaluation of the Nowcasting Activity

As mentioned above in *Section 1.4.8*, the news sentiment's effectiveness in nowcasting the BCI and CSI values will be examined using the Pearson Correlation Coefficients, and the significance value of the sentiment index as shown in the results of the Linear Regression model. For the Pearson Correlation Coefficients measure, a high and positive correlation value between the target variables and the news sentiment index is desirable. As for the statistical significance evaluation measure, it is desirable when the news sentiment index is found to be statistically significant in predicting the BCI and CSI values.

3.5.2 Evaluation of the Forecasting Activity

To assess the predictive performance of the various machine learning models employed in this study, the ratio RMSE and ratio MAE will be employed, as was mentioned in the earlier section.

A ratio below 1 would indicate a forecast improvement over the baseline, signifying a more accurate model, while a ratio above 1 would indicate the opposite.

CHAPTER 4

IMPLEMENTATION

The discussion on the implementation process will follow the stages of the CRISP-DM framework and the processes outlined in *CHAPTER 3*. The complete coding scripts and their respective outputs will be provided in the appendix. For each subsection, readers will be directed to the corresponding appendix if they need to refer to the coding and output snippets. Only the outputs will be presented in the body of *CHAPTER 4*; coding snippets will not be included in the body of this section.

4.1 Data Collection Process

This section is divided into three parts, namely MIER Dataset, Macroeconomics Dataset (DOSM), and News Articles (Web Scraping). Each part details the source and collection methods for the respective datasets used in this research.

4.1.1 MIER Dataset

Survey-based sentiment indices from the Malaysian Institute of Economic Research (MIER) for the years 2022 and 2023, on a quarterly basis were obtained through a data purchase from the institution. The data is stored in an Excel file named "*Data BCI_CSI_2022_2023.xlsx*". The proof of payment for the data purchase is included in *Appendix A*.

4.1.2 Macroeconomics Dataset (DOSM)

Macroeconomic data for imports, exports, private investment, private consumption, and GDP for the years 2022 and 2023 on a quarterly basis were obtained from OpenDOSM (2024-b). As outlined in the PDF document downloaded from the reference provided, the Private Final Consumption Expenditure is a measure of private consumption, while the Gross Fixed Capital Formation (Private Sector) represents private investment. The data provided reflects the Type of Expenditure at Constant 2015 Prices, expressed as the percentage change from the corresponding quarter of the preceding year. This adjustment ensures that seasonality and inflation have already been accounted for, making the data ready for immediate use. For details on the data for private consumption, private investment, imports, and exports, please refer to

Table 9B (*Page 57* of the PDF document). Data for the YoY GDP can be found on *Page xiii* of the same document. Note that all values represent the Year-over-Year (YoY) growth rates.

4.1.3 News Articles (Web Scrapping)

While the target number of articles to be scraped was set at 10,000, only 3,361 articles were successfully scraped from See Hua Daily News. The primary reason is that Asia Times, Guangmin Daily, and Overseas Chinese Daily News do not have a dedicated finance/business section, resulting in all articles, regardless of category, being aggregated together. Hence, these news portals were ruled out as sources for scraping. Additionally, Kwong Wah Jit Poh, Nanyang Siang Pau, Sin Chew Jit Poh, and China Press did not provide comparable coverage for the 2022 to 2023 period on their public platforms. These publications operate in a manner where newer articles replace older ones, making older articles no longer accessible on the website.

Efforts to expand the data sources through direct requests to news organizations or attempts to purchase archived articles were unsuccessful, despite offering to provide Research Letters from the university to validate the legitimacy of the data request. The Research Letters can be found in *Appendix B.* Although China Press offered the option to purchase individual articles, the associated costs were prohibitive for this research project. The estimation of approximately 3,000 articles for a full two-year period was based on the actual number of articles scraped from See Hua Daily News, which totalled to 3,361 articles. Using this as a benchmark, and considering the price of RM10 per article, the total cost for acquiring a similar dataset from China Press would have been approximately RM30,000, which significantly exceeded the available budget for this research.

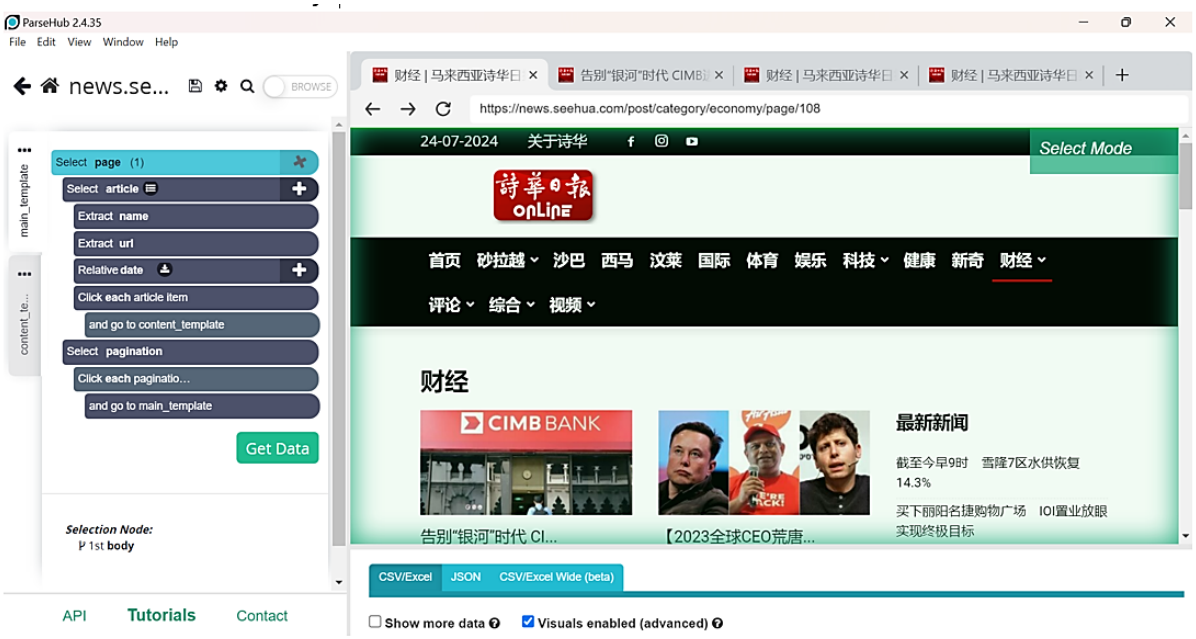


Figure 2: Initial Setup in ParseHub for Scraping Articles from See Hua Daily News

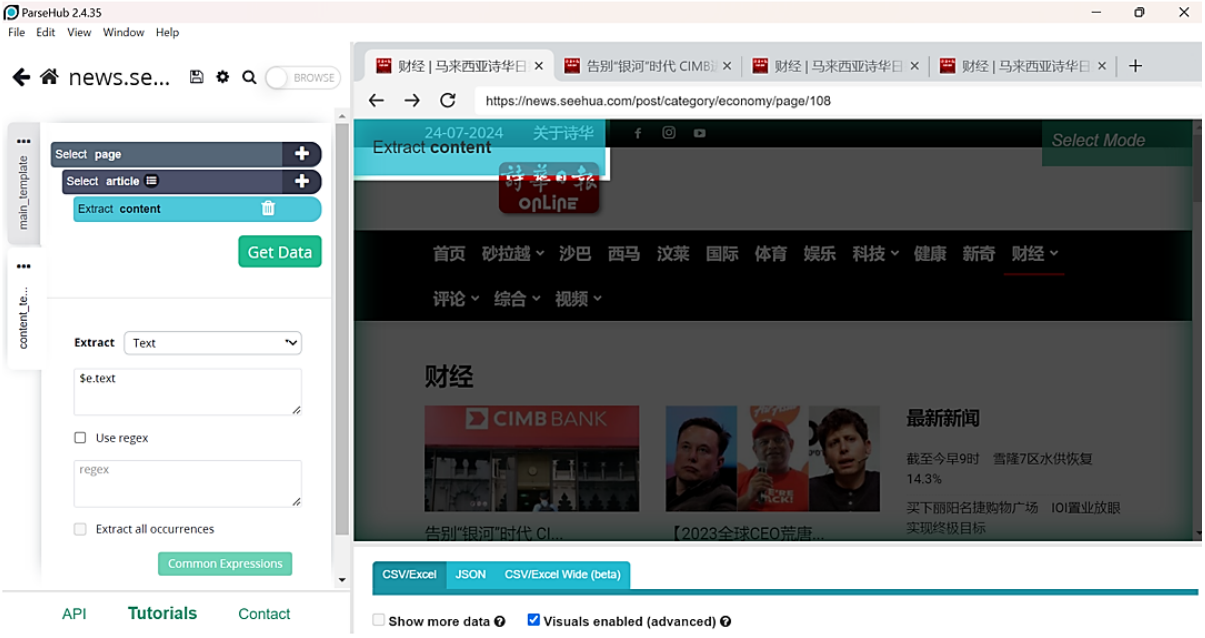


Figure 3: Configuring Content Extraction in ParseHub for See Hua Daily News

To carry out the scraping, ParseHub, a web scraping tool, was used to collect the news articles from the See Hua Daily News website. The two figures above provide a snapshot of the scraping setup process. For a detailed, step-by-step guide of the process, please refer to *Appendix C*. The scraped data is stored in a file named "*Scraped (See Hua).csv*".

4.2 Initialization of the Data Analysis Process

In this initial step of the data analysis process, the necessary libraries for data manipulation, statistical modeling, text processing, plotting, and machine learning were imported. Refer to *Appendix D* for the code snippet.

4.3 Initial Exploratory Data Analysis of the Textual Dataset

Refer to *Appendix E* for the complete code and output snippets.

4.3.1 Loading the Data

The dataset was initially read from the CSV file named "*Scrapped (See Hua).csv*", which contains the scraped news articles. Various encodings were attempted ('utf-8', 'gb18030', 'big5') to ensure proper reading of the file. The successful encoding used was 'utf-8'.

4.3.2 Initial Exploration of the Structure of the Dataset and Preliminary Cleaning of the Dataset

The first few rows of the dataset were displayed to understand its structure and content, revealing columns for *article_name*, *article_url*, *article_date*, and *article_article_content*.

```
First few rows of the dataset:
                article_name                          article_url  \
0         全球富豪身价狂涨 中国富人财产反而缩水  https://news.seehua.com/post/787364
1       开市即迎来套利压力  马股早盘跌18.85点  https://news.seehua.com/post/788101
2  2022首个交易日出师不利  马股全天下滑18.48点  https://news.seehua.com/post/788243
3      亚航拟改名为CAPITAL A BERHAD  https://news.seehua.com/post/788313
4      与大市背道而驰  富时大马综合指数跌9.52点  https://news.seehua.com/post/788684

  article_date                          article_article_content
0    2022年1月1日  这2年在新冠病毒肆虐期间，全球富豪资产大幅增长，不过中国科技富豪资产大失血，根据彭博亿万富豪...
1    2022年1月3日  （吉隆坡3日讯）2022年首个交易日，亚洲股市开盘表现普遍平平无奇，唯独马股出师不利，上周五...
2    2022年1月3日  （吉隆坡3日讯）由于区域多个股市仍未开市，导致亚洲市场淡静。马股首个交易日出师不利，全天下滑...
3    2022年1月3日  （吉隆坡3日讯）亚洲航空（AirAsia，5099，主板消费股）董事局建议将公司的名字改为C...
4    2022年1月4日  （吉隆坡4日讯）美国隔夜股市上涨，带动亚洲股市周二升多跌少。不过，马股仍延续周一跌势，与大市...
```

Figure 4: First Few Rows of the Textual Dataset

The *article_url* column was dropped as it was not necessary for further analysis. The columns were then renamed for clarity (*article_name* to *Title*, *article_date* to *Date*, and *article_article_content* to *Content*), and the order was rearranged to improve readability and structure. The cleaned DataFrame was saved to a new CSV file named "*See Hua (New).csv*" with 'utf-8' encoding, which was subsequently read again to ensure correctness.

4.3.3 Exploration of the Structure of the Dataset After Preliminary Cleaning of the Dataset

```
First few rows of the dataset:
        Date                          Title  \
0  2022年1月1日           全球富豪身价狂涨 中国富人财产反而缩水
1  2022年1月3日         开市即迎来套利压力  马股早盘跌18.85点
2  2022年1月3日  2022首个交易日出师不利  马股全天下滑18.48点
3  2022年1月3日       亚航拟改名为CAPITAL A BERHAD
4  2022年1月4日       与大市背道而驰  富时大马综合指数跌9.52点

                                             Content
0  这2年在新冠病毒肆虐期间，全球富豪资产大幅增长，不过中国科技富豪资产大失血，根据彭博亿万富豪...
1  （吉隆坡3日讯）2022年首个交易日，亚洲股市开盘表现普遍平平无奇，唯独马股出师不利，上周五...
2  （吉隆坡3日讯）由于区域多个股市仍未开市，导致亚洲市场淡静。马股首个交易日出师不利，全天下滑...
3  （吉隆坡3日讯）亚洲航空（AirAsia，5099，主板消费股）董事局建议将公司的名字改为C...
4  （吉隆坡4日讯）美国隔夜股市上涨，带动亚洲股市周二升多跌少。不过，马股仍延续周一跌势，与大市...
```

Figure 5: First Few Rows of the Modified Textual Dataset

```
Summary of the DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3361 entries, 0 to 3360
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     3361 non-null   object
 1   Title    3361 non-null   object
 2   Content  3361 non-null   object
dtypes: object(3)
memory usage: 78.9+ KB
None
```

Figure 6: Summary of the Modified DataFrame

```
Summary statistics:
              Date                  Title  \
count         3361                   3361
unique         576                   3354
top     2023年8月25日  次季业绩符预期 天地通数码实现3大财务目标
freq            28                      2

                                                  Content
count                                                3361
unique                                               3349
top     （吉隆坡22日讯）美联储主席鲍威尔暗示5月再升息，加上美债收益率上涨导致美股承压，隔夜美股3...
freq                                                    2
```

Figure 7: Summary Statistics of the Modified DataFrame

The first few rows of the modified dataset were displayed to verify the changes (*Figure 5*). Summary statistics and information about the DataFrame were obtained to understand the dataset's characteristics. The summary indicated a total of 3,361 entries across three columns namely, *Date*, *Title*, and *Content*, and confirmed that there were no missing entries in any of the columns (*Figure 6* and *Figure 7*).

4.4 Initial Exploratory Data Analysis of the Numerical Datasets

Refer to *Appendix F* for the complete code and output snippets.

4.4.1 MIER Datasets

4.4.1.1 Loading the Data

The BCI (Business Condition Index) and CSI (Consumer Sentiment Index) datasets were loaded from the Excel file "*Data_BCI_CSI_2022_2023.xlsx*" using the sheet names *'BCI Formatted* ' and *'CSI Formatted*', respectively.

4.4.1.2 Exploratory Data Analysis (EDA) on the BCI and CSI Data

```
First few rows of the BCI dataset:
  Quarter  BCI Values
0  2022Q1       101.0
1  2022Q2        96.2
2  2022Q3        99.8
3  2022Q4        85.9
4  2023Q1        95.4
```

Figure 8: First Few Rows of the BCI Dataset

```
First few rows of the CSI dataset:
  Quarter  CSI Values
0  2022Q1       108.9
1  2022Q2        86.0
2  2022Q3        98.4
3  2022Q4       105.3
4  2023Q1        99.2
```

Figure 9: First Few Rows of the CSI Dataset

The first few rows of both the BCI (*Figure 8*) and CSI (*Figure 9*) datasets were displayed to understand their structure and content. Each dataset included columns for *Quarter* and *BCI/CSI Values*.

```
Summary of the bci_df_eda DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Quarter      8 non-null      object
 1   BCI Values   8 non-null      float64
dtypes: float64(1), object(1)
memory usage: 260.0+ bytes
None
```

Figure 10: Summary Statistics of BCI Dataset

```
Summary of the csi_df_eda DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Quarter      8 non-null      object
 1   CSI Values   8 non-null      float64
dtypes: float64(1), object(1)
memory usage: 260.0+ bytes
None
```

Figure 11: Summary Statistics of CSI Dataset

The summary information for both BCI (*Figure 10*) and CSI (*Figure 11*) DataFrames was generated, showing that each dataset contained 8 entries with no missing values.

4.4.1.3 Checking for Outliers

```
Outliers in BCI Values:
Empty DataFrame
Columns: [Quarter, BCI Values ]
Index: []
```

Figure 12: Outlier Detection (BCI Dataset)

```
Outliers in CSI Values:
Empty DataFrame
Columns: [Quarter, CSI Values ]
Index: []
```

Figure 13: Outlier Detection (CSI Dataset)

Outliers in the BCI (*Figure 12*) and CSI (*Figure 13*) values were checked using the Z-score method. The analysis indicated that there were no outliers in either dataset.

4.4.2 Macroeconomics Dataset

4.4.2.1 Loading the Data

The macroeconomic dataset, covering imports, exports, GDP, private consumption, and private investment, was loaded from the Excel file named "*Macroeconomics_Data.xlsx*". Each variable was loaded from its respective sheet and combined into a single DataFrame based on the *Quarter* column.

4.4.2.2 Exploratory Data Analysis (EDA) on the Macroeconomics Data

First few rows of the Macroeconomics dataset:

	Quarter	Imports	Exports	GDP	Private Consumption	Private Investment
0	2022Q1	16.1	12.3	4.8	5.3	0.4
1	2022Q2	20.1	15.9	8.8	18.3	6.3
2	2022Q3	21.1	21.5	14.1	14.8	13.2
3	2022Q4	7.2	8.6	7.1	7.3	10.3
4	2023Q1	-6.5	-3.3	5.6	5.9	4.7

Figure 14: First Few Rows of the Macroeconomics Dataset

```
Summary of the Macroeconomics DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Quarter              8 non-null      object
 1   Imports              8 non-null      float64
 2   Exports              8 non-null      float64
 3   GDP                  8 non-null      float64
 4   Private Consumption  8 non-null      float64
 5   Private Investment   8 non-null      float64
dtypes: float64(5), object(1)
memory usage: 516.0+ bytes
None
```

Figure 15: Summary Statistics of the Macroeconomics DataFrame

The first few rows of the combined macroeconomics dataset were displayed to understand its structure and content, revealing columns for *Quarter*, *Imports*, *Exports*, *GDP*, *Private Consumption*, and *Private Investment* (*Figure 14*). The summary information for the Macroeconomics DataFrame was generated, showing that the dataset contained 8 entries for each variable with no missing values (*Figure 15*).

4.4.2.3 Checking for Outliers

```
Outliers in Imports:
No outliers found.

Outliers in Exports:
No outliers found.

Outliers in GDP:
No outliers found.

Outliers in Private Consumption:
No outliers found.

Outliers in Private Investment:
No outliers found.
```

Figure 16: Outlier Detection (Macroeconomics Data)

Outliers in the macroeconomic variables were checked using the Z-score method as well. The analysis indicated that there were no outliers in any of the variables.

4.5 Preprocessing Steps for the Textual Dataset

Refer to *Appendix G* for the complete code and output snippets.

4.5.1 Stripping Whitespace from Column Names

Leading and trailing whitespaces were stripped from all column names, ensuring clean and standardized column headers.

4.5.2 Converting Chinese Dates to Standard Format

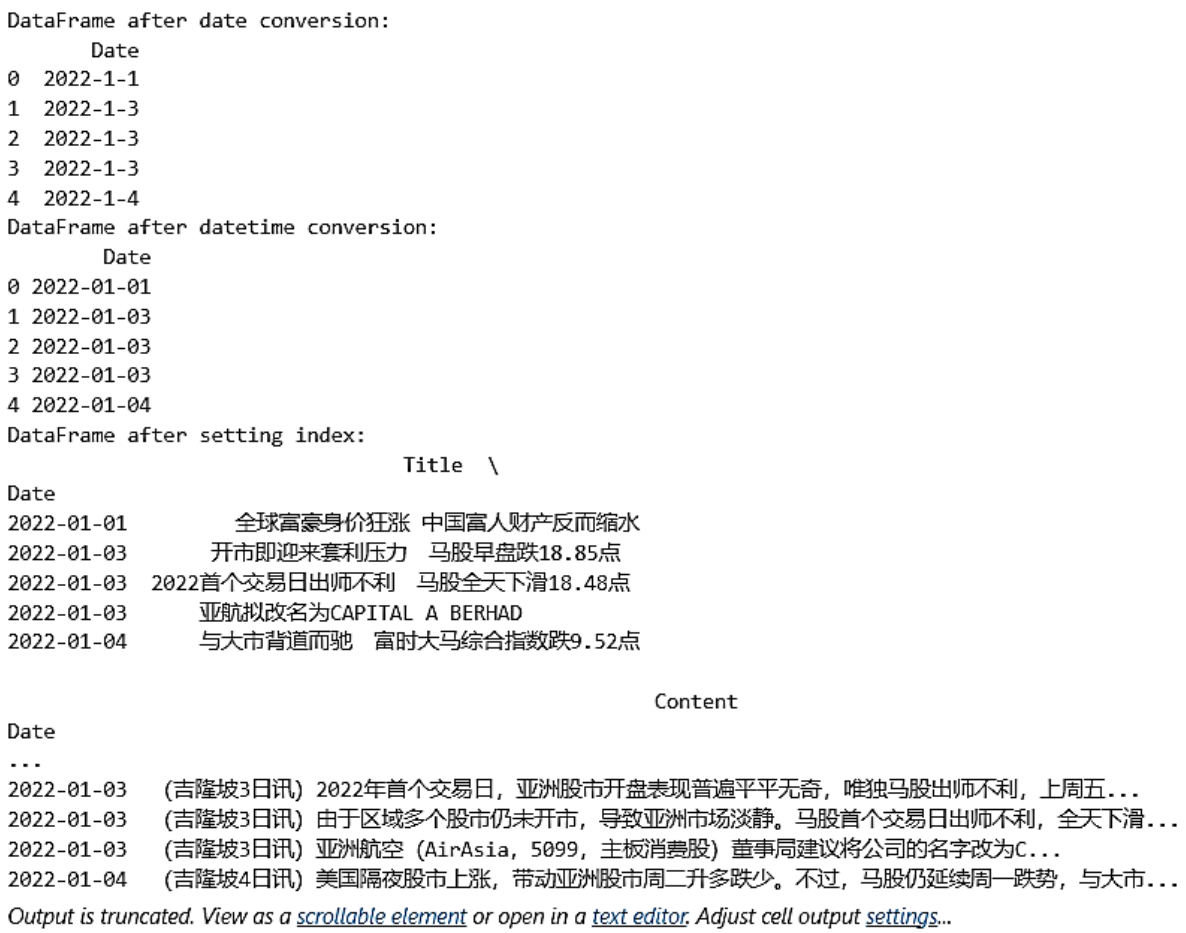


Figure 17: Chinese Dates Conversion to Standard Format

A function was applied to convert Chinese dates in the *Date* column to a standard format. This step facilitated further date manipulations and analyses. After conversion, the dates were set as the index of the DataFrame, confirming the proper format and structure.

4.5.3 Text Normalization

Leading and trailing whitespaces were stripped from all string columns to ensure clean data entries. Potential Unicode issues were handled by normalizing the text, further ensuring data consistency.

4.5.4 Loading and Displaying Sentiment Dictionary

```
Sample positive words: ['天籁', '可爱的', '可靠性', '划一', '豪华', '变好', '进步', '精粹', '范文', '超强']
Sample negative words: ['声讨', '损毁', '起诉书', '犯罪者', '埋葬', '缺席', '等闲视之', '蹂躏', '昏迷', '庸医']
```

Figure 18: First Few Positive and Negative Words According to the JMZ Dictionary

```
Positive words starting with '不': ['不懈', '不同凡响', '不含糊', '不愧', '不虚此行', '不乏', '不屈不挠', '不可思议的', '不拘泥',
Positive words starting with '没': []
Negative words starting with '不': ['不满', '不便的', '不充足的', '不怀', '不经', '不完备的', '不过关', '不畏', '不堪重负', '不切
Negative words starting with '没': ['没完没了', '没把握', '没用', '没事', '没得说', '没劲', '没有理由的', '没关系', '没法子', '没
```

Figure 19: Filtering Words with Negation

The JMZ sentiment dictionary was loaded from the Excel file named "*中文金融情感词典_姜富伟等(2021).xlsx*", containing positive and negative words. A sample of these words was displayed to verify the content, showing words classified into their respective sentiment categories (*Figure 18*). To show that the dictionary includes negators, words starting with "不" and "没" were filtered and displayed (*Figure 19*). This confirmed that negation was accounted for in the sentiment analysis.

4.5.5 Loading Stop Words and Defining Preprocessing Function

Stop words were loaded from a JSON file named "*stopwords-zh.json*" to remove common, non-informative words from the text. A preprocessing function was defined and applied to tokenize the text, remove stop words, and prepare the data for sentiment analysis.

4.5.6 Concatenating Title and Content and Performing Textual Preprocessing

The *Title* and *Content* columns were concatenated into a single *combined_text* column, which was then tokenized using the preprocessing function. This step prepared the text data for subsequent sentiment analysis and machine learning models.

4.6 Exploratory Data Analysis of the Preprocessed Textual Dataset

Refer to *Appendix H* for the complete code and output snippets.

4.6.1 Data Verification

```
First few rows of the preprocessed dataset:
                                Title  \
Date
2022-01-01          全球富豪身价狂涨 中国富人财产反而缩水
2022-01-03        开市即迎来套利压力 马股早盘跌18.85点
2022-01-03  2022首个交易日出师不利 马股全天下滑18.48点
2022-01-03       亚航拟改名为CAPITAL A BERHAD
2022-01-04       与大市背道而驰 富时大马综合指数跌9.52点

                                                 Content  \
Date
2022-01-01  这2年在新冠病毒肆虐期间,全球富豪资产大幅增长,不过中国科技富豪资产大失血,根据彭博亿万富豪...
2022-01-03  (吉隆坡3日讯)2022年首个交易日,亚洲股市开盘表现普遍平平无奇,唯独马股出师不利,上周五...
2022-01-03  (吉隆坡3日讯)由于区域多个股市仍未开市,导致亚洲市场淡静。马股首个交易日出师不利,全天下滑...
2022-01-03  (吉隆坡3日讯)亚洲航空(AirAsia,5099,主板消费股)董事局建议将公司的名字改为C...
2022-01-04  (吉隆坡4日讯)美国隔夜股市上涨,带动亚洲股市周二升多跌少。不过,马股仍延续周一跌势,与大市...

                                             combined_text  \
Date
2022-01-01  全球富豪身价狂涨 中国富人财产反而缩水 这2年在新冠病毒肆虐期间,全球富豪资产大幅增长,不过...
2022-01-03  开市即迎来套利压力 马股早盘跌18.85点 (吉隆坡3日讯)2022年首个交易日,亚洲股市开...
2022-01-03  2022首个交易日出师不利 马股全天下滑18.48点 (吉隆坡3日讯)由于区域多个股市仍未开...
2022-01-03  亚航拟改名为CAPITAL A BERHAD (吉隆坡3日讯)亚洲航空(AirAsia,50...
2022-01-04  与大市背道而驰 富时大马综合指数跌9.52点 (吉隆坡4日讯)美国隔夜股市上涨,带动亚洲股市...
...
2022-01-03  [开市, 迎来, 套利, 压力,  , 马股, 早盘, 跌, 1885,  , 吉隆坡, 3...
2022-01-03  [2022, 首个, 交易日, 出师不利,  , 马股, 全天, 下滑, 1848,  , ...
2022-01-03  [亚航, 拟, 改名, CAPITAL,  , A,  , BERHAD,  , 吉隆坡, ...
2022-01-04  [大市, 背道而驰,  , 富时, 大马, 综合, 指数, 跌, 952,  , 吉隆坡, ...
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Figure 20: First Few Rows of the Preprocessed Textual Data

```
Summary of the preprocessed DataFrame:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3361 entries, 2022-01-01 to 2023-12-30
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Title          3361 non-null   object
 1   Content        3361 non-null   object
 2   combined_text  3361 non-null   object
 3   tokens         3361 non-null   object
dtypes: object(4)
memory usage: 131.3+ KB
None
```

Figure 21: Summary of the Preprocessed Textual Data

The first few rows of the preprocessed dataset were displayed to verify the changes. The summary of the preprocessed DataFrame was obtained, showing a total of 3,361 entries across four columns namely, *Title*, *Content*, *combined_text*, and *tokens* (*Figure 20* and *Figure 21*).

4.6.2 Saving Preprocessed Textual Data

The preprocessed data was saved to a CSV file named "*preprocessed_data.csv*" with 'utf-8' encoding for future use.

4.7 Computation of the News Sentiment Index

Refer to *Appendix I* for the complete code and output snippets.

4.7.1 Sentiment Analysis

A sentiment analysis function, *compute_sentiment*, was defined to calculate the sentiment score for each article. The function counts the positive and negative words within the tokens and computes a sentiment score, normalized by the total number of words. The *compute_sentiment* function was then applied to the *tokens* column of the DataFrame. This process generated new columns namely, *sentiment_score*, *positive_count*, *negative_count*, and *total_words*. These columns were added to the DataFrame to store the sentiment analysis results.

4.7.2 Inspecting Sentiment Calculation

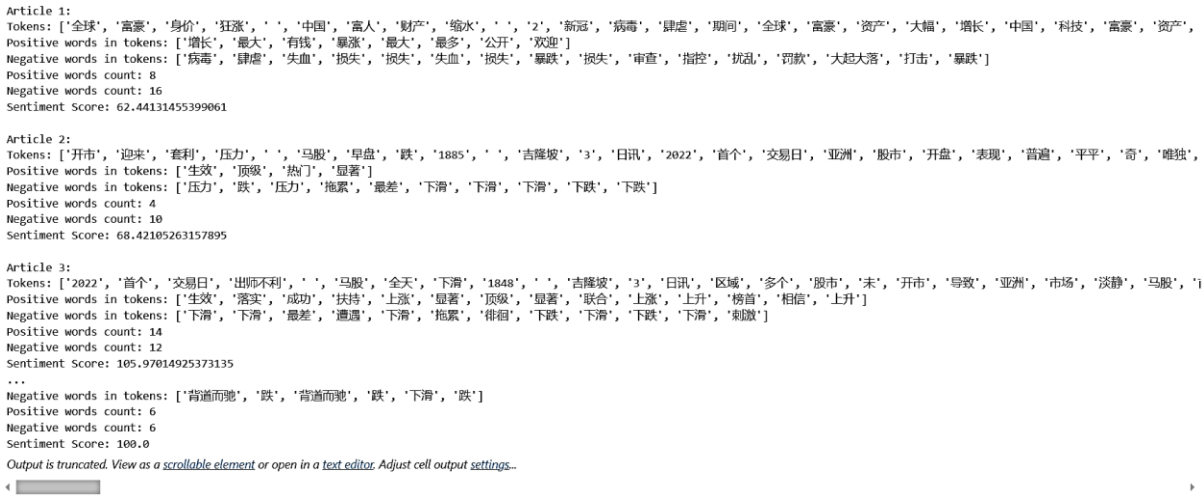


Figure 22: Inspect Sentiment Calculation for a Few Articles

To ensure the correctness of the sentiment scores, a sample of articles was inspected (*Figure 22*). For each selected article, the tokens, positive and negative words, and sentiment scores were printed. This step confirmed that the sentiment computation accurately identified and counted sentiment words.

4.7.3 Distribution of Sentiment Scores

```
Sample sentiment scores:
            sentiment_score
Date
2022-01-01        62.441315
2022-01-03        68.421053
2022-01-03       105.970149
2022-01-03       171.428571
2022-01-04       100.000000
2022-01-04       114.388489
2022-01-04        64.497041
2022-01-05       148.543689
2022-01-05       100.000000
2022-01-05        57.446809
```

Figure 23: Sample Sentiment Scores

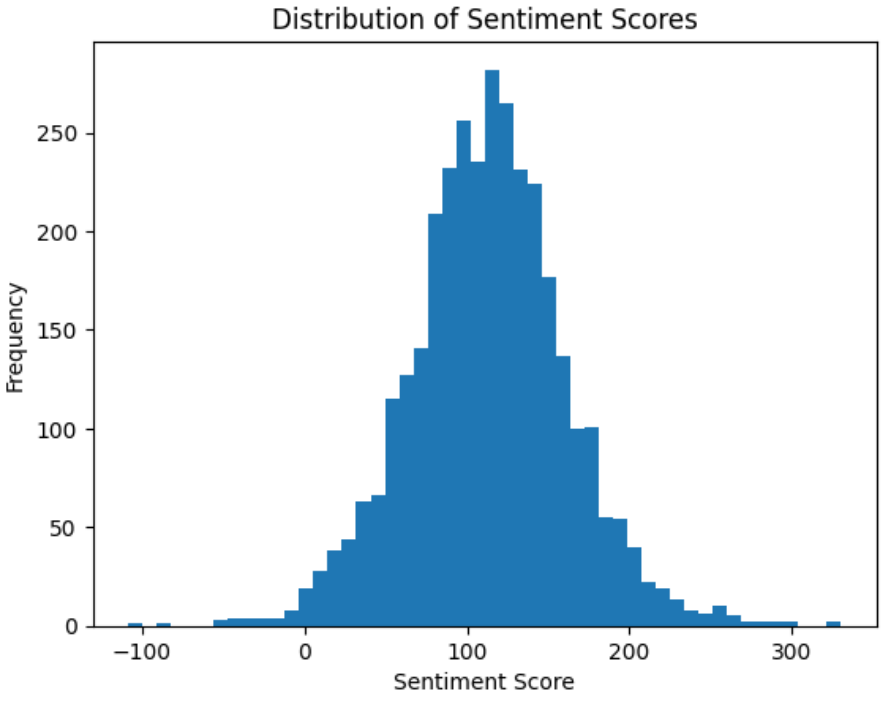


Figure 24: Distribution of Sentiment Scores

Sample sentiment scores were printed to verify the calculations (*Figure 23*). The scores for the first few articles demonstrated a range of sentiments, reflecting the diversity of the article content. A histogram was plotted to visualize the distribution of sentiment scores (*Figure 24*). The scores were approximately normally distributed, indicating a balanced sentiment across the articles.

4.7.4 Creating the Final DataFrame for Quarterly Sentiment Indices

```
Quarterly Sentiment Index:
     Date  quarterly_sentiment_index
0  2022Q1                 104.752807
1  2022Q2                 107.689848
2  2022Q3                 108.340161
3  2022Q4                 114.046902
4  2023Q1                 114.676678
```

Figure 25: Displaying the Resulting DataFrame for Quarterly Sentiment Indices

The sentiment scores were resampled to compute quarterly averages, aligning with the frequency of economic indicators. This step involved calculating the mean sentiment score for each quarter. A final DataFrame was created to store the quarterly sentiment indices. The *Date* column was converted to a quarterly format (e.g., 2022Q1), and the DataFrame was organized with the *Date* column as the first column. The first few rows of the resulting DataFrame were displayed to verify the computation of the quarterly sentiment index (*Figure 25*). This table showed the quarterly sentiment scores for each quarter from Q1 2022 to Q1 2023. The resulting quarterly sentiment index was saved to a CSV file named "*quarterly_sentiment_index.csv*" with 'utf-8' encoding. This file contains the computed quarterly sentiment indices, ready for further analysis.

4.8 Nowcasting Activity of the BCI and CSI Figures Using the News Sentiment Index

Refer to *Appendix J* for the complete code and output snippets. Note also that the results (plots, regression outputs, correlation matrix, and Variance Inflation Factor results) for the nowcasting activity will only be presented in *CHAPTER 5* and subsequently discussed in *CHAPTER 6*. This approach will be consistently applied to all sections from this point onward in *CHAPTER 4*.

4.8.1 Preparing the Data for the Nowcasting Activity

```
Column names in merged_df: Index(['Date', 'quarterly_sentiment_index', 'Quarter_BCI', 'BCI Values ',
       'Quarter_CSI', 'CSI Values '],
      dtype='object')
```

Figure 26: Verify Column Names in the DataFrame

The BCI and CSI data from MIER were loaded from an Excel file named "*Data BCI_CSI_2022_2023.xlsx*". The respective sheets '*BCI Formatted* ' and '*CSI Formatted*' were specified to extract the data. The quarterly sentiment index was then merged with the BCI and CSI data. The *Date* column from the sentiment index was matched with the *Quarter* column from the BCI and CSI data. The column names in the merged DataFrame were verified (*Figure 26*), and leading and trailing whitespaces were stripped to ensure consistency.

4.8.2 Time Series Plotting

A time series plot was created to visually compare the News Sentiment Index with MIER's Business Condition Index and Consumer Sentiment Index. The plot displayed the index values over time, facilitating visual analysis.

4.8.3 Nowcasting BCI and CSI Values Using the News Sentiment Index

The DataFrame was sorted by date. Lagged variables for BCI and CSI were created to account for temporal dependencies, with one lag for each variable. Rows with missing values created by lagging were dropped. Regression analysis was performed to evaluate the relationship between the BCI values and the quarterly sentiment index. The Ordinary Least Squares (OLS) regression model was applied, and the results were printed to assess the model's performance. Similarly, regression analysis was conducted for the CSI values using the quarterly sentiment index.

4.8.4 Plotting Actual vs Predicted Values

Separate plots were generated to compare the actual BCI and CSI values with the predicted values based on the regression models. These plots provided a visual representation of the nowcasting performance of the sentiment index.

4.8.5 Multicollinearity Check

The condition number was assessed to check for potential multicollinearity issues in the BCI nowcasting activity. The correlation matrix and Variance Inflation Factor (VIF) were calculated to further investigate multicollinearity.

4.9 Evaluating the Pearson Correlation Between the Macroeconomics Variables and the News Sentiment Index

Refer to *Appendix K* for the complete code and output snippets.

4.9.1 Preparing the Data for the Pearson Correlation Activity

The quarterly sentiment index data was loaded from a CSV file named "*quarterly_sentiment_index.csv*". The macroeconomic data was merged with the quarterly sentiment index data on the *Date* and *Quarter* columns. The *Quarter* and *Date* columns were then dropped.

4.9.2 Compute Pearson Correlation Coefficients

Pearson correlation coefficients between the quarterly sentiment index and various macroeconomic variables (imports, exports, GDP, private consumption, private investment) were computed and displayed.

4.10 Modelling Process for the Forecasting Activity of the 5 Target Variables Using Machine Learning Models

Refer to *Appendix L* for the complete code and output snippets.

4.10.1 Modelling Process without Hyperparameter Tuning and Performance Evaluation

4.10.1.1 Initialization for the Modelling Process

Functions *compute_rmse(y_true, y_pred)* and *compute_mae(y_true, y_pred)* were defined to calculate the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), respectively. Variables for the window size (4 quarters), forecast horizons (1, 2, 3), and macroeconomic variables (imports, exports, GDP, private consumption, private investment) were initialized. The data was copied to a new DataFrame, and the *Date* column was excluded from the features.

Dictionaries for storing results, RMSE ratios, and MAE ratios were initialized for various machine learning models, including Linear Regression, LASSO, Ridge, SVR, Random Forest Regressor, and XGBoost.

4.10.1.2 Rolling Window Approach

A rolling window approach was used to train and test the models. The OLS-AR(1) model was trained as a benchmark, and other models were trained and tested. RMSE and MAE were calculated and stored for each model and variable.

4.10.1.3 Display of Results

RMSE and MAE ratios for each macroeconomic variable were plotted for the different models and forecast horizons. Additionally, the results, including RMSE and MAE ratios for each model, variable, and forecast horizon, were displayed.

4.10.2 Modelling Process with Hyperparameter Tuning and Performance Evaluation

4.10.2.1 Initialization for the Modelling Process

A function *compute_metrics(y_true, y_pred)* was defined to calculate the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for the model predictions. Variables were initialized for the window size (*window_size_tuned = 4 quarters*), forecast horizons (*horizons_tuned = [1, 2, 3]*), and macroeconomic variables (*macro_vars_tuned = ['Imports', 'Exports', 'GDP', 'Private Consumption', 'Private Investment']*). The data was copied to a new DataFrame '*data_tuned'* for further processing. The *Date* column was excluded from the features. Parameter grids for hyperparameter tuning were defined for LASSO, Ridge, SVR, Random Forest, and XGBoost models. Models were initialized using pipelines to standardize the data and apply the respective regression algorithms. Dictionaries *best_models_tuned* and *best_params_tuned* were initialized to store the best models and parameters identified during hyperparameter tuning.

4.10.2.2 Hyperparameter Tuning with Grid Search

Grid search with 5-fold cross-validation was performed to identify the best models and parameters for each macroeconomic variable and forecast horizon.

4.10.2.3 Use Best Models for Rolling Window Approach

The best models identified during tuning were used in a rolling window approach to predict the target variables for each horizon. The OLS-AR(1) model was trained as a benchmark to compare the performance of other models. RMSE and MAE were calculated for each model and stored in dictionaries for further analysis. RMSE and MAE ratios were also computed to compare the performance of the tuned models with the benchmark. A check was performed to identify any empty lists in the results to ensure all variables and horizons had valid predictions.

4.10.2.4 Display of Results

Plots were generated to visualize the RMSE and MAE ratios for each macroeconomic variable across different models and forecast horizons. The average RMSE and MAE ratios for each model, variable, and horizon were displayed. The best parameters identified for each model during hyperparameter tuning were displayed for reference.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Nowcasting the BCI and CSI Figures using the News Sentiment Index

5.1.1 Time Series Plot

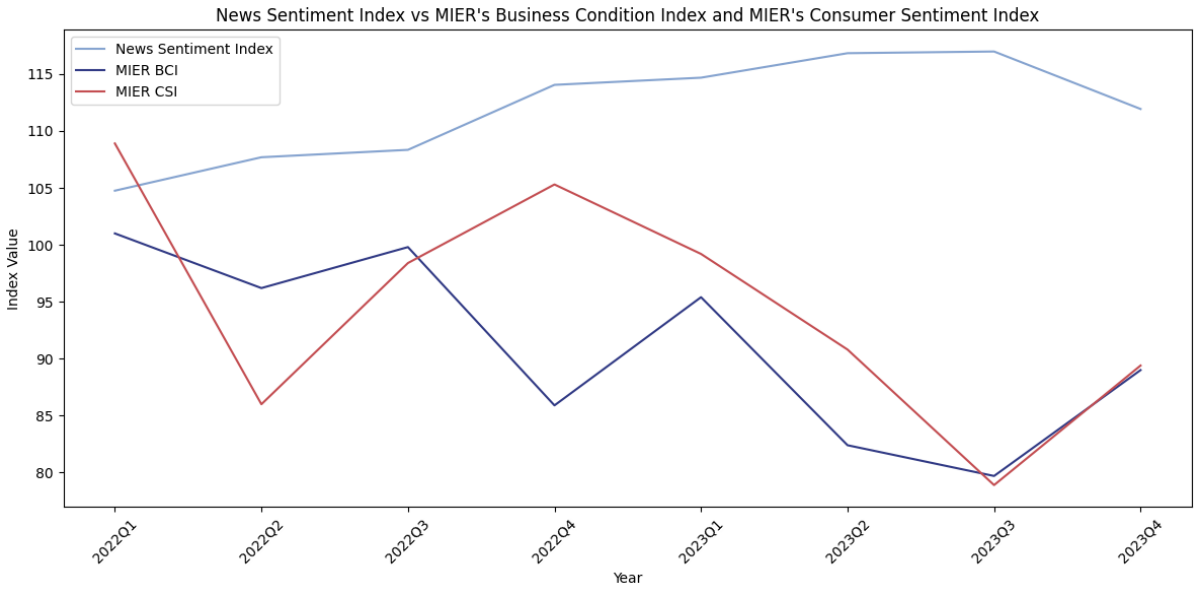


Figure 27: Time Series Plot Between the News Sentiment Index and the MIER Indices

The figure above illustrates the relationship between the News Sentiment Index, MIER's Business Condition Index (BCI), and MIER's Consumer Sentiment Index (CSI) from 2022 Q1 to 2023 Q4. The News Sentiment Index generally trends upward, indicating increased positive sentiment. In contrast, the BCI and CSI exhibit more volatility. From 2022 Q1 to 2022 Q3, the News Sentiment Index and BCI show some alignment, both declining and then recovering. However, post-2022 Q3, the BCI sharply declines until 2023 Q3, diverging from the sentiment index, suggesting a moderately weak correlation.

The CSI, initially higher than the sentiment index, drops steeply in 2022 Q2, partially recovers, and then declines again until 2023 Q3 before a slight recovery. This pattern indicates even less correlation with the News Sentiment Index.

Overall, the News Sentiment Index shows a moderately weak correlation with the BCI and a very weak correlation with the CSI.

5.1.2 Regression Output for Nowcasting MIER's BCI

OLS Regression Results

Dep. Variable:	BCI Values	R-squared:	0.697
Model:	OLS	Adj. R-squared:	0.545
Method:	Least Squares	F-statistic:	4.592
Date:	Fri, 26 Jul 2024	Prob (F-statistic):	0.0921
Time:	14:54:19	Log-Likelihood:	-19.399
No. Observations:	7	AIC:	44.80
Df Residuals:	4	BIC:	44.64
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	289.5625	81.926	3.534	0.024	62.099	517.026
BCI_Lag	-0.0541	0.265	-0.204	0.848	-0.791	0.682
quarterly_sentiment_index	-1.7254	0.609	-2.834	0.047	-3.416	-0.035

Omnibus:	nan	Durbin-Watson:	2.089
Prob(Omnibus):	nan	Jarque-Bera (JB):	1.849
Skew:	1.250	Prob(JB):	0.397
Kurtosis:	3.303	Cond. No.	6.16e+03

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 6.16e+03. This might indicate that there are strong multicollinearity or other numerical problems.

c:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\statsmodels\stats\stattools.py:74: ValueWarning: omni_normtest is not valid with less than 8 observations; 7 samples were given.

warn("omni_normtest is not valid with less than 8 observations; %i "

Figure 28: OLS Regression Results for BCI Values Against Lagged BCI and Quarterly Sentiment Index

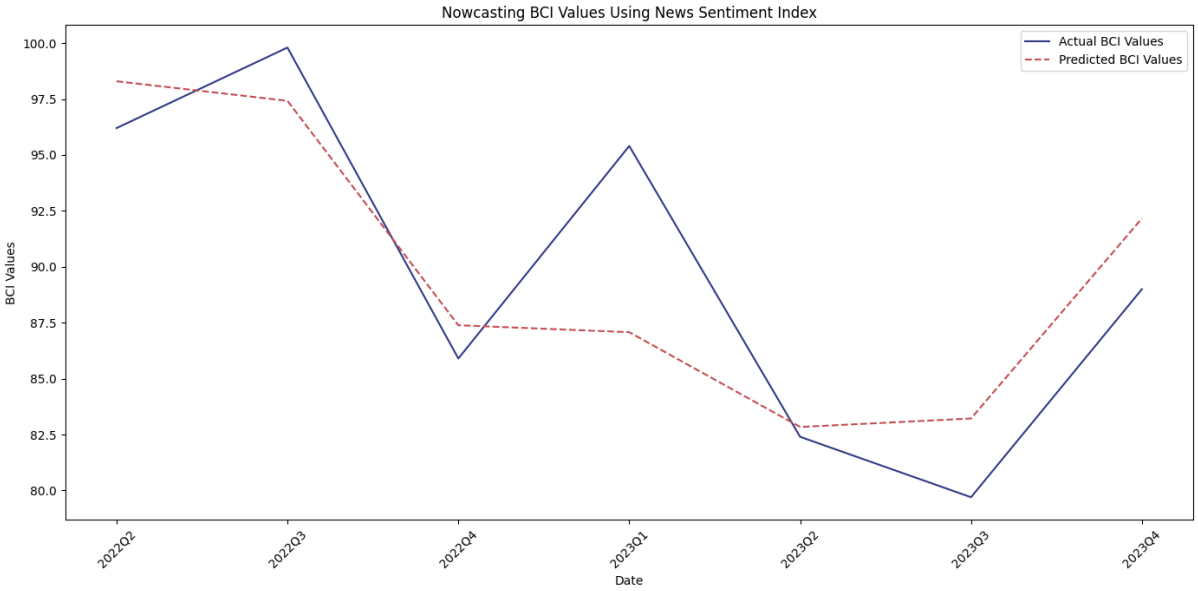


Figure 29: Comparison of Actual vs Predicted BCI Values Over Time

The regression results (*Figure 28*) for the Business Condition Index (BCI) demonstrate a moderately strong model fit, with an R-squared value of 0.697, suggesting that approximately 69.7% of the variance in BCI values is explained by the model. However, the adjusted R-squared is 0.545, indicating that when adjusted for the number of predictors, the model's explanatory power decreases slightly.

The F-statistic of 4.592 and its corresponding p-value of 0.0921 suggest that the overall model is marginally significant at the 10% level. The lagged BCI variable (*BCI_Lag*) has a coefficient of -0.0541 and a p-value of 0.848, indicating it is not statistically significant and has little impact on current BCI values. The quarterly sentiment index variable has a negative coefficient of -1.7254 with a p-value of 0.047, making it statistically significant at the 5% level. This suggests that higher sentiment index values are associated with lower BCI values, an unexpected negative relationship.

The Durbin-Watson statistic of 2.089 indicates no significant autocorrelation in the residuals. However, the large condition number (6.16e+03) raises concerns about potential multicollinearity or numerical issues within the model. The normality test (Omnibus and Jarque-Bera) is not valid due to the small sample size of seven observations. The plot (*Figure 29*) for nowcasting BCI values using the News Sentiment Index shows that the predicted BCI values follow the general trend of the actual BCI values but with less volatility. The predicted values are generally higher than the actual values in most quarters, indicating a potential

overestimation by the model during this period. This discrepancy further suggests that while the model captures some trends, it fails to account for more nuanced fluctuations in the BCI values.

Overall, the model shows some explanatory power but also highlights significant issues, particularly with the unexpected negative relationship between the sentiment index and BCI, and the potential of multicollinearity or numerical problems as indicated by the high condition number. The concerns about potential multicollinearity or numerical issues will be further addressed in *CHAPTER 6*.

5.1.3 Regression Output for Nowcasting MIER's CSI

OLS Regression Results

Dep. Variable:	CSI Values	R-squared:	0.031
Model:	OLS	Adj. R-squared:	-0.454
Method:	Least Squares	F-statistic:	0.06367
Date:	Fri, 26 Jul 2024	Prob (F-statistic):	0.939
Time:	14:54:25	Log-Likelihood:	-24.644
No. Observations:	7	AIC:	55.29
Df Residuals:	4	BIC:	55.13
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	117.3405	137.175	0.855	0.441	-263.517	498.198
CSI_Lag	0.1029	0.413	0.249	0.816	-1.045	1.251
quarterly_sentiment_index	-0.3063	1.171	-0.261	0.807	-3.558	2.946

Omnibus:	nan	Durbin-Watson:	1.274
Prob(Omnibus):	nan	Jarque-Bera (JB):	0.394
Skew:	-0.034	Prob(JB):	0.821
Kurtosis:	1.840	Cond. No.	4.96e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.96e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
c:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\statsmodels\stats\stattools.py:74: ValueWarning: omni_normtest is not valid with less than 8 observations; 7 samples were given.
warn("omni_normtest is not valid with less than 8 observations; %i "

Figure 30: OLS Regression Results for CSI Values Against Lagged CSI and Quarterly Sentiment Index

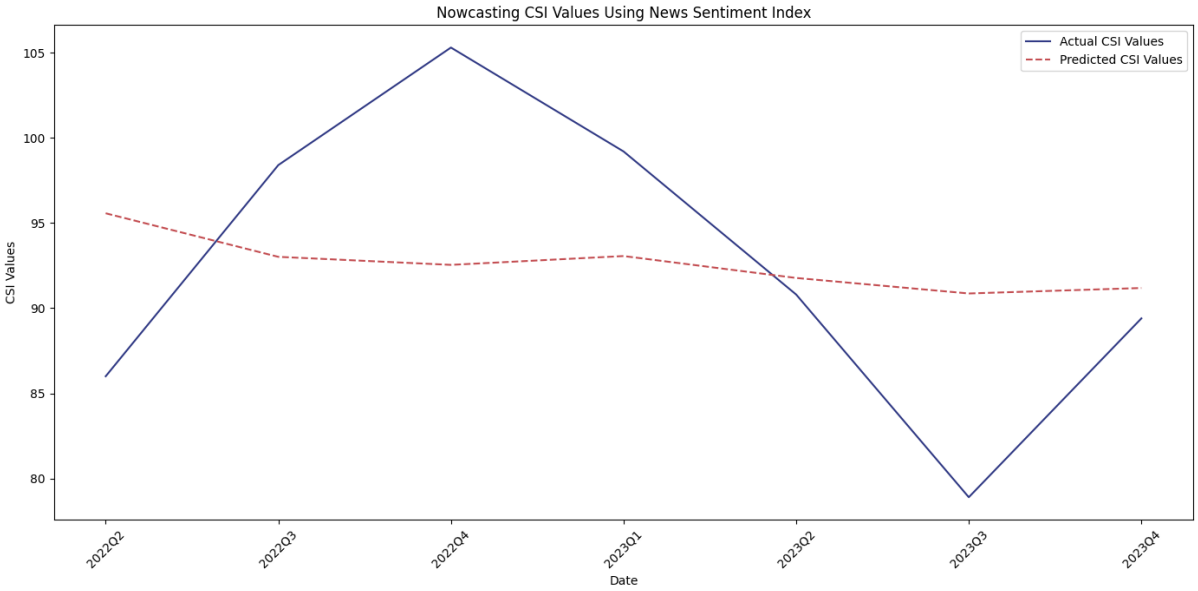


Figure 31: Comparison of Actual vs Predicted CSI Values Over Time

The regression results (*Figure 30*) for the Consumer Sentiment Index (CSI) demonstrate a weak model fit, with an R-squared value of 0.031, suggesting that only 3.1% of the variance in CSI values is explained by the model. The adjusted R-squared is -0.454, indicating that when adjusted for the number of predictors, the model's explanatory power significantly decreases, implying a poor fit. The F-statistic of 0.06367 and its corresponding p-value of 0.939 suggest that the overall model is not significant, indicating that the predictors do not explain the variability in the CSI values.

The lagged CSI variable (*CSI_Lag*) has a coefficient of 0.1029 and a p-value of 0.816, suggesting that it is not statistically significant and has little impact on current CSI values. The quarterly sentiment index variable also has a negative coefficient of -0.3063 with a p-value of 0.807, indicating that it is not statistically significant and does not meaningfully predict CSI values. The Durbin-Watson statistic of 1.274 suggests a slight positive autocorrelation in the residuals. The large condition number (4.96e+03) raises concerns about potential multicollinearity or numerical issues within the model. Additionally, the normality test (Omnibus and Jarque-Bera) is not valid due to the small sample size of seven observations.

The plot (*Figure 31*) for nowcasting CSI values using the News Sentiment Index shows that the predicted CSI values generally follow a downward trend, whereas the actual CSI values exhibit more variability, with significant peaks and troughs. This discrepancy indicates that the

model fails to capture the dynamic fluctuations in the actual CSI values, leading to a less accurate prediction.

Overall, the model shows very little explanatory power and fails to accurately predict the CSI values. Therefore, it is unnecessary to perform further checks for potential multicollinearity or numerical issues, as the model's inherent lack of fit renders such checks redundant.

5.2 Forecasting the 5 Target Variables using the News Sentiment Index

5.2.1 Pearson Correlation Between the Macroeconomics Variables and the News Sentiment Index

```
Pearson Correlation Coefficients with Quarterly Sentiment Index:
quarterly_sentiment_index    1.000000
Imports                     -0.889319
Exports                     -0.817534
GDP                         -0.494618
Private Consumption         -0.532231
Private Investment           0.052724
Name: quarterly_sentiment_index, dtype: float64
```

Figure 32: Pearson Correlation Results

The Pearson correlation coefficients indicate a strong negative correlation between the quarterly sentiment index and imports (-0.889319), and exports (-0.817534). There is a moderate negative correlation with GDP (-0.494618) and private consumption (-0.532231). Private investment shows a very weak positive correlation (0.052724) with the quarterly sentiment index. These results suggest that as the sentiment index increases, imports, exports, GDP, and private consumption tend to decrease, while private investment remains largely unaffected.

The negative correlations between the quarterly sentiment index and imports, exports, GDP, and private consumption are unexpected. This is because generally, a higher sentiment index, indicating more positive news sentiment, might be expected to correlate positively with economic activity indicators such as GDP, private consumption, and trade volumes. The weak positive correlation with private investment is also unexpected, as more positive sentiment might typically encourage higher investment levels.

5.2.2 Performance Evaluation for Machine Learning Models Without Hyperparameter Tuning

Due to the extensive nature of both Ratio RMSE and Ratio MAE outputs, whether displayed graphically or in textual format, these results will thereby only be included in the appendix (*Appendix M* and *Appendix N* respectively). For the textual display, models with pink borders indicate that both RMSE and MAE ratios are below 1 for all horizons. The models that meet this criterion, demonstrating robust predictive performance across all forecasting horizons, are indicated as follows.

For the imports and exports variables, none of the models achieved RMSE and MAE ratios below 1 across all horizons. The GDP variable on the other hand, saw the LASSO model consistently performing well, with ratios below 1 at all three of the horizons. Private consumption had the LASSO, Random Forest, and XGBoost models meeting the criterion. Lastly, private investment has had consistent performance with the LASSO and Ridge models.

To select the most robust models, the criterion is to choose those with at least three best performances out of the five macroeconomic variables. This principle of majority performance ensures that the selected models are not only effective in one or two specific variables, but instead demonstrate a broader and more consistent predictive capability across most of the variables analysed. By requiring strong performance in at least three out of five of the variables, this method prioritizes models that have proven to be reliable and adaptable across different economic indicators, enhancing their overall robustness and reliability for practical applications.

Narrowing down to models with at least three best performances across the five variables, the standout model is LASSO. The LASSO model excelled in predicting GDP, private consumption, and private investment for all of the forecasting horizon.

5.2.3 Performance Evaluation for Machine Learning Models with Hyperparameter Tuning

Similar to the approach in the previous subsection, both graphical and textual results of the RMSE and MAE ratios will be presented in *Appendix O* and *Appendix P*, respectively. In the same manner, for the text display, models highlighted with pink borders indicate that both

RMSE and MAE ratios are below 1 across all forecasting horizons. The following models are the ones that meet this criterion.

For the imports, exports, and GDP variables, none of the models meet the criterion. Private consumption had the Random Forest model with all ratios below 1 across all horizons, indicating strong predictive performance. Lastly, private investment on the other hand, showed consistent performance with the LASSO and SVR models having ratios below 1 across all horizons.

Upon narrowing down to models with at least three best performances across the five variables, it is clear that none of the models has managed to meet the criterion.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

6.1 Discussion of Nowcasting Findings

	Variable	VIF
0	const	1795.654814
1	BCI_Lag	1.210022
2	quarterly_sentiment_index	1.210022

Figure 33: Variance Inflation Factor (VIF) Analysis

Correlation Matrix:

	BCI_Lag	quarterly_sentiment_index
BCI_Lag	1.000000	-0.416616
quarterly_sentiment_index	-0.416616	1.000000

Figure 34: Correlation Matrix between BCI_Lag and Quarterly Sentiment Index

The regression output for the Consumer Sentiment Index (CSI) nowcasting activity will not be discussed given its poor model performance. The weak model fit, with an R-squared value of only 0.031 and a non-significant F-statistic, indicates that the predictors do not adequately explain the variability in CSI values. Therefore, further analysis of this model is deemed unnecessary, as it fails to provide meaningful insights into consumer sentiment prediction. Moreover, the findings that the news sentiment index is not predictive of MIER's CSI are consistent with the findings of Chong et al. (2021), which also indicated a weak relationship between the news sentiment of their study and consumer sentiment.

Multicollinearity testing, including Variance Inflation Factor (VIF) analysis, was conducted to check for multicollinearity issues in the Business Condition Index (BCI) regression output. As shown in *Figure 33*, both the *BCI_Lag* and quarterly sentiment index variables have VIF values of 1.210022 which is well below the commonly accepted threshold of 10. This indicates that no multicollinearity issues exist among the predictors. Additionally, the correlation matrix (*Figure 34*) further supports this finding, with a moderate negative correlation of -0.416616 between the *BCI_Lag* and the quarterly sentiment index. Despite the absence of multicollinearity, the higher condition number observed in the regression output could be

attributed to numerical issues. These issues are likely due to the small sample size of only seven observations, covering two years on a quarterly basis with one lag. This small sample size might lead to numerical instability in the regression analysis, causing higher condition numbers. Nonetheless, it is still worthwhile discussing the findings and comparing them with existing literature.

As pointed out in *Section 5.1*, the findings indicate that the quarterly news sentiment index is significant in predicting MIER's Business Condition Index (BCI). This is consistent with Chong et al. (2021), who found that the sentiment reflected in business and financial news from online sources does correspond to the survey-based measures of business sentiment, represented by the BCI figures. This consistency reinforces the validity of using news sentiment as a predictive tool for business sentiment, highlighting its potential utility for real-time economic forecasting.

Nevertheless, the negative coefficient of the quarterly sentiment index was unexpected. This outcome has two possible implications - either there are other significant factors influencing the BCI during the examined timeframe (2022 to 2023), thereby resulting in the negative direction, or the small sample size might have caused this anomaly. Potential influencing factors could include political events (Aisen & Veiga, 2013), economic conditions, or global incidents (Walker, 2024). For instance, Malaysia's hung parliament situation which took place at the end of 2022 (Gahagan, 2022), the formation of a new coalition government after the 15th General Election (Wong, 2023), and the ongoing Russian-Ukraine war which started at the beginning of 2022 (Kaya, 2024) could have impacted economic agents' decisions, essentially leading to negative sentiment. These events may have created uncertainty, subsequently affecting business confidence adversely despite positive news sentiment. Such political instability and international conflicts can lead to caution and reduced investment, ultimately affecting business conditions negatively.

That said, to better understand this relationship, it is crucial to examine a larger sample size covering more years and to include additional factors in the analysis. This would help determine if the negative coefficient does persist and if it does remain significant in forecasting the BCI. Moreover, incorporating other variables such as political stability, global economic conditions, and domestic policy changes could provide a more comprehensive model that accounts for these possible influential factors.

In summary, the analysis highlights the significance of the news sentiment index in predicting business conditions, which is consistent with previous research. However, the unexpected negative coefficient and potential numerical issues underscore the need for further investigation with a larger sample size and additional variables. Future work should focus on extending the data period, incorporating more diverse factors, and re-evaluating the model to ensure robustness and accuracy in forecasting business sentiment. This will provide a more comprehensive understanding of the factors driving business confidence and enhance the predictive power of news sentiment indices.

6.2 Discussion of Forecasting Findings

The analysis of the Pearson correlation coefficients reveals unexpected negative correlations between the quarterly news sentiment index and several macroeconomic variables, namely imports (-0.889319), exports (-0.817534), GDP (-0.494618), and private consumption (-0.532231). These negative correlations are unexpected because a higher sentiment index, which indicates more positive news sentiment, would typically be expected to correlate positively with economic activity indicators such as GDP, private consumption, private investment, and trade volumes. The reasons for these unexpected correlations may be similar to those discussed in the previous subsection, hence further investigation is necessary to validate whether these correlations are accurate and to understand the underlying causes better.

Contrary to the 4 macroeconomic variables mentioned above, the private investment variable shows a positive correlation (0.052724 with the quarterly sentiment index), albeit the correlation being very weak. While this positive correlation aligns more closely with expectations, its strength is minimal, suggesting that other factors may play a more significant role in influencing private investment than news sentiment alone. This warrants a deeper examination of the variables and factors that drive private investment decisions.

Comparing these findings to those of Chong et al. (2021), notable differences and similarities emerge. The authors found that news sentiments generally had a higher and expected positive correlation with aggregate GDP growth and private sector economic activities. They observed weaker correlations with trade activities (exports and imports), potentially due to the news coverage focusing more on domestic economic activities or a stronger lead-lag relationship between news sentiment and trade activities. Furthermore, after excluding periods of financial

crises, they noticed a decline in correlations between macroeconomic growth variables and news sentiments, except for private investment. This suggests that news sentiment may still provide valuable information regarding private investment activities even during non-crisis periods. The current study's unexpected negative correlations contrast with Chong et al. (2021)'s findings, indicating a need for further research to validate these differences.

In terms of the evaluation of the forecasting activity, the performance of machine learning models without hyperparameter tuning revealed better results than the those that are tuned. The untuned models showed robust predictive capabilities by achieving a majority performance across the evaluated metrics, whereas the tuned models did not achieve a similar level of consistency. Consequently, the discussion will focus on the findings from the untuned models. The LASSO model emerged as the most robust model, as it has managed to effectively predict GDP, private investment, and private consumption across all forecasting horizons. This consistent performance across multiple economic indicators highlights the LASSO model's reliability and adaptability.

In contrast, Chong et al. (2021) investigated the predictive power of news sentiment on economic growth outcomes and found that news sentiment consistently forecasted private investment growth better than a benchmark OLS-AR(1) model, particularly for forecasts two to three quarters ahead. However, while they did not identify a single "best" model in their study, they did note that news sentiment had limited predictive power for broader economic components beyond private investment. Making comparisons to their study, this study's findings partially align with their results, particularly regarding the predictive power of news sentiment for private investment but differ in identifying the LASSO model as robust for other variables as well.

In summary, the analysis underscores the complexities and challenges of using news sentiment indices to forecast macroeconomic variables. While the LASSO model has demonstrated strong predictive capabilities for GDP, private investment, and private consumption, the unexpected negative correlations and the need for further validation suggest areas for future research. Expanding the data period, incorporating additional factors, and re-evaluating the models can enhance robustness and accuracy.

6.3 Conclusion

Findings of this study suggest that the regression output for the Consumer Sentiment Index (CSI) nowcasting activity showed poor model performance, with a weak R-squared value and a non-significant F-statistic. This indicated that the predictors did not adequately explain the variability in CSI values, making further analysis unnecessary. The findings that the news sentiment index does not predict MIER's CSI align with Chong et al. (2021), who also found a weak relationship between news sentiment and consumer sentiment. For the BCI values, however, the quarterly news sentiment index was significant in predicting the Business Condition Index, aligning with Chong et al. (2021)'s results and reinforcing the potential of news sentiment as a useful tool for forecasting business sentiment. Additionally, the Pearson Correlation test revealed negative correlations between the sentiment index and imports, exports, GDP, and private consumption, suggesting that higher sentiment scores were linked to decreases in these economic activities. It is only for private investment that there is a very weak positive correlation between itself and the news sentiment index. In terms of the forecasting activity, the LASSO model proved to be the most robust among the machine learning models, effectively predicting GDP, private investment, and private consumption across all forecasting horizons of one, two, and three quarters ahead. These findings align partially with Chong et al. (2021), where the authors found that the news sentiment index is predictive of private investment 2 to 3-quarters ahead.

While comparisons between the findings of this study and that of Chong et al. (2021) are not directly equivalent, that is, since Chong et al. (2021) used English news sentiment data over a longer period (2006 to 2021), whereas this study used Mandarin news sentiment data over a shorter period (2022 to 2023), the intention was to highlight the potential of using Mandarin news sentiments for forecasting macroeconomic variables in Malaysia. Despite the differences, this study suggests that Mandarin news sentiment can be a valuable tool in economic forecasting. Referring back to the objectives of the study as outlined in *Section 1.3*, the study has successfully demonstrated that the Mandarin-based news sentiment index could predict the Business Condition Index (BCI) but was not effective for the Consumer Sentiment Index (CSI). Additionally, the LASSO model showed strong performance in predicting GDP, private investment, and private consumption across all forecasting horizons, thereby achieving the objective of performance comparison. Lastly, the study also provided insights into the correlation between the news sentiment index and the demand-side components of GDP.

In this study, a significant challenge faced was the reliance on a single news source, See Hua Newspaper. Efforts to broaden the data sources from other news portals were unsuccessful, resulting in limitations in capturing the full spectrum of economic sentiment in Malaysia's Chinese-language media. Additionally, focusing on the 2022 to 2023 period, necessitated by project time constraints, limited the scope of historical data that could be analyzed, potentially affecting the robustness of the findings by not accounting for longer-term economic cycles or sentiment trends. Future research should address these limitations by incorporating a broader range of news sources to capture more diverse economic sentiments. Extending the data period to include more years would provide a more comprehensive view of long-term economic cycles and trends. Additionally, incorporating other influential variables such as political stability, global economic conditions, and domestic policy changes could enhance the predictive models' robustness and accuracy. In summary, addressing these areas can provide more reliable and actionable insights for policymakers and business leaders, essentially improving decision-making processes through the effective use of news sentiment indices.

REFERENCES

Abir Masmoudi, Hamdi Jamila, & Lamia Hadrich Belguith. (2021). Deep learning for sentiment analysis of Tunisian dialect. *Computación y Sistemas*, *25*(1), 129–148. doi: 10.13053/CyS-25-1-3472

Agu, S. C., Onu, F. U., Ezemagu, U. K., & Oden, D. (2022). Predicting gross domestic product to macroeconomic indicators. *Intelligent Systems with Applications*, *14*, 200082. https://doi.org/10.1016/j.iswa.2022.200082

Aisen, A., & Veiga, F. J. (2013). How does political instability affect economic growth? *European Journal of Political Economy*, *29*, 151–167. https://doi.org/10.1016/j.ejpoleco.2012.11.001

Al Shamsi, A. A., & Sherief Abdallah. (2022). Sentiment analysis of Emirati dialect. *Big Data and Cognitive Computing*, *6*(2), 57–57. https://doi.org/10.3390/bdcc6020057

Aoki, G., Kazuto Ataka, Doi, T., & Kota Tsubouchi. (2023). Data-driven estimation of economic indicators with search big data in discontinuous situation. *The Journal of Finance and Data Science*, *9*, 100106–100106. https://doi.org/10.1016/j.jfds.2023.100106

Aprigliano, V., Emiliozzi, S., Guaitoli, G., Luciani, A., Marcucci, J., & Monteforte, L. (2023). The power of text-based indicators in forecasting Italian economic activity. *International Journal of Forecasting*, *39*(2), 791–808. https://doi.org/10.1016/j.ijforecast.2022.02.006

Ardia, D., Bluteau, K., & Boudt, K. (2019). Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values. *International Journal of Forecasting*, *35*(4), 1370–1386. https://doi.org/10.1016/j.ijforecast.2018.10.010

Ashwin, J., Eleni, K., Lorena, S. (2021, November). *Nowcasting euro area GDP with news sentiment: A tale of two crises.* (ECB Working Paper, No. 2616). doi:10.2866/240669

Barbaglia, L., Consoli, S., & Sebastiano Manzan. (2024). Forecasting GDP in Europe with textual data. *Journal of Applied Econometrics*, *39*(2), 338-355. https://doi.org/10.1002/jae.3027

Barbaglia, L., Frattarolo, L., Luca Onorante, Filippo Maria Pericoli, Ratto, M., & Luca Tiozzo Pezzoli. (2023). Testing big data in a big crisis: Nowcasting under Covid-19. *International Journal of Forecasting*, *39*(4), 1548–1563. https://doi.org/10.1016/j.ijforecast.2022.10.005

Buckman, S. R., Shapiro, A. H., Sudhof, M., & Wilson, D. J. (2020). *News sentiment in the time of COVID-19.* FRBSF Economic Letter. https://www.frbsf.org/wp-content/uploads/el2020-08.pdf

Bureau of Economic Analysis. (2023, December 18). *Gross domestic product.* https://www.bea.gov/resources/learning-center/what-to-know-gdp

Bybee, L., Kelly, B. T., Manela, A., & Xiu, D. (2021, October). *Business news and business cycles.* (Working Paper 29344). DOI 10.3386/w29344

Chong, E., Ho, C. C., Ong, Z. F., & Ong, H. H. (2021). *Using news sentiment for economic forecasting: A Malaysian case study*. https://www.bis.org/ifc/publ/ifcb57_17.pdf

Chow, C. K. (2023). *Chinese linguistics 汉语语言学: Singapore-Malaysia Chinese newspapers 新马华文报章*. Retrieved from https://libguides.nus.edu.sg/c.php?g=145587&p=956390

Chu, B., & Qureshi, S. (2023). Comparing out-of-sample performance of machine learning methods to forecast U.S. GDP growth. *Computational Economics*, *62*, 1567-1609. https://doi.org/10.1007/s10614-022-10312-z

Committee for the Coordination of Statistical Activities. (2020). *How COVID-19 is changing the world: a statistical perspective.* https://unstats.un.org/unsd/ccsa/documents/covid19-report-ccsa.pdf

Correa, R., Garud, K., Londono-Yarce, J.-M., & Mislang, N. (2017). Constructing a dictionary for financial stability. *IFDP Notes, 2017*(33), 1–7. https://doi.org/10.17016/2573-2129.33

Dauphin, J-F., Dybczak, K., Maneely, Sanjani, M. T., Suphaphiphat, N., Wang, Y., & Zhang, H. (2022, March). *Nowcasting GDP-A scalable approach using DFM, machine learning and novel data, applied to European economies.* (Working Paper No.

2022/052). https://www.imf.org/en/Publications/WP/Issues/2022/03/11/Nowcasting-GDP-A-Scalable-Approach-Using-DFM-Machine-Learning-and-Novel-Data-Applied-to-513703

Diaz, G., & fseasy. (2020). *stopwords-iso / stopword-zh*. GitHub. https://github.com/stopwords-iso/stopwords-zh

Gahagan, J. (2022, November 20). *Malaysia elects first ever hung parliament*. BBC. https://www.bbc.com/news/world-asia-63694710

Ghosh, S. (2021). Consumer confidence and consumer spending in Brazil: a nonlinear autoregressive distributed lag model analysis. *Arthaniti: Journal of Economic Theory and Practice*, *20*(1), 53-85. https://journals.sagepub.com/doi/full/10.1177/0976747919898906

Ghosh, S., & Ranjan, A. (2021). *A machine learning (ml) approach to GDP nowcasting: An Emerging market experience.* Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3980188

Gupta, V., Jain, N., Shubhm, S., Madan, A., Chaudhary, A., & Qin, X. (2021). Toward integrated CNN-based sentiment analysis of tweets for scarce-resource language—Hindi. *ACM Transactions on Asian Low-Resource Language Information Processing, 20*(5), 80(1)-80(23). https://doi.org/10.1145/3450447

Hegde, A., Chakravarthi, B. R., Shashirekha1, H. L., Ponnusamy, R., Navaneethakrishnan, S. C., Kumar, L. S., Thenmozhi, D., Karunakar, M., Sriram, S., & Aymen. S. (2023). Findings of the shared task on sentiment analysis in Tamil and Tulu code-mixed text. *Third Workshop on Speech and Language Technologies for Dravidian Languages,* 64-71. https://aclanthology.org/2023.dravidianlangtech-1.pdf

Huang, A., Wu, W., & Yu, T. (2020). Textual analysis for China's financial markets: a review and discussion. *China Finance Review International, 10*(1), 1-15. https://doi.org/10.1108/CFRI-08-2019-0134

Jasni, S. K., Aris, F. E. M., & Jamilat, V. S. (2022). *Nowcasting Malaysia's GDP with machine learning*. Department of Statistics Malaysia. https://drive.google.com/file/d/1o0InpFfcqyYlBbk4YaD9t0ntvWCKxff5/view

Jiang, F., Lee, J., Martin, X., & Zhou, G. (2019). Manager sentiment and stock returns. *Journal of Financial Economics*, *132*(1), 126–149. https://doi.org/10.1016/j.jfineco.2018.10.001

Juhro, S. M., & Iyke, B. N. (2020). Consumer confidence and consumption expenditure in Indonesia. *Economic Modelling*, *89*, 367–377. https://doi.org/10.1016/j.econmod.2019.11.001

Kalamara, E., Turrell, A., Redl, C., Kapetanios, G., & Kapadia, S. (2022). Making text count: Economic forecasting using newspaper text. *Journal of Applied Econometrics*, *37*(5), 896–919. https://doi.org/10.1002/jae.2907

Kant, D., Pick, A., & Winter, J. D. (2022, November). *Nowcasting GDP using machine learning algorithms.* (DNB Working Paper No. 754). https://www.dnb.nl/media/kq4pe4cr/working_paper_no_754.pdf

Karim, B., Choi, S. M., Iyer, T., Li, J., Ouattara, F., Tiffin, A. J., & Yao, J. (2022, May). *Overcoming data sparsity: A machine learning approach to track the real-time impact of COVID-19 in Sub-Saharan Africa.* (Working Paper No. 2022/088). https://www.imf.org/en/Publications/WP/Issues/2022/05/07/Sub-Saharan-Africa-Economic-Activity-GDP-Machine-Learning-Nowcasting-COVID-19-517646

Kaya, A. (2024, March 1). *How are geopolitical risks affecting the world economy?* Economics Observatory. https://www.economicsobservatory.com/how-are-geopolitical-risks-affecting-the-world-economy

Khan, H., & Upadhayaya, S. (2020). Does business confidence matter for investment?. *Empirical Economics, 59*, 1633–1665. https://doi.org/10.1007/s00181-019-01694-5

Kogilavani Shanmugavadivel, Sai Haritha Sampath, Pramod Nandhakumar, Mahalingam, P., Subramanian, M., Prasanna Kumar Kumaresan, & Ruba Priyadharshini. (2022). An analysis of machine learning models for sentiment analysis of Tamil code-mixed data. *Computer Speech & Language*, *76*, 101407–101407. https://doi.org/10.1016/j.csl.2022.101407

Lioudis, N. (2023, August 22). *How the balance of trade affects currency exchange rates.* Investopedia. https://www.investopedia.com/ask/answers/041515/how-does-balance-trade-impact-currency-exchange-rates.asp

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, *66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

Malaysian Institute of Economic Research. (2023, August 11). *MIER: Consumer sentiment and business conditions remain on downtrend in 2Q2023.* https://mier.org.my/mier-consumer-sentiment-and-business-conditions-remain-on-downtrend-in-2q2023/

Martin, L. C. (2019, August). *Machine learning vs traditional forecasting methods: An application to South African GDP*. (Stellenbosch Economic Working Papers WP12/2019). https://ideas.repec.org/p/sza/wpaper/wpapers326.html

Moudjari, L., & Akli-Astouati, K. (2020). An Experimental Study on Sentiment Classification of Algerian Dialect Texts. *Procedia Computer Science*, *176*, 1151–1159. https://doi.org/10.1016/j.procs.2020.09.111

Muchisha, N. D., Tamara, N., Andriansyah, & Soleh, A. M. (2021). Nowcasting Indonesia's GDP using machine learning algorithms. *Indonesian Journal of Statistics and Its Applications*, *5*(2), 355-368. https://mpra.ub.uni-muenchen.de/105235/1/MPRA_paper_105235.pdf

Muhammad Fakhrur Razi Abu Bakar, Norisma Idris, Liyana Shuib, & Norazlina Khamis. (2020). Sentiment analysis of noisy Malay text: State of art, challenges and future work. *IEEE Access*, *8*, 24687–24696. https://doi.org/10.1109/access.2020.2968955

Nakazawa, T. (2022, July). *Constructing GDP nowcasting models using alternative data*. (Bank of Japan Working Paper Series No.22-E-5). https://www.boj.or.jp/en/research/wps_rev/wps_2022/data/wp22e09.pdf

Nothman, J., Qin, H., & Yurchak, R. (2019). *Stop word lists in free open-source software packages*. https://doi.org/10.18653/v1/w18-2502

Office for National Statistics. (n.d.). *GDP monthly estimate, UK statistical bulletins.* https://www.ons.gov.uk/economy/grossdomesticproductgdp/bulletins/gdpmonthlyestimateuk/previousReleases

Ong, J. Y., Muhammad Mun'im Zabidi, Norhafizah Ramli, & Sheikh, U. U. (2020). Sentiment analysis of informal Malay tweets with deep learning. *IAES International Journal of Artificial Intelligence (IJ-AI), 9*(2), 212-220. DOI:10.11591/ijai.v9.i2.pp212-220

OpenDOSM. (2024-a). *Gross domestic product (GDP).* https://open.dosm.gov.my/dashboard/gdp

OpenDOSM. (2024-b). *[4Q 2023] Gross domestic product*. https://open.dosm.gov.my/publications/gdp_2023-q4

Putra, R. A. A., & Arini, S. (2020). *Measuring the economics of a pandemic: How people mobility depict economics?. An evidence of people's mobility data towards economic activities.* https://www.semanticscholar.org/paper/Measuring-the-Economics-of-a-Pandemic%3A-How-People-Achyunda-Putra/1c8f314cd2854af3aa73f9336799714edaf721c7

Qeqe, B., & Sibanda, K. (2022). The relationship between business confidence and private sector investment: Evidence from selected sectors of the South African Economy. *African Journal of Business and Economic Research, 17*(1), 205-218. DOI: https://doi.org/10.31920/1750-4562/2022/v17n1a9

Rambaccussing, D., & Kwiatkowski, A. (2020). Forecasting with news sentiment: Evidence with UK newspapers. *International Journal of Forecasting, 36*(4), 1501–1516. https://doi.org/10.1016/j.ijforecast.2020.04.002

Rana, T. A., Shahzadi, Rana, T., K., Arshad, A., & Tubishat, M. (2021). An unsupervised approach for sentiment analysis on social media short text classification in Roman Urdu. *ACM Transactions on Asian Low-Resource Language Information Processing, 21*(2), 28(1)-28(16). https://doi.org/10.1145/3474119

Richardson, A., Mulder, T. V. F., & Vehbi, T. (2021). Nowcasting GDP using machine-learning algorithms: A real-time assessment. *International Journal of Forecasting, 37*(2), 941-948. https://doi.org/10.1016/j.ijforecast.2020.10.005

Shapiro, A. H., Sudhof, M., & Wilson, D. (2022). Measuring news sentiment. *Journal of Econometrics,* 228(2022), 221-243. https://linkinghub.elsevier.com/retrieve/pii/S0304407620303535

Statistics Explained. (2023, July 21). *Beginners: GDP - What is gross domestic product (GDP)?*. https://ec.europa.eu/eurostat/statistics-explained/SEPDF/cache/63211.pdf

Takahashi, K. (2022). *Nowcasting economic activity with mobility data*. https://www.bis.org/ifc/publ/ifcb59_36_rh.pdf

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications*, *34*(4), 2622–2629. https://doi.org/10.1016/j.eswa.2007.05.028

The Economic Times. (2024, March 14). What is 'Fiscal policy.' https://economictimes.indiatimes.com/definition/fiscal-policy

Thorsrud, L. A. (2016, December). Words are the new numbers: A newsy coincident index of business cycles. (Working Paper, No. 21/2016). https://hdl.handle.net/11250/2495606

Tripathi, S., Muhr, D., Brunner, M., Jodlbauer, H., Dehmer, M., & Emmert-Streib, F. (2021). Ensuring the robustness and reliability of data-driven knowledge discovery models in production and manufacturing. *Frontiers in Artificial Intelligence*, *4*, 576892. https://doi.org/10.3389/frai.2021.576892

Walker, N. (2024). *Conflict in Ukraine: A timeline (current conflict, 2022-present)*. House of Commons Library. https://researchbriefings.files.parliament.uk/documents/CBP-9847/CBP-9847.pdf

Wang, D., & Alfred, R. (2020). A review on sentiment analysis model for Chinese Weibo text, *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 456-463, https://www.researchgate.net/publication/342650210_A_Review_on_Sentiment_Analysis_Model_for_Chinese_Weibo_Text

Wei, Z., Liu, W., Zhu, G., Zhang, S., & Hsieh, M-Y. (2022) Sentiment classification of Chinese Weibo based on extended sentiment dictionary and organisational structure of

comments. *Connection Science, 34*(1), 409-428. DOI: 10.1080/09540091.2021.2006146

Wong, C-H. (2023). Introduction: Hung parliament, coalition government and the rise of the Islamists-Malaysia after the 2022 election. *The Round Table: The Commonwealth Journal of International Affairs*, *112*(3), 207-212. https://doi.org/10.1080/00358533.2023.2219522

Xiao, R., & McEnery, T. (2008). Negation in Chinese: A corpus-based study. *Journal of Chinese Linguistics*, *36*(2), 274-330. https://eprints.lancs.ac.uk/id/eprint/70/1/Negation_in_Chinese_for_JEAL.pdf

Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE Access*, *8*, 23522–23530. https://doi.org/10.1109/access.2020.2969854

Zhang, Q., He, N., & Xu, H. (2023). Nowcasting Chinese GDP in a data-rich environment: Lessons from machine learning algorithms. *Economic Modelling*, *122*, 106204. https://doi.org/10.1016/j.econmod.2023.106204

Zhang, S., Wei, Z., Wang, Y., & Liao, T. (2018). Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems, 81*, 395–403. https://doi.org/10.1016/j.future.2017.09.048

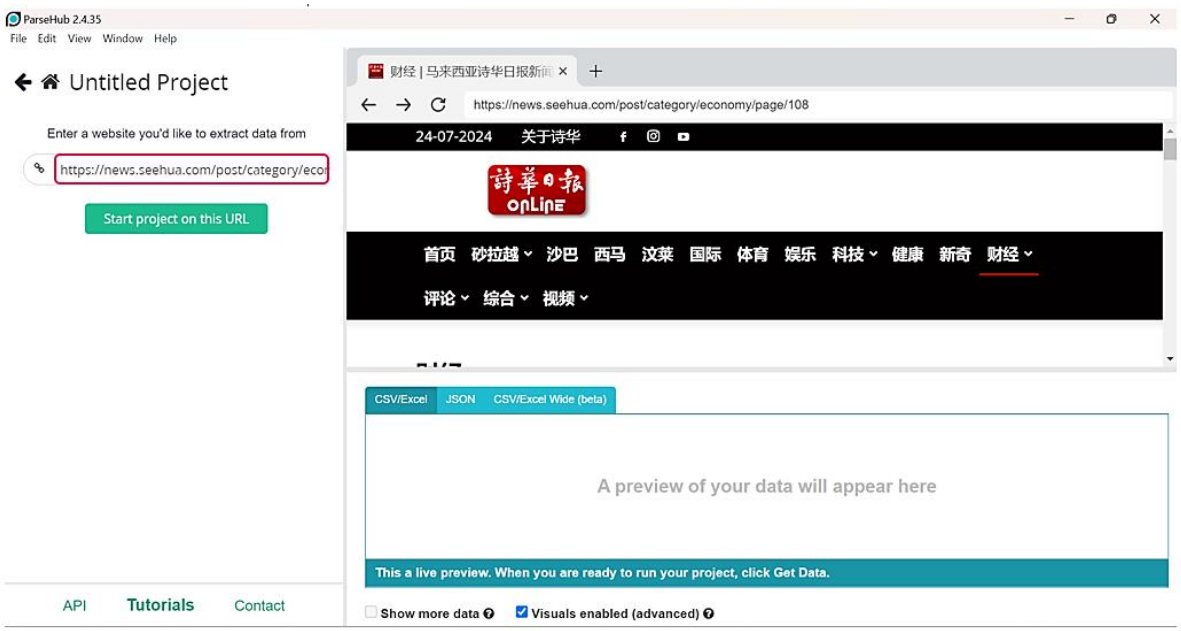
Zulkefly Abdul Karim, Fairul Shah Rizat Muhamad Fahmi, Bakri Abdul Karim & Mohamed Aseel Shokr. (2022). Market sentiments and firm-level equity returns: Panel evidence of Malaysia. *Economic Research-Ekonomska Istraživanja*, *35*(1), 5253-5272. DOI: 10.1080/1331677X.2021.2025126

APPENDICES

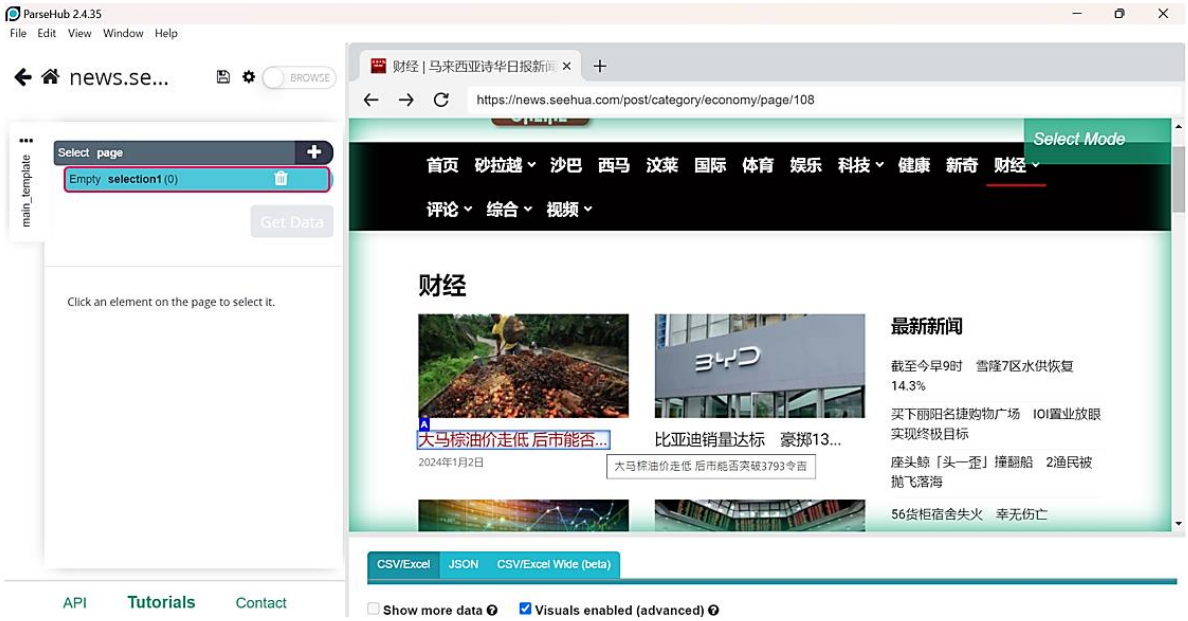
Appendix A: Proof of Payment (MIER Data Purchase)

Appendix B: Research Letters

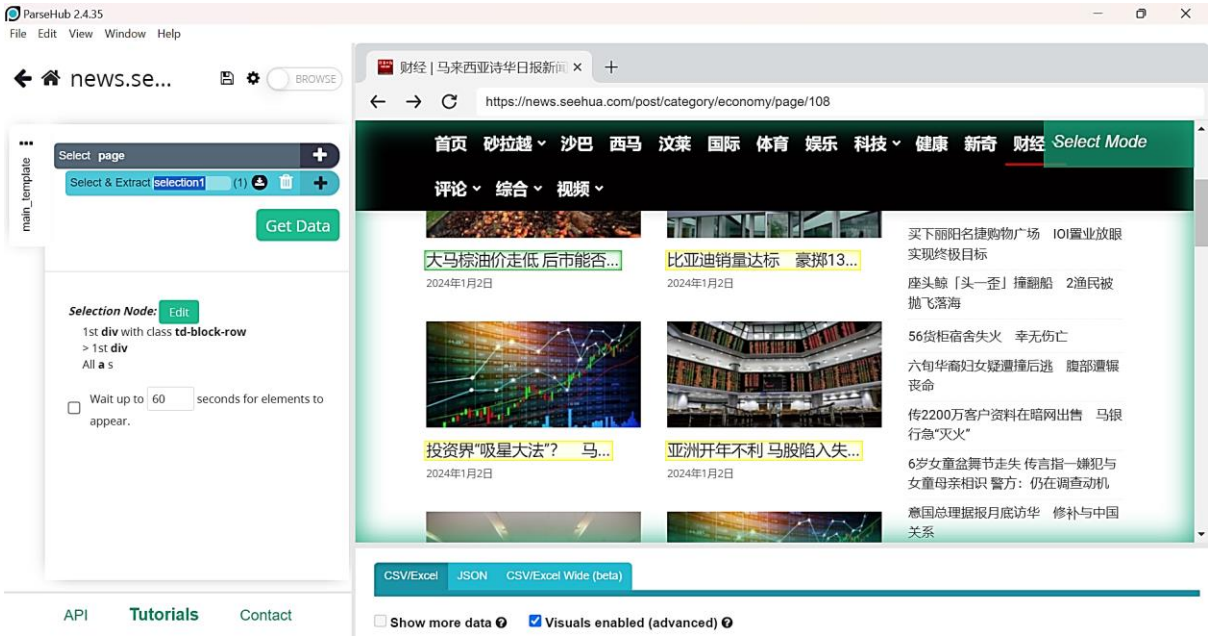
Appendix C: Step-by-Step Guide in ParseHub



Step 1: Enter the website URL as indicated by the pink highlighted area and click on the "Start project on this URL" option.

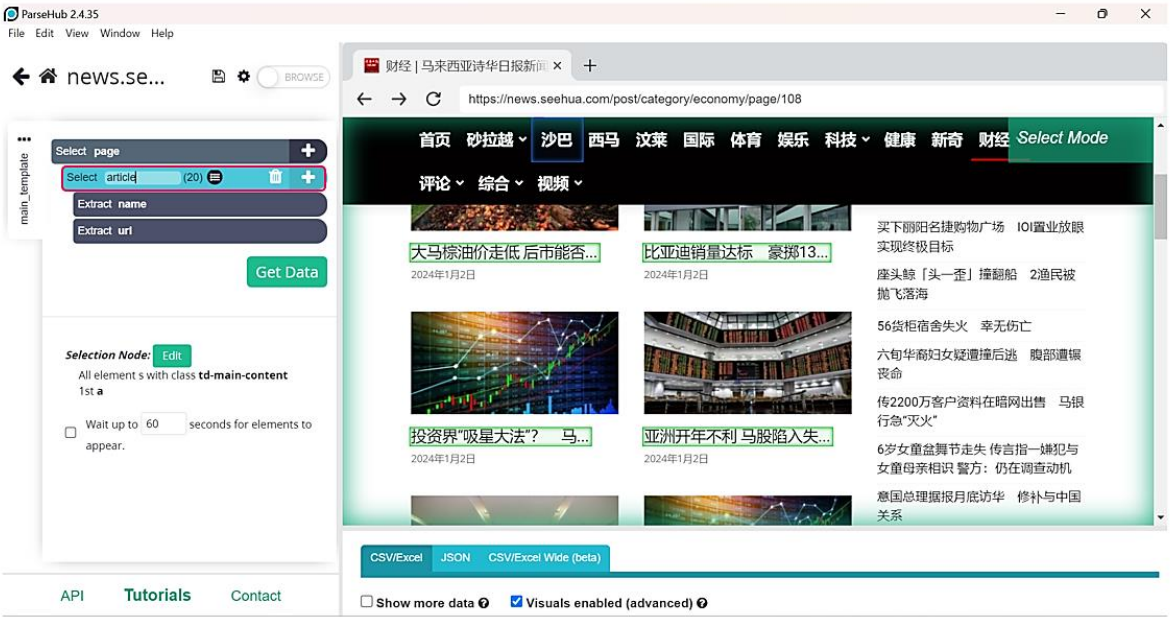


Step 2: Click on the first article title (“大马棕油价走低后市能否…”) as indicated by the blue highlighted area.

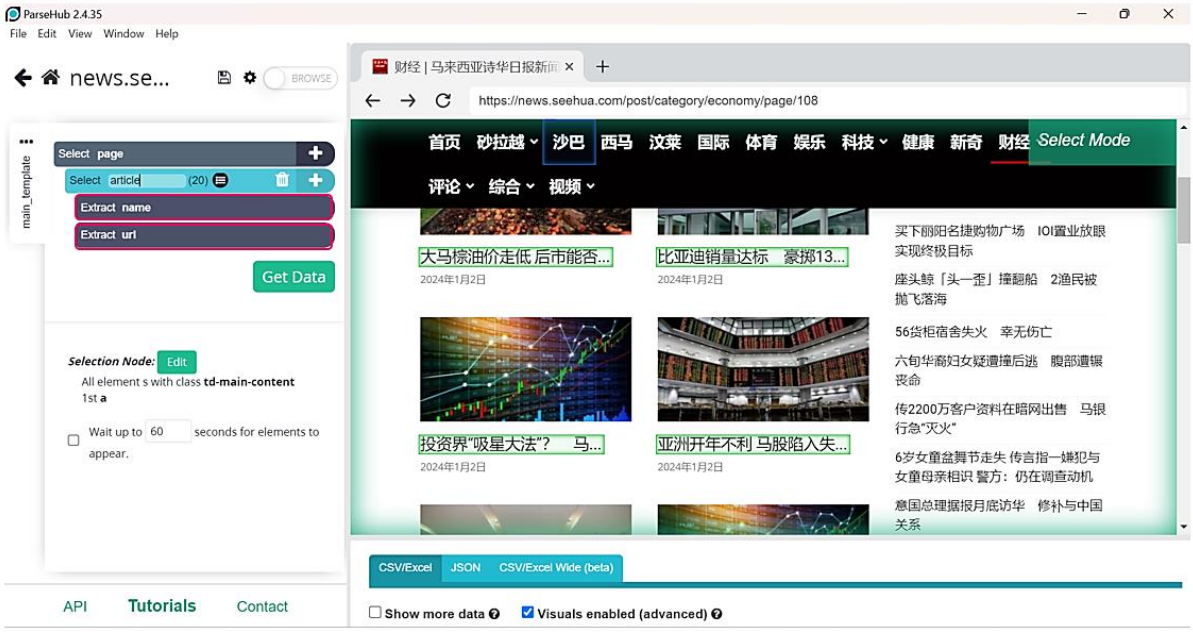


Step 3: Click on *selection1* and change its name to *article*.

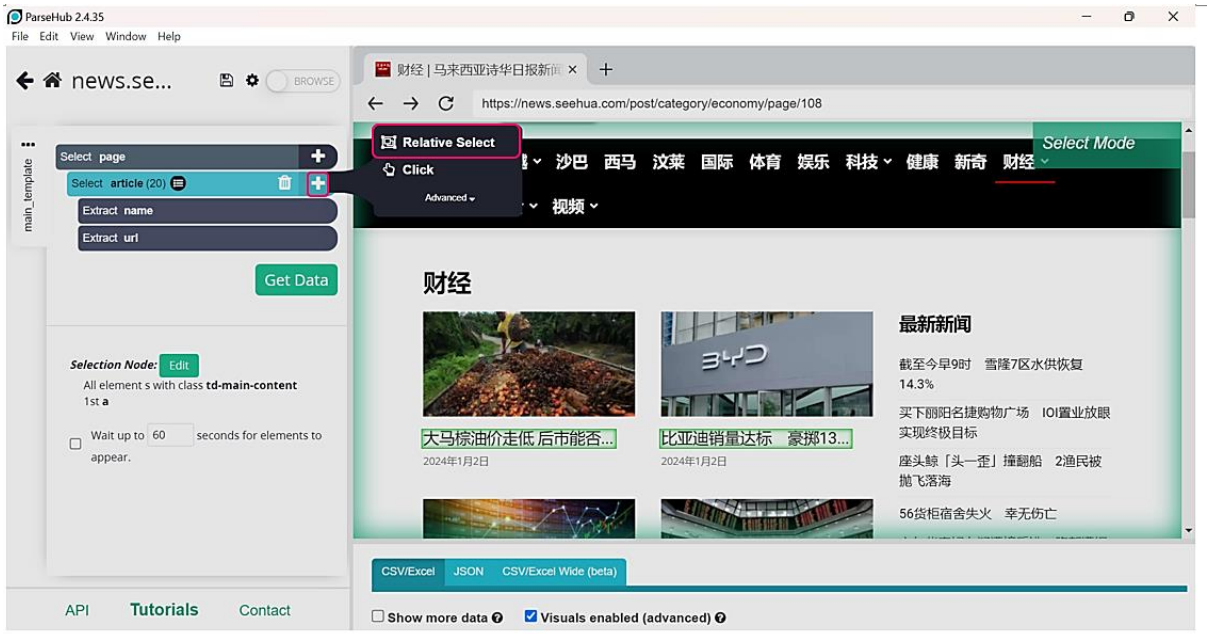
Step 4: Elements currently selected are highlighted in green. Similar elements suggested by ParseHub for selection will appear in yellow borders. Click on the next suggested article title (“比亚迪销量达标 豪掷 13…”).



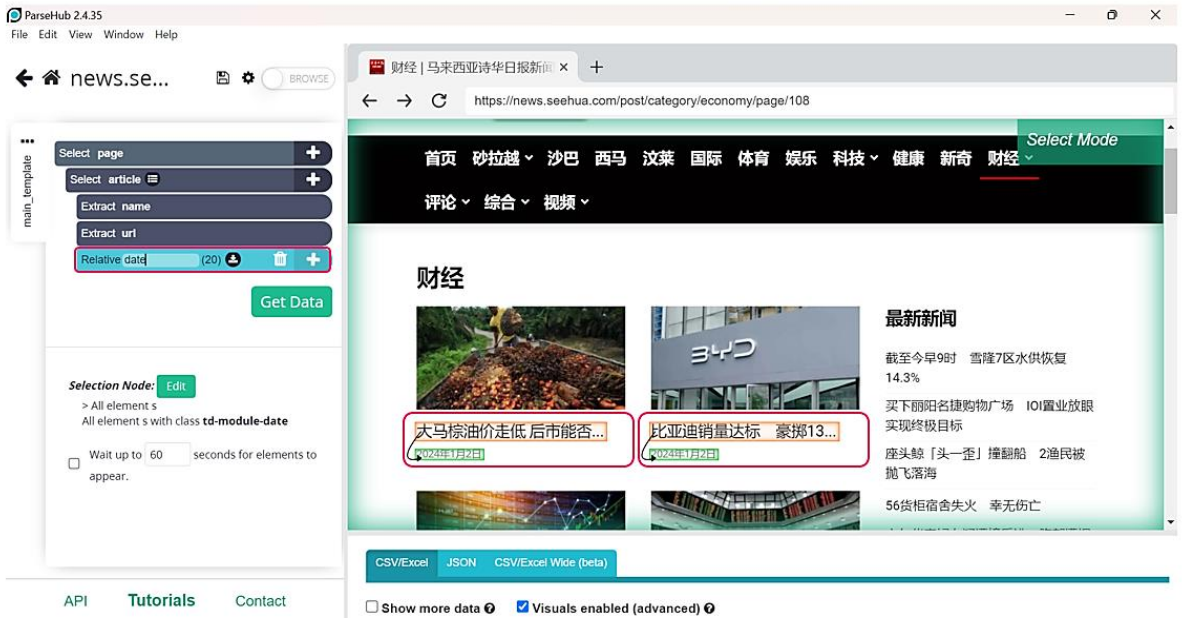
Step 5: The number (20) to the right of the *select articles* command indicates the number of elements currently selected. Check for the total number of articles present on the page. The number of elements currently selected should be the same as the number of the desired elements to be selected. If not, the process outlined in *Step 4* should be repeated.



Step 6: The two *Extract name* and *Extract url* commands will appear.



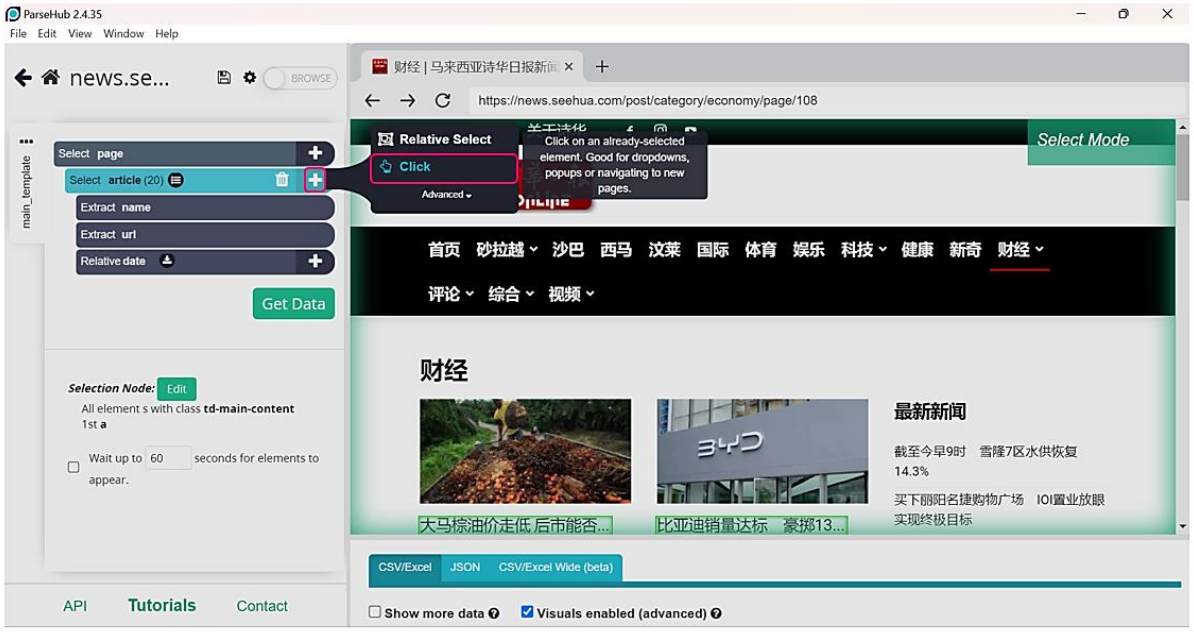
Step 7: To extract the date of publication for each article, click on the pink highlighted area of the plus sign next to the *Select article* command. Proceed to click on the *Relative Select* option.



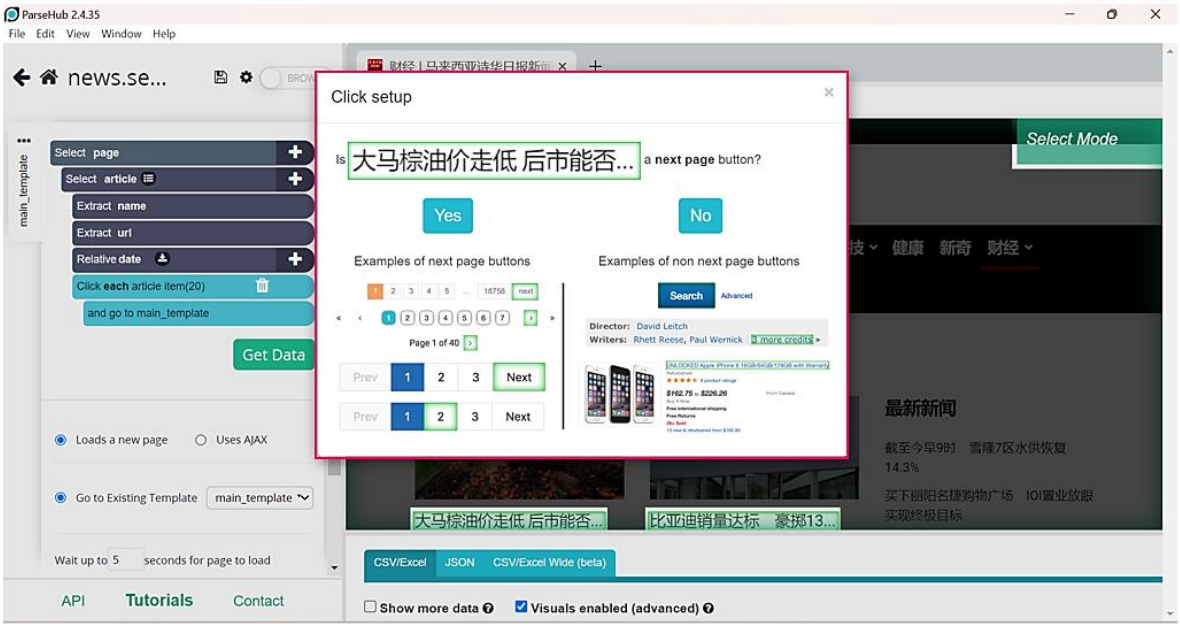
Step 8: Click on *selection1* and rename it to *date*.

Step 9: To create a relative selection, click on one of the article titles (orange highlighted area) and then then click on the article publication date (green highlighted area) corresponding to that article. A black arrow will appear for the pair.

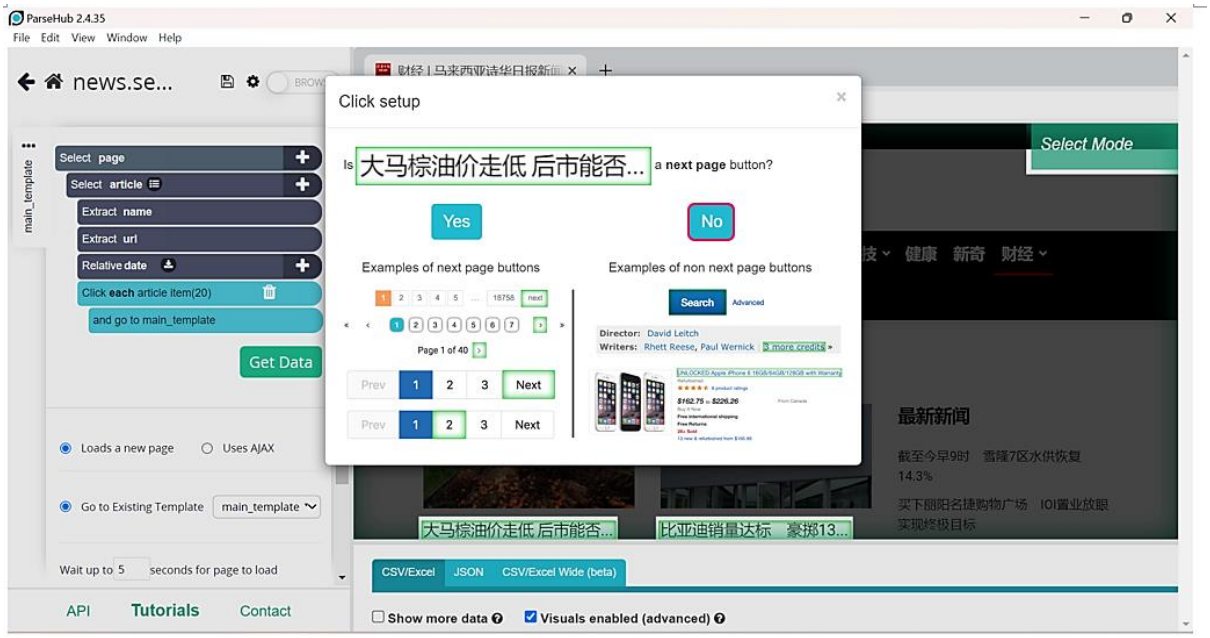
Step 10: To help ParseHub learn about the pattern of extraction, click on another article title (orange highlighted area) then on its publication date (green highlighted area). A black arrow will appear for each of the subsequent pairs.



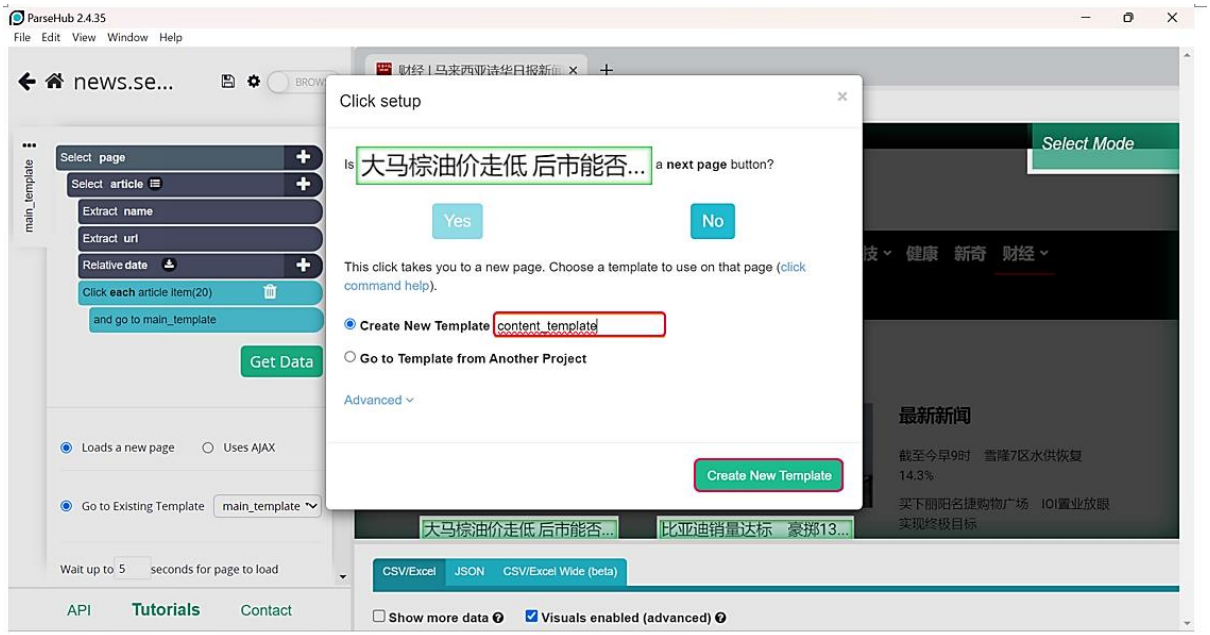
Step 11: To get the contents of the articles, ParseHub needs to be redirected to another page for the extraction to take place. Since all of the article titles have already been selected, ParseHub can now be directed to click on each article and extract information from its corresponding details page. To do that, click on the pink highlighted area of the plus sign next to the *Select article* command. Proceed to click on the *Click* option.



Step 12: A *Click setup* pop-up window would appear as indicated by the pink highlighted area.

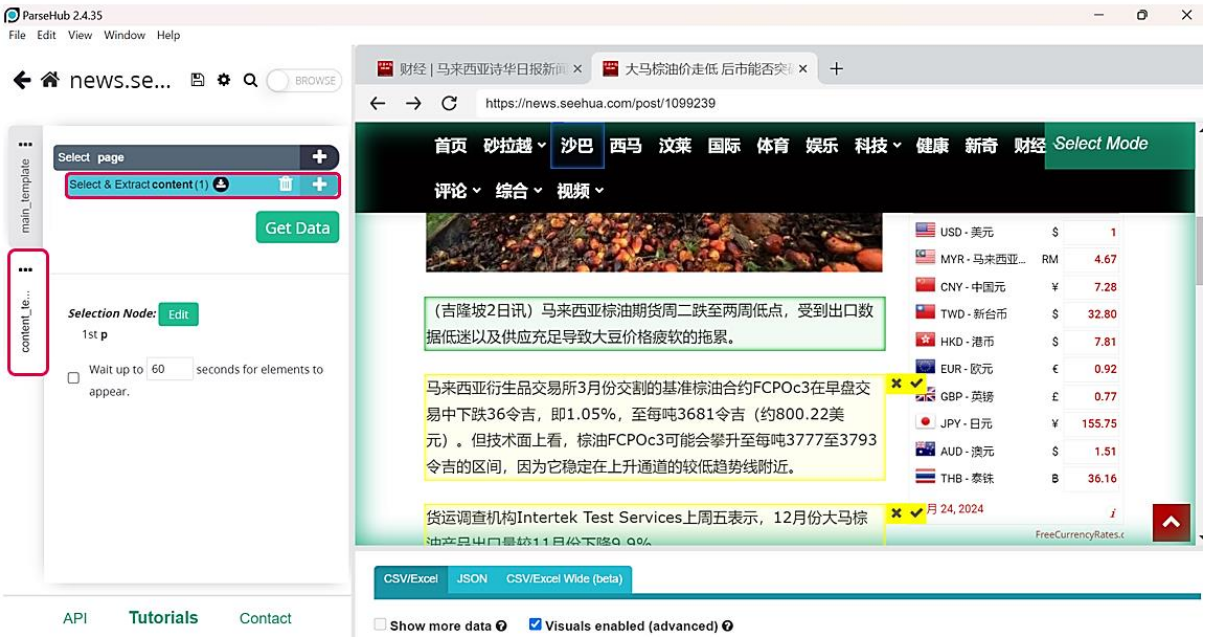


Step 13: Since ParseHub needs to click on a link to get to the details page, choose the *No* option as shown in the pink highlighted area.



Step 14: Pick the *Create New Template* option and next to it, rename the template to *content_template*.

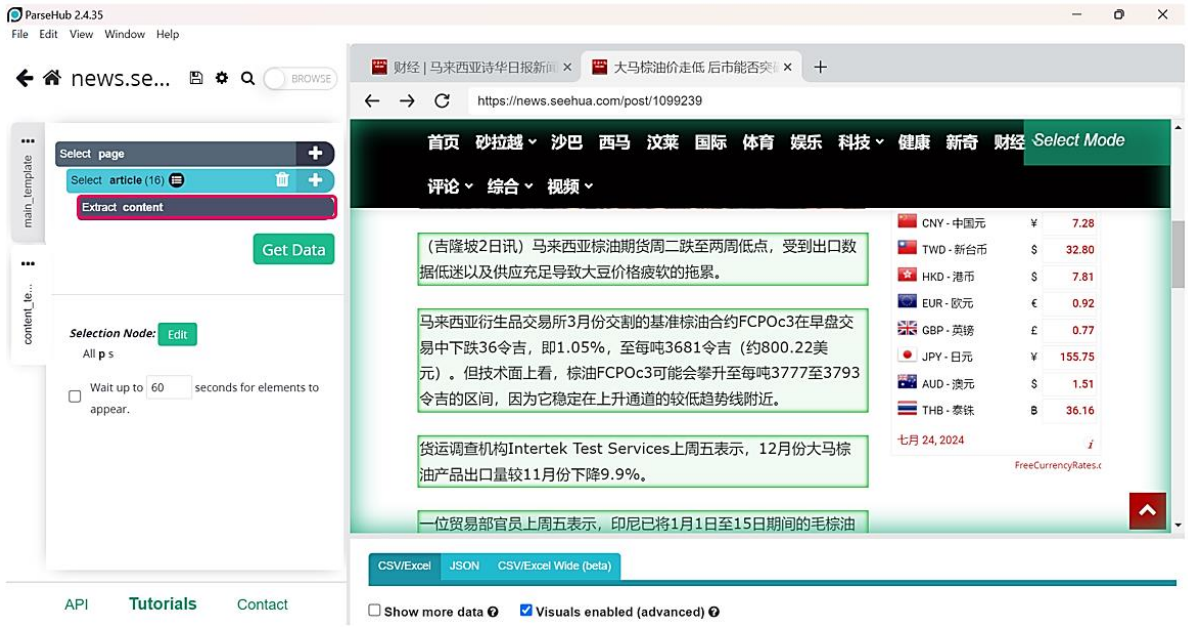
Step 15: Click on the *Create New Template* option.



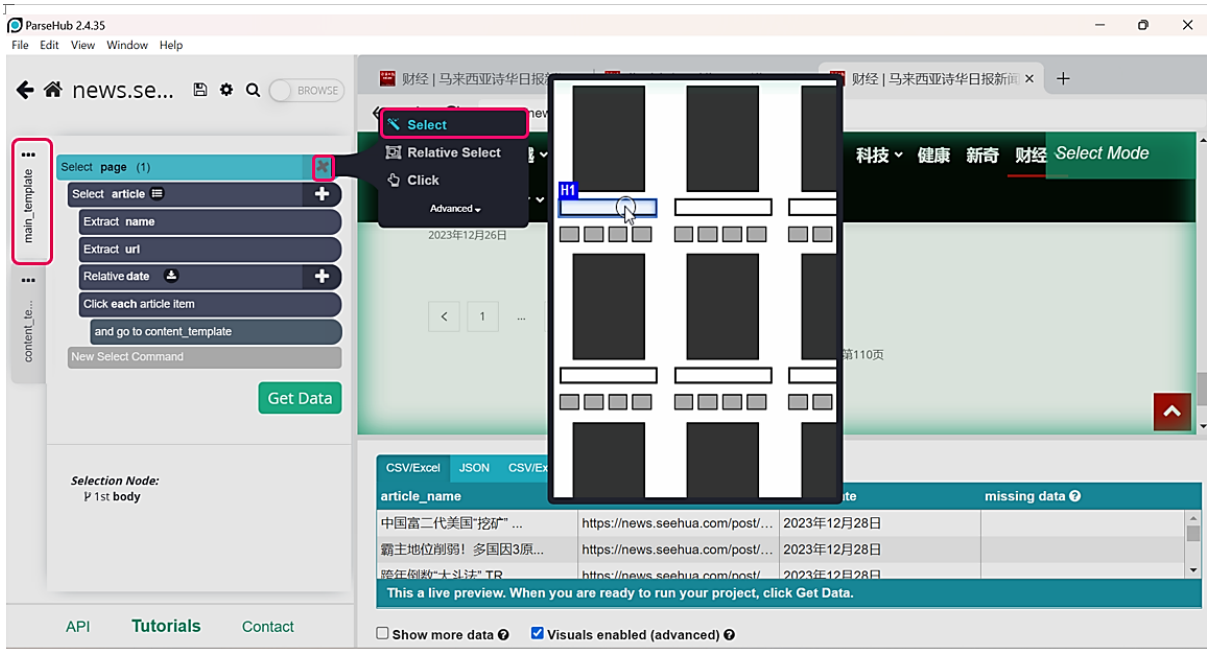
Step 16: ParseHub will be directed to the *content_template* where it can now extract data from the article details pages.

Step 17: Hover over the first paragraph of the content and click on it. A green border would appear around it. Continue to click on the rest of the yellow highlighted area to teach ParseHub the pattern of extraction.

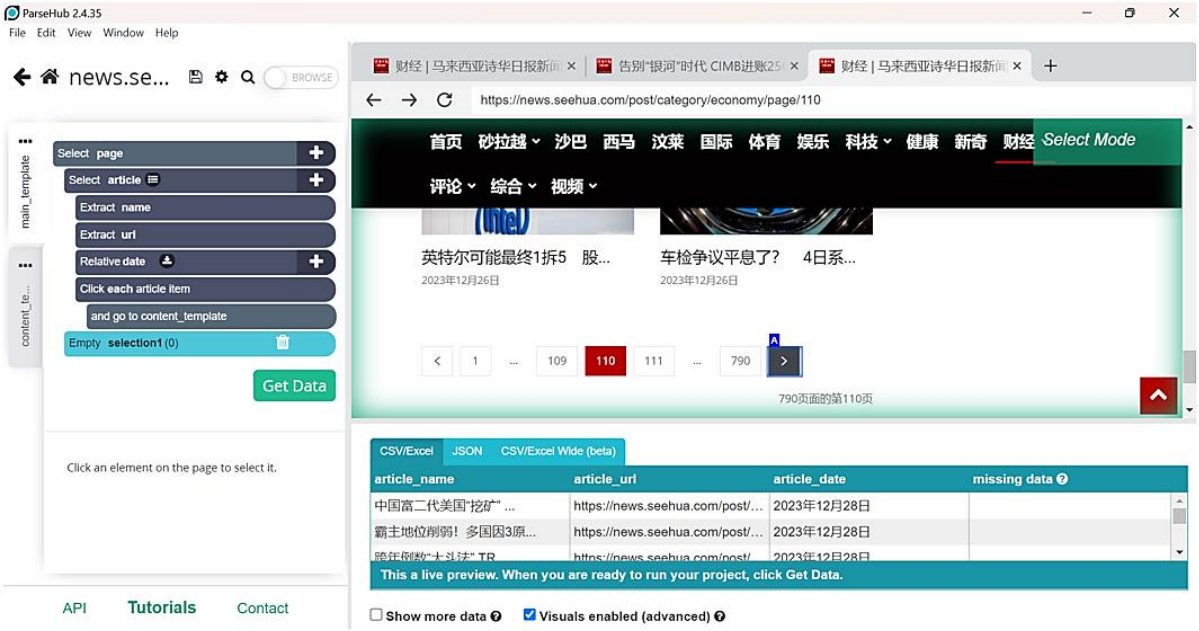
Step 18: Click on *selection1* of the *Select & Extract* command and rename it to *content*.



Step 19: The *Extract* command will appear. Click on *name* of the *Extract* command and rename it to *content*.

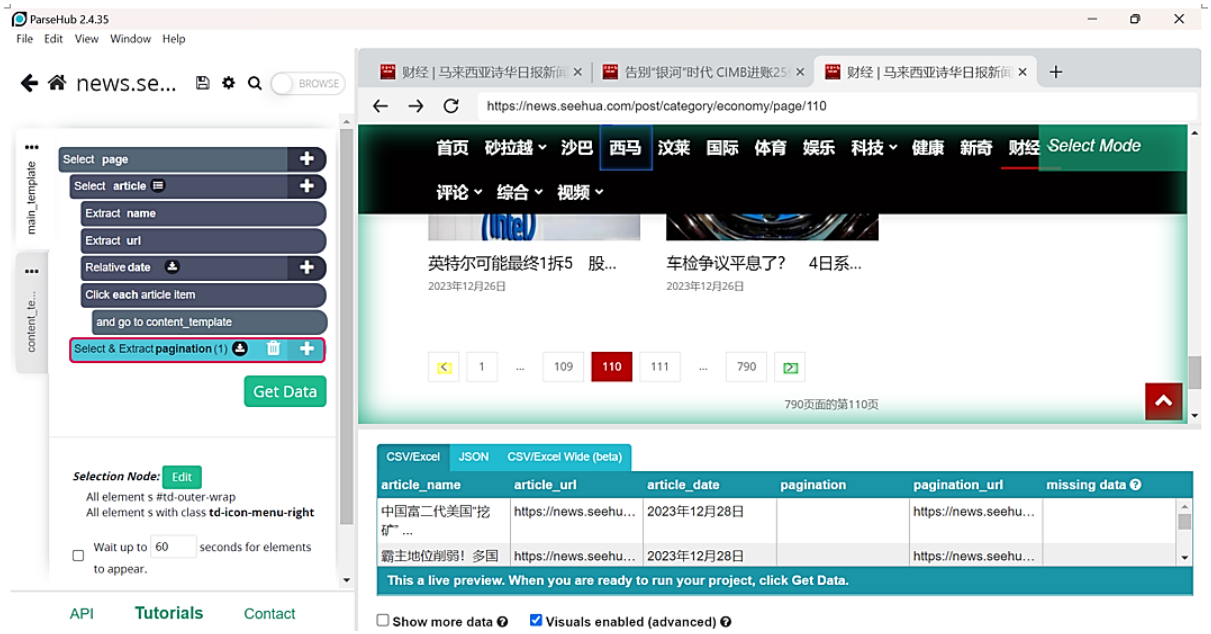


Step 20: Go back to the *main_template* tab and click on the plus sign next to the *Select page* command. Proceed to click on the *Select* option.

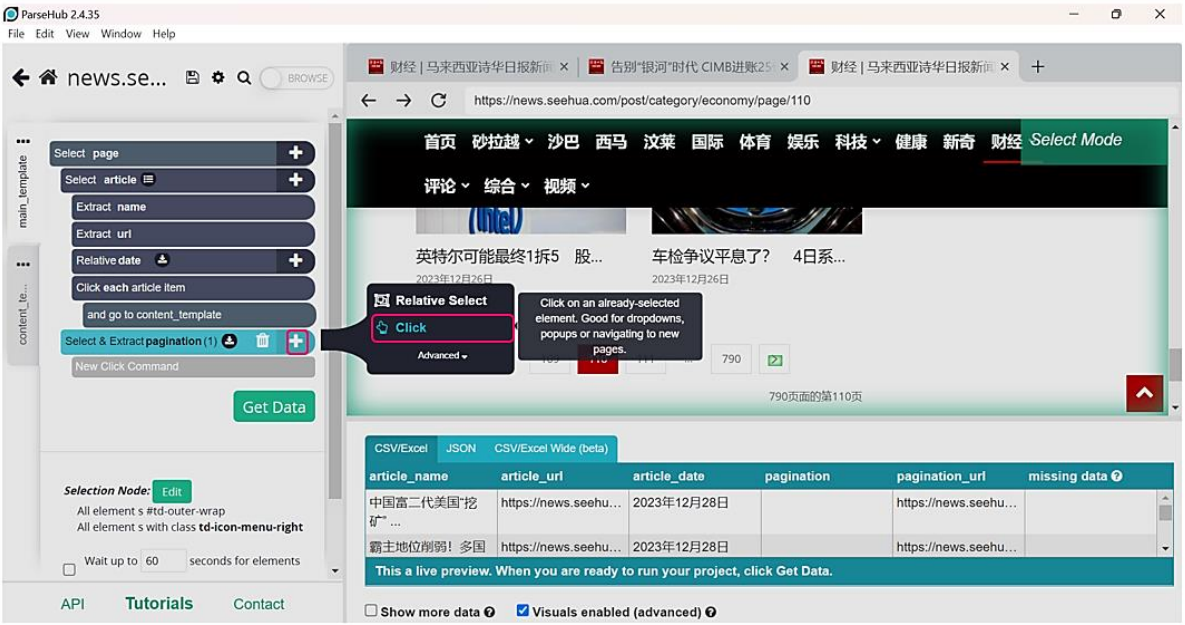


Step 21: Click on *selection1* of the *Select & Extract* command and rename it to *pagination*.

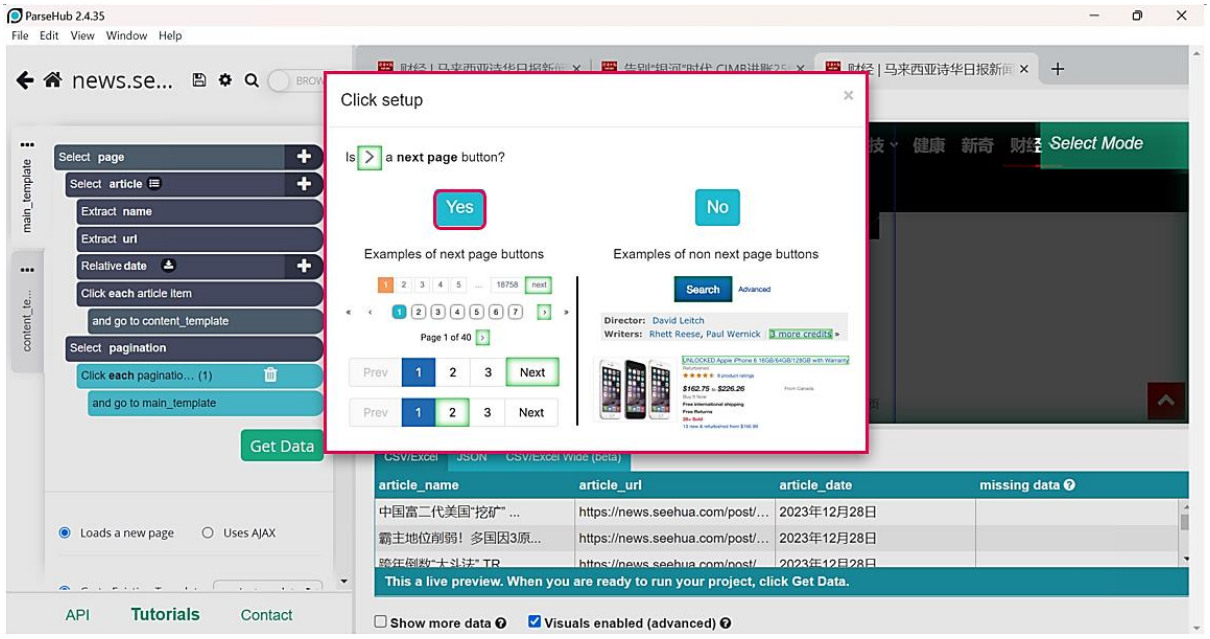
Step 22: Click on the arrow icon on the website as indicated by the blue highlighted area.



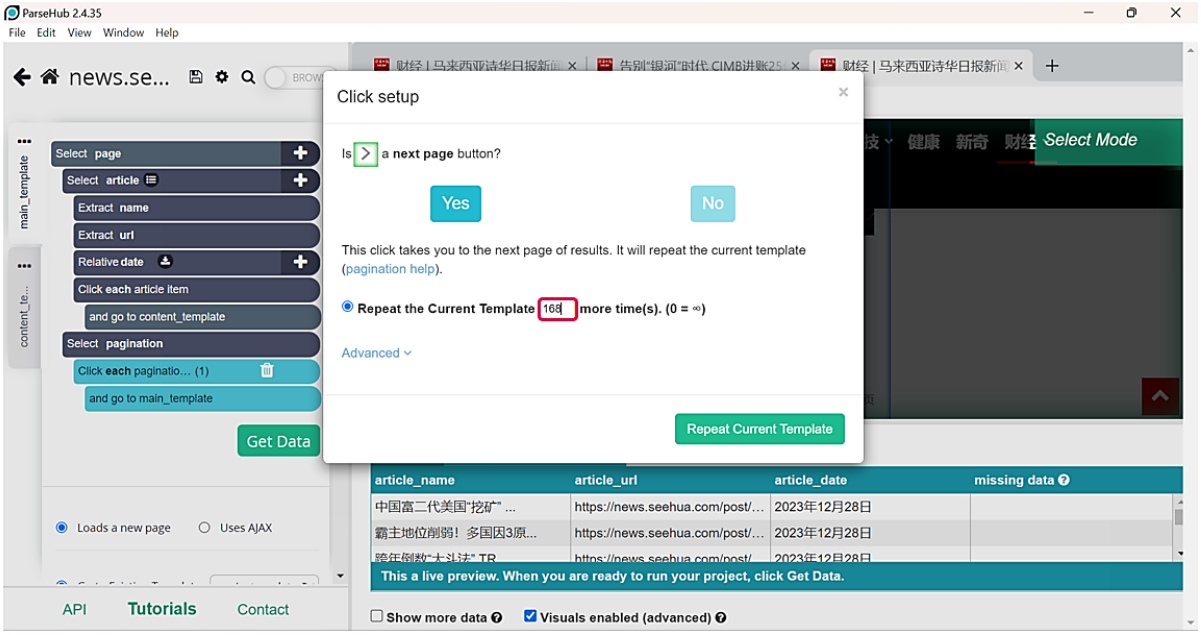
Step 23: The *Select & Extract pagination* command will appear.



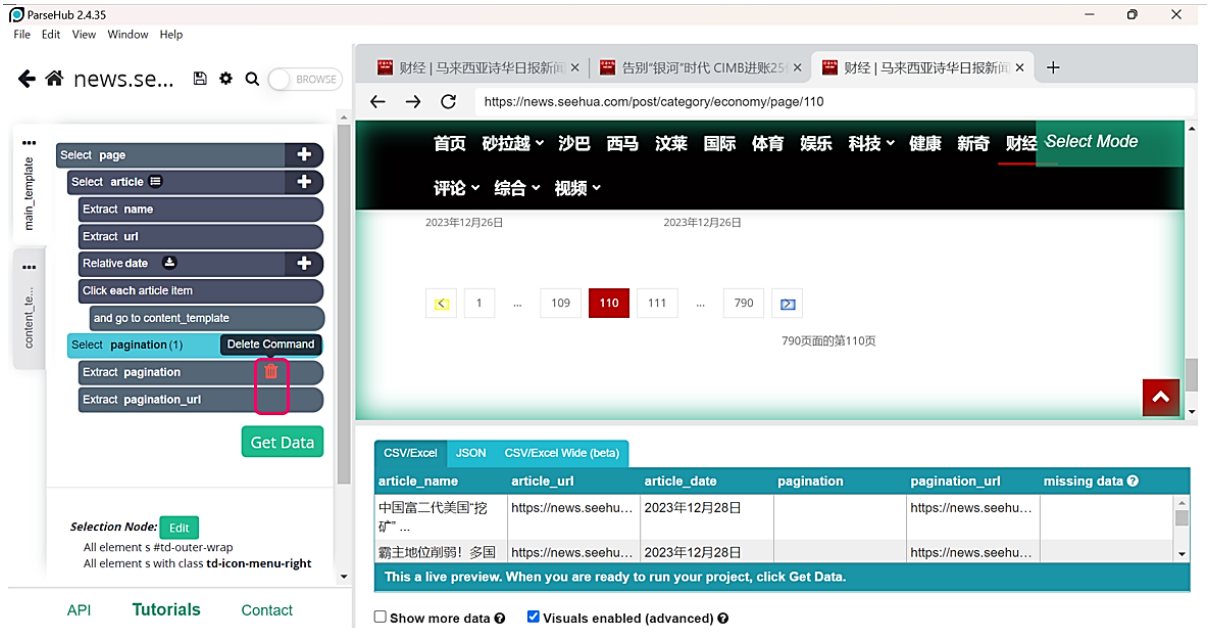
Step 24: Click on the pink highlighted area of the plus sign next to the *Select & Extract pagination* command. Proceed to click on the *Click* option.



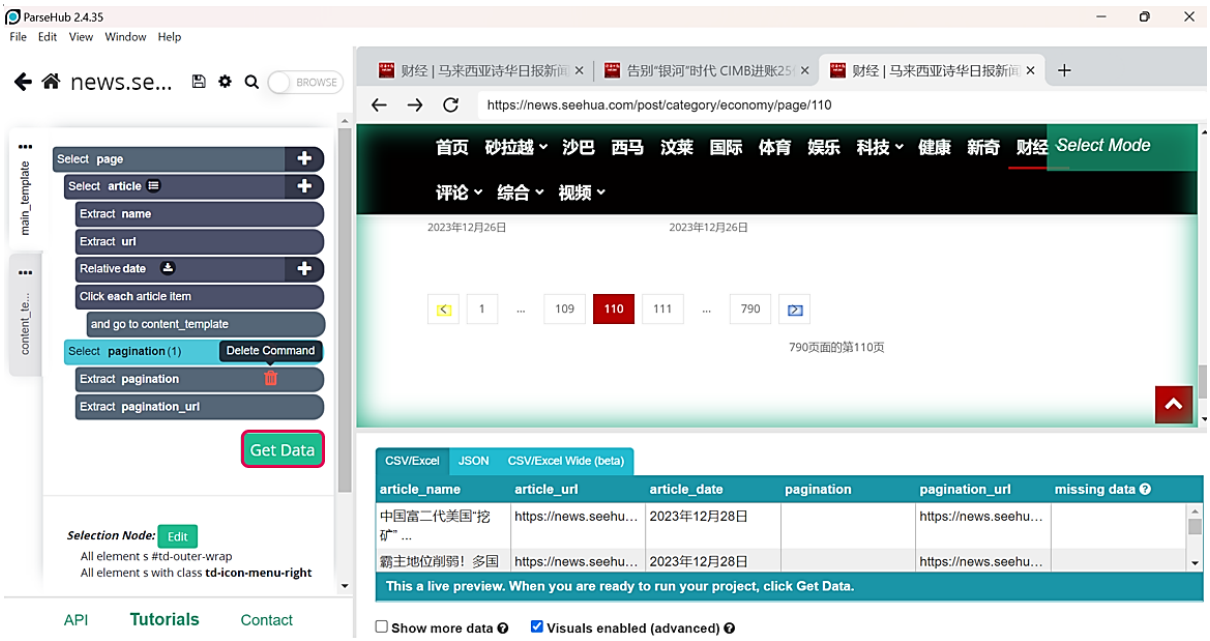
Step 25: A *Click setup* pop-up window would appear as indicated by the pink highlighted area. Choose the *Yes* option.



Step 26: In the pink highlighted area, enter the number of times to repeat the template.



Step 27: The *Extract pagination* and *Extract Pagination_url* command will appear. Delete both the commands by clicking on the rubbish bin icon next to them.



Step 28: Lastly, click on the *Get Data* option and the scrapping process will be initiated.

Appendix D: Initialization of the Data Analysis Process

```
PART A: Initialization of the Data Analysis Process
```

```
# Import necessary libraries for data manipulation, statistical modeling, text processing, plotting, and machine learning models
import pandas as pd  # For data manipulation and analysis
import numpy as np  # For numerical operations
import statsmodels.api as sm  # For statistical models and tests
import unicodedata  # For Unicode character database
import jieba  # For Chinese text segmentation
import re  # For regular expression operations
import json  # For JSON manipulation
import matplotlib.pyplot as plt  # For plotting and visualization
from sklearn.metrics import mean_squared_error, mean_absolute_error  # For evaluation metrics
from sklearn.svm import SVR  # For Support Vector Regressor model
from sklearn.ensemble import RandomForestRegressor  # For Random Forest Regressor model
from sklearn.linear_model import LinearRegression, Lasso, Ridge  # For linear models
from sklearn.preprocessing import StandardScaler  # For feature scaling
from sklearn.model_selection import GridSearchCV  # For hyperparameter tuning
from sklearn.pipeline import make_pipeline  # For creating machine learning pipelines
from statsmodels.stats.outliers_influence import variance_inflation_factor  # For calculating Variance Inflation Factor (VIF)
from xgboost import XGBRegressor  # For XGBoost Regressor model
```

Appendix E: Initial Exploratory Data Analysis of the Textual Dataset

```
PART B: Initial Exploratory Data Analysis of the Textual Dataset
```

```
# Read the Scrapped CSV file
encodings = ['utf-8', 'gb18030', 'big5']
for encoding in encodings:
    try:
        scrapped_df = pd.read_csv('Scrapped (See Hua).csv', encoding=encoding)
        print(f"Successfully read the file with encoding: {encoding}")
        break
    except Exception as e:
        print(f"Failed to read the file with encoding: {encoding}")
        print(f"Error: {e}")
```

```
Successfully read the file with encoding: utf-8
```

```
# Perform Exploratory Data Analysis (EDA) on the scrapped data
# Display the first few rows
print("First few rows of the dataset:")
print(scrapped_df.head())
```

```
First few rows of the dataset:
                article_name                          article_url  \
0         全球富豪身价狂涨 中国富人财产反而缩水  https://news.seehua.com/post/787364
1       开市即迎来套利压力  马股早盘跌18.85点  https://news.seehua.com/post/788101
2  2022首个交易日出师不利  马股全天下滑18.48点  https://news.seehua.com/post/788243
3      亚航拟改名为CAPITAL A BERHAD  https://news.seehua.com/post/788313
4      与大市背道而驰  富时大马综合指数跌9.52点  https://news.seehua.com/post/788684

  article_date                          article_article_content
0    2022年1月1日  这2年在新冠病毒肆虐期间，全球富豪资产大幅增长，不过中国科技富豪资产大失血，根据彭博亿万富豪...
1    2022年1月3日  （吉隆坡3日讯）2022年首个交易日，亚洲股市开盘表现普遍平平无奇，唯独马股出师不利，上周五...
2    2022年1月3日  （吉隆坡3日讯）由于区域多个股市仍未开市，导致亚洲市场淡静。马股首个交易日出师不利，全天下滑...
3    2022年1月3日  （吉隆坡3日讯）亚洲航空（AirAsia，5099，主板消费股）董事局建议将公司的名字改为C...
4    2022年1月4日  （吉隆坡4日讯）美国隔夜股市上涨，带动亚洲股市周二升多跌少。不过，马股仍延续周一跌势，与大市...
```

```
# Drop the 'article_url' column
scrapped_df = scrapped_df.drop(columns=['article_url'])
```

```python
# Rename the columns
scrapped_df = scrapped_df.rename(columns={
    'article_name': 'Title',
    'article_date': 'Date',
    'article_article_content': 'Content'
})
```

```python
# Rearrange the columns
scrapped_df = scrapped_df[['Date', 'Title', 'Content']]
```

```python
# Save the modified DataFrame to a new CSV file with utf-8 encoding
scrapped_df.to_csv('See Hua (New).csv', index=False, encoding='utf-8')
```

```python
# Read the now modified CSV file
encodings = ['utf-8', 'gb18030', 'big5']
for encoding in encodings:
    try:
        df = pd.read_csv('See Hua (New).csv', encoding=encoding)
        print(f"Successfully read the file with encoding: {encoding}")
        break
    except Exception as e:
        print(f"Failed to read the file with encoding: {encoding}")
        print(f"Error: {e}")
```

```
Successfully read the file with encoding: utf-8
```

```python
# Perform Exploratory Data Analysis (EDA) on the data
# Display the first few rows
print("First few rows of the dataset:")
print(df.head())
```

```
First few rows of the dataset:
        Date                        Title  \
0  2022年1月1日          全球富豪身价狂涨 中国富人财产反而缩水
1  2022年1月3日        开市即迎来套利压力  马股早盘跌18.85点
2  2022年1月3日  2022首个交易日出师不利  马股全天下滑18.48点
3  2022年1月3日       亚航拟改名为CAPITAL A BERHAD
4  2022年1月4日       与大市背道而驰  富时大马综合指数跌9.52点

                                             Content
0  这2年在新冠病毒肆虐期间，全球富豪资产大幅增长，不过中国科技富豪资产大失血，根据彭博亿万富豪...
1  （吉隆坡3日讯）2022年首个交易日，亚洲股市开盘表现普遍平平无奇，唯独马股出师不利，上周五...
2  （吉隆坡3日讯）由于区域多个股市仍未开市，导致亚洲市场淡静。马股首个交易日出师不利，全天下滑...
3  （吉隆坡3日讯）亚洲航空（AirAsia，5099，主板消费股）董事局建议将公司的名字改为C...
4  （吉隆坡4日讯）美国隔夜股市上涨，带动亚洲股市周二升多跌少。不过，马股仍延续周一跌势，与大市...
```

```python
# Display the summary of the DataFrame
print("\nSummary of the DataFrame:")
print(df.info())
```

```
Summary of the DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3361 entries, 0 to 3360
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Date     3361 non-null   object
 1   Title    3361 non-null   object
 2   Content  3361 non-null   object
dtypes: object(3)
memory usage: 78.9+ KB
None
```

```python
# Display summary statistics of the DataFrame
print("\nSummary statistics:")
print(df.describe(include='all'))
```

```
Summary statistics:
              Date                  Title  \
count         3361                   3361   
unique         576                   3354   
top     2023年8月25日  次季业绩符预期 天地通数码实现3大财务目标   
freq            28                      2   

                                                  Content
count                                                3361
unique                                               3349
top     （吉隆坡22日讯）美联储主席鲍威尔暗示5月再升息，加上美债收益率上涨导致美股承压，隔夜美股3...
freq                                                    2
```

Appendix F: Initial Exploratory Data Analysis of the Numerical Datasets

```
PART C: Initial Exploratory Data Analysis of the Numerical Datasets
```

```
PART C(1): MIER Dataset
```

```
# Load MIER BCI and CSI data from Excel
bci_df_eda = pd.read_excel('Data BCI_CSI_2022_2023.xlsx', sheet_name='BCI Formatted ')
csi_df_eda = pd.read_excel('Data BCI_CSI_2022_2023.xlsx', sheet_name='CSI Formatted')
```

```
# Perform EDA on the BCI data
print("\nFirst few rows of the BCI dataset:")
print(bci_df_eda.head())
```

```
First few rows of the BCI dataset:
  Quarter  BCI Values
0  2022Q1       101.0
1  2022Q2        96.2
2  2022Q3        99.8
3  2022Q4        85.9
4  2023Q1        95.4
```

```
# Perform EDA on the CSI data
print("\nFirst few rows of the CSI dataset:")
print(csi_df_eda.head())
```

```
First few rows of the CSI dataset:
  Quarter  CSI Values
0  2022Q1       108.9
1  2022Q2        86.0
2  2022Q3        98.4
3  2022Q4       105.3
4  2023Q1        99.2
```

```python
# Display the summary statistics of the bci_df_eda DataFrame
print("\nSummary of the bci_df_eda DataFrame:")
print(bci_df_eda.info())
```

```
Summary of the bci_df_eda DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Quarter      8 non-null      object 
 1   BCI Values   8 non-null      float64
dtypes: float64(1), object(1)
memory usage: 260.0+ bytes
None
```

```python
# Display the summary statistics of the csi_df_eda DataFrame
print("\nSummary of the csi_df_eda DataFrame:")
print(csi_df_eda.info())
```

```
Summary of the csi_df_eda DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Quarter      8 non-null      object 
 1   CSI Values   8 non-null      float64
dtypes: float64(1), object(1)
memory usage: 260.0+ bytes
None
```

```python
# Function to detect outliers using Z-score method
def detect_outliers_z_score(df, column):
    mean = df[column].mean()
    std = df[column].std()
    z_scores = (df[column] - mean) / std
    outliers = df[np.abs(z_scores) > 3]
    return outliers
```

```
# Detect outliers in BCI Values
bci_outliers = detect_outliers_z_score(bci_df_eda, 'BCI Values ')
print("Outliers in BCI Values:")
print(bci_outliers)
```

```
Outliers in BCI Values:
Empty DataFrame
Columns: [Quarter, BCI Values ]
Index: []
```

```
# Detect outliers in CSI Values
csi_outliers = detect_outliers_z_score(csi_df_eda, 'CSI Values ')
print("Outliers in CSI Values:")
print(csi_outliers)
```

```
Outliers in CSI Values:
Empty DataFrame
Columns: [Quarter, CSI Values ]
Index: []
```

```
PART C(2): Macroeconomics Dataset
```

```
# Load the macroeconomic data from each sheet
macro_data = pd.DataFrame()
sheets = ['Imports', 'Exports', 'GDP', 'Private Consumption', 'Private Investment']

for sheet in sheets:
    data = pd.read_excel('Macroeconomics Data.xlsx', sheet_name=sheet)
    data.rename(columns={data.columns[0]: 'Quarter', data.columns[1]: sheet}, inplace=True)
    if macro_data.empty:
        macro_data = data
    else:
        macro_data = pd.merge(macro_data, data, on='Quarter', how='outer')
```

```
# Perform EDA on the Macroeconomics data
print("\nFirst few rows of the Macroeconomics dataset:")
print(macro_data.head())
```

```python
# Perform EDA on the Macroeconomics data
print("\nFirst few rows of the Macroeconomics dataset:")
print(macro_data.head())
```

```
First few rows of the Macroeconomics dataset:
  Quarter  Imports  Exports   GDP  Private Consumption  Private Investment
0  2022Q1     16.1     12.3   4.8                  5.3                 0.4
1  2022Q2     20.1     15.9   8.8                 18.3                 6.3
2  2022Q3     21.1     21.5  14.1                 14.8                13.2
3  2022Q4      7.2      8.6   7.1                  7.3                10.3
4  2023Q1     -6.5     -3.3   5.6                  5.9                 4.7
```

```python
# Display the summary statistics of Macroeconomics DataFrame
print("\nSummary of the Macroeconomics DataFrame:")
print(macro_data.info())
```

```
Summary of the Macroeconomics DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Quarter              8 non-null      object 
 1   Imports              8 non-null      float64
 2   Exports              8 non-null      float64
 3   GDP                  8 non-null      float64
 4   Private Consumption  8 non-null      float64
 5   Private Investment   8 non-null      float64
dtypes: float64(5), object(1)
memory usage: 516.0+ bytes
None
```

```python
# Function to detect outliers using Z-score method
def detect_outliers_z_score(df, column):
    mean = df[column].mean()
    std = df[column].std()
    z_scores = (df[column] - mean) / std
    outliers = df[np.abs(z_scores) > 3]
    return outliers
```

```python
# Perform outlier detection for all variables
for sheet in sheets:
    print(f"\nOutliers in {sheet}:")
    outliers = detect_outliers_z_score(macro_data, sheet)
    if outliers.empty:
        print("No outliers found.")
    else:
        print(outliers)
```

```
Outliers in Imports:
No outliers found.

Outliers in Exports:
No outliers found.

Outliers in GDP:
No outliers found.

Outliers in Private Consumption:
No outliers found.

Outliers in Private Investment:
No outliers found.
```

Appendix G: Preprocessing Steps for the Textual Dataset

```
PART D: Preprocessing Steps for the Textual Dataset
```

```
# Strip leading and trailing whitespaces from all column names
df.columns = df.columns.str.strip()
print("Column names after stripping:", df.columns)
```

```
Column names after stripping: Index(['Date', 'Title', 'Content'], dtype='object')
```

```
# Convert Chinese dates to standard format
def convert_chinese_date(chinese_date):
    if not isinstance(chinese_date, str):
        return ""
    chinese_date = re.sub(r'年', '-', chinese_date)
    chinese_date = re.sub(r'月', '-', chinese_date)
    chinese_date = re.sub(r'日', '', chinese_date)
    return chinese_date
```

```
# Apply date conversion
if 'Date' in df.columns:
    df['Date'] = df['Date'].apply(convert_chinese_date)
    # Print to inspect after date conversion
    print("DataFrame after date conversion:")
    print(df[['Date']].head())

    # Convert the string dates to datetime objects
    df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d', errors='coerce')
    # Print to inspect after datetime conversion
    print("DataFrame after datetime conversion:")
    print(df[['Date']].head())

    # Ensure dates are set as index
    df.set_index('Date', inplace=True)
else:
    print("Column 'Date' not found in DataFrame.")

# Print the DataFrame to inspect after setting index
print("DataFrame after setting index:")
print(df.head())
```

```
DataFrame after date conversion:
      Date
0  2022-1-1
1  2022-1-3
2  2022-1-3
3  2022-1-3
4  2022-1-4
DataFrame after datetime conversion:
       Date
0 2022-01-01
1 2022-01-03
2 2022-01-03
3 2022-01-03
4 2022-01-04
DataFrame after setting index:
                                                Title  \
Date
2022-01-01                  全球富豪身价狂涨 中国富人财产反而缩水
2022-01-03                开市即迎来套利压力  马股早盘跌18.85点
2022-01-03           2022首个交易日出师不利  马股全天下滑18.48点
2022-01-03               亚航拟改名为CAPITAL A BERHAD
2022-01-04               与大市背道而驰  富时大马综合指数跌9.52点

                                                                                                Content
Date
2022-01-01  这2年在新冠病毒肆虐期间，全球富豪资产大幅增长，不过中国科技富豪资产大失血，根据彭博亿万富豪...
2022-01-03  （吉隆坡3日讯）2022年首个交易日，亚洲股市开盘表现普遍平平无奇，唯独马股出师不利，上周五...
2022-01-03  （吉隆坡3日讯）由于区域多个股市仍未开市，导致亚洲市场淡静。马股首个交易日出师不利，全天下滑...
2022-01-03  （吉隆坡3日讯）亚洲航空（AirAsia，5099，主板消费股）董事局建议将公司的名字改为C...
2022-01-04  （吉隆坡4日讯）美国隔夜股市上涨，带动亚洲股市周二升多跌少。不过，马股仍延续周一跌势，与大市...
```

```python
# Strip leading and trailing whitespaces from all string columns
df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

```python
# Handle potential Unicode issues by normalizing the text
df = df.apply(lambda x: x.apply(lambda y: unicodedata.normalize('NFKC', y) if isinstance(y, str) else y))

print("DataFrame after normalization:")
print(df.head())
```

```
DataFrame after normalization:
                                Title  \
Date
2022-01-01          全球富豪身价狂涨 中国富人财产反而缩水
2022-01-03        开市即迎来套利压力 马股早盘跌18.85点
2022-01-03  2022首个交易日出师不利 马股全天下滑18.48点
2022-01-03      亚航拟改名为CAPITAL A BERHAD
2022-01-04      与大市背道而驰 富时大马综合指数跌9.52点

                                                 Content
Date
2022-01-01  这2年在新冠病毒肆虐期间,全球富豪资产大幅增长,不过中国科技富豪资产大失血,根据彭博亿万富豪...
2022-01-03  (吉隆坡3日讯)2022年首个交易日,亚洲股市开盘表现普遍平平无奇,唯独马股出师不利,上周五...
2022-01-03  (吉隆坡3日讯)由于区域多个股市仍未开市,导致亚洲市场淡静。马股首个交易日出师不利,全天下滑...
2022-01-03  (吉隆坡3日讯)亚洲航空(AirAsia,5099,主板消费股)董事局建议将公司的名字改为C...
2022-01-04  (吉隆坡4日讯)美国隔夜股市上涨,带动亚洲股市周二升多跌少。不过,马股仍延续周一跌势,与大市...
```

```
# Load dictionary from Excel
dictionary_df = pd.read_excel('中文金融情感词典_姜富伟等(2021).xlsx', sheet_name=None)
positive_words = set(dictionary_df['positive']['Positive Word'].dropna().apply(lambda x: unicodedata.normalize('NFKC', str(x)).strip()))
negative_words = set(dictionary_df['negative']['Negative Word'].dropna().apply(lambda x: unicodedata.normalize('NFKC', str(x)).strip()))
```

```
# Print the first few positive and negative words to check
print("Sample positive words:", list(positive_words)[:10])
print("Sample negative words:", list(negative_words)[:10])
```

```
Sample positive words: ['天籁', '可爱的', '可靠性', '划一', '豪华', '变好', '进步', '精粹', '范文', '超强']
Sample negative words: ['声讨', '损毁', '起诉书', '犯罪者', '埋葬', '缺席', '等闲视之', '蹂躏', '昏迷', '庸医']
```

```
# To show that the dictionary does contain negators and has been classified into their respective sentiment categories
# Display words starting with "不" (bu) and "没" (mei) in the dictionary
# Function to filter words starting with specific characters
def filter_words(words_set, start_chars):
    return [word for word in words_set if word.startswith(start_chars)]
```

```
# Filter words starting with "不" and "没" in the positive words
positive_bu_words = filter_words(positive_words, "不")
positive_mei_words = filter_words(positive_words, "没")
```

```
# Filter words starting with "不" and "没" in the negative words
negative_bu_words = filter_words(negative_words, "不")
negative_mei_words = filter_words(negative_words, "没")
```

```
# Display the filtered words
print("Positive words starting with '不':", positive_bu_words)
print("Positive words starting with '没':", positive_mei_words)
print("Negative words starting with '不':", negative_bu_words)
print("Negative words starting with '没':", negative_mei_words)
```

```
Positive words starting with '不': ['不懈', '不同凡响', '不含糊', '不愧', '不虚此行', '不乏', '不屈不挠', '不可思议的', '不拘泥', '不凡', '不吝', '不错', '不俗', '不亦乐乎']
Positive words starting with '没': []
Negative words starting with '不': ['不满', '不便的', '不充足的', '不怀', '不经', '不完备的', '不过关', '不畏', '不堪重负', '不切实际', '不能医治', '不齿', '不注意', '不完美的', '不可理喻', '不
Negative words starting with '没': ['没完没了', '没把握', '没用', '没事', '没得说', '没劲', '没有理由的', '没关系', '没法子', '没落', '没门', '没法儿', '没有利润的', '没脚', '没想到', '没有防备
```

```
# Load stop words dictionary from JSON
with open('stopwords-zh.json', 'r', encoding='utf-8') as f:
    stop_words = set(json.load(f))
```

```
# Define function to preprocess the textual data
def preprocess_text(text):
    if not isinstance(text, str):
        text = ""
    text = unicodedata.normalize('NFKC', text)
    text = re.sub(r'\s+', ' ', text)
    text = re.sub(r'[^\w\s\u4e00-\u9fff]', '', text)
    words = jieba.lcut(text)
    words = [word for word in words if word not in stop_words]
    return words
```

```python
# Concatenate title and content for tokenization and sentiment analysis
if 'Title' in df.columns and 'Content' in df.columns:
    df['combined_text'] = df['Title'] + " " + df['Content']
    df['tokens'] = df['combined_text'].apply(preprocess_text)
else:
    print("Columns 'Title' or 'Content' not found in DataFrame.")
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\User\AppData\Local\Temp\jieba.cache
Loading model cost 0.970 seconds.
Prefix dict has been built successfully.
```

Appendix H: Exploratory Data Analysis of the Preprocessed Textual Dataset

```
PART E: Exploratory Data Analysis of the Preprocessed Textual Dataset
```

```
# Perform EDA on the preprocessed data
print("\nFirst few rows of the preprocessed dataset:")
print(df.head())
```

```

First few rows of the preprocessed dataset:
                                                Title  \
Date
2022-01-01          全球富豪身价狂涨 中国富人财产反而缩水
2022-01-03       开市即迎来套利压力 马股早盘跌18.85点
2022-01-03  2022首个交易日出师不利 马股全天下滑18.48点
2022-01-03      亚航拟改名为CAPITAL A BERHAD
2022-01-04      与大市背道而驰 富时大马综合指数跌9.52点

                                                Content  \
Date
2022-01-01  这2年在新冠病毒肆虐期间,全球富豪资产大幅增长,不过中国科技富豪资产大失血,根据彭博亿万富豪...
2022-01-03  (吉隆坡3日讯)2022年首个交易日,亚洲股市开盘表现普遍平平无奇,唯独马股出师不利,上周五...
2022-01-03  (吉隆坡3日讯)由于区域多个股市仍未开市,导致亚洲市场淡静。马股首个交易日出师不利,全天下滑...
2022-01-03  (吉隆坡3日讯)亚洲航空(AirAsia,5099,主板消费股)董事局建议将公司的名字改为C...
2022-01-04  (吉隆坡4日讯)美国隔夜股市上涨,带动亚洲股市周二升多跌少。不过,马股仍延续周一跌势,与大市...

                                                combined_text  \
Date
2022-01-01  全球富豪身价狂涨 中国富人财产反而缩水 这2年在新冠病毒肆虐期间,全球富豪资产大幅增长,不过...
2022-01-03  开市即迎来套利压力 马股早盘跌18.85点 (吉隆坡3日讯)2022年首个交易日,亚洲股市开...
2022-01-03  2022首个交易日出师不利 马股全天下滑18.48点 (吉隆坡3日讯)由于区域多个股市仍未开...
2022-01-03  亚航拟改名为CAPITAL A BERHAD (吉隆坡3日讯)亚洲航空(AirAsia,50...
2022-01-04  与大市背道而驰 富时大马综合指数跌9.52点 (吉隆坡4日讯)美国隔夜股市上涨,带动亚洲股市...

                                                tokens
Date
2022-01-01  [全球, 富豪, 身价, 狂涨,  , 中国, 富人, 财产, 缩水,  , 2, 新冠, ...
2022-01-03  [开市, 迎来, 套利, 压力,  , 马股, 早盘, 跌, 1885,  , 吉隆坡, 3...
2022-01-03  [2022, 首个, 交易日, 出师不利,  , 马股, 全天, 下滑, 1848,  , ...
2022-01-03  [亚航, 拟, 改名, CAPITAL,  , A,  , BERHAD,  , 吉隆坡, ...
2022-01-04  [大市, 背道而驰,  , 富时, 大马, 综合, 指数, 跌, 952,  , 吉隆坡, ...

```

```python
# Display the summary statistics of the preprocessed DataFrame
print("\nSummary of the preprocessed DataFrame:")
print(df.info())
```

```
Summary of the preprocessed DataFrame:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3361 entries, 2022-01-01 to 2023-12-30
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Title          3361 non-null   object
 1   Content        3361 non-null   object
 2   combined_text  3361 non-null   object
 3   tokens         3361 non-null   object
dtypes: object(4)
memory usage: 131.3+ KB
None
```

```python
# Save the preprocessed data to a CSV file with UTF-8 encoding
df.to_csv('preprocessed_data.csv', index=True, encoding='utf-8')
```

Appendix I: Computation of the News Sentiment Index

```
PART F: Computation of the News Sentiment Index
```

```python
# Sentiment analysis
def compute_sentiment(words):
    positive = sum(1 for word in words if word in positive_words)
    negative = sum(1 for word in words if word in negative_words)
    total_words = len(words)
    score = (positive - negative) / total_words * 1000 if total_words > 0 else 0
    return 100 + score, positive, negative, total_words

if 'tokens' in df.columns:
    sentiment_data = df['tokens'].apply(compute_sentiment)
    df[['sentiment_score', 'positive_count', 'negative_count', 'total_words']] = pd.DataFrame(sentiment_data.tolist(), index=df.index)
```

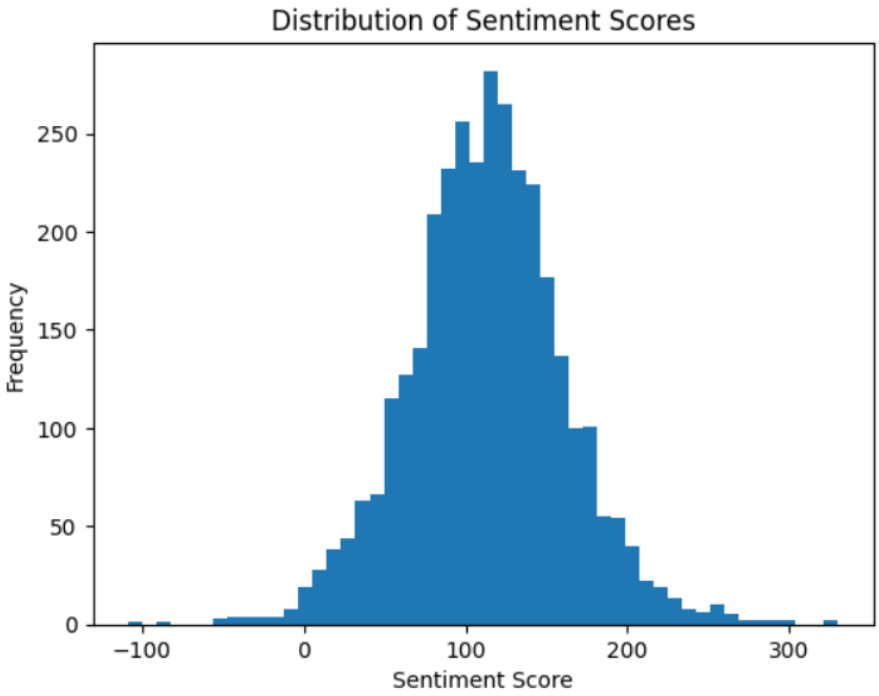
```python
# Inspect sentiment calculation for a few articles
for index in range(min(5, len(df))):
    row = df.iloc[index]
    print(f"\nArticle {index + 1}:")
    print(f"Tokens: {row['tokens']}")
    pos_words_in_tokens = [word for word in row['tokens'] if word in positive_words]
    neg_words_in_tokens = [word for word in row['tokens'] if word in negative_words]
    print(f"Positive words in tokens: {pos_words_in_tokens}")
    print(f"Negative words in tokens: {neg_words_in_tokens}")
    print(f"Positive words count: {row['positive_count']}")
    print(f"Negative words count: {row['negative_count']}")
    print(f"Sentiment Score: {row['sentiment_score']}")
```

```
Article 1:
Tokens: ['全球', '富豪', '身价', '狂涨', ' ', '中国', '富人', '财产', '缩水', ' ', '2', '新冠', '病毒', '肆虐', '期间', '全球', '富豪', '资产', '大幅', '增长', '中国', '科技', '富豪', '资产',
Positive words in tokens: ['增长', '最大', '有钱', '暴涨', '最大', '最多', '公开', '欢迎']
Negative words in tokens: ['病毒', '肆虐', '失血', '损失', '损失', '失血', '损失', '暴跌', '损失', '审查', '指控', '扰乱', '罚款', '大起大落', '打击', '暴跌']
Positive words count: 8
Negative words count: 16
Sentiment Score: 62.44131455399061

Article 2:
Tokens: ['开市', '迎来', '套利', '压力', ' ', '马股', '早盘', '跌', '1885', ' ', '吉隆坡', '3', '日讯', '2022', '首个', '交易日', '亚洲', '股市', '开盘', '表现', '普遍', '平平', '奇', '唯独',
Positive words in tokens: ['生效', '顶级', '热门', '显著']
Negative words in tokens: ['压力', '跌', '压力', '拖累', '最差', '下滑', '下滑', '下滑', '下跌', '下跌']
Positive words count: 4
Negative words count: 10
Sentiment Score: 68.42105263157895

Article 3:
Tokens: ['2022', '首个', '交易日', '出师不利', ' ', '马股', '全天', '下滑', '1848', ' ', '吉隆坡', '3', '日讯', '区域', '多个', '股市', '未', '开市', '导致', '亚洲', '市场', '淡静', '马股', '
Positive words in tokens: ['生效', '落实', '成功', '扶持', '上涨', '显著', '顶级', '显著', '联合', '上涨', '上升', '榜首', '相信', '上升']
Negative words in tokens: ['下滑', '下滑', '最差', '遭遇', '下滑', '拖累', '徘徊', '下跌', '下滑', '下跌', '下滑', '刺激']
Positive words count: 14
Negative words count: 12
Sentiment Score: 105.97014925373135

Article 4:
Tokens: ['亚航', '拟', '改名', 'CAPITAL', ' ', 'A', ' ', 'BERHAD', ' ', '吉隆坡', '3', '日讯', '亚洲', '航空', 'AirAsia5099', '主板', '消费', '股', '董事局', '建议', '公司', '名字', '改为',
Positive words in tokens: ['建议', '批准', '批准']
Negative words in tokens: []
Positive words count: 3
Negative words count: 0
Sentiment Score: 171.42857142857144

Article 5:
Tokens: ['大市', '背道而驰', ' ', '富时', '大马', '综合', '指数', '跌', '952', ' ', '吉隆坡', '4', '日讯', '美国', '隔夜', '股市', '上涨', '带动', '亚洲', '股市', '周二', '升多', '跌少', '马股
Positive words in tokens: ['上涨', '带动', '顶级', '最大', '热门', '最大']
Negative words in tokens: ['背道而驰', '跌', '背道而驰', '跌', '下滑', '跌']
Positive words count: 6
Negative words count: 6
Sentiment Score: 100.0
```

```python
# Print sample sentiment scores
print("Sample sentiment scores:")
print(df[['sentiment_score']].head(10))
```

```
Sample sentiment scores:
            sentiment_score
Date
2022-01-01        62.441315
2022-01-03        68.421053
2022-01-03       105.970149
2022-01-03       171.428571
2022-01-04       100.000000
2022-01-04       114.388489
2022-01-04        64.497041
2022-01-05       148.543689
2022-01-05       100.000000
2022-01-05        57.446809
```

```python
# Ensure 'sentiment_score' is correctly formatted as a tuple
df['sentiment_score_value'] = df['sentiment_score'].apply(lambda x: x[0] if isinstance(x, tuple) else x)

# Plot histogram of sentiment scores
plt.hist(df['sentiment_score_value'], bins=50)
plt.xlabel('Sentiment Score')
plt.ylabel('Frequency')
plt.title('Distribution of Sentiment Scores')
plt.show()
```



```python
# Resample sentiment scores to quarterly averages
quarterly_sentiment = df['sentiment_score'].resample('Q').mean()
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_33100\3153474309.py:2: FutureWarning: 'Q' is deprecated and will be removed in a future version, please use 'QE' instead.
  quarterly_sentiment = df['sentiment_score'].resample('Q').mean()
```

```python
# Create the final DataFrame for quarterly sentiment indices
quarterly_sentiment_df = quarterly_sentiment.reset_index().rename(columns={'sentiment_score': 'quarterly_sentiment_index'})
```

```python
# Convert 'Date' to show Quarterly format (e.g., 2022Q1)
def convert_date_to_quarter(date):
    return f"{date.year}Q{(date.month - 1) // 3 + 1}"

quarterly_sentiment_df['Quarter'] = quarterly_sentiment_df['Date'].apply(convert_date_to_quarter)
```

```python
# Reorder columns to have 'Date' as the first column
quarterly_sentiment_df = quarterly_sentiment_df[['Quarter', 'quarterly_sentiment_index']].rename(columns={'Quarter': 'Date'})
```

```python
# Display the first few rows of the resulting DataFrame
print("\nQuarterly Sentiment Index:")
print(quarterly_sentiment_df.head())
```

```python
# Display the first few rows of the resulting DataFrame
print("\nQuarterly Sentiment Index:")
print(quarterly_sentiment_df.head())
```

```
Quarterly Sentiment Index:
     Date  quarterly_sentiment_index
0  2022Q1                 104.752807
1  2022Q2                 107.689848
2  2022Q3                 108.340161
3  2022Q4                 114.046902
4  2023Q1                 114.676678
```

```python
# Save the results to CSV files with UTF-8 encoding
quarterly_sentiment_df.to_csv('quarterly_sentiment_index.csv', index=False, encoding='utf-8')
```

Appendix J: Nowcasting Activity of the BCI and CSI Figures Using the News Sentiment Index

```
PART G: Evaluation of the Performance of the Sentiment Index on its Nowcasting Ability of the BCI and CSI figures (Visual Plot and Regression Output)
```

```
PART G(1): Preparing the Data and Plotting the Time Series Plot
```

```
# Load MIER BCI and CSI data from Excel
bci_df = pd.read_excel('Data BCI_CSI_2022_2023.xlsx', sheet_name='BCI Formatted ')
csi_df = pd.read_excel('Data BCI_CSI_2022_2023.xlsx', sheet_name='CSI Formatted')
```

```
# Merge the sentiment index with BCI and CSI data
merged_df = quarterly_sentiment_df.merge(bci_df, left_on='Date', right_on='Quarter')
merged_df = merged_df.merge(csi_df, left_on='Date', right_on='Quarter', suffixes=('_BCI', '_CSI'))
```

```
# Verify the column names in the merged DataFrame
print("Column names in merged_df:", merged_df.columns)
```

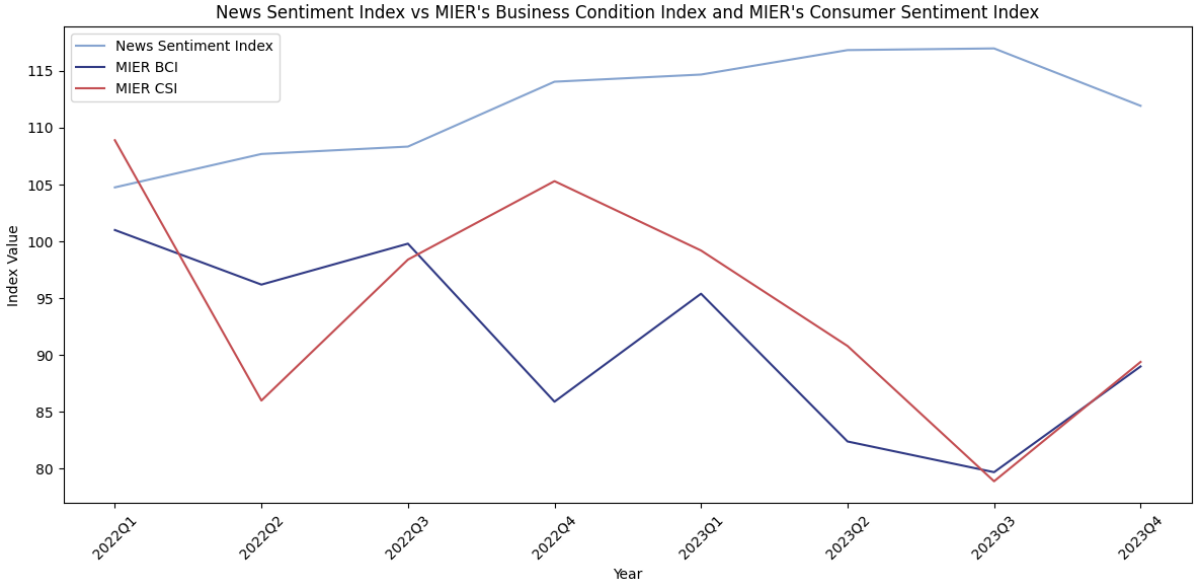
```
Column names in merged_df: Index(['Date', 'quarterly_sentiment_index', 'Quarter_BCI', 'BCI Values ',
       'Quarter_CSI', 'CSI Values '],
      dtype='object')
```

```
# Strip leading and trailing whitespaces from all column names in merged_df
merged_df.columns = merged_df.columns.str.strip()
```

```
# Verify the column names in the merged DataFrame after stripping
print("Column names in merged_df after stripping:", merged_df.columns)
```

```
Column names in merged_df after stripping: Index(['Date', 'quarterly_sentiment_index', 'Quarter_BCI', 'BCI Values',
       'Quarter_CSI', 'CSI Values'],
      dtype='object')
```

```
# Plot the time series data
plt.figure(figsize=(12, 6))

plt.plot(merged_df['Date'], merged_df['quarterly_sentiment_index'], label='News Sentiment Index', color='#86A4D0')
plt.plot(merged_df['Date'], merged_df['BCI Values'], label='MIER BCI', color='#313A85')
plt.plot(merged_df['Date'], merged_df['CSI Values'], label='MIER CSI', color='#C44E52')

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Index Value')
plt.title("News Sentiment Index vs MIER's Business Condition Index and MIER's Consumer Sentiment Index")
plt.legend()

# Customize x-axis to show the quarter format
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



PART G(2): Nowcast BCI and CSI values Using the News Sentiment Index

```
# Sort the merged DataFrame by date
merged_df.sort_values('Date', inplace=True)
```

```
# Create lagged variables
merged_df['BCI_Lag'] = merged_df['BCI Values'].shift(1)
merged_df['CSI_Lag'] = merged_df['CSI Values'].shift(1)
```

```
# Drop rows with missing values created by lagging
merged_df.dropna(inplace=True)
```

```python
# Regression analysis for BCI
X_bci = merged_df[['BCI_Lag', 'quarterly_sentiment_index']]
y_bci = merged_df['BCI Values']
X_bci = sm.add_constant(X_bci)  # adding a constant

model_bci = sm.OLS(y_bci, X_bci).fit()
predictions_bci = model_bci.predict(X_bci)
```

```python
# Print out the statistics
print(model_bci.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             BCI Values   R-squared:                       0.697
Model:                            OLS   Adj. R-squared:                  0.545
Method:                 Least Squares   F-statistic:                     4.592
Date:                Fri, 26 Jul 2024   Prob (F-statistic):             0.0921
Time:                        14:54:19   Log-Likelihood:                -19.399
No. Observations:                   7   AIC:                             44.80
Df Residuals:                       4   BIC:                             44.64
Df Model:                           2
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                          289.5625     81.926      3.534      0.024      62.099     517.026
BCI_Lag                         -0.0541      0.265     -0.204      0.848      -0.791       0.682
quarterly_sentiment_index       -1.7254      0.609     -2.834      0.047      -3.416      -0.035
==============================================================================
Omnibus:                        nan   Durbin-Watson:                   2.089
Prob(Omnibus):                  nan   Jarque-Bera (JB):                1.849
Skew:                         1.250   Prob(JB):                        0.397
Kurtosis:                     3.303   Cond. No.                     6.16e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.16e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
c:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\statsmodels\stats\stattools.py:74: ValueWarning: omni_normtest is not valid with less than 8 observations; 7 samples were given.
  warn("omni_normtest is not valid with less than 8 observations; %i "
```

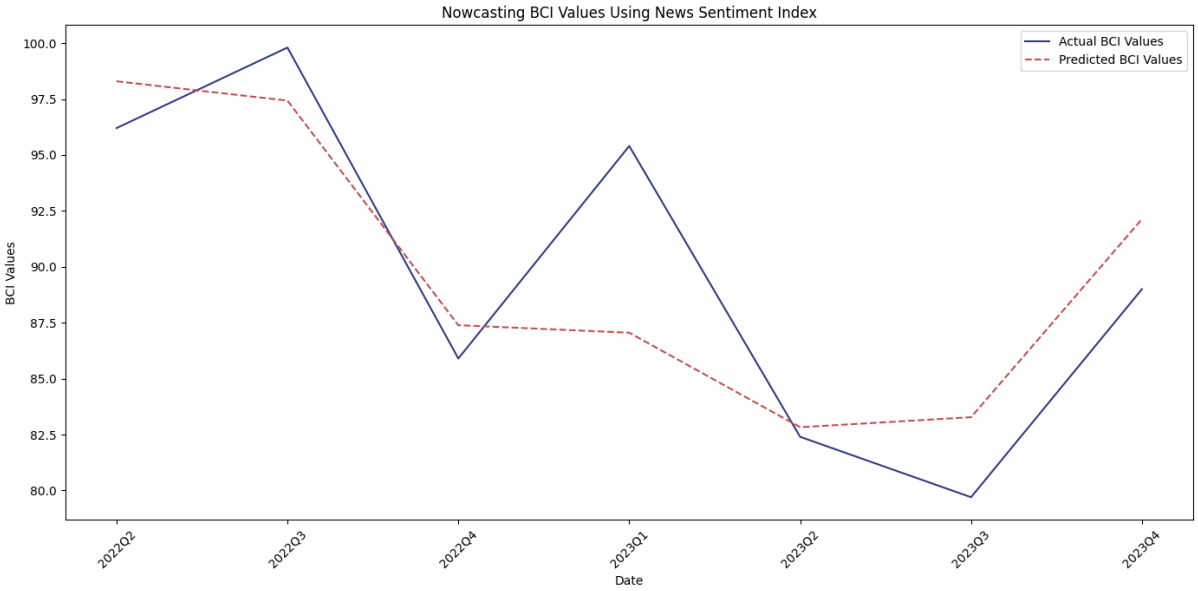
```python
# Regression analysis for CSI
X_csi = merged_df[['CSI_Lag', 'quarterly_sentiment_index']]
y_csi = merged_df['CSI Values']
X_csi = sm.add_constant(X_csi)  # adding a constant

model_csi = sm.OLS(y_csi, X_csi).fit()
predictions_csi = model_csi.predict(X_csi)
```

```python
# Print out the statistics
print(model_csi.summary())
```

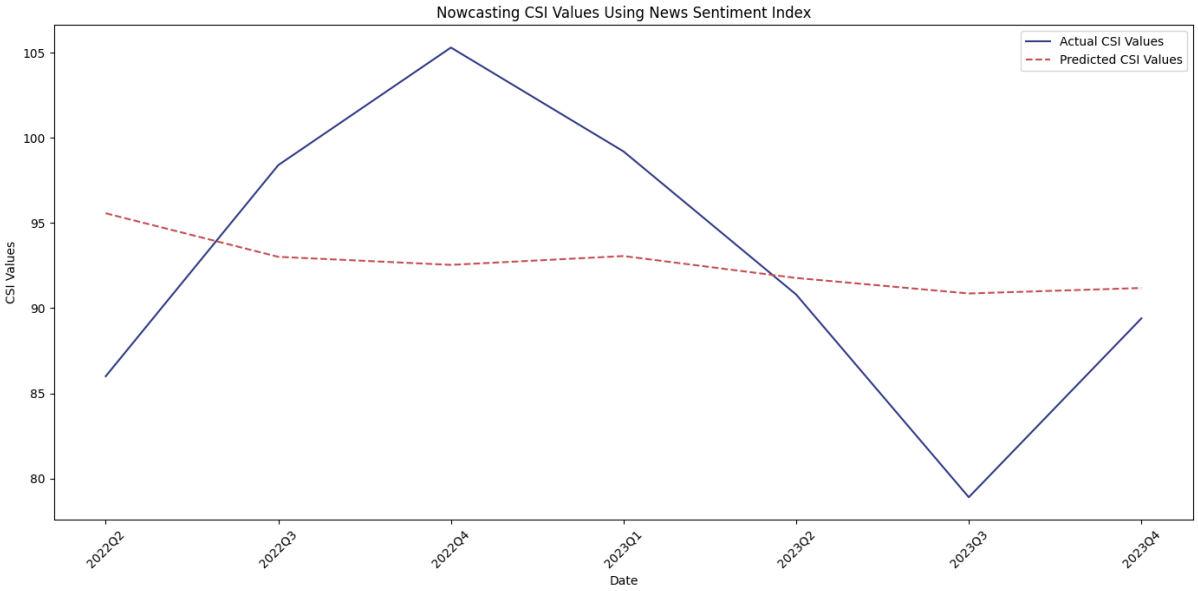
```
                            OLS Regression Results
==============================================================================
Dep. Variable:             CSI Values   R-squared:                       0.031
Model:                            OLS   Adj. R-squared:                 -0.454
Method:                 Least Squares   F-statistic:                   0.06367
Date:                Fri, 26 Jul 2024   Prob (F-statistic):              0.939
Time:                        14:54:25   Log-Likelihood:                -24.644
No. Observations:                   7   AIC:                             55.29
Df Residuals:                       4   BIC:                             55.13
Df Model:                           2
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                          117.3405    137.175      0.855      0.441    -263.517     498.198
CSI_Lag                          0.1029      0.413      0.249      0.816      -1.045       1.251
quarterly_sentiment_index       -0.3063      1.171     -0.261      0.807      -3.558       2.946
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   1.274
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.394
Skew:                          -0.034   Prob(JB):                        0.821
Kurtosis:                       1.840   Cond. No.                     4.96e+03
==============================================================================
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.96e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
c:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\statsmodels\stats\stattools.py:74: ValueWarning: omni_normtest is not valid with less than 8 observations; 7 samples were given.
  warn("omni_normtest is not valid with less than 8 observations; %i "

```python
# Plotting actual vs predicted for BCI
plt.figure(figsize=(14, 7))
plt.plot(merged_df['Date'], y_bci, label='Actual BCI Values', color='#313A85')
plt.plot(merged_df['Date'], predictions_bci, label='Predicted BCI Values', color='#C44E52', linestyle='--')
plt.xlabel('Date')
plt.ylabel('BCI Values')
plt.title('Nowcasting BCI Values Using News Sentiment Index')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Plotting actual vs predicted for CSI
plt.figure(figsize=(14, 7))
plt.plot(merged_df['Date'], y_csi, label='Actual CSI Values', color='#313A85')
plt.plot(merged_df['Date'], predictions_csi, label='Predicted CSI Values', color='#C44E52', linestyle='--')
plt.xlabel('Date')
plt.ylabel('CSI Values')
plt.title('Nowcasting CSI Values Using News Sentiment Index')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



For the BCI nowcasting activity, the condition number suggests potential multicollinearity issues, which may need further investigation.

Check Correlation Matrix (BCI)

```
# Given that 'merged_df' is the DataFrame containing the predictors and the response variable
correlation_matrix = merged_df[['BCI_Lag', 'quarterly_sentiment_index']].corr()

print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
                           BCI_Lag  quarterly_sentiment_index
BCI_Lag                   1.000000                  -0.416616
quarterly_sentiment_index -0.416616                  1.000000
```

Variance Inflation Factor (VIF) for BCI Data

```
# Given that 'merged_df' is the DataFrame containing the predictors
X = merged_df[['BCI_Lag', 'quarterly_sentiment_index']]

# Adding a constant term for the intercept
X = sm.add_constant(X)

# Calculate VIF for each predictor
vif = pd.DataFrame()
vif["Variable"] = X.columns
vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(vif)
```

```
                   Variable          VIF
0                     const  1795.654814
1                   BCI_Lag     1.210022
2  quarterly_sentiment_index     1.210022
```

There is no multicollinearity concern. Since the VIF values for the predictors are low, multicollinearity is not a significant issue in the regression model.

Appendix K: Evaluating the Pearson Correlation Between the Macroeconomics Variables and the News Sentiment Index

```
PART H: Evaluation of the Pearson Correlation Between the Macroeconomics Variables and the News Sentiment Index
```

```
# Load the quarterly sentiment index data
quarterly_sentiment_df = pd.read_csv('quarterly_sentiment_index.csv')
```

```
# Merge the macroeconomic data with the quarterly sentiment index data
merged_data = pd.merge(quarterly_sentiment_df, macro_data, left_on='Date', right_on='Quarter')
```

```
# Drop the 'Quarter' column
merged_data.drop('Quarter', axis=1, inplace=True)
```

```
# Drop the 'Date' column
numeric_data = merged_data.drop(columns=['Date'])
```

```
# Compute Pearson correlation coefficients
correlation_matrix = numeric_data.corr()
```

```
# Display the correlation matrix
print("Pearson Correlation Matrix:")
print(correlation_matrix)
```

```
Pearson Correlation Matrix:
                         quarterly_sentiment_index   Imports   Exports  \
quarterly_sentiment_index                   1.000000 -0.889319 -0.817534
Imports                                    -0.889319  1.000000  0.980870
Exports                                    -0.817534  0.980870  1.000000
GDP                                        -0.494618  0.775915  0.853863
Private Consumption                        -0.532231  0.776856  0.780318
Private Investment                          0.052724  0.362085  0.479134

                                GDP  Private Consumption  Private Investment
quarterly_sentiment_index -0.494618            -0.532231            0.052724
Imports                    0.775915             0.776856            0.362085
Exports                    0.853863             0.780318            0.479134
GDP                        1.000000             0.826276            0.790622
Private Consumption        0.826276             1.000000            0.551795
Private Investment         0.790622             0.551795            1.000000
```

```python
# Extract the correlations with the quarterly sentiment index
correlation_with_sentiment = correlation_matrix['quarterly_sentiment_index']
print("\nPearson Correlation Coefficients with Quarterly Sentiment Index:")
print(correlation_with_sentiment)
```

```
Pearson Correlation Coefficients with Quarterly Sentiment Index:
quarterly_sentiment_index    1.000000
Imports                     -0.889319
Exports                     -0.817534
GDP                         -0.494618
Private Consumption         -0.532231
Private Investment           0.052724
Name: quarterly_sentiment_index, dtype: float64
```

Appendix L: Modelling Process for the Forecasting Activity of the 5 Target Variables Using Machine Learning Models

```
PART I: Modelling Process for the Forecasting Activity of the 5 Target Variables Using Machine Learning Models
```

```
PART I(1): Modelling Process Without Hyperparameter Tuning and Performance Evaluation
```

```
# Define functions to compute RMSE and MAE
def compute_rmse(y_true, y_pred):
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    return rmse

def compute_mae(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    return mae
```

```
# Initialize variables
window_size = 4  # 4 quarters as training period
horizons = [1, 2, 3]  # Forecast horizons
macro_vars = ['Imports', 'Exports', 'GDP', 'Private Consumption', 'Private Investment']
```

```
# Prepare the data
data = merged_data.copy()
```

```
# Ensure 'Date' column is excluded from features
features = data.columns.difference(['Date'])
```

```
# Initialize dictionaries to store results
models = {
    'OLS-AR(1)': LinearRegression(),
    'LASSO': Lasso(),
    'Ridge': Ridge(),
    'SVR': SVR(),
    'RandomForest': RandomForestRegressor(),
    'XGBoost': XGBRegressor()
}

results = {model_name: {var: {horizon: [] for horizon in horizons} for var in macro_vars} for model_name in models}
rmse_ratios = {model_name: {var: {horizon: [] for horizon in horizons} for var in macro_vars} for model_name in models if model_name != 'OLS-AR(1)'}
mae_ratios = {model_name: {var: {horizon: [] for horizon in horizons} for var in macro_vars} for model_name in models if model_name != 'OLS-AR(1)'}
```

```python
# Rolling window approach
for var in macro_vars:
    for horizon in horizons:
        for start in range(0, len(data) - window_size - horizon + 1):
            end = start + window_size
            train_window = data.iloc[start:end]
            test_window = data.iloc[end:end+horizon]

            X_train = train_window[features].drop(columns=[var])
            y_train = train_window[var]
            X_test = test_window[features].drop(columns=[var])
            y_test = test_window[var]

            # Train OLS-AR(1) as benchmark
            ols_ar1 = LinearRegression()
            ols_ar1.fit(X_train, y_train)
            y_pred_ols = ols_ar1.predict(X_test)
            rmse_ols = compute_rmse(y_test, y_pred_ols)
            mae_ols = compute_mae(y_test, y_pred_ols)

            for model_name, model in models.items():
                model.fit(X_train, y_train)
                y_pred = model.predict(X_test)
                rmse = compute_rmse(y_test, y_pred)
                mae = compute_mae(y_test, y_pred)

                # Store results
                results[model_name][var][horizon].append((rmse, mae))
                if model_name != 'OLS-AR(1)':
                    rmse_ratios[model_name][var][horizon].append(rmse / rmse_ols)
                    mae_ratios[model_name][var][horizon].append(mae / mae_ols)
```

```python
# Function to safely calculate the mean of non-empty lists
def safe_mean(lst):
    return np.mean(lst) if lst else float('nan')
```

```
# Plotting RMSE ratios for each macroeconomic variable
for var in macro_vars:
    plt.figure(figsize=(10, 6))
    for model_name, horizon_dict in rmse_ratios.items():
        if model_name != 'OLS-AR(1)':
            horizons = list(horizon_dict[var].keys())
            ratios = [safe_mean(horizon_dict[var][h]) for h in horizons]
            plt.bar([f'{model_name} (h={h})' for h in horizons], ratios, label=model_name)
    plt.axhline(y=1, color='r', linestyle='--')
    plt.title(f'RMSE Ratios for {var}')
    plt.xlabel('Model (Forecast Horizon)')
    plt.ylabel('RMSE Ratio')
    plt.xticks(rotation=45)
    plt.legend()
    plt.show()
```

** Note that the outputs for the above code could be found in *Appendix M*.

```
# Plotting MAE ratios for each macroeconomic variable
for var in macro_vars:
    plt.figure(figsize=(10, 6))
    for model_name, horizon_dict in mae_ratios.items():
        if model_name != 'OLS-AR(1)':
            horizons = list(horizon_dict[var].keys())
            ratios = [safe_mean(horizon_dict[var][h]) for h in horizons]
            plt.bar([f'{model_name} (h={h})' for h in horizons], ratios, label=model_name)
    plt.axhline(y=1, color='r', linestyle='--')
    plt.title(f'MAE Ratios for {var}')
    plt.xlabel('Model (Forecast Horizon)')
    plt.ylabel('MAE Ratio')
    plt.xticks(rotation=45)
    plt.legend()
    plt.show()
```

** Note that the outputs for the above code could be found in *Appendix M*.

```
# Display results
for var in macro_vars:
    print(f'\nVariable: {var}')
    for model_name, horizon_dict in rmse_ratios.items():
        if model_name != 'OLS-AR(1)':
            print(f'\n{model_name}:')
            for horizon in horizon_dict[var]:
                rmse_ratio = safe_mean(rmse_ratios[model_name][var][horizon])
                mae_ratio = safe_mean(mae_ratios[model_name][var][horizon])
                print(f'  Horizon {horizon}: RMSE Ratio={rmse_ratio}, MAE Ratio={mae_ratio}')
```

** Note that the outputs for the above code could be found in *Appendix N*.

```
PART I(2): Modelling Process With Hyperparameter Tuning and Performance Evaluation
```

```
# Define a function to compute RMSE and MAE
def compute_metrics(y_true, y_pred):
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mae = mean_absolute_error(y_true, y_pred)
    return rmse, mae
```

```
# Initialize variables
window_size_tuned = 4  # 4 quarters as training period
horizons_tuned = [1, 2, 3]  # Forecast horizons
macro_vars_tuned = ['Imports', 'Exports', 'GDP', 'Private Consumption', 'Private Investment']
```

```
# Prepare the data
data_tuned = merged_data.copy()
```

```
# Ensure 'Date' column is excluded from features
features_tuned = data_tuned.columns.difference(['Date'])
```

```
# Define parameter grids for each model
param_grids = {
    'LASSO': {'lasso__alpha': [0.1, 0.5, 1.0, 5.0, 10.0]},
    'Ridge': {'ridge__alpha': [0.1, 0.5, 1.0, 5.0, 10.0]},
    'SVR': {'svr__C': [0.1, 1.0, 10.0], 'svr__epsilon': [0.1, 0.2, 0.5], 'svr__kernel': ['linear', 'rbf']},
    'RandomForest': {'randomforestregressor__n_estimators': [100, 200], 'randomforestregressor__max_depth': [None, 10, 20], 'randomforestregressor__min_samples_split': [2, 5]},
    'XGBoost': {'xgbregressor__n_estimators': [100, 200], 'xgbregressor__max_depth': [3, 6, 9], 'xgbregressor__learning_rate': [0.01, 0.1, 0.2]}
}
```

MagicPython

```
# Initialize models with pipelines
models = {
    'LASSO': make_pipeline(StandardScaler(), Lasso(max_iter=10000)),
    'Ridge': make_pipeline(StandardScaler(), Ridge(max_iter=10000)),
    'SVR': make_pipeline(StandardScaler(), SVR()),
    'RandomForest': make_pipeline(StandardScaler(), RandomForestRegressor()),
    'XGBoost': make_pipeline(StandardScaler(), XGBRegressor())
}
```

```python
# Initialize dictionaries to store results
best_models_tuned = {}
best_params_tuned = {}
for var in macro_vars_tuned:
    best_models_tuned[var] = {}
    best_params_tuned[var] = {}
    for model_name, model in models.items():
        param_grid = param_grids[model_name]
        grid_search = GridSearchCV(model, param_grid, cv=5, scoring='neg_mean_squared_error')
        grid_search.fit(data_tuned[features_tuned], data_tuned[var])
        best_models_tuned[var][model_name] = grid_search.best_estimator_
        best_params_tuned[var][model_name] = grid_search.best_params_
```

```python
# Now use best_models for the rolling window approach
results_tuned = {model_name: {var: {horizon: [] for horizon in horizons_tuned} for var in macro_vars_tuned} for model_name in models}
metrics_tuned = {model_name: {var: {horizon: {'RMSE': [], 'MAE': []} for horizon in horizons_tuned} for var in macro_vars_tuned} for model_name in models}
rmse_ratios_tuned = {model_name: {var: {horizon: [] for horizon in horizons_tuned} for var in macro_vars_tuned} for model_name in models if model_name != 'OLS-AR(1)'}
mae_ratios_tuned = {model_name: {var: {horizon: [] for horizon in horizons_tuned} for var in macro_vars_tuned} for model_name in models if model_name != 'OLS-AR(1)'}
```

```python
# Rolling window approach
for var in macro_vars_tuned:
    for horizon in horizons_tuned:
        for start in range(0, len(data_tuned) - window_size_tuned - horizon + 1):
            end = start + window_size_tuned
            train_window = data_tuned.iloc[start:end]
            test_window = data_tuned.iloc[end:end + horizon]

            X_train = train_window[features_tuned].drop(columns=[var])
            y_train = train_window[var]
            X_test = test_window[features_tuned].drop(columns=[var])
            y_test = test_window[var]

            # Train OLS-AR(1) as benchmark
            ols_ar1 = LinearRegression()
            ols_ar1.fit(X_train, y_train)
            y_pred_ols = ols_ar1.predict(X_test)
            rmse_ols, mae_ols = compute_metrics(y_test, y_pred_ols)

            for model_name, model in best_models_tuned[var].items():
                model.fit(X_train, y_train)
                y_pred = model.predict(X_test)
                rmse, mae = compute_metrics(y_test, y_pred)

                # Store results
                results_tuned[model_name][var][horizon].append(rmse)
                metrics_tuned[model_name][var][horizon]['RMSE'].append(rmse)
                metrics_tuned[model_name][var][horizon]['MAE'].append(mae)
                if model_name != 'OLS-AR(1)':
                    rmse_ratios_tuned[model_name][var][horizon].append(rmse / rmse_ols)
                    mae_ratios_tuned[model_name][var][horizon].append(mae / mae_ols)
```

```
# Check for empty lists
for model_name, var_dict in rmse_ratios_tuned.items():
    for var, horizon_dict in var_dict.items():
        for horizon, ratios in horizon_dict.items():
            if not ratios:
                print(f'Empty list found: Model={model_name}, Variable={var}, Horizon={horizon}')
```

```
# Function to safely calculate the mean of non-empty lists
def safe_mean(lst):
    return np.mean(lst) if lst else float('nan')
```

```
# Plotting RMSE ratios for each macroeconomic variable
for var in macro_vars_tuned:
    plt.figure(figsize=(10, 6))
    for model_name, horizon_dict in rmse_ratios_tuned.items():
        if model_name != 'OLS-AR(1)':
            horizons = list(horizon_dict[var].keys())
            ratios = [safe_mean(horizon_dict[var][h]) for h in horizons]
            plt.bar([f'{model_name} (h={h})' for h in horizons], ratios, label=model_name)
    plt.axhline(y=1, color='r', linestyle='--')
    plt.title(f'RMSE Ratios for {var}')
    plt.xlabel('Model (Forecast Horizon)')
    plt.ylabel('RMSE Ratio')
    plt.xticks(rotation=45)
    plt.legend()
    plt.show()
```

** Note that the outputs for the above code could be found in *Appendix O*.

```
# Plotting MAE ratios for each macroeconomic variable
for var in macro_vars_tuned:
    plt.figure(figsize=(10, 6))
    for model_name, horizon_dict in mae_ratios_tuned.items():
        if model_name != 'OLS-AR(1)':
            horizons = list(horizon_dict[var].keys())
            ratios = [safe_mean(horizon_dict[var][h]) for h in horizons]
            plt.bar([f'{model_name} (h={h})' for h in horizons], ratios, label=model_name)
    plt.axhline(y=1, color='r', linestyle='--')
    plt.title(f'MAE Ratios for {var}')
    plt.xlabel('Model (Forecast Horizon)')
    plt.ylabel('MAE Ratio')
    plt.xticks(rotation=45)
    plt.legend()
    plt.show()
```

** Note that the outputs for the above code could be found in *Appendix O*.

```python
# Display results for RMSE and MAE ratios
for var in macro_vars_tuned:
    print(f'\nVariable: {var}')
    for model_name in rmse_ratios_tuned:
        print(f'\n{model_name}:')
        for horizon in horizons_tuned:
            avg_rmse_ratio = safe_mean(rmse_ratios_tuned[model_name][var][horizon])
            avg_mae_ratio = safe_mean(mae_ratios_tuned[model_name][var][horizon])
            print(f'  Horizon {horizon}: RMSE Ratio={avg_rmse_ratio}, MAE Ratio={avg_mae_ratio}')
```

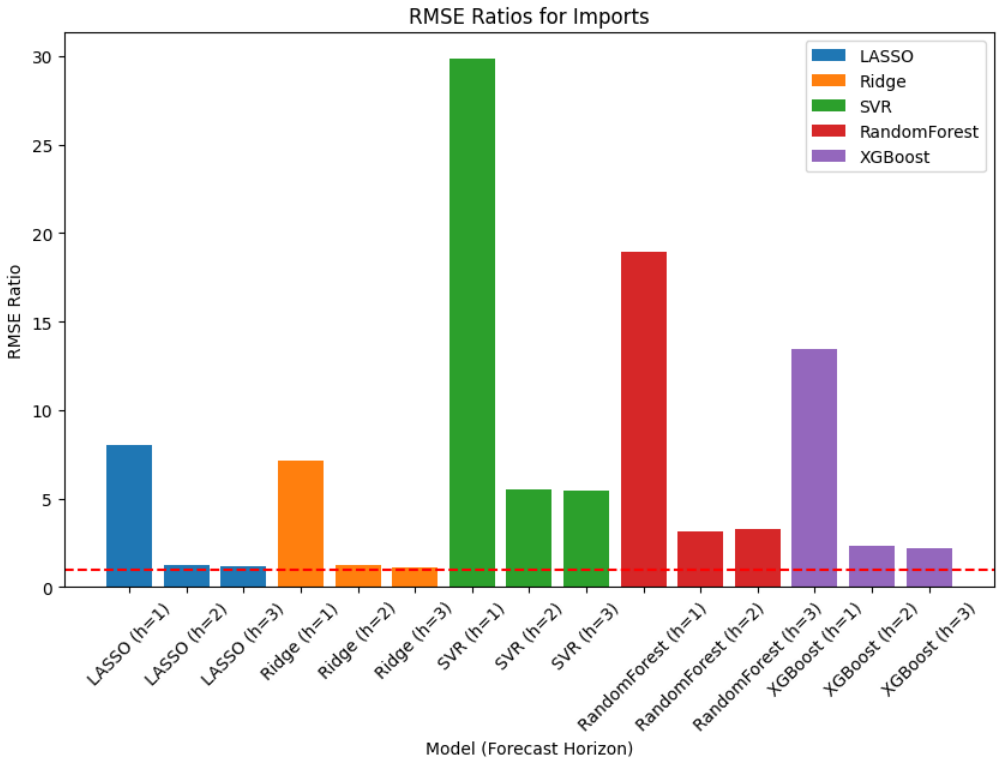
** Note that the outputs for the above code could be found in *Appendix P*.

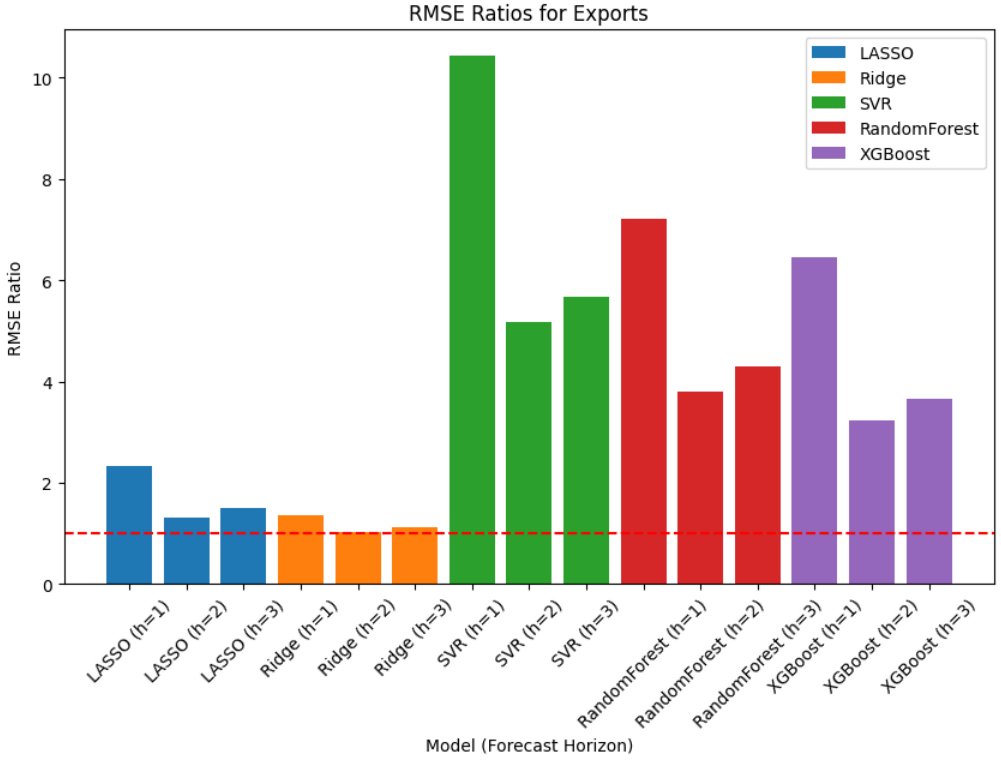
```python
# Display the best parameters identified for each model
print("\nBest parameters for each model and variable:")
for var, model_params in best_params_tuned.items():
    print(f'\nVariable: {var}')
    for model_name, params in model_params.items():
        print(f'{model_name}: {params}')
```

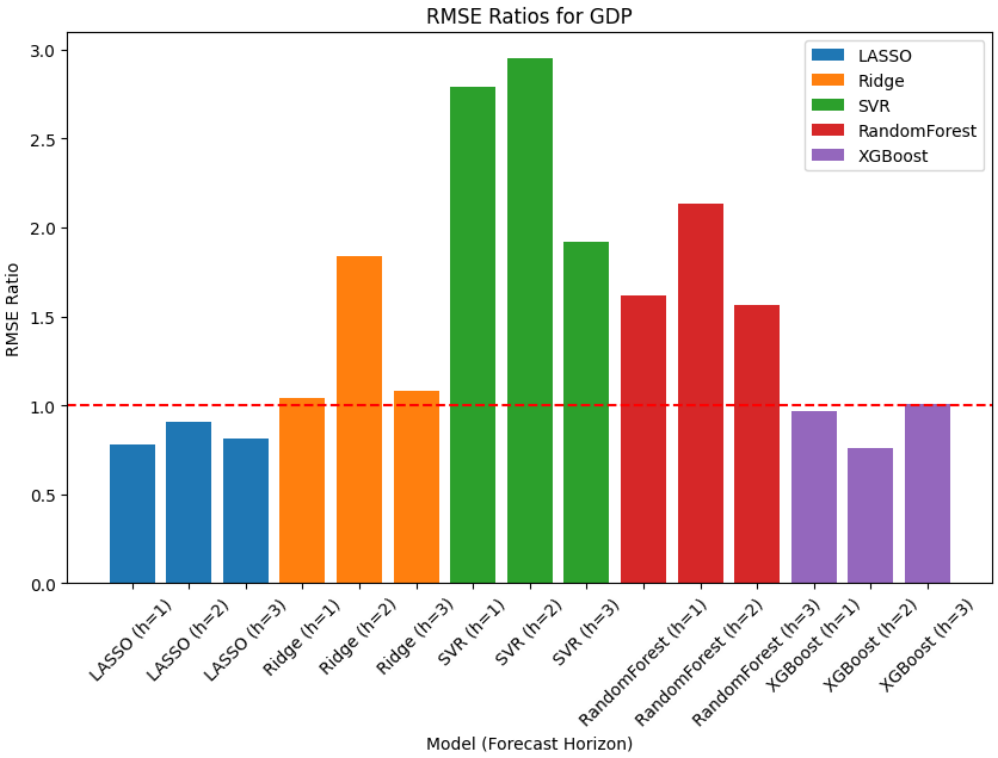
```
Best parameters for each model and variable:

Variable: Imports
LASSO: {'lasso__alpha': 0.1}
Ridge: {'ridge__alpha': 0.5}
SVR: {'svr__C': 1.0, 'svr__epsilon': 0.5, 'svr__kernel': 'linear'}
RandomForest: {'randomforestregressor__max_depth': 10, 'randomforestregressor__min_samples_split': 2, 'randomforestregressor__n_estimators': 200}
XGBoost: {'xgbregressor__learning_rate': 0.2, 'xgbregressor__max_depth': 6, 'xgbregressor__n_estimators': 200}

Variable: Exports
LASSO: {'lasso__alpha': 0.1}
Ridge: {'ridge__alpha': 0.1}
SVR: {'svr__C': 10.0, 'svr__epsilon': 0.5, 'svr__kernel': 'linear'}
RandomForest: {'randomforestregressor__max_depth': 10, 'randomforestregressor__min_samples_split': 2, 'randomforestregressor__n_estimators': 200}
XGBoost: {'xgbregressor__learning_rate': 0.2, 'xgbregressor__max_depth': 6, 'xgbregressor__n_estimators': 200}

Variable: GDP
LASSO: {'lasso__alpha': 0.1}
Ridge: {'ridge__alpha': 0.1}
SVR: {'svr__C': 10.0, 'svr__epsilon': 0.2, 'svr__kernel': 'linear'}
RandomForest: {'randomforestregressor__max_depth': 10, 'randomforestregressor__min_samples_split': 2, 'randomforestregressor__n_estimators': 200}
XGBoost: {'xgbregressor__learning_rate': 0.2, 'xgbregressor__max_depth': 6, 'xgbregressor__n_estimators': 200}

Variable: Private Consumption
LASSO: {'lasso__alpha': 0.1}
Ridge: {'ridge__alpha': 0.1}
SVR: {'svr__C': 10.0, 'svr__epsilon': 0.1, 'svr__kernel': 'linear'}
RandomForest: {'randomforestregressor__max_depth': 20, 'randomforestregressor__min_samples_split': 2, 'randomforestregressor__n_estimators': 100}
XGBoost: {'xgbregressor__learning_rate': 0.01, 'xgbregressor__max_depth': 3, 'xgbregressor__n_estimators': 200}


Variable: Private Investment
LASSO: {'lasso__alpha': 0.1}
Ridge: {'ridge__alpha': 0.1}
SVR: {'svr__C': 10.0, 'svr__epsilon': 0.1, 'svr__kernel': 'linear'}
RandomForest: {'randomforestregressor__max_depth': 20, 'randomforestregressor__min_samples_split': 2, 'randomforestregressor__n_estimators': 100}
XGBoost: {'xgbregressor__learning_rate': 0.01, 'xgbregressor__max_depth': 3, 'xgbregressor__n_estimators': 100}

```

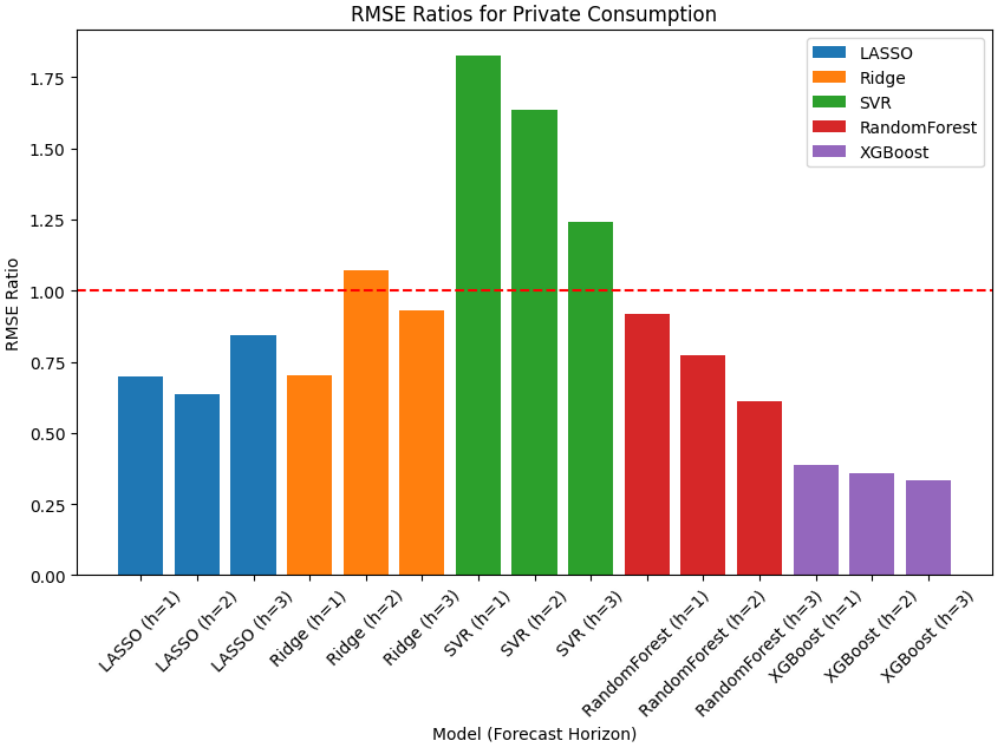
Appendix M: Performance Evaluation for Machine Learning Models Without Hyperparameter Tuning (Graphical Representation)

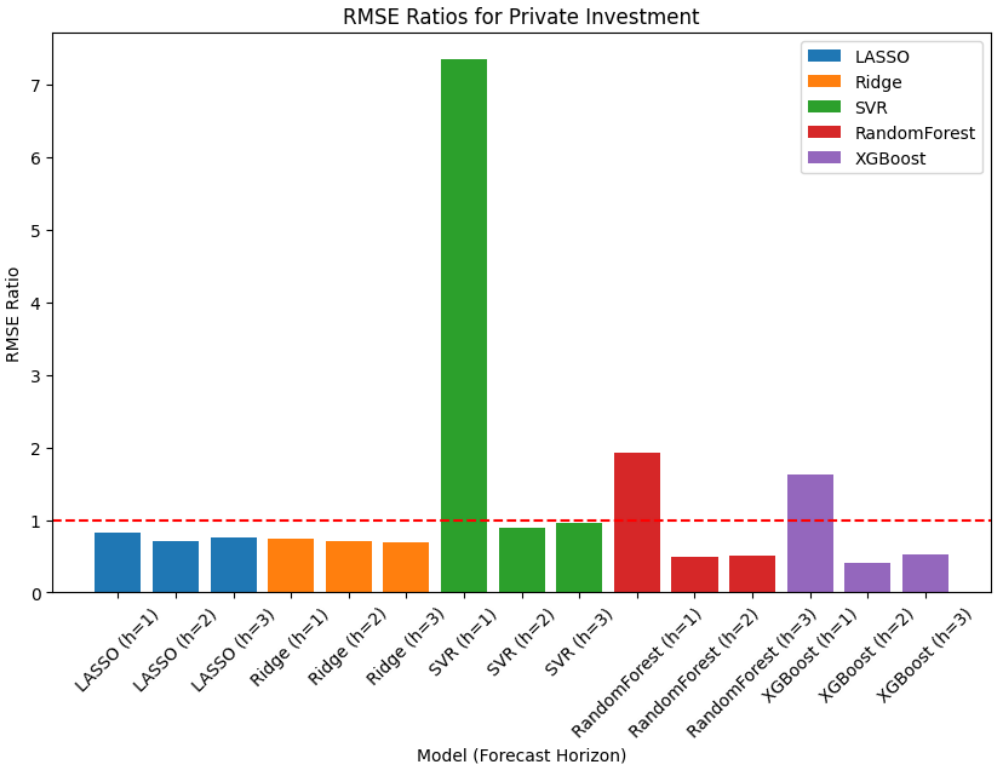
RMSE Ratio



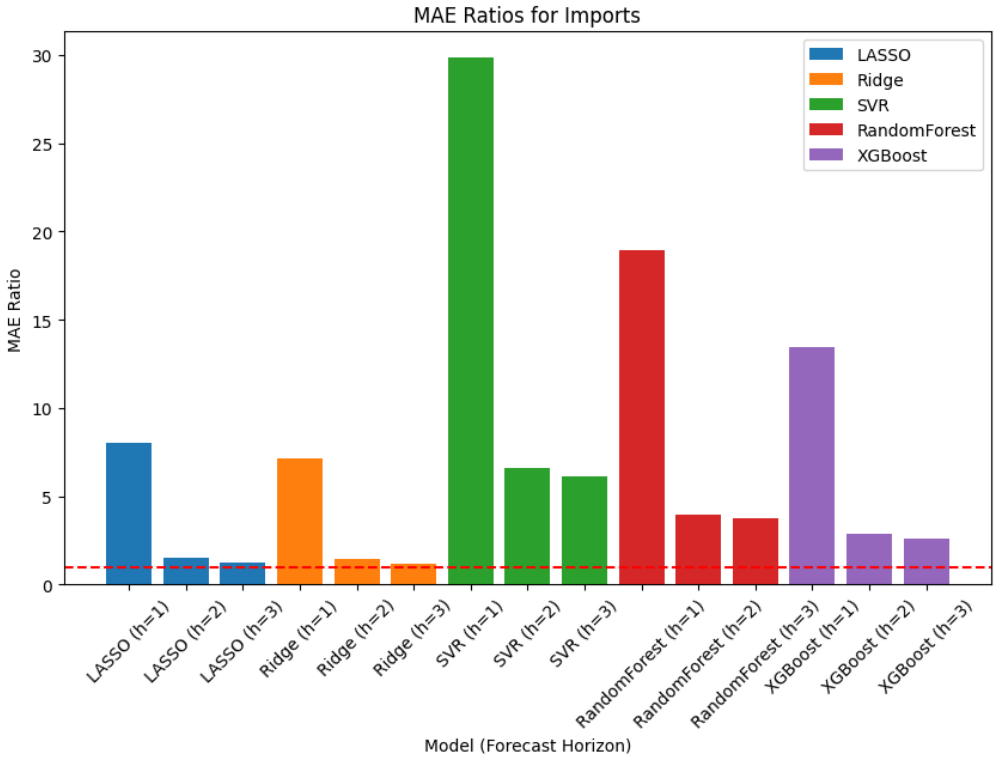


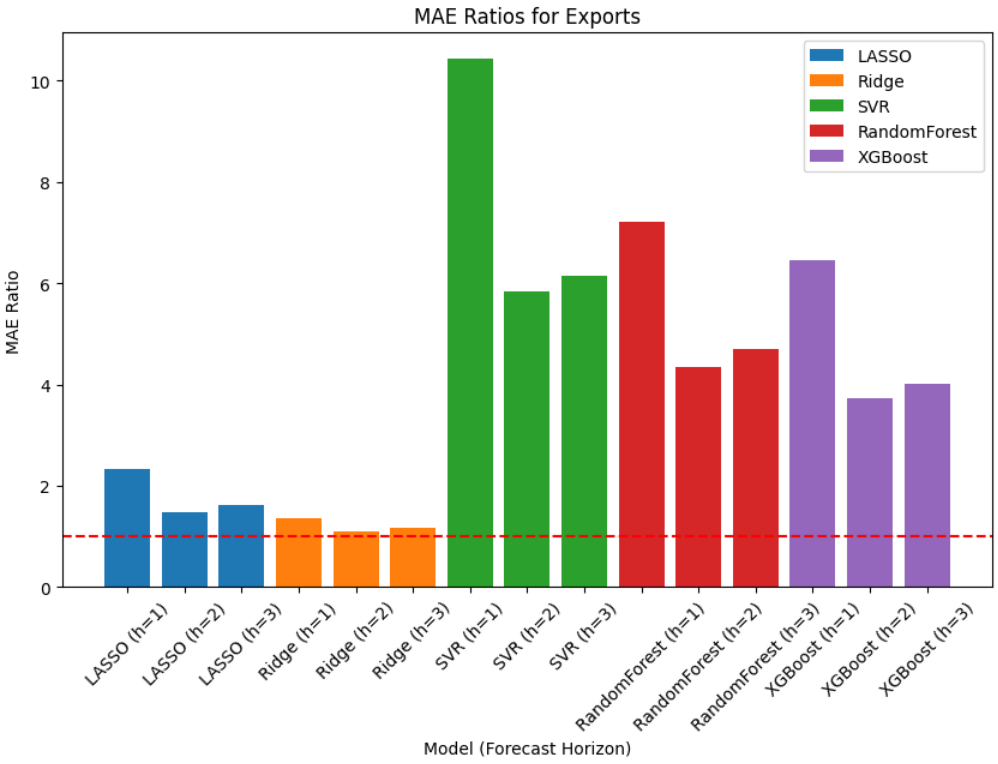

RMSE Ratios for GDP
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
3.0
2.5
2.0
1.5
1.0
0.5
0.0
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

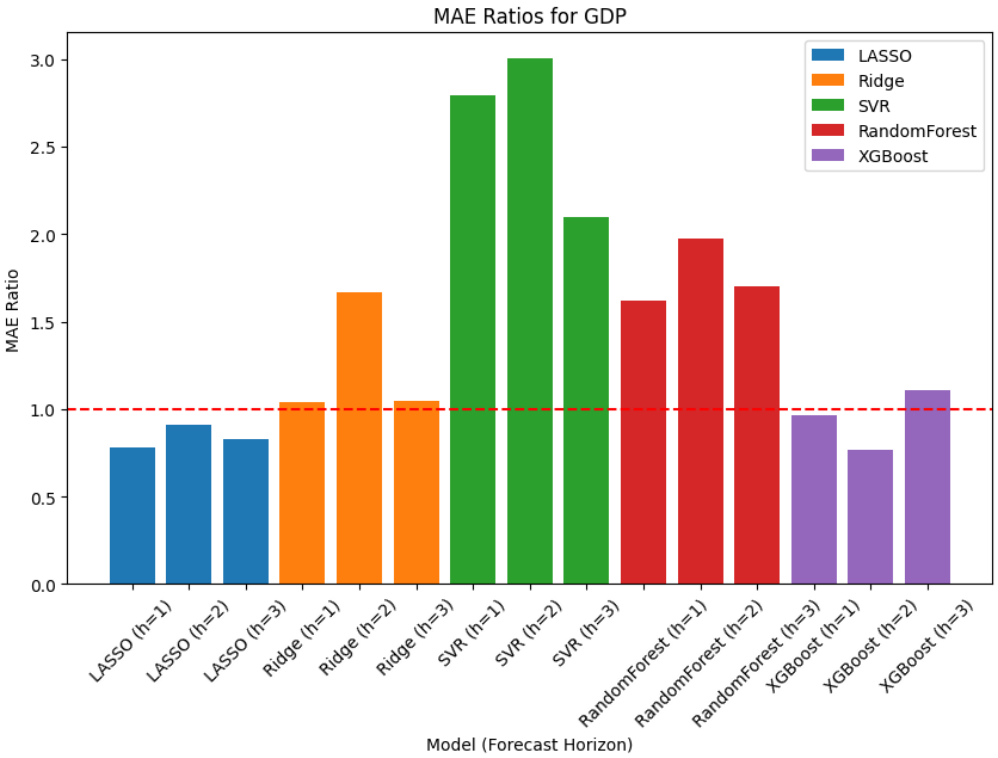

RMSE Ratios for Private Consumption
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

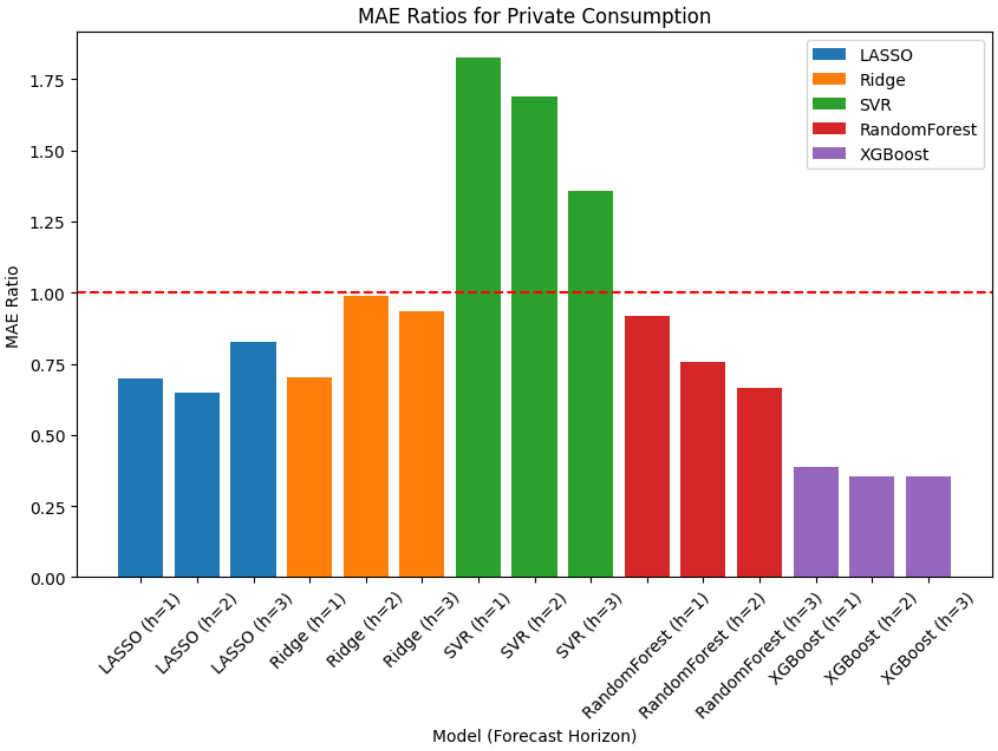

RMSE Ratios for Private Investment
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
0
1
2
3
4
5
6
7
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

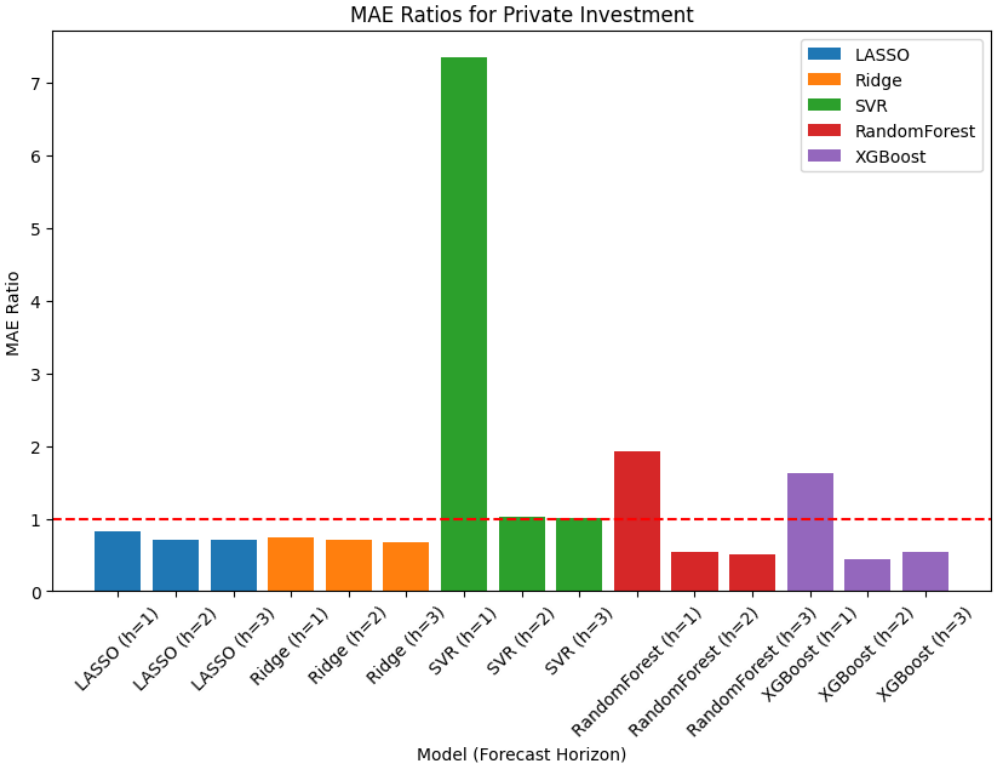
MAE Ratio


MAE Ratios for Imports
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0
5
10
15
20
25
30
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Exports
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0
2
4
6
8
10
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for GDP
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0.0
0.5
1.0
1.5
2.0
2.5
3.0
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Private Consumption
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Private Investment
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0
1
2
3
4
5
6
7
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

Appendix N: Performance Evaluation for Machine Learning Models Without Hyperparameter Tuning (Textual Representation)

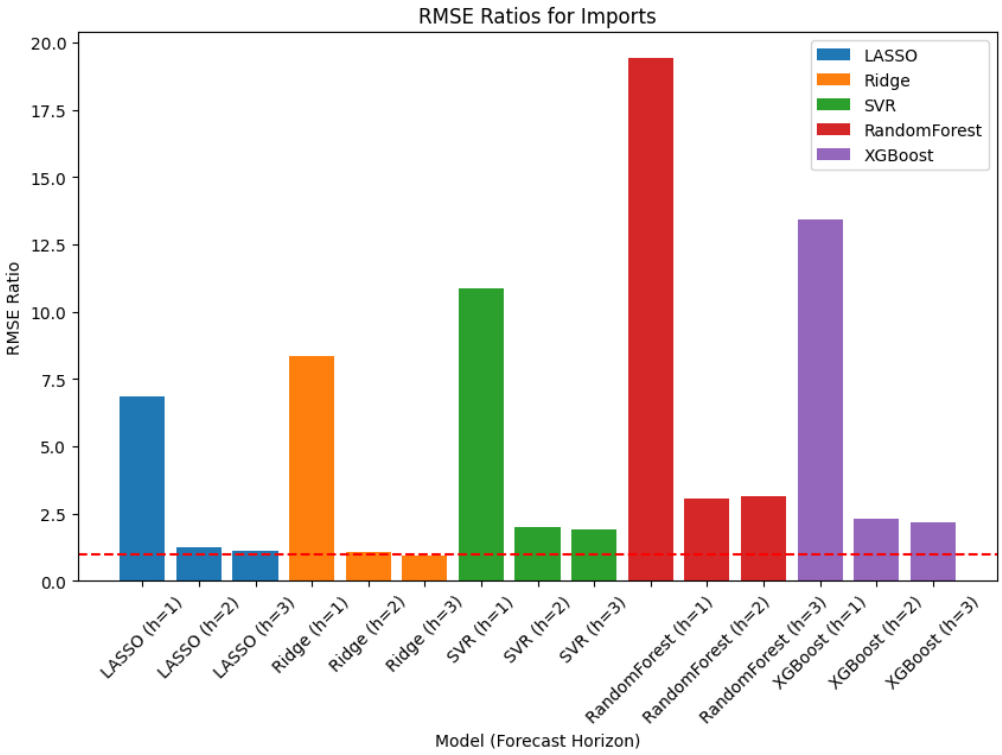
```
Variable: Imports

LASSO:
  Horizon 1: RMSE Ratio=8.062281660686319, MAE Ratio=8.062281660686319
  Horizon 2: RMSE Ratio=1.2214297024700742, MAE Ratio=1.5045688467663425
  Horizon 3: RMSE Ratio=1.1552946454941775, MAE Ratio=1.2619297724130785

Ridge:
  Horizon 1: RMSE Ratio=7.153449142207594, MAE Ratio=7.153449142207594
  Horizon 2: RMSE Ratio=1.2686993414588443, MAE Ratio=1.4644321360470245
  Horizon 3: RMSE Ratio=1.0977836900430233, MAE Ratio=1.194255121865858

SVR:
  Horizon 1: RMSE Ratio=29.8218896159722, MAE Ratio=29.8218896159722
  Horizon 2: RMSE Ratio=5.52157593663728, MAE Ratio=6.579982453805068
  Horizon 3: RMSE Ratio=5.460134116844499, MAE Ratio=6.134595930037081

RandomForest:
  Horizon 1: RMSE Ratio=18.973846086827454, MAE Ratio=18.973846086827454
  Horizon 2: RMSE Ratio=3.174695466244947, MAE Ratio=3.9539597615003537
  Horizon 3: RMSE Ratio=3.276781773277424, MAE Ratio=3.725999585027502

XGBoost:
  Horizon 1: RMSE Ratio=13.440747412134172, MAE Ratio=13.440747412134172
  Horizon 2: RMSE Ratio=2.3205875403048744, MAE Ratio=2.882573619011952
  Horizon 3: RMSE Ratio=2.1973579077882093, MAE Ratio=2.573535199900526


Variable: Exports

LASSO:
  Horizon 1: RMSE Ratio=2.326291862862216, MAE Ratio=2.326291862862216
  Horizon 2: RMSE Ratio=1.3152334241305854, MAE Ratio=1.4825480486266567
  Horizon 3: RMSE Ratio=1.5110437954542975, MAE Ratio=1.6193028874748934

Ridge:
  Horizon 1: RMSE Ratio=1.3540906094153136, MAE Ratio=1.3540906094153136
  Horizon 2: RMSE Ratio=1.0328646740514968, MAE Ratio=1.1063638727223346
  Horizon 3: RMSE Ratio=1.1280017788198773, MAE Ratio=1.1637501613918033

SVR:
  Horizon 1: RMSE Ratio=10.42712094092089, MAE Ratio=10.42712094092089
  Horizon 2: RMSE Ratio=5.1734933061211485, MAE Ratio=5.839480817066229
  Horizon 3: RMSE Ratio=5.685991492955201, MAE Ratio=6.147485883626585

RandomForest:
  Horizon 1: RMSE Ratio=7.203804283984363, MAE Ratio=7.203804283984363
  Horizon 2: RMSE Ratio=3.8092302830469014, MAE Ratio=4.352284010330703
  Horizon 3: RMSE Ratio=4.294345372705995, MAE Ratio=4.708034152948718

XGBoost:
  Horizon 1: RMSE Ratio=6.46835941102377, MAE Ratio=6.46835941102377
  Horizon 2: RMSE Ratio=3.2398641323251507, MAE Ratio=3.7345896313134497
  Horizon 3: RMSE Ratio=3.6529741067466728, MAE Ratio=4.013726860893075
```

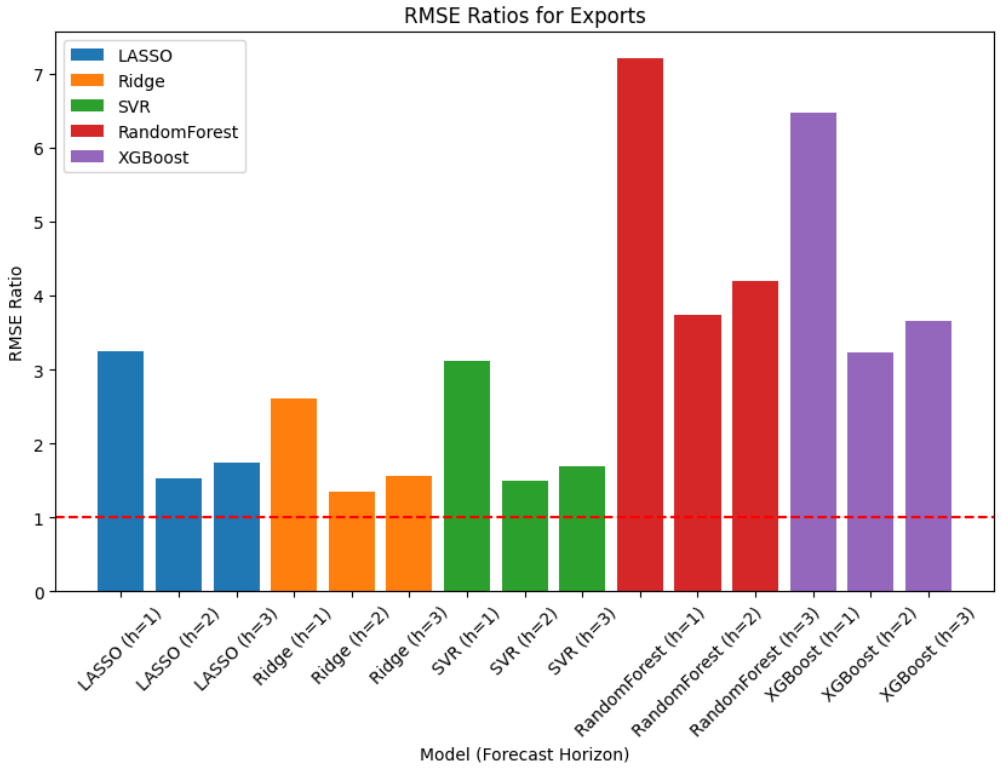
```
Variable: GDP

LASSO:
  Horizon 1: RMSE Ratio=0.7805330002274586, MAE Ratio=0.7805330002274586
  Horizon 2: RMSE Ratio=0.9069400173820447, MAE Ratio=0.908158003302547
  Horizon 3: RMSE Ratio=0.814759474055427, MAE Ratio=0.8315159907771614

Ridge:
  Horizon 1: RMSE Ratio=1.0389645031838963, MAE Ratio=1.0389645031838963
  Horizon 2: RMSE Ratio=1.836018248046334, MAE Ratio=1.667285232008652
  Horizon 3: RMSE Ratio=1.081277511627757, MAE Ratio=1.0447863603817358

SVR:
  Horizon 1: RMSE Ratio=2.7897369823609224, MAE Ratio=2.7897369823609224
  Horizon 2: RMSE Ratio=2.9480284716343825, MAE Ratio=3.0011359953279655
  Horizon 3: RMSE Ratio=1.9176050450174822, MAE Ratio=2.100415923424472

RandomForest:
  Horizon 1: RMSE Ratio=1.6210784298927372, MAE Ratio=1.6210784298927372
  Horizon 2: RMSE Ratio=2.13133270757502, MAE Ratio=1.9734931836653729
  Horizon 3: RMSE Ratio=1.565866522284449, MAE Ratio=1.7020788269294864

XGBoost:
  Horizon 1: RMSE Ratio=0.9683224601872289, MAE Ratio=0.9683224601872289
  Horizon 2: RMSE Ratio=0.7613757261103992, MAE Ratio=0.7695135315987853
  Horizon 3: RMSE Ratio=1.0051155672388936, MAE Ratio=1.1090016263415405


Variable: Private Consumption

LASSO:
  Horizon 1: RMSE Ratio=0.6968461228533559, MAE Ratio=0.6968461228533559
  Horizon 2: RMSE Ratio=0.6355006885336676, MAE Ratio=0.6506524258365922
  Horizon 3: RMSE Ratio=0.8440954777073724, MAE Ratio=0.827776636751802

Ridge:
  Horizon 1: RMSE Ratio=0.70220839517731, MAE Ratio=0.70220839517731
  Horizon 2: RMSE Ratio=1.070514045850506, MAE Ratio=0.988621141007337
  Horizon 3: RMSE Ratio=0.9323866762779056, MAE Ratio=0.9341914784891039

SVR:
  Horizon 1: RMSE Ratio=1.8245466251269242, MAE Ratio=1.8245466251269242
  Horizon 2: RMSE Ratio=1.6360841528757397, MAE Ratio=1.6902020273333314
  Horizon 3: RMSE Ratio=1.2403998789150976, MAE Ratio=1.3568561130061287

RandomForest:
  Horizon 1: RMSE Ratio=0.9170925492261618, MAE Ratio=0.9170925492261618
  Horizon 2: RMSE Ratio=0.7725943196716502, MAE Ratio=0.7554900342609893
  Horizon 3: RMSE Ratio=0.6124200238585203, MAE Ratio=0.6664886852731239

XGBoost:
  Horizon 1: RMSE Ratio=0.38626615447460066, MAE Ratio=0.38626615447460066
  Horizon 2: RMSE Ratio=0.3566957029486573, MAE Ratio=0.35557822843189785
  Horizon 3: RMSE Ratio=0.3316453097881615, MAE Ratio=0.3563342316263795
```

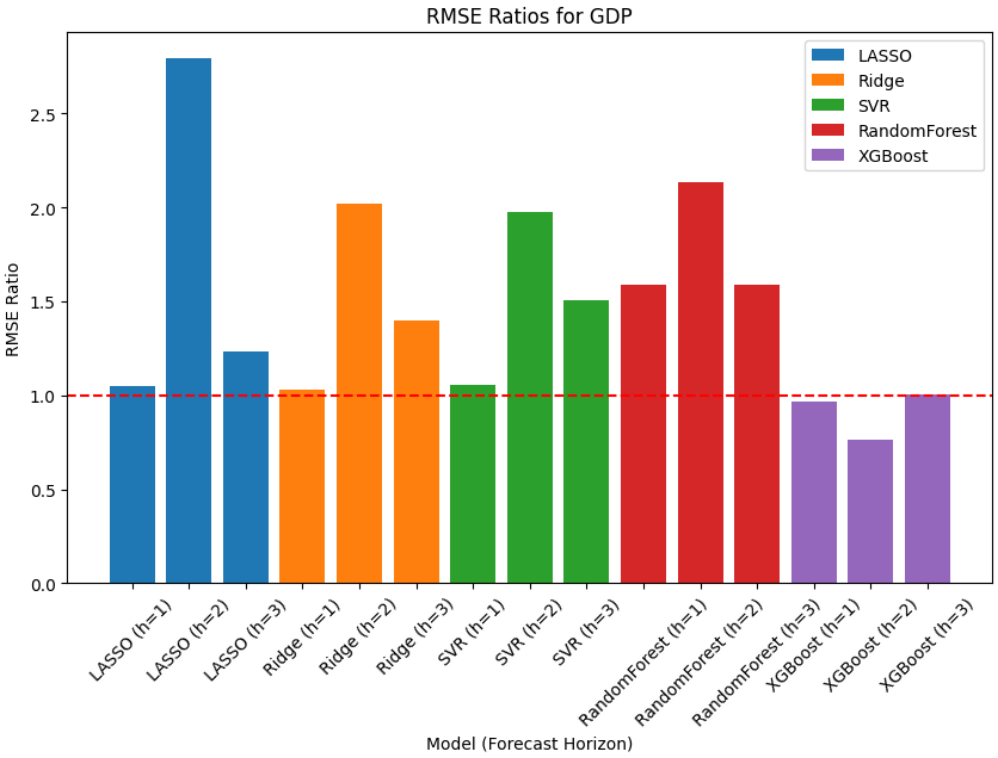
```
Variable: Private Investment

LASSO:
  Horizon 1: RMSE Ratio=0.826428142738611, MAE Ratio=0.826428142738611
  Horizon 2: RMSE Ratio=0.707531945643069, MAE Ratio=0.7146329185923435
  Horizon 3: RMSE Ratio=0.7613506718865624, MAE Ratio=0.7112094478812314

Ridge:
  Horizon 1: RMSE Ratio=0.7420266448964463, MAE Ratio=0.7420266448964463
  Horizon 2: RMSE Ratio=0.7144675489631411, MAE Ratio=0.7055065439153577
  Horizon 3: RMSE Ratio=0.6997119114826631, MAE Ratio=0.6805786381089314

SVR:
  Horizon 1: RMSE Ratio=7.339901841412565, MAE Ratio=7.339901841412565
  Horizon 2: RMSE Ratio=0.884980726402218, MAE Ratio=1.0280825542145777
  Horizon 3: RMSE Ratio=0.9587539885897879, MAE Ratio=1.0061695023206503

RandomForest:
  Horizon 1: RMSE Ratio=1.9215034512788773, MAE Ratio=1.9215034512788773
  Horizon 2: RMSE Ratio=0.49553038687898415, MAE Ratio=0.5429784271501888
  Horizon 3: RMSE Ratio=0.5005324021701076, MAE Ratio=0.5056799470321833

XGBoost:
  Horizon 1: RMSE Ratio=1.6301282483699469, MAE Ratio=1.6301282483699469
  Horizon 2: RMSE Ratio=0.40029583640315547, MAE Ratio=0.44041662121606895
  Horizon 3: RMSE Ratio=0.532473332471259, MAE Ratio=0.5433102594988414
```

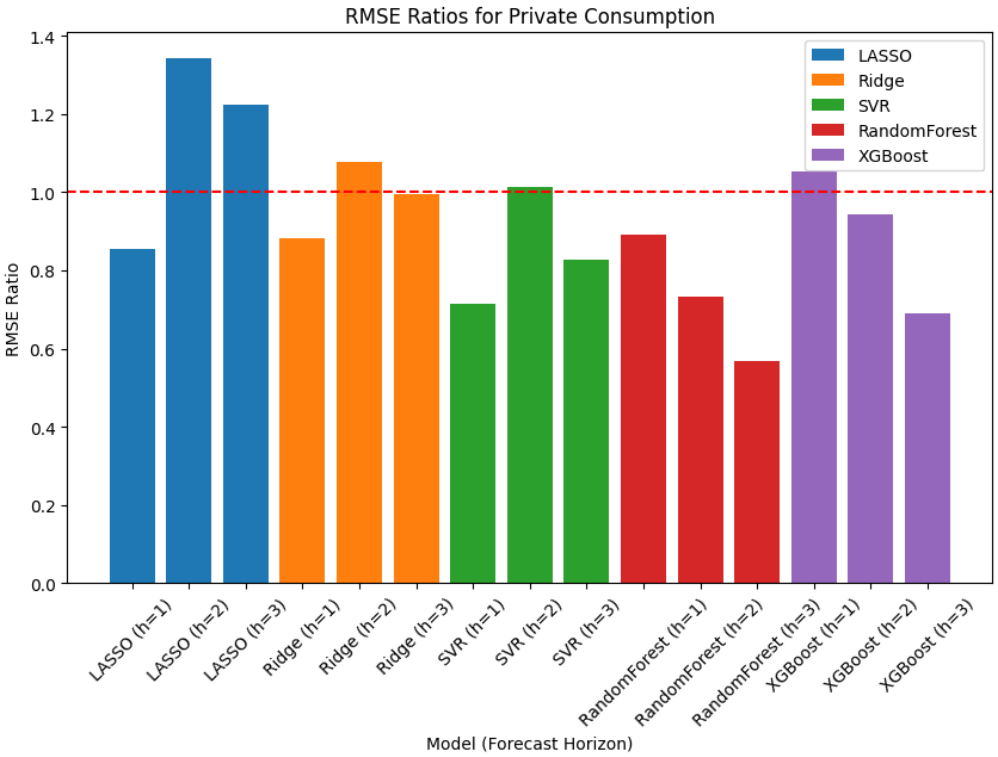
Appendix O: Performance Evaluation for Machine Learning Models with Hyperparameter Tuning (Graphical Representation)

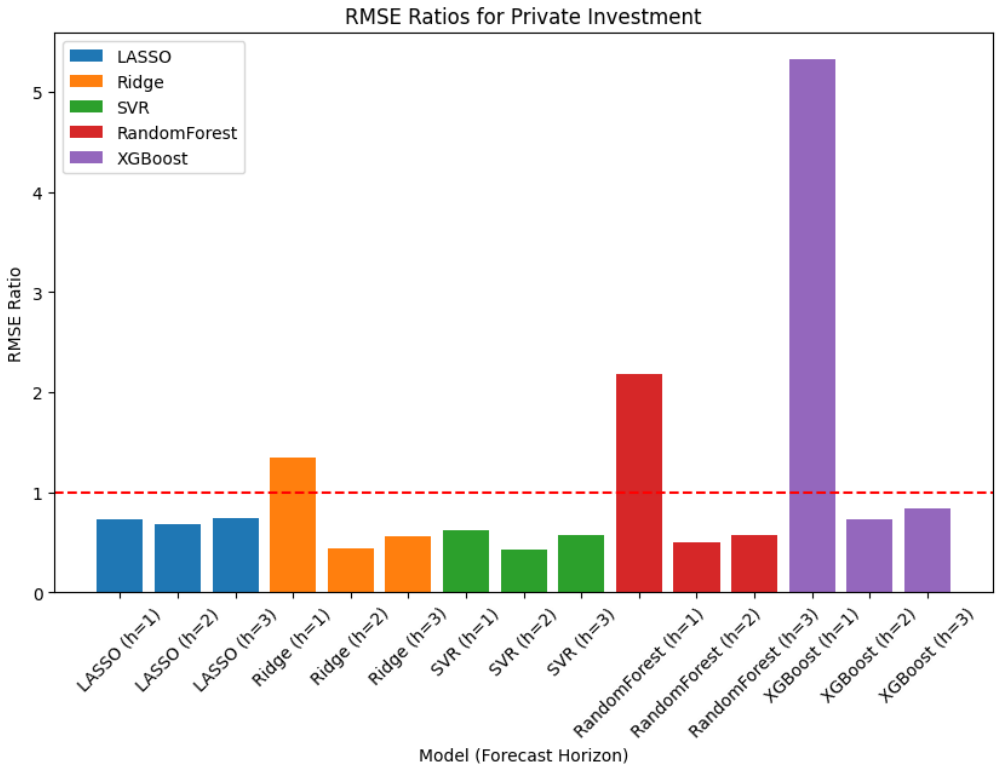
RMSE Ratio



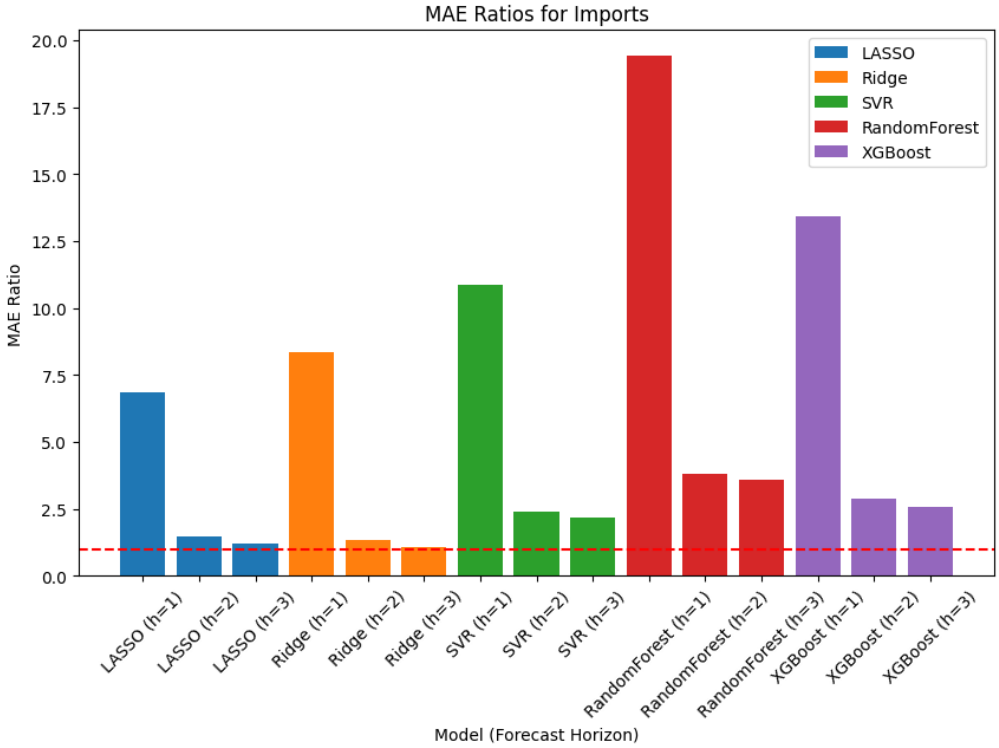


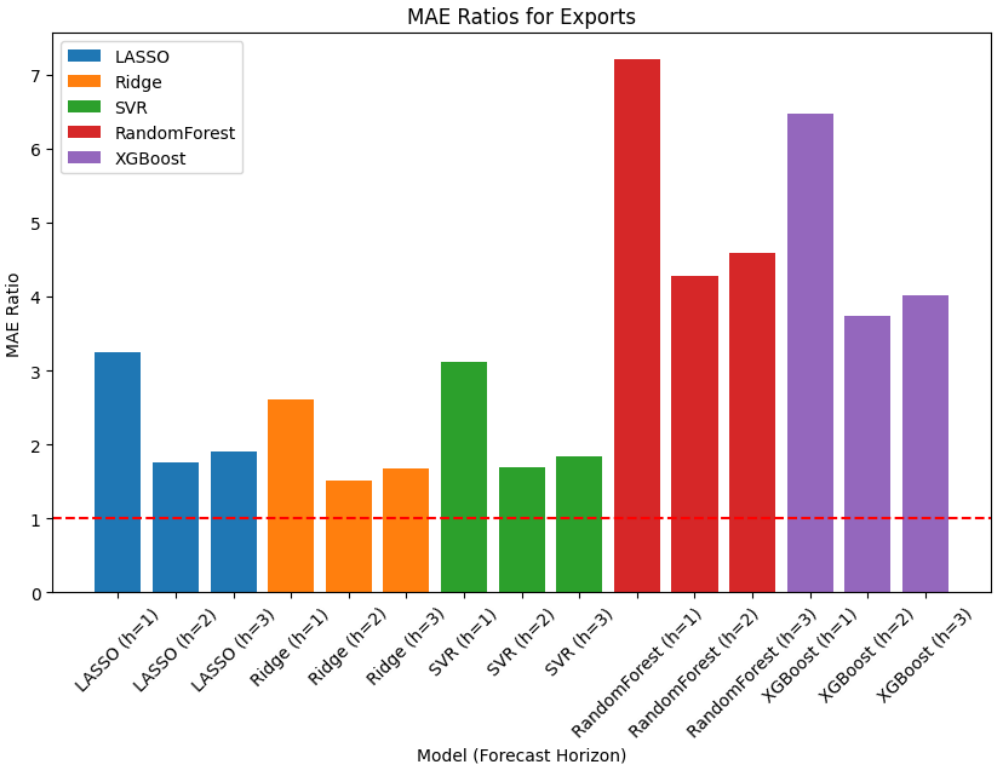

RMSE Ratios for GDP
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
0.0
0.5
1.0
1.5
2.0
2.5
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

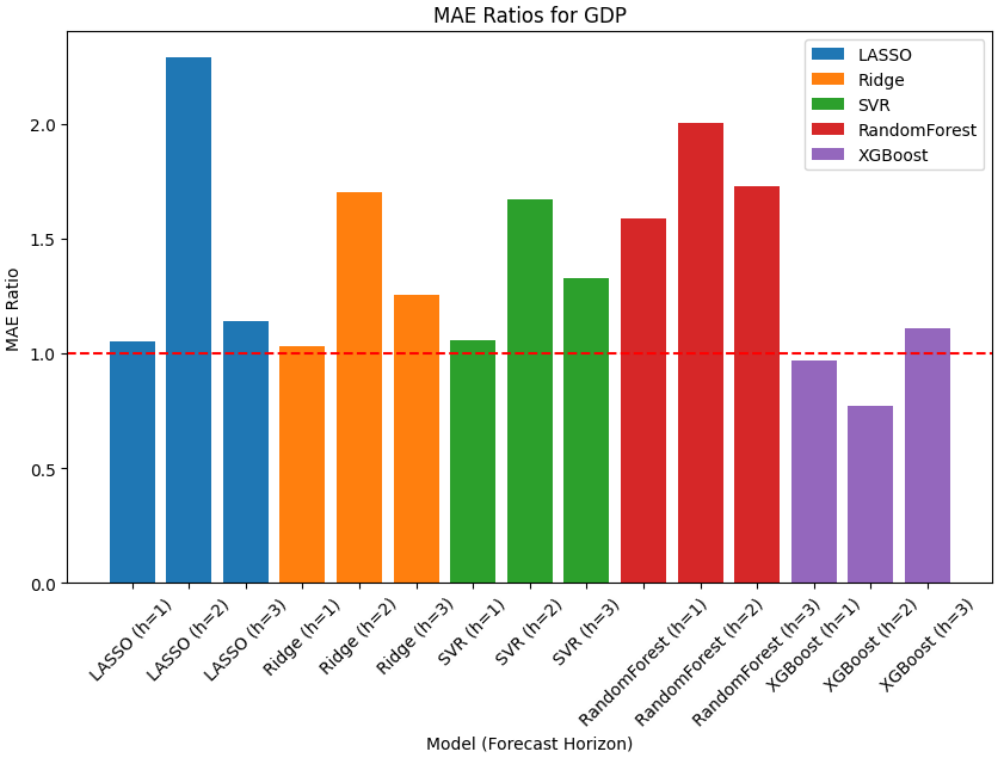

RMSE Ratios for Private Consumption
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

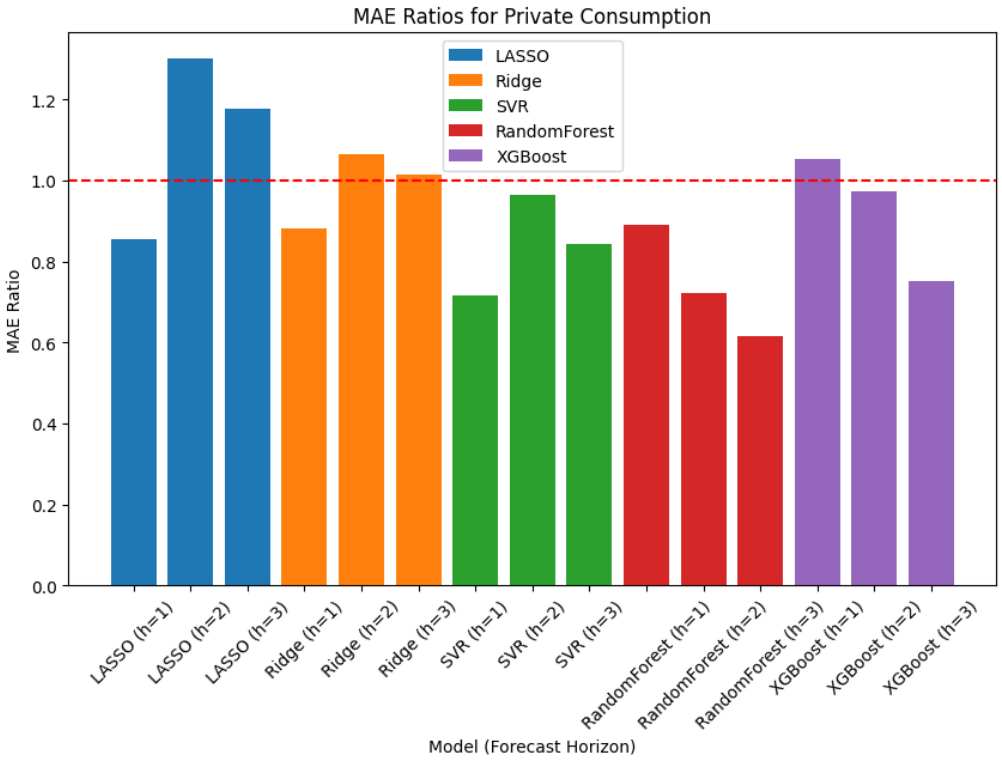

RMSE Ratios for Private Investment
LASSO
Ridge
SVR
RandomForest
XGBoost
RMSE Ratio
0
1
2
3
4
5
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

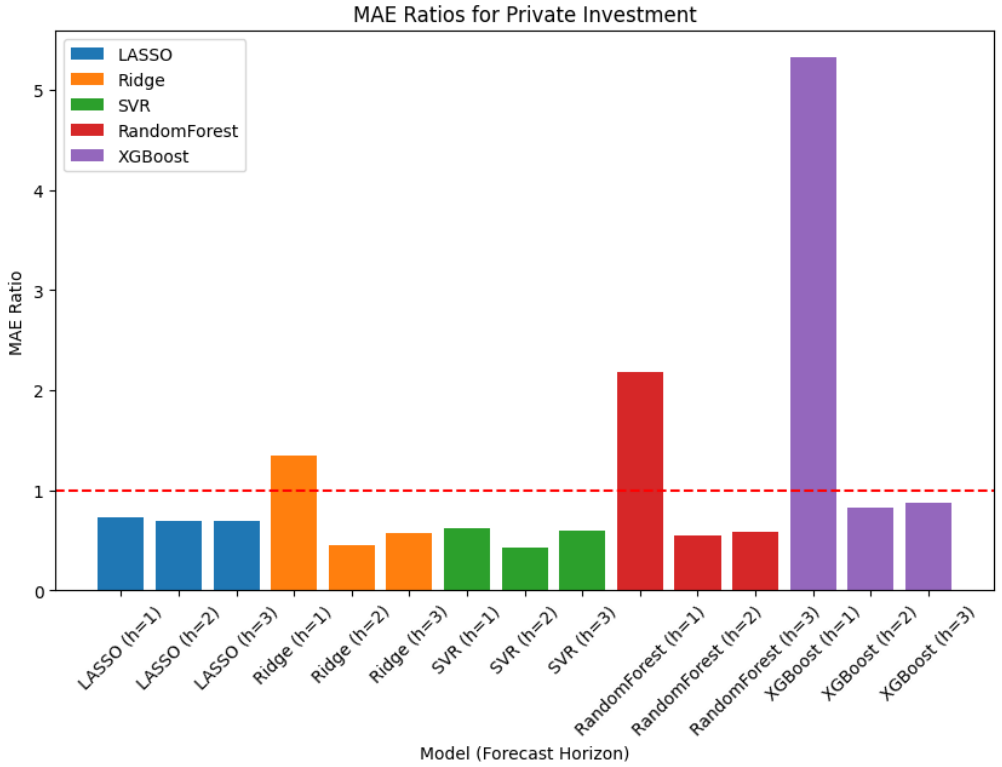
MAE Ratio


MAE Ratios for Imports
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Exports
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0
1
2
3
4
5
6
7
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for GDP
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0.0
0.5
1.0
1.5
2.0
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Private Consumption
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0.0
0.2
0.4
0.6
0.8
1.0
1.2
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)


MAE Ratios for Private Investment
LASSO
Ridge
SVR
RandomForest
XGBoost
MAE Ratio
0
1
2
3
4
5
LASSO (h=1)
LASSO (h=2)
LASSO (h=3)
Ridge (h=1)
Ridge (h=2)
Ridge (h=3)
SVR (h=1)
SVR (h=2)
SVR (h=3)
RandomForest (h=1)
RandomForest (h=2)
RandomForest (h=3)
XGBoost (h=1)
XGBoost (h=2)
XGBoost (h=3)
Model (Forecast Horizon)

Appendix P: Performance Evaluation for Machine Learning Models with Hyperparameter Tuning (Textual Representation)

```
Variable: Imports

LASSO:
  Horizon 1: RMSE Ratio=6.873486295800818, MAE Ratio=6.873486295800818
  Horizon 2: RMSE Ratio=1.273217497618627, MAE Ratio=1.4626844635699217
  Horizon 3: RMSE Ratio=1.12633602869413, MAE Ratio=1.232514069431971

Ridge:
  Horizon 1: RMSE Ratio=8.33135890724605, MAE Ratio=8.33135890724605
  Horizon 2: RMSE Ratio=1.0716518638793118, MAE Ratio=1.3349344316623954
  Horizon 3: RMSE Ratio=0.9516190960563062, MAE Ratio=1.066548191505463

SVR:
  Horizon 1: RMSE Ratio=10.872780949455754, MAE Ratio=10.872780949455754
  Horizon 2: RMSE Ratio=1.9915136631597707, MAE Ratio=2.3826152697348273
  Horizon 3: RMSE Ratio=1.9323983219473573, MAE Ratio=2.165580458454172

RandomForest:
  Horizon 1: RMSE Ratio=19.397379495036315, MAE Ratio=19.397379495036315
  Horizon 2: RMSE Ratio=3.0757662171743547, MAE Ratio=3.7959009435613225
  Horizon 3: RMSE Ratio=3.141984910497537, MAE Ratio=3.5786575584455864

XGBoost:
  Horizon 1: RMSE Ratio=13.44083782690895, MAE Ratio=13.44083782690895
  Horizon 2: RMSE Ratio=2.3205940711947797, MAE Ratio=2.882585167080404
  Horizon 3: RMSE Ratio=2.1973655089944413, MAE Ratio=2.573544028330296

 Variable: Exports

LASSO:
  Horizon 1: RMSE Ratio=3.2491818158296004, MAE Ratio=3.2491818158296004
  Horizon 2: RMSE Ratio=1.530477774567889, MAE Ratio=1.763485379806295
  Horizon 3: RMSE Ratio=1.7430660889841227, MAE Ratio=1.907700245128555

Ridge:
  Horizon 1: RMSE Ratio=2.612423987600157, MAE Ratio=2.612423987600157
  Horizon 2: RMSE Ratio=1.341594038040444, MAE Ratio=1.5211434063647609
  Horizon 3: RMSE Ratio=1.555702825983027, MAE Ratio=1.6745563687343545

SVR:
  Horizon 1: RMSE Ratio=3.1130300013160563, MAE Ratio=3.1130300013160563
  Horizon 2: RMSE Ratio=1.4932277506387663, MAE Ratio=1.7015819154399534
  Horizon 3: RMSE Ratio=1.6915292605657806, MAE Ratio=1.8326851268600264

RandomForest:
  Horizon 1: RMSE Ratio=7.20133732750948, MAE Ratio=7.20133732750948
  Horizon 2: RMSE Ratio=3.7324967430828586, MAE Ratio=4.271301384554824
  Horizon 3: RMSE Ratio=4.192325092508515, MAE Ratio=4.5925402832189715

XGBoost:
  Horizon 1: RMSE Ratio=6.468349966515203, MAE Ratio=6.468349966515203
  Horizon 2: RMSE Ratio=3.239855681828402, MAE Ratio=3.7345796261104076
  Horizon 3: RMSE Ratio=3.652965833016153, MAE Ratio=4.013717031457258
```

```
Variable: GDP

LASSO:
  Horizon 1: RMSE Ratio=1.0509722928441745, MAE Ratio=1.0509722928441745
  Horizon 2: RMSE Ratio=2.792014087644741, MAE Ratio=2.286052764257826
  Horizon 3: RMSE Ratio=1.2329930923199015, MAE Ratio=1.1395120386482969

Ridge:
  Horizon 1: RMSE Ratio=1.032982511192271, MAE Ratio=1.032982511192271
  Horizon 2: RMSE Ratio=2.0228256980025594, MAE Ratio=1.7014971757751374
  Horizon 3: RMSE Ratio=1.395894240861621, MAE Ratio=1.25425800577459

SVR:
  Horizon 1: RMSE Ratio=1.0547598441602073, MAE Ratio=1.0547598441602073
  Horizon 2: RMSE Ratio=1.9774220028603535, MAE Ratio=1.667821678580572
  Horizon 3: RMSE Ratio=1.5062569389212594, MAE Ratio=1.3272137404789244

RandomForest:
  Horizon 1: RMSE Ratio=1.5873296369503773, MAE Ratio=1.5873296369503773
  Horizon 2: RMSE Ratio=2.1365491183396697, MAE Ratio=2.001833528735937
  Horizon 3: RMSE Ratio=1.5867512134013535, MAE Ratio=1.7272638975058738

XGBoost:
  Horizon 1: RMSE Ratio=0.9682714108241772, MAE Ratio=0.9682714108241772
  Horizon 2: RMSE Ratio=0.7613871461540985, MAE Ratio=0.7694973234669319
  Horizon 3: RMSE Ratio=1.005188587525058, MAE Ratio=1.10905456614279


Variable: Private Consumption

LASSO:
  Horizon 1: RMSE Ratio=0.8565247815669649, MAE Ratio=0.8565247815669649
  Horizon 2: RMSE Ratio=1.3413425957316327, MAE Ratio=1.2998739326180375
  Horizon 3: RMSE Ratio=1.2228608968435404, MAE Ratio=1.1760403647253046

Ridge:
  Horizon 1: RMSE Ratio=0.8825650252613588, MAE Ratio=0.8825650252613588
  Horizon 2: RMSE Ratio=1.078418185400769, MAE Ratio=1.0638506096138782
  Horizon 3: RMSE Ratio=0.9965712194377128, MAE Ratio=1.0158719643090166

SVR:
  Horizon 1: RMSE Ratio=0.7151068298545828, MAE Ratio=0.7151068298545828
  Horizon 2: RMSE Ratio=1.0120357498221664, MAE Ratio=0.9637561544560574
  Horizon 3: RMSE Ratio=0.827935296340025, MAE Ratio=0.8424419030723407

RandomForest:
  Horizon 1: RMSE Ratio=0.8906753794727774, MAE Ratio=0.8906753794727774
  Horizon 2: RMSE Ratio=0.7320506623963116, MAE Ratio=0.7214942063366666
  Horizon 3: RMSE Ratio=0.5680855331146235, MAE Ratio=0.6163731048498771

XGBoost:
  Horizon 1: RMSE Ratio=1.0516779762172226, MAE Ratio=1.0516779762172226
  Horizon 2: RMSE Ratio=0.9420832056951198, MAE Ratio=0.9738626821157258
  Horizon 3: RMSE Ratio=0.6889834195479233, MAE Ratio=0.7523057029395029
```

```
Variable: Private Investment

LASSO:
  Horizon 1: RMSE Ratio=0.7301112406702358, MAE Ratio=0.7301112406702358
  Horizon 2: RMSE Ratio=0.6878735271306624, MAE Ratio=0.6955063501267901
  Horizon 3: RMSE Ratio=0.7428010348877225, MAE Ratio=0.6929061336506945

Ridge:
  Horizon 1: RMSE Ratio=1.3446445019361948, MAE Ratio=1.3446445019361948
  Horizon 2: RMSE Ratio=0.4381420292715695, MAE Ratio=0.4574795345394122
  Horizon 3: RMSE Ratio=0.5638102778125978, MAE Ratio=0.5789767087281336

SVR:
  Horizon 1: RMSE Ratio=0.6256372305330968, MAE Ratio=0.6256372305330968
  Horizon 2: RMSE Ratio=0.4256400172875328, MAE Ratio=0.42638933219956615
  Horizon 3: RMSE Ratio=0.5771522826382013, MAE Ratio=0.5924538780697381

RandomForest:
  Horizon 1: RMSE Ratio=2.182966554599528, MAE Ratio=2.182966554599528
  Horizon 2: RMSE Ratio=0.4985602141053553, MAE Ratio=0.5457176195721898
  Horizon 3: RMSE Ratio=0.5734441187987802, MAE Ratio=0.5884096281228127

XGBoost:
  Horizon 1: RMSE Ratio=5.322339041623887, MAE Ratio=5.322339041623887
  Horizon 2: RMSE Ratio=0.7260819948589576, MAE Ratio=0.8319825665436472
  Horizon 3: RMSE Ratio=0.8443470349626382, MAE Ratio=0.8796523564073588
```

Appendix Q: Log Sheets