UNIVERSITY OF TWENTE.

ScanmarQED
Marketing.Illuminated

Efficient Marketing: Economically Integrated Strategies and Short-Term Forecasting

A.J. Stok

2024

Efficient Marketing: Economically Integrated Strategies and Short-Term Forecasting

A thesis presented in fulfillment of the requirements for the degree of
Master of Science

Author:
A.J. Stok
Master: Industrial Engineering and Management
Specialisation: Financial Engineering and Management

University of Twente

Drienerlolaan 5
7522 NB Enschede
The Netherlands

University supervisors:
Dr. R.A.M.G. Joosten
Faculty of Behavioural, Management and Social Sciences (BMS)

Dr. B. Roorda
Faculty of Behavioural, Management and Social Sciences (BMS)

ScanmarQED

Papiermodel 32
3994 DK Houten
The Netherlands

External supervisor:
M. de Koning, MSc.
SVP Product

June 10, 2024

Preface

Dear reader,

Before you lies my Master thesis *"Efficient Marketing: Economically Integrated Strategies and Short-Term Forecasting"*. I conduct this research at ScanmarQED, a provider of marketing software tools. This thesis is the final assignment in fulfillment of the requirements for a Master's degree in Financial Engineering and Management, which is part of the specialisation track in Industrial Engineering and Management at the University of Twente.

I would like to thank ScanmarQED for the opportunity to carry out my research. The company welcomed me with warm hospitality. I specifically want to thank Marieke de Koning for being my supervisor, providing me with an interesting and challenging research topic, in which I could also implement my Econometrics knowledge, and giving me valuable feedback and discussions. Your guidance helped me to improve my thesis. I would also like to thank Brian Cusick for giving feedback on my research and for the interesting discussions about my results and research approach. Moreover, I would like to thank Kenneth Wailes for providing me the data set and valuable insights in the data.

Furthermore, I would like to thank my university supervisor Reinoud Joosten for his guidance, time, detailed feedback, and long discussions that improved my thesis a lot. I would also like to thank my second supervisor Berend Roorda for his feedback.

Finally, I want to thank my family, friends and boyfriend for their advice and support during my studies at the University of Twente and Tilburg University.

I hope you will enjoy reading my Master thesis.

Anouschka Stok

Houten, June 2024

Management summary

Problem Statement: We conduct this study at ScanmarQED, a provider of marketing software tools. ScanmarQED currently relies on a Marketing Mix Model (MMM), a statistical model that uses aggregated data to asses efficacy and effectiveness of marketing and advertising investments. However, there persists uncertainty regarding the suitability of the model and the current forecasting system. This study focusses on the creation of a time-saving automated forecasting system called the AutoForecaster. This algorithm predicts a marketing Key Performance Indicator (KPI), incorporating relevant economic factors when there is a correlation with the KPI. The AutoForecaster selects the best forecasting method from initial prediction models and variable selection techniques found in the literature, while considering a set of predetermined economic factors, performance measures, and the initial data set from the user. We consider a cross-sectional data set, which includes a city variable, dates, marketing variables, non-marketing variables and economic factors such as Vehicle Miles Travelled (VMT), oil price, unemployment rate and Personal Consumption Expenditure (PCE). The research question of this study is:

> *How can an efficient implementation of a marketing AutoForecaster optimise marketing strategies while simultaneously providing up-to-date financial insights and forecasting economic factors?*

Method: We choose to implement different forecasting techniques in our algorithm. Here, we make a distinction between the forecasting techniques for economic factors and the forecasting techniques for the marketing KPI, *Units Serviced*, due to their different natures. Furthermore, we apply different variable selection techniques to identify the important variables that influence the *Units Serviced*. For the economic factors we apply interpolation and implement AR(2), OLS, Lasso, RR, XGBoost, SVR, and KNN as the forecasting techniques. To forecast *Units Serviced*, we select four different variable selection processes to identify the most important variables influencing the KPI: (i) Principle Component Analysis (PCA), (ii) Forward Stepwise Selection (FSS), (iii) Backward Stepwise Selection (BSS) and (iv) Random Forest (RF). We also implement eight different forecasting techniques for the KPI, namely, (i) AR, (ii) OLS, (iii) XGBoost, (iv) SVR, (v) Bayesian, (vi) ARIMA, (vii) RF, (viii) Panel Model. If one of the variable selection methods identifies an economic factor as one of the variables that influences the KPI, the algorithm forecasts this factor. Based on the ratio of the performance measures Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC), the algorithm selects the best model.

Results: We categorise our variables into different marketing communication channels: (i) *Social*, (ii) *Display*, (iii) *Search*, (iv) *Video*, (v) Out-of-Home (*OOH*), (vi) Customer Relationship Management (*CRM*), (vii) *PlayFly*, (viii) Connected TV (*CTV*), (ix) *Direct*, (x) *Radio*, (xi) *Audio*, and (xii) *Print*. Among these, *Social*, *Display*, *Video*, *Search*, and *CRM* show the highest contributions to clicks, Gross Rating Points (GRPs), impressions,

and messages sent.

Although the ARIMA model demonstrated high accuracy for MSE, RMSE, and MAE, the Bayesian model outperformed in terms of BIC and AIC, highlighting a trade-off between model complexity and accuracy. The AutoForecaster selected the Bayesian model because of its comparable accuracy performance measures with ARIMA.

The most effective variable selection method, BSS, identifies the most influential variables for our model, including two economic factors: (i) the unemployment rate, forecast using OLS, and (ii) Vehicle Miles Travelled, forecast using Lasso.

Our analysis reveals that, on average, coefficients associated with media channels tend to be negative, except for *Display*. Economic factors, on the other hand, show positive coefficients on average, with the unemployment rate exhibiting a slightly negative coefficient.

Discussion: We aim to create an automated model to forecast a marketing dependent variable, where various financial models are integrated within a marketing framework, introducing a novel methodology that implements diverse models into an automated system that can select the most suitable model. Although the individual techniques are not novel on their own, their adaptation and integration within the marketing framework increases the novelty of this approach. Additionally, this capability enables a comparison between the results and variables derived from ScanmarQED's internal model and those produced by the automated algorithm.

Conclusion: Our research streamlines the implementation process of the AutoForecaster, contributing to the optimisation of marketing strategies. Our analysis shows the importance of the implementation of marketing variables, non-marketing variables, and economic factors in driving marketing strategy decisions for a forecast horizon of 30/60/90-days.

By creating a dashboard in PowerBI, we ensure a real-time dashboard with an extension to financial dashboards as additional data becomes available. The AutoForecaster provides coefficients of the selected variables, offering actionable insights for optimising marketing strategies.

This first iteration of the AutoForecaster provides a foundation to build upon. Future versions should incorporate more conditions and constraints to enhance its functionality, improving both predictability and accuracy.

Key words: AutoForecaster, Marketing, PCA, FSS, BSS, RF, AR, OLS, KNN, Lasso, SVR, XGBoost, Bayesian, ARIMA, Panel Model, MSE, RMSE, MAE, BIC, AIC.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AIC	Akaike Information Criterion
AR	AutoRegressive
ARF	Adaptive Random Forest
ARIMA	AutoRegressive Integrated Moving Average
BIC	Bayesian Information Criterion
BSS	Backward Stepwise Selection
FSS	Forward Stepwise Selection
GRP	Gross Rating Point
i.i.d.	independently and identically distributed
KNN	K-Nearest Neighbours
KPI	Key Performance Indicator
Lasso	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
MMM	Marketing Mix Model
MSE	Mean Squared Error
OLS	Ordinary Least Squares
PCA	Principle Component Analysis
PCE	Personal Consumption Expenditures
RF	Random Forest
RMSE	Root Mean Squeared Error
ROI	Return On Investment
RR	Ridge Regression
RSS	Residual Sum of Squares
SVR	Super Vector Regression
VMT	Vehicle Miles Travelled
XGBoost	eXtreme Gradient Boosting

Chapter 1

Introduction

"When most people think about the future, they ignore that the future is a distribution of possibilities."

— Howard Marks

The most frequently used marketing instrument remains advertising, billions of dollars are invested in different channels, commonly accounting for 3% of firm sales (Gijsenberg & Nijs, 2019). Incorporating and creating accurate predictions and market dynamics in business forecasting is important, as these forecasts can impact decision-making processes, resource allocation, and overall strategic planning for companies. However, as the quote above describes, there are many potential outcomes of the future. In our initial version of the Autoforecaster, we concentrate on a single outcome, mindful that the future consists of a spectrum of potentialities, as articulated by Howard Marks.

Our goal is to develop an AutoForecaster that helps predict the KPI while considering the influence of market dynamics. Recognising diverse possibilities the future might bring is key for creating a flexible forecasting model. In this chapter, we introduce the structure of our thesis, starting with a definition of the research problem and the associated objectives. It provides a structured framework to define the scope and methodology, ensuring a systematic exploration of the research questions.

We outline the challenges facing the employer, ScanmarQED, in forecasting marketing KPI, hereby, emphasising the complexities of integrating market dynamics into the existing model. By acknowledging the importance of considering various future possibilities in forecasting accuracy, our research aims to provide an improvement of the current implemented models and their predictive abilities.

1.1 Background information

ScanmarQED is a provider of marketing software tools, specialising in practical analytical solutions tailored to the needs of marketing and sales professionals (ScanmarQED, n.d.-a). ScanmarQED uses data-driven software and consultancy services to help users make informed decisions to optimise marketing and sales strategies and achieve their business goals. It helps users to derive insights, facilitating the optimisation of marketing and media strategies, brand development, precise sales forecasting, and the formulation of effective pricing and promotional strategies through the application of a Marketing Mix Model (MMM).

Definition 1.1.1 (Marketing Mix Model)**.** A statistical model that uses aggregated data to assess the efficacy and effectiveness of marketing and advertising investments, aiming to determine their impact on a company's Return On Investment (ROI) (Pandey et al., 2021).

In this definition, efficacy refers to the potential impact of marketing and advertising investments as predicted by theoretical models, while effectiveness examines their real-world impact on the company's ROI. MMM uses historical information to quantify the sales impact of various marketing activities and defines the effectiveness of each of the marketing elements in terms of its contribution to sales volume, effectiveness, efficiency, and ROI. MMM is helpful in the following ways (ScanmarQED, n.d.-b);

- It helps marketeers to identify which marketing channels work best.
- It understands which brands are more/less reactive to marketing activities, which of the latter have been the most profitable, how to better spend the budget and how economic, competitive, seasonal, weather, and operational factors impact sales.
- It links the forecast to the marketing planning process.
- It quantifies the impact of emerging marketing channels such as social media.

1.2 ScanmarQED and MMM

ScanmarQED uses machine learning techniques to develop an approach that offers customers more focus on the interpretation of the results of MMM (ScanmarQED, n.d.-a). With this model, ScanmarQED forecasts a marketing Key Performance Indicator (KPI) for their customers, aiming to help them make better marketing decisions. MMM and forecasting data are applications in the StrataQED desktop tool. The philosophy behind this tool is that it should be a quick-to-learn and user-friendly application which offers complete transparency. With this software, it is possible to test hundreds of models using the automated model search function. By setting a series of parameters about how customers are likely to respond to different marketing activities, the application can implement a behavioural model. The application shows the contribution of the marketing variable to the modelled KPI and the ROI for each marketing channel (e.g. *TV* and *Radio*). Furthermore, every time new data are available, the software automatically refreshes so that the user can easily assess the impact of the changes made to the model. With this model, the user can create a forecast to predict the impact of the marketing variables. Figure 1.1 shows the foundation of ScanmarQED's forecasting model. It starts with the data processing, followed by the model creation with its associated coefficients, resulting the generation of the forecast using this model. ScanmarQED uses Ordinary Least Squares (OLS) for its forecast process and to determine coefficients.

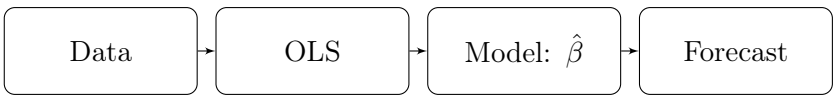


Figure 1.1: StrataQED forecasting process uses an OLS approach. It uses city-specific data from the customer, including daily records of marketing activities, non-marketing activities, economic factors, and KPIs. The model is modified based on the layout and specifics of this data set.

1.3 Problem identification

While MMM can provide valuable insights into historical data and how different marketing variables affect sales or performance, it usually does not provide automatic forecasting. Forecasts are usually generated separately based on historical data and MMM insights. These insights can help the development of more accurate forecasting models or advise in improvements to marketing strategies for the future. MMM is a tool for analysis and understanding, and can be integrated into forecasting processes, but it is not an AutoForecaster by itself.

Definition 1.3.1 (AutoForecaster)**.** An algorithm that selects the best forecasting method from initial prediction models and variable selection techniques found in the literature, considering a set of predetermined market dynamics, performance measures, and the initial data set from the user.

The AutoForecaster represents an innovative algorithm developed specifically for this thesis. It uses information from the immediate past to update its predictions for the future, it integrates new data, with the aim to ensure that the forecast remains accurate and adaptive with the optimal forecasting methodology in place.

As mentioned prior, ScanmarQED uses regression models to forecast data based on MMM results. They want to know whether other methods achieve the same results as their current model. The AutoForecaster aims to simplify the process of integrating new data while offering financial insights into the model's performance. By automating processes with the AutoForecaster, the aim is to reduce the time spent on model development and achieve faster insights for decision-making.

In existing StrataQED models, market dynamics, also known hereafter as economic factors, is regularly reported on a monthly or quarterly basis. These factors play a crucial role in establishing a reliable forecast. However, an issue arises due to the periodic publication of these factors, which does not align with the weekly updated data of the client. Given the client's emphasis on their specific variables, this misalignment poses a significant challenge for ScanmarQED. Developing a model that incorporates economic factors at a more detailed level requires substantial investments of both time and financial resources.

Definition 1.3.2 (Economic factors)**.** Forces that impact prices and consumer behaviour, which are often external factors that result in the creation of pricing signals that affect business or industry (Banton, n.d.; Andersen, 2020).

Economic factors function within StrataQED as a mechanism to prevent potential biases inherent in forecasting models. These economic factors, such as oil prices or inflation, are in the data set periodically reported on a monthly or quarterly basis and hold significance in predicting the KPI. To make a distinction between the economic factors and marketing variables in the models, we create a basic function, which we modify per model,

$$y_t = f(\boldsymbol{X}_t, \boldsymbol{Z}_t) + \varepsilon_t.$$

Here, y_t is the KPI at time t, $\boldsymbol{X}_t$ represents the vector of marketing variables (e.g. *TV* and *Radio*) at time t, $\boldsymbol{X}_t = \begin{bmatrix} X_{1,t} & X_{2,t} & \dots & X_{n,t} \end{bmatrix}$, where n is the total number of selected marketing variables. $\boldsymbol{Z}_t$ corresponds to economic factors at time

t, $\boldsymbol{Z}_t = \begin{bmatrix} Z_{1,t} & Z_{2,t} & \dots & Z_{m,t} \end{bmatrix}$, where m is the total number of selected economic factors. $f(\cdot)$ represents the function that relates the independent variables $\boldsymbol{X}_t$ and $\boldsymbol{Z}_t$ to the dependent variable y_t, and ε_t is the error term with zero mean, which captures the difference between the observed value of y_t and the predicted value by the model $f(\boldsymbol{X}_t, \boldsymbol{Z}_t)$ at time t.

Figure 1.2 shows the quarterly representation of economic factor Z_j, together with its transformation to weekly data. ScanmarQED currently lacks a forecasting capability for economic factors that could otherwise contribute to the prediction of the dependent variable. Due to the investment of developing an AutoForecaster, the uncertainty regarding the future use of the results, and customers prioritising controllable marketing variables, there is an absence of this forecasting capability.

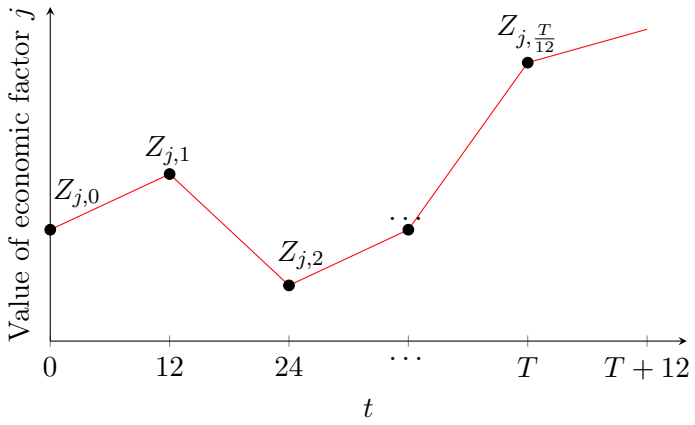


Figure 1.2: Quarterly published economic factor, Z_j, with transformed weekly data: The plotted black markers depict quarterly data of economic factor Z_j at weekly time t, while the red line illustrates the desired weekly data. Positioned above the markers, nodes represent the quarterly values, while the line illustrates the transition between these points, offering a continuous trajectory of the data set's evolution over time. We use information from $t = 0, \dots, T$ to forecast forthcoming trends in economic factors.

1.3.1 Core problem

As mentioned prior, ScanmarQED favours an AutoForecaster that uses marketing data to predict the Key Performance Indicator (KPI), incorporating economic factors when there is a correlation between these factors and the KPI. As such, this model is expected to use marketing objectives and underlying variables to generate forecasts. However, uncertainties persist regarding the suitability of MMM for the current dataset, particularly concerning the integration of economic factors. The challenge lies in the optimising the forecasting model to efficiently use both marketing and economic data. This is crucial for making forecasts, optimising marketing strategies, and providing useful financial insights.

1.4 Problem approach

This study uses a single data set obtained from ScanmarQED and different forecasting methodologies outlined in Chapter 2, *'Theoretical Framework'*. The forecasting analysis focusses on short term periods of 30/60/90 days in accordance with the specifications of ScanmarQED. Given the limited two-year historical data, this approach enhances forecast

accuracy, effectively captures short-term patterns, and facilitates responsive and detailed planning. The data set consists of marketing variables, non-marketing variables, KPIs and economic factors. Table 1.1 shows the data set structure, featuring city-specific date data and metrics such as impressions, clicks, sent, and Gross Rating Point (GRP). Please note that the values provided are illustrative. Additionally, the table incorporates one social media channel, Facebook, measured in impressions. Chapter 3 elaborates further on the data set and measurement metrics. In Chapter 3.3, we examine economic variables that may impact the KPI, including factors such as Vehicle Miles Travelled (VMT), oil price, inflation, unemployment rate, interest rates, population income, and personal consumption expenditures.

City	Date	Facebook Impressions	Economic Factor
1	3/4/2021	10	290
	10/4/2021	5	450
	⋮	⋮	⋮
	30/9/2023	200	305
2	3/4/2021	0	0
	10/4/2021	0	0
	⋮	⋮	⋮
	30/9/2023	500	305
⋮			
39	3/4/2021	350	56
	10/4/2021	0	48
	⋮	⋮	⋮
	30/9/2023	256	150

Table 1.1: Example of the data set with different values, one marketing variable (Facebook impressions), and one economic factor.

Our exploration leads us to formulate the overarching research question for this thesis:

> *How can an efficient implementation of a marketing AutoForecaster optimise marketing strategies while simultaneously providing up-to-date financial insights and forecasting economic factors?*

To find a conclusion for our general research question, we need additional research questions. These questions align with specific chapters within the study. Research Question (RQ) 1 includes the theory needed to create an AutoForecaster by investigating several techniques. RQ 2 addresses the data chapter, clarifying the data set. RQ 3 discusses the methodology, aiming to clarify and provide the foundation of the models. Lastly, RQ 4 elaborates on the findings and results of the AutoForecaster.

RQ 1 How do time-sensitive forecasting techniques, integrating economic factors, alongside variable selection methods and complying to classical assumptions, collectively contribute to accurate short-term predictions in marketing forecasting models?

RQ 2 How are data transformation complexities, missing values, outliers, and multicollinearity issues addressed in the preprocessing steps of the data set for developing the AutoForecaster?

RQ 3 How does the development and implementation of a marketing AutoForecaster contribute to enhancing marketing strategies, and what quantitative insights, including performance metrics, are obtained?

RQ 4 What are the findings and implications of the marketing AutoForecaster across the varying 30/60/90-day time intervals, and how do these findings contribute to understanding marketing strategy dynamics and formulations?

1.5 Research outline

This thesis provides the foundation for the AutoForecaster. Our chapters begin with a theoretical framework, offering a broad overview, and then become more detailed in subsequent chapters, focusing on the algorithm and the development of the AutoForecaster. Figure 1.3 illustrates the different stages of the model in a simplified way, beginning with the path from data set creation to the formulation of forecasts for the dependent variable.

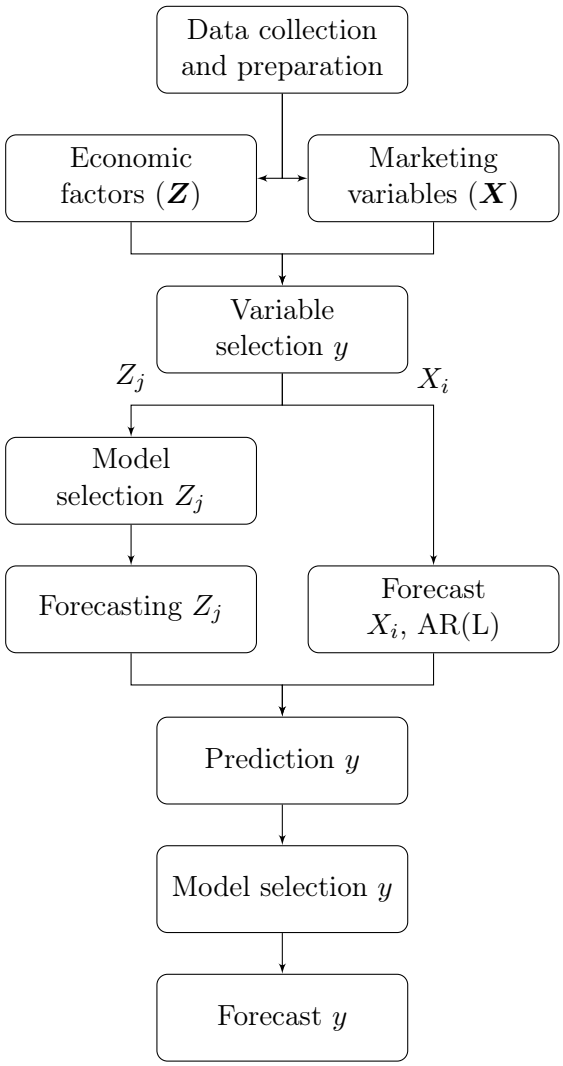


Figure 1.3: AutoForecaster workflow: user input to an automated forecast. Our setup incorporates variable selection, where the selected variables are inputs for prediction and forecasting. $\boldsymbol{Z}$ represents economic factors, y is the KPI, and $\boldsymbol{X}$ are the media variables. Z_j and X_i are the selected variables. Both Z_j and y use different prediction models to identify the best forecasting model.

Beginning with the initiation of the data set, the variable selection process for the target variable y incorporates both the media variables and economic factors. This method identifies the most influential variables for the KPI. The algorithm only forecasts an economic factor, if it is identified through the variable selection process. The marketing variables use AR(L) to forecast. We proceed to predict our KPI, using the final prediction model to forecast its values. We explain all steps in the different chapters each serving a different purpose in our research:

Ch. 2 Theoretical Framework adresses the theoretical foundation for the AutoForecaster and forecasting techniques. It addresses RQ 1.

Ch. 3 Data covers details about the data, pre-processing steps, and the data analysis process, addressing RQ 2.

Ch. 4 Methodology includes information about the approaches and the in-depth formulations of the models, covering RQ 3.

Ch. 5 Results presents the algorithm's outcomes. RQ 4 serves as the foundational basis for this chapter, providing guidance on data presentation.

Ch. 6 Conclusion addresses the main research question. In addition, it includes discussions of research limitations and proposals for further investigation.

The approach begins with an exploration of theoretical frameworks to acquire insights into the subject. Subsequently, the data section explains the data set in detail. Additionally, the methodology section defines the algorithm and integrates supplementary theoretical research for the development of the AutoForecaster. Furthermore, this chapter covers data processing and code development, with an explanation of the results provided in the results chapter. During the research, we use the data of an anonymous client of ScanmarQED. We use different tools mentioned in Appendix A.

Summarising, throughout this thesis, we develop the creation of an AutoForecaster. The next chapters aim to answer the questions about the AutoForecaster's effectiveness and impact by analysing different techniques. Each chapter contributes in understanding how to integrate economic factors, build forecasting algorithms and gain valuable insights for improving marketing strategies. This structured approach aims to provide a comprehensive understanding of the AutoForecaster's capabilities and its practical uses in marketing analytics.

Chapter 2

Theoretical framework

> *"Forecasts usually tell us more about the forecaster than of the future."*
>
> — Warren Buffett

Our research investigates various forecasting models and corresponding variables. The selection of these models do not only reflect the qualities of the data but also on the biases, perspectives, and expertise of the forecaster. The objective of this chapter is to identify a range of forecasting models and variable selection methods applicable for both economic variables and marketing KPI forecasting. Throughout this chapter, we aim to answer the corresponding research question.

> *How do time-sensitive forecasting techniques, integrating economic factors, alongside variable selection methods and complying to classical assumptions, collectively contribute to accurate short-term predictions in marketing forecasting models?*

This research aims to provide a complete understanding of the decision making process, acknowledging the relationship between the chosen models and the expertise integrated within the automated forecasting system. The quote implies that forecasts often show the biases, viewpoints, and expertise of those who make them, rather than precisely predicting the future. Since the AutoForecaster is an automated system, it is expected to reduce biases and subjective influences, thereby aiming for more objective and accurate predictions.

2.1 Economic factors

Economic factors, characterised by their external nature (beyond the control of individual businesses) and their dynamic nature (changing continuously and unpredictable), significantly impact businesses and industries (Andersen, 2020). Fluctuations in the consumer purchasing behaviour directly impact market sales of particular products (Hartanto et al., 2022). To gain competitive advantages in sales and marketing strategies, organisations must understand and address the challenges within consumer sales patterns. Integrating economic factors like inflation, currency dynamics, and fluctuations in oil prices, could enhance the sophistication of forecasting a marketing related KPI.

2.1.1 Challenges

During the economic crisis of 2009 and the COVID-19 pandemic, prompt adaptation was critical to address the marketing uncertainty in consumer demand and behaviour

(Ishrat et al., 2023). Here, businesses faced the need to quickly adjust their marketing tactics to meet customer needs. This economic uncertainty poses a significant challenge in forecasting, as these factors are unpredictable and exclusion can cause inaccurate forecasts. For example, the price of cacao demonstrates a high correlation with the dollar's value, contributing to product volatility (Setiawan et al., 2020); a similar scenario can occur with our marketing related KPI and other economic factors like oil prices.

Morlotti et al. (2024) illustrate that increased price volatility negatively influences consumer purchasing behaviour, particularly concerning revenue managed goods. Incorporating economic factors in marketing strategies involves various risks, including currency, credit, liquidity, political, legal, competitive, technological, and economic risks (Yankovoy et al., 2023). While our model uses different prediction techniques such as linear regression, obtaining an accurate and high-quality data set becomes critical for successful data-driven marketing initiatives, as low quality information can influence unfavourable decision-making processes, negatively impacting marketing and communication outcomes (Rosário & Dias, 2023).

2.1.2 Forecast

Forecasting economic factors can play a crucial role in supporting the forecast analysis of a marketing KPI and in informed decision making (Guégan & Rakotomarolahy, 2010). Ismail et al. (2009) employ forecast factors, as independents for revenue forecasting, when these factors are not available for future time periods.

Economic forecasting models represent sophisticated systems of relationships between variables, with equations estimated from available data to construct a time series (Hendry & Clements, 2003). However, the prediction of economic factors faces the challenge of structural breaks, as relationships between economic factors tend to change over time (Thu & Leon-Gonzalez, 2021).

Given this thesis objective of achieving a marketing KPI forecast of 30/60/90 days, comparison primarily involves one-month error among various short and long-term forecasting models. Table 2.1 incorporates the outcomes of these model comparisons, highlighting top-performing models in the studies cited.

Table 2.1: Top performing forecasting models for economic factors from papers cited in the literature column.

Top-performers	**Literature**
AR(2), OLS	Thu & Leon-Gonzalez (2021)
OLS, KNN	Maccarrone et al. (2021)
DFM, Lasso	Bantis et al. (2023)
SVR, VAR	Zhao et al. (2023)
RR, Lasso	Zhang et al. (2023)
XGBoost	Yang et al. (2023)

Abbreviations: AutoRegressive (AR); Ordinary Least Squares with exogenous variables (OLS); K-Nearest Neighbours (KNN); Dynamic Factor Models (DFM); Least Absolute Shrinkage and Selection Operator (Lasso); Super Vector Regression (SVR); Vector AutoRegressive (VAR); Ridge Regression (RR); eXtreme Gradient Boosting (XGBoost).

Sarwar et al. (2023) and Wichitaksorn (2022) confirm the efficiency of SVR and Lasso for forecasting economic factors. While, VAR and DFM are particularly advantageous for

capturing the interdependencies and interactions among multiple dependent variables over time, our analysis focusses on predicting a single variable. Therefore, we exclude VAR and DFM models. Consequently, we employ the remaining methods listed in Table 2.1 - AR(2), OLS, Lasso, RR, XGBoost, SVR, and KNN - in the AutoForecaster to predict economic factors.

2.2 Marketing dependent variable

We introduce various marketing forecasting techniques, to identify the most suitable method for implementation in the AutoForecaster. We begin this section with a brief overview of the forecasting timeframe. Subsequently, we explain the different variable selection and forecasting techniques identified in the relevant literature within the same context as our research.

Here, the primary objective is to identify the most suitable variable selection and forecasting techniques and adjust these methods to forecast the marketing related KPI, at a 30/60/90-days timeframe. We employ a single-step forecasting technique, in which the algorithm selects the same model for each step within the forecast horizon.

2.2.1 30/60/90-days timeframe

We focus in this research on a short-term horizon of 30/60/90 days, covering a time frame of up to three months at maximum. Accordingly, we use a time based split which covers two years to discover seasonality, patterns and trends. Short-term forecasting is based on historical data, current trends, and reasonable future assumptions to predict cash flows. Peng et al. (2018) concludes that short-term forecasting tends to perform well, often resulting in more precision compared to horizons spanning twelve months.

We assume a long-term forecast spans a twelve month period or more. As indicated by Hendry & Clements (2003), the accuracy of the forecast tends to decrease with increasing forecast horizon. This decline occurs due to the accumulation of innovation errors and a reduction in predictability as the forecast period extends.

2.2.2 Variable selection

Variable selection methods help identify relevant correlated variables associated with the forecast of the KPI. This approach provides better completeness and accuracy of the forecasting model by exclusively incorporating variables correlated with the dependent marketing variable.

According to Wang et al. (2023), there are pros and cons between newer techniques like Lasso and more classical procedures like Forward Stepwise Selection (FSS) and Backward Stepwise Selection (BSS). Lasso relies on certain assumptions to achieve optimality, which are challenging to verify in practical applications. Moreover, while FSS and BSS do not entail tuning parameters, more advanced algorithms typically display sensitivity to the choice of these parameters and initial values. Thus, we compare classical procedures with modern algorithms based on their empirical performance.

Cui et al. (2010) favours FSS and BSS over stepwise selection, as this method is computationally inefficient, especially with a large data set. Li & Shi (2023) favours the efficiency of Random Forest (RF), while Asad et al. (2021) identifies Principle Component Analysis (PCA) and Recursive Feature Elimination (RFE), which is similar to BSS, as

efficient variable selection methods.

Consequently, the methods employed for selecting variables aimed at forecasting the marketing dependent variable include PCA, FSS, BSS and RF.

2.2.3 Forecasting

In marketing forecasting techniques, two distinct types emerge: static and dynamic models. The static model involves a learning process embedded in both the hidden and the output layers (Moshiri et al., 1999). Contrarily, dynamic models facilitate feedback from various layers to the input layer, enabling the dynamic behaviour of the series to be captured. The models differ in their approach to handling time-dependent relationships and updating information over time. In static models, the parameters are fixed and do not change over time. It assumes that the relationship between variables remains constant over time. Once the model is trained on historical data, the same set of parameters is used to forecast future time periods. Dynamic models, allow the parameters to change over time. The model adapts to the evolving relationships between variables and is updated as new data becomes available. Table 2.2 shows the models that perform best in predicting the marketing KPI.

Table 2.2: Optimal performing forecasting models for the marketing KPI from various research articles.

Forecast models	**Literature**
Dynamic Bayesian	Migon et al. (2023); Martin et al. (2023)
AR	James et al. (2013)
ARIMA	Wang & Liu (2022)
XGBoost, RF	Sajawal et al. (2023); Singh & Srivastava (2020); Gunjal et al. (2022)
SVR	Wang & Gu (2022)
ARF, SVR, OLS	Rožanec et al. (2021)

Abbreviations: AutoRegressive (AR); Autoregressive Integrated Moving Average (ARIMA); eXtreme Gradient Boosting (XGBoost); Random Forest (RF); Super Vector Regression (SVR); Adaptive Random Forest (ARF); Ordinary Least Squares (OLS).

Both the ARF and RF models rank among the top performers in the literature. Given that ARF represents an adaptive variant of RF, we prioritise RF for integration into our AutoForecaster, as Python currently lacks the support of the ARF function. Nonetheless, our intention is to merge both techniques in future research to develop a dynamic model.

We also explore a novel approach not found in the marketing literature; a panel model regression applied to marketing data. Considering the potential information of various entities within the data set, the use of Panel models becomes relevant due to the inclusion of panels. Here, $c = 1, \ldots, C$ denotes the entities (cities) and $t = 1, \ldots, T$ represents the time variable. Billé et al. (2023) shows the efficacy of dynamic panel models in forecasting regional GDP.

We exclude VAR, as discussed in Section 2.1.2, due to the focus of predicting only one marketing KPI, for which AR is more suitable. Therefore, the selected marketing forecasting models are AR, OLS, XGBoost, SVR, Dynamic Bayesian, ARIMA, RF, and Panel model.

2.3 Classical attributes

We focus on key attributes within statistical modelling that are crucial for building reliable and accurate forecasting models. Although not all characteristics need to be met for forecasting, certain models may require additional specific properties, detailed in Chapter 4. Here, we emphasise the main classical attributes: zero unconditional mean, stationarity, exogeneity, and homoscedasticity, along with their relevance to respective models. We start by explaining the zero unconditional mean, which is essential for understanding the error term in regression models. Then, we discuss stationarity, emphasising its importance for maintaining consistent data patterns over time. Next, we introduce exogeneity, highlighting the significance of independent predictors in regression analysis. Although equally important, we end with exogeneity as it relates to residual distribution and parameter estimate accuracy.

2.3.1 Zero Unconditional Mean

$\mathbb{E}(\varepsilon_t) = 0$ is the zero unconditional mean of a random error (Das, 2019). It implies that the factors influencing the dependent marketing variable are not correlated to the independent variables, eliminating bias. This condition ensures that the model errors average to zero. A nonzero mean suggests bias in the forecasting method, impacting model accuracy.

2.3.2 Stationarity

van Greunen & Heymans (2023) state that data can manifest two forms of stationarity: strict stationarity and covariance stationarity. Strict stationarity implies that time series joint distribution remain unchanged within a specific period. However, the most commonly observed form, covariance stationarity, requires a constant mean, variance, and covariance over time. Typically, raw time series data are non-stationary and needs transformation to achieve stationary before analysis. One method to achieve stationarity involves employing a first difference approach when handling time series data. The stationarity characteristic facilitates the application of suitable models, ensuring consistent statistical properties, and improves the reliability of forecasts and insights obtained from the data.

2.3.3 Exogeneity

Exogeneity assumes that the expected noise value is not a function of independent variables (Das, 2019). $\mathbb{E}(\varepsilon_t|x_1, x_2, \ldots, x_k) = 0$, shows that the expected value of the error term given the independent variables is zero. This indicates that the error term is not related to or influenced by the independent variables in the model. It ensures that the model accurately captures the relationship between the dependent and independent variables, without the error term being biased or influenced by the predictors. When this assumption is violated, this is known as endogeneity. The exogeneity condition is equivalent to $cov(\varepsilon_t, x_i) = 0$. Exogeneity violations can lead to biased and inefficient forecasts.

2.3.4 Homoscedasticity

Homoskedasticity represents the constant variance in random noise across the entire sample of a regression model (Das, 2019). This assumption implies that the noise term maintains a finite variance σ^2, i.e., $\mathbb{E}(\varepsilon_t^2) = \sigma^2$, where $\mathbb{E}(\varepsilon_t) = 0$. Heteroscedasticity arises when this assumption is violated, leading to unequal variances across predictions, impacting the forecast precision.

2.4 Conclusion

In summary, we found short-term forecasting methodologies, particularly for predicting a marketing related KPI and including economic factors into predictive models. Time series forecasting is a valuable tool, using historical data patterns to extrapolate future values. Machine learning techniques such as Regression, SVR, RF, and XGBoost adapt to change quickly, while methods like OLS and AR capture time-sensitive trends, significantly impacting short-term variable predictions.

Incorporating economic factors into forecasting models enhances the accuracy and reliability of the forecast by illustrating their impact on the marketing KPI. Forecasting methodologies such as AR(2), OLS, KNN, Lasso, SVR, RR, and XGBoost are helpful techniques to to forecast these economic factors.

Classical attributes, including zero unconditional mean, stationarity, exogeneity, and homoscedasticity, contribute to creating reliable and accurate forecasts. While not all attributes need to apply to a specific model, additional attributes may exist.

Various forecasting models are available for predicting the marketing KPI within a short-term horizon of 30/60/90 days. These models, including Dynamic Bayesian, AR, ARIMA, XGBoost, RF, SVR, OLS, and Panel model, are applicable within the specified timeframe horizon. The marketing KPI forecast includes variables selected by FSS, BSS, PCA, or RF.

Moreover, the AutoForecaster automatically selects the best models to predict economic factors and the marketing KPI. This chapter highlights the significance of time-series analysis, the inclusion of economic factors, the classical attributes, various variable selection techniques, and different forecasting models. These elements collectively form the foundation for this research.

Chapter 3

Data

> *"Information is the oil of the 21st century, and analytics is the combustion engine."*
>
> — Peter Sondergaard

Peter Sondergaard's quote emphasises the 21st century's reliance on data, comparing it to oil and analytics to the engine driving process. This analogy captures the role of data in shaping our world and indicates its potential in enhancing our understanding, decision-making, and innovation. In this chapter, we investigate the correlation between the variables in the data set and external economic factors, exploring how they influence marketing strategies. We answer in this chapter the following research question:

> *How are data transformation complexities, missing values, outliers, and multicollinearity issues addressed in the preprocessing steps of the data set for developing the AutoForecaster?*

By addressing this question, our aim is to resolve the complexities of the data environment and solve unclear data gaps. Furthermore, this chapter is the basis for continuing our research of the AutoForecaster.

3.1 Data Source and acquisition

We acquired this data set from ScanmarQED, which serves as the basis for our research in developing the AutoForecaster. We maintain strict confidentiality by disclosing actual values and by transposing them prior to publication. Additionally, we convert the unstructured text columns into integers to create a structured data set. In this chapter, we explore the data set to discover multiple dependencies between variables.

In our research, our objective is to assemble a data set that includes the economic factors sourced from the Internet. We conduct thorough research to ensure the transparency of our data, explicitly citing all sources and maintaining clear traceability of our data collection process.

3.2 Data Description

In this study, we analyse a data set from a major U.S.service retailer, operating in three distinct geographical regions, each with its unique sales pattern and model. Our research focusses on the region with the most extensive data, excluding the other markets.

We consider three KPIs: net sales, gross sales and the number of units serviced. Gross sales represent the total revenue generated before deductions, providing a comprehensive view of overall revenue performance, while net sales reflect revenue after accounting for returns, discounts, and allowances, focussing on effective revenue after adjustments. The variable *Units Serviced* tracks the actual number of units serviced.

Both net and gross sales present a distorted picture of reality. Specifically, when a unit undergoes a free service, gross sales include its potential earnings, while net sales do not account for this scenario at all. In contrast, the number of units serviced provides a more reliable variable as it takes into account all units being serviced, making it our dependent variable.

The initial analysis examines trends in the dependent variable to gain insights into campaign effectiveness and service performance. Our goal is to develop a financial dashboard that provides insights into the performance of the AutoForecaster. We exclude calculations like ROI as the earnings and expenditure data is not accessible.

The data set comprises 172 variables with 5102 entries, reflecting the regional granularity across 39 cities, spanning from April 3, 2021, to September 30, 2023, resulting in 131 data points per city. Variables include dummy variables, economic factors, and marketing variables, such as clicks, impressions, sent, and Gross Rating Point (GRP) of a platform. Impressions denote the number of times an advertisement or content is displayed, while sent refers to the number of messages or communications dispatched. Clicks represent the instances where users interact with the content by clicking on it, and GRP quantifies the exposure level of an advertisement in a given media outlet. Additionally, it includes two economic factors, Vehicle Miles Travelled (VMT) and oil price, providing insight into the market dynamics and potential impact on sales or business operations.

3.2.1 Dependent variable: the number of units serviced

Figure 3.1 shows the progression of unit service volumes, presenting trends, seasonal fluctuations, and residuals. This data set originates from historical records and employs a Python package to assess potential seasonality. Aggregating data from multiple cities, the total number of units offers a company-wide performance overview.

The trend component represents the long-term movement or direction in the data, disregarding short-term fluctuations. In Figure 3.1, the upward trend indicates a continuous growth in the *Unit Serviced* over time.

By eye-balling the seasonal component in Figure 3.1, we see a recurring pattern at regular intervals. The pattern manifests annually, with declines typically observed around May and July, and a spike in March.

The residuals graph shows unexplained variability in the data after removing the trend and seasonal components. A recurring annual cycle is followed by a horizontal line at zero, succeeded by a anti-symmetry pattern; a repeating pattern of values that are mirrored around the middle of the line. This suggests that the decomposition process failed to capture all inherent patterns adequately. Such inadequacy may arise from additional components, variables, or patterns not considered in this analysis.

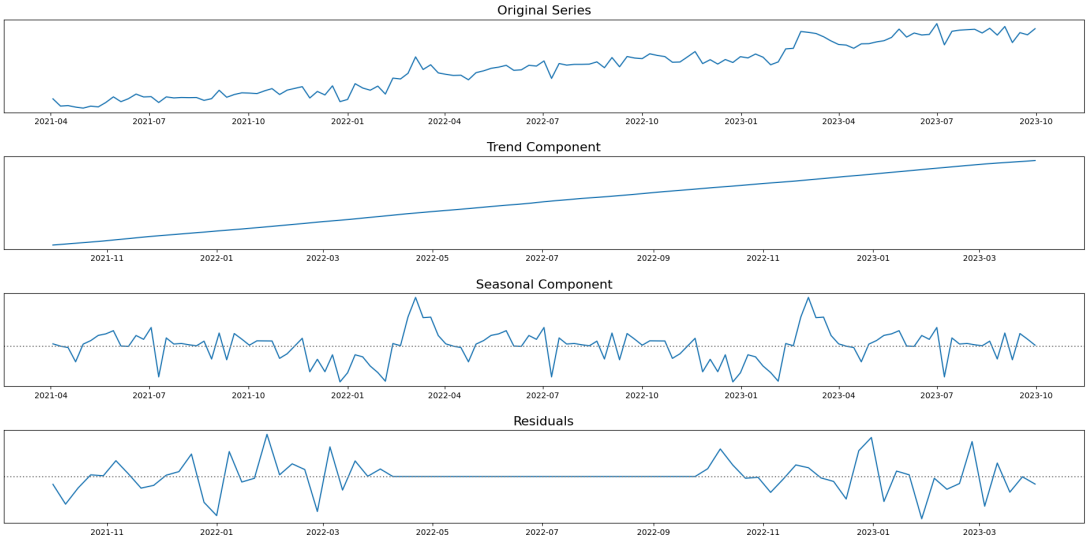
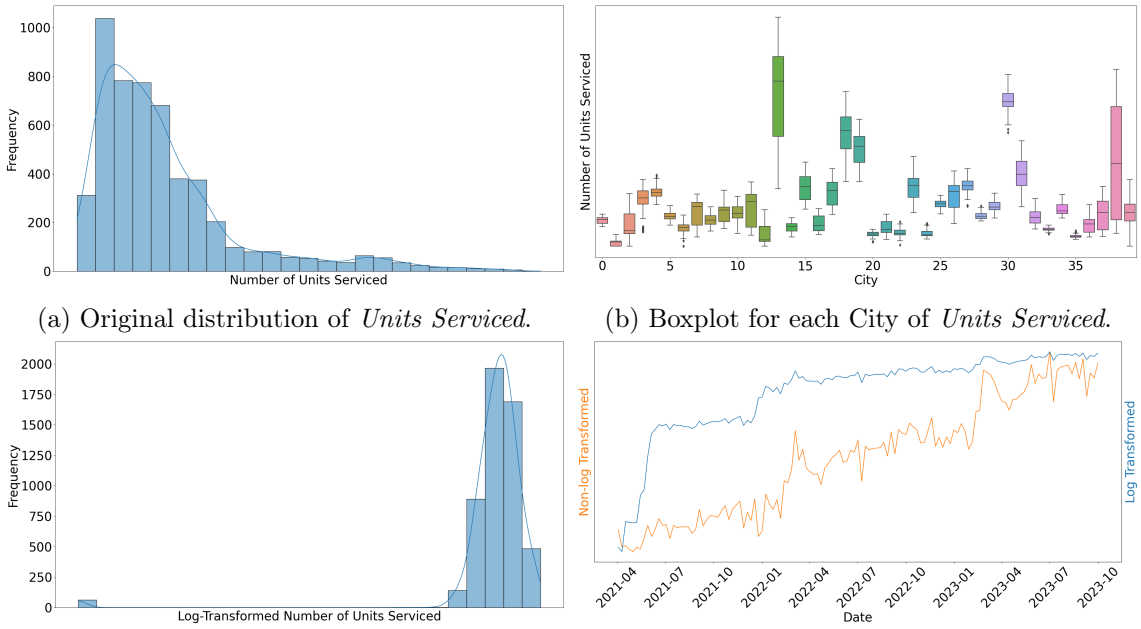


Figure 3.1: The plot illustrates the progression of *Unit Serviced* over time, including the original series, trend component, seasonal fluctuations, and residuals, with the dotted line representing the zero line. Due to confidentiality issues, the scales of the graphs are removed.

Logarithmic Transformation

We examine the characteristics of *Units Serviced*, particularly regarding variation and distributions across different cities. Figure 3.2 shows these insights.



(a) Original distribution of *Units Serviced*.

(b) Boxplot for each City of *Units Serviced*.

(c) Log-Transformed distribution of *Units Serviced*.

(d) Comparison of Original (orange) and Log-Transformed (blue) values.

Figure 3.2: The variable *Unit Serviced* with the original distribution (a), the distribution per city visualised in box plots (b), the Log-Transformed distribution (c), and a comparison between the original and Log-Transformed values (d). The scale of this figure is removed due to confidentiality issues.

Figure 3.2a shows a right skewed distribution of our data, with a high frequency on the left and a tail on the right. Figure 3.2b illustrates the city-wise boxplots, revealing varying heights and medians, indicating differences between cities. Additionally, the skewness of the boxplots differs among cities, with some positively skewed (the whisker is shorter on the lower end of the box) and others negatively skewed (the whisker is shorter on the upper end of the box), along with varying outliers. When comparing interquartile ranges, we observe a greater distribution of data for expanded boxes. As some statistical models assume normality, transforming the data could enhance model reliability.

Figure 3.2c presents the transformed data, showing a more symmetric or normal distribution with one unexplained outlier. Figure 3.2d compares the non-log transformed and log-transformed variables, each with its own axis. Although we have transformed the dependent variable, confirming trend, seasonality, and residuals remains essential. Upon examination, we find that the figure exhibits the same patterns as the original series, so we exclude it from further analysis. Given the variability in our city variables and the skewness of our original distribution, we choose to apply a Log-Transformation to our data to attain a more normal distribution.

Stationarity assumption

After identifying seasonality, trend, and residual patterns in our dependent variable, *Unit Serviced*, we examine the stationarity due to the trend in the data. Since we have cross-sectional data, we apply the panel data stationarity check with the Levin-Lin-Chu panel unit root test. For testing the unit root hypothesis, the test considers pooling cross-section time series data, where the intercept and trend coefficients are allowed to vary across individuals (Levin et al., 2002). We consider the following model:

$$\Delta y_{i,t} = \alpha_{0,t} + \alpha_{1,i}t + \sum_{l=1}^{L} \delta_i y_{i,t-l} + \zeta_{i,t}$$

In this formula, t represents the time, where $\alpha_{1,i}t$ allows each individual unit in the panel data to have its own linear trend over time, L is the lag order. The error process ζ follows a stationary Autoregressive Moving-Average (ARMA) process for each individual and is distributed independently across individuals. The hypothesis is defined as:

H_0: $\delta_i = 0$ and $\alpha_{1,i} = 0$ for all i

H_1: $\delta_i < 0$ and $\alpha_{1,i} \in \mathbb{R}$ for all i

Here, the null hypothesis states that the time series contains a unit root, indicating non-stationarity, the alternative hypothesis is that the time series is stationary according to the test; it does not contain a unit root. The statistic is computed as: $\Lambda_{LLC} = \frac{S^2_{\hat{\gamma}}}{\hat{\sigma}^2}$. $S^2_{\hat{\gamma}}$ is the variance of the coefficient estimate for the lagged series and $\hat{\sigma}^2$ is the estimated residual variance from the regression.

We implement this method in RStudio, since Python does not support this test. We find a $p - value < 2.2 \times 10^{-16}$, which is very small. This suggests strong evidence against the null hypothesis. Since the p-value is less than the significance level (0.05), we reject the null hypothesis in favour of the alternative hypothesis. According to this test, the data set does not contain a unit root.

Furthermore, differencing in marketing forecasts changes the original scale and interpretation of the data. Maintaining interpretability is crucial for understanding the

impact of marketing campaigns, ad spending, and other factors. Differencing leads to losing valuable information, especially in marketing data, where trends and seasonality provide important insights into consumer behaviour and market trends. Additionally, marketing variables often involve complex relationships and dynamics, and differencing may add complexity, particularly during the implementation of marketing transformations.

3.2.2 Dummy variables

The data set includes dummy variables for each month, holidays, and specific weather events such as blizzards, extreme wind, floods, hail, hurricanes, tornadoes, and tropical storms (levels 1 and 2). These variables capture the occurrence of these weather phenomena and their potential impact on the business operations, providing insight into how natural events might affect sales patterns or operational efficiency.

Dummy variables are introduced for seasonality, incorporating these variables helps in the analysis to see how various weather events correlate with fluctuations in sales or operational changes within the business. Seasonality appears as regular, periodic changes in the series mean, commonly aligning with the clock and calendar, involving repetitions over a day, week, or year (Gan et al., 2014).

3.2.3 Characteristics

The data set consists of four types of data: object (2), integer (52), datetime (1), and floating (117). We factorise object data, representing regions and cities, and focus, as mentioned earlier, on one specific region. In addition, marketing variables, measured in impressions, clicks, sent, and GRP, are in the data set.

There are 13 channels, each with its campaigns and platforms. For instance, the *Video* channel contains information about campaigns advertised on YouTube and The Trade Desk (TTD), including promo, non-promo, and other ads. Two other channels, *OOH* (out-of-home advertising) and *Search*, represent billboards, wallscapes, posters, and the Google search engine. Google, being a specific case, models two metrics: impressions and clicks. To ensure consistency and calculation, only one metric should be included. Considering Google's purpose of providing information through search, we exclude the impressions metric and include the number of clicks.

Further investigation reveals essential relationships among variables (see Appendix B). For example, the relationship between 'HoursActual', 'HoursOT' (overtime), and 'HoursRegular': $HoursActual = HoursOT + HoursRegular$. Figure 3.3 shows these variables, with actual values on the y-axis removed due to confidentiality issues.

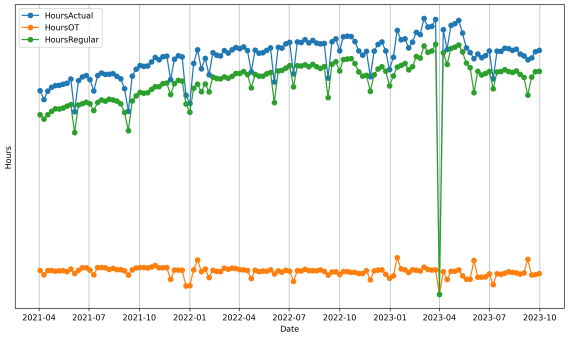


Figure 3.3: Aggregated hourly data, with 'HoursActual', 'HoursOT', and 'HoursRegular'.

3.3 Economic factors

This section defines the economic factors extracted from variables acquired from online sources. We identify and integrate these variables into the data set to facilitate the prediction of economic factors. The factors included in this data set are VMTAvg2, VMTAvg5, VMT_NSA_State and oil price. VMT reflects the traffic near retailers, influencing sales, while oil price can influence transportation costs, energy prices and overall inflation levels. In our investigation, we omit VMTAvg2 and VMTAvg5 due to their manipulated nature.

3.3.1 Potential factors

Araujo & Gaglianone (2023) include inflation, unemployment, and interest rates in their research. These economic indicators have broad relevance and could significantly impact the retail service industry. Additionally, factors such as population income (higher incomes potentially correlating with increased use of retailer services) and personal consumption expenditure might be important. These elements could affect both industry growth and profitability.

Economic conditions directly affect the retailer service industry; during economic downturns, individuals may cut spending on luxuries, affecting revenue. Interest rates directly influence consumer spending capacity and borrowing potential. Higher interest rates could diminish expenditure on services, while lower interest rates might increase disposable income, benefiting the services. Inflation rates affect the cost structure, elevated inflation rates tend to raise material and labour costs, impacting profit margins.

3.3.2 Implementation

Incorporating economic factors into our data set requires some considerations. First, we need to determine whether the data represent percentages or actual values. If we are working with monthly data in actual values, assuming uniformity across all cities may not be appropriate. In such cases, we should consider using percentage values for each city that can be uniformly applied across the data set. We obtain our data from the Federal Reserve Bank of St. Louis (FRED), an online database featuring economic time-series data from various sources.

Due to time constraints, we focus on implementing two additional economic factors to our data set: Personal Consumption Expenditures (PCE) and the unemployment rate.

Personal Consumption Expenditures (PCE)

PCE tracks spending on goods and services in the U.S. economy (Bureau of Economic Analysis (BEA), 2023), accounting about two-thirds of domestic final spending and driving future economic growth. It indicates the portion of household income spent on current consumption versus saved for future use. We implement the percentage change per month obtained from U.S. Bureau of Economic Analysis (2024a) to accurately reflect city-specific variations.

Unemployment rate

This metric includes both employed and unemployed persons, as well as those temporarily absent from work. We consider the percentage unemployment rate per county, assuming

uniformity within each city's county. We retrieve the data from U.S. Bureau of Labor Statistics (2024b).

3.3.3 Interpolation

Section 3.3 mentions the inclusion of two economic factors in the data set; the oil price and VMT spanning multiple years. In cases where data is available quarterly and we possess weekly data, we apply interpolation. Interpolation involves generating new data points within the range of a discrete set of known data points. In this study, our focus is on the linear interpolation method, represented by the formula:

$$y = y_t + (y_{t+1} - y_t)\frac{x - x_t}{x_{t+1} - x_t}$$

Here, y represents the interpolated value of the dependent variable x, x_t and x_{t+1} are the neighbouring values ($x_t \leq x < x_{t+1}$), y_t and y_{t+1} are the variable values at x_t and x_{t+1}. In our data set, VMT is reported quarterly, and all monthly values are identical. To establish a continuous variable, we apply interpolation. Figure 3.4 illustrates the principle of VMT interpolation from January 2022 to November 2022. The orange line denotes the original values, while the blue line represents the interpolated values. While the figure summarises values from all cities for clarity, it is important to note that our data set includes diverse cities with distinct values. Consequently, we determine that all values per city undergo interpolation and be replaced in the data set with the city-specific interpolated values. Appendix I.4 includes this pseudocode.

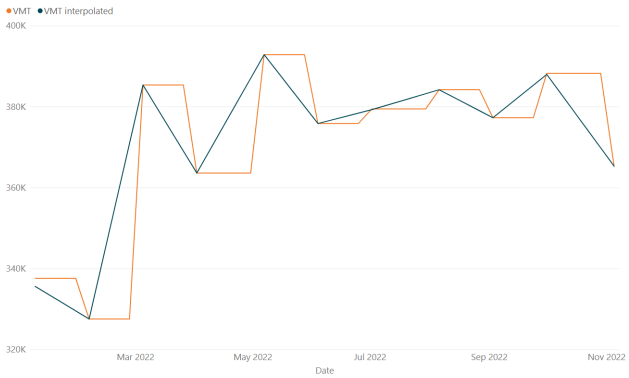


Figure 3.4: Aggregated interpolated VMT vs the aggregated original VMT values from January 2022 to November 2022.

3.4 Data Preprocessing

This section outlines the preprocessing steps required before applying any algorithm to the data. The received data set is complete, without any missing values. Although there are duplicate values in the economic factors, these have been addressed by interpolation, as previously explained in Section 3.3.3.

Furthermore, Appendix B documents all variable dependencies and removal of duplicated variables. Regarding the dummy variables representing months, each month has been included as a dummy within the data set. However, it is important to note that, for multicollinearity concerns, only $(k-1)$ dummies of k groups should be used for modelling, ensuring a unique estimation of the model's coefficients. The excluded category serves as

a reference point for comparison with other categories. We incorporate dummies after the variable selection process to mitigate seasonality trends within our model.

3.4.1 Outliers

Figure 3.3 shows Total Hours, Regular Hours and Overtime Hours, with a notable drop that needs further inspection. Figure 3.5 focusses on the aggregated data of 'HoursRegular'. Given the data set's size of 39 cities, showing all would compromise visual clarity.

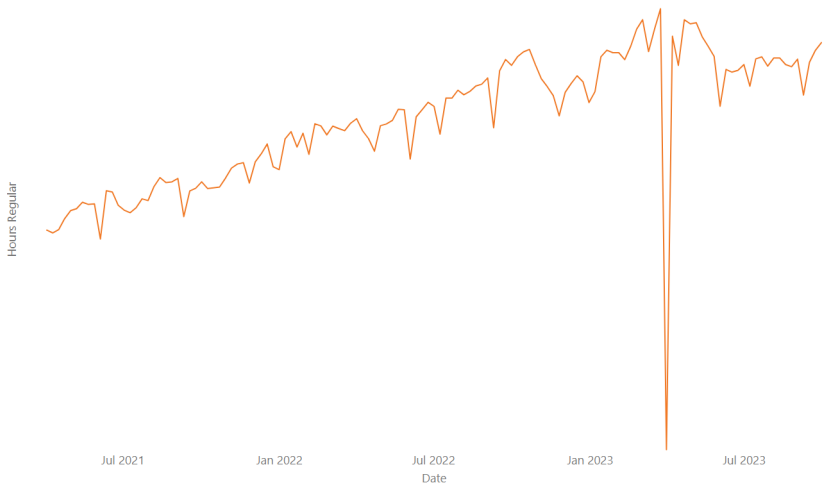


Figure 3.5: This figure illustrates the aggregated Regular Hours for all cities. Due to confidentiality constraints, we are unable to disclose the values. This figure is based on the raw data.

Figure 3.5 illustrates the aggregated regular working hours across all cities, with an identified outlier. On April Third, all cities experienced a drop to zero hours, possibly linked to Good Friday. However, this explanation is insufficient, as it cannot explain why all employees had zero hours throughout the entire week. Thus, we classify this point as an outlier — an anomalous data point inconsistent with the rest.

To address this anomaly, we consider different imputation methods, such as mean/median imputation, interpolation, statistical models like KNN, or custom imputation. Investigation reveals that the drop results from a data processing error in the raw data. To maintain consistency in our research, especially considering the common use of interpolation for economic factors, we decide to apply interpolation exclusively to these values on these dates for this variable across all cities.

3.4.2 Multicollinearity and Stationary

In this data set, we encounter dependent variables that ideally should be independent. For example, consider the 'Hours' variable discussed earlier. Retaining all three variables, 'HoursOT', 'HoursRegular', and 'HoursActual', leads to issues with multicollinearity in certain modelling techniques, as these variables are linearly dependent. Multicollinearity occurs when predictor variables in a regression model are highly correlated, potentially destabilising the model's coefficients and reducing its predictive power.

Variable selection methods, such as stepwise selection, LASSO, or feature importance analysis, can help identify which variables contribute the most to our model. Therefore, we decide to drop all total values, as described in Appendix B, and work with all remaining

variables, as they may contain unique information that the totals do not capture. In total, we exclude 26 variables. Appendix C displays the remaining correlations between the marketing variables to verify the absence of collinearity.

In our research, we have 'HoursActual' which is derived from 'HoursOT' and 'HoursRegular'. As we still observe a high correlation between 'HoursOT' and 'HoursRegular', we drop these two variables and retain 'HoursActual'. After conducting a stationarity check on our data set, we find that all variables do not have a unit root. Appendix D shows the marketing variables that are included in the algorithm.

3.4.3 Marketing variables

According to Joosten et al. (2023), "advertising has both gradual long term effects through the height and shape of the market potential functions, but also sharp immediate effects on demand". To address the effects of advertising carryover, we use AdStock as a metric, which measures cumulative advertising expenditures over time (Bayer et al., 2020). AdStock considers both the immediate impact of advertising on brand choice and its long-term effect on brand awareness, accumulating influence over successive periods. Mathematically, we define AdStock as (Joseph, 2006):

$$X_{i,t} = A_{i,t} + \lambda_i * X_{i,t-1},\ t = 1, \ldots, n$$

Here, λ_i denotes the decay of the variable i, $X_{i,t}$ represents the AdStock at time t for the variable i, and $A_{i,t}$ denotes the raw advertising variable i at time t. This model captures the carryover impact of advertising on consumer awareness and sales volume. The delay effect, or lag component, recognises the prolonged impact after an ad ceases to air (Beltran-Royo et al., 2016), arising from consumers who continue to influence purchases.

Variables transformed by AdStock can be integrated as predictors in various modelling approaches, such as regression models (e.g., linear regression), used as exogenous inputs in time series models (e.g., ARIMA), or incorporated as features in machine learning models (e.g., XGBoost). Our aim in this algorithm is to include AdStock, and address lag for the chosen variables. Given the integration of multiple models, we employ diverse techniques to accurately capture and apply the marketing effects.

3.5 Conclusion

We acquire the data set from ScanmarQED, representing a major U.S. retail service. This data set forms the foundation for our research, containing 172 variables and 5102 entries spanning from April 3, 2021, to September 30, 2023. To ensure confidentiality, we apply strict measures, refraining from disclosing actual values and transposing them before any publication.

Focussing on one of the three regions of the sales pattern, we identify three KPIs. Due to distortions in both net and gross sales, and the reliability of *Units Serviced*, we identify the latter as our dependent marketing KPI for modelling, omitting the other two. The data set includes relevant marketing variables, which we can use to forecast the marketing KPI.

Economic factors such as VMT and oil prices provide insight into market dynamics that influence sales and operational efficiency. Other potential economic factors include inflation, unemployment, interest rates, population income, and consumer spending.

However, due to time limitations, we only implement two additional economic factors, PCE and unemployment rate. After implementation and transformation, we have a total of four different economic factors within our data set. We use interpolation for economic factors with discrete data points to create continuous data and manage outliers in marketing variables.

The limitations of this data set revolve around complexities in data transformation and confidentiality assurance. We have made assumptions, such as stationarity and log transformation, based on *Units Serviced*, for analysing marketing impacts, while economic factors assist in understanding broader trends and their influence on business operations.

The data set's relevance lies in its integration of *Units Serviced* and marketing variables (e.g., TV, Google, Instagram), measured through impressions, clicks, sent, or GRPs. Our objective is to pinpoint the most influential marketing techniques regarding *Units Serviced*.

Chapter 4

Methodology

"Future forecasting is all about testing strategies - it's like a wind tunnel."

— Jamais Cascio

In this chapter, we explain our research methodology to develop an AutoForecaster. We aim to answer the following RQ:

> *How does the development and implementation of a marketing AutoForecaster contribute to enhancing marketing strategies, and what quantitative insights, including performance metrics, are obtained?*

We must take several steps to create the AutoForecaster, initially, we detail the research design, including the creation of a process chart outlining the algorithm's structure and the necessary programming steps. Additionally, we explore diverse forecasting methods, clarifying their applications and mathematical foundations. Furthermore, we explain the intricate aspects of the model, integrating insights derived from data analysis and the AutoForecaster's evaluation.

Finally, we discuss the experimental design and interpretation of results. The quote by Jamais Cascio, reflects our approach as the AutoForecaster tests diverse strategies to discern the most effective approaches and outcomes.

4.1 Process chart of the AutoForecaster

Our aim is to develop an AutoForecaster that automatically identifies the best prediction model with marketing variables. As mentioned prior, our dependent variable is *Units Serviced*. By integrating economic factors, our objective is to enhance the accuracy of our models when there is a correlation between the economic factor and the dependent variable. We implement variable selection techniques to identify the variables affecting our dependent variable the most. Should an economic factor belong to the top variables, we conduct forecasts for these economic factors. Additionally, we incorporate lagged values of these variables to determine the optimal prediction method and forecast. Subsequently, we integrate these forecast values into the AutoForecaster alongside other selected variables. With these values, we develop a forecaster based on the most effective prediction method.

Appendix E shows the detailed process of determining the optimal forecasting model for both the dependent variable and the economic factors. This figure explains the entire process of generating a forecast. For clarity, the process starts and ends at the green boxes.

Initially, we load data and select one specific market. Simultaneously, economic factors are identified and pre-processed to align with the original data frame format. Subsequently, we merge both datasets and predetermined variables are interpolated. The dependent variable undergoes logarithmic scaling, followed by variable selection procedures. Data are examined for multicollinearity and stationarity, and split into the marketing variables and other variables. AdStock transformation is applied to the marketing variables, which are then combined with the other variables. Next, the dependent variable is predicted to identify the optimal model, and the best model is selected. We add dummy variables in a later stage, to find out if adding these dummy variables increases the accuracy of the model. If economic factors be present throughout the variable selection process, we forecast these factors. Incorporating all forecast factors and predicted dummy variables, we apply an AR model to the marketing variables, and the dependent variable is forecast accordingly.

4.2 Cross-validation

We use cross-validation with different grid searches during the algorithm process to determine the parameters of different prediction methods. James et al. (2013) describes cross-validation as a resampling method that uses different portions of the data to test and train a model on different iterations to choose the penalty parameter based on this predictive performance. The data set is repeatedly divided into a training and testing set to find the fit of the model and to assess the predictive performance. For our model we use `GridSearchCV` to find the hyperparameters of different models, this method performs a K-fold cross-validation. The data set is split into K portions, each fold is used once as the testing sample, and the remainder is used for training the model for some value of λ. Figure 4.1 shows the idea behind cross-validation.

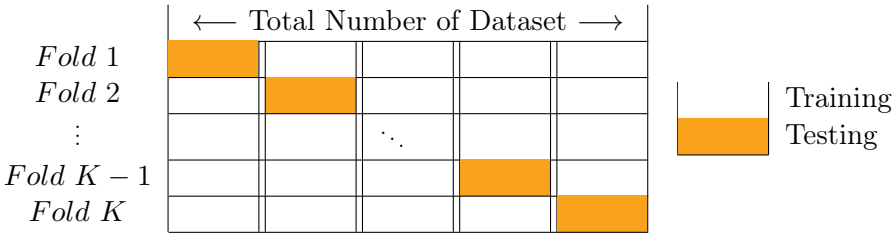


Figure 4.1: Cross-validation diagram depicting K-fold validation where each fold serves alternately as the validation set while the remaining folds are used for training. The white sections represent training data, and the orange sections represent validation data.

After performing cross-validation for each parameter combination, the `GridSeachCV` selects the combination that results in the best performance metric. With the best parameters, we estimate the model. Cross-validation is computationally intensive as it repeatedly estimates the model and checks the performance across different folds and values for λ. However, we apply this method to find the best parameters in an automatic way.

4.3 Economic factor - Forecast models

We investigate several prediction methods to identify the most effective for the prediction of economic factors. We provide explanations for the following models: AR(2), XGBoost, SVR, OLS, KNN, Lasso, and RR. We do not implement independent variables to the forecast of our economic factor, since our main goal is to forecast the marketing KPI. We forecast economic factors based on their historic values.

4.3.1 AutoRegressive model (AR)

AR(2) is an autoregressive model, denoted as AR(L) in a general context (James et al., 2013). T represents the total number of time points in the data, and L is the maximum lag. Fitting a AR regression of y for each city c at time t results in the formula:

$$y_{c,t} = \beta_0 + \beta_1 y_{c,t-1} + \beta_2 y_{c,t-2} + \cdots + \beta_L y_{c,t-L} + \varepsilon_t,$$

This is called an order-L autoregressive model (AR(L)). We formulate the AR(2) model for economic factors a bit differently, the response vector $y_{c,t}$ is replaced with the $z_{j,c,t}$, indicating the economic factor j. We define the model as: $z_{j,c,t} = \gamma_{j,0} + \gamma_{j,1} z_{j,t-1} + \gamma_{j,2} z_{j,c,t-2} + \varepsilon_{j,t}$, where γ_j represents the parameters for each economic factor j. We define this as $\gamma_j = \left[\gamma_{j,0}, \gamma_{j,1}, \gamma_{j,2}\right]$ and $\tilde{Z}^T_{j,c,t} = \left[1, z_{j,c,t-1}, z_{j,c,t-2}\right]$ where we want to obtain:

$$\hat{\gamma}_j = \arg\min_{\gamma_j} \left|\left| z_{j,c,t} - \gamma_j \tilde{Z}_{j,c,t} \right|\right|_2^2$$

The $||x||_2$ notation is the L2 norm, defined as $||x||_2 = \sqrt{x_1^2 + x_2^2, \ldots, x_n^2}$. Squaring this norm results in $||x||_2^2 = (x_1^2 + x_2^2), \ldots, x_n^2$.

4.3.2 Ordinary Least Squares (OLS)

The least squares method selects parameters to minimise the Residual Sum of Squares (RSS). In simple linear regression, this involves minimising the RSS equation. We consider L as the number of lags for predicting $z_{j,c,t}$, $t \geq 2$, j represents a specific economic factor, and c represents the city at time t. We use $\tilde{Z}^T_{j,c,t} = [1, z_{j,c,t-1}, z_{j,c,t-2}, \ldots, z_{j,c,t-L}]$ as our exogenous variable, and define $\gamma_{j,c} = [\gamma_{j,c,0}, \gamma_{j,c,t-1}, \gamma_{j,c,t-2}, \ldots, \gamma_{j,c,t-L}]$.

$$z_{j,c,t} = \gamma_{j,c} \tilde{Z}_{j,c,t} + \varepsilon_{j,c,t} \tag{4.1}$$

$$\hat{\gamma}_{j,c} = \arg\min_{\gamma_{j,c}} \left|\left| z_{j,c,t} - \gamma_{j,c} \tilde{Z}_{j,c,t} \right|\right|_2^2 \tag{4.2}$$

In linear regression, we assume that the errors are independently and identically distributed (i.i.d.), indicating homoscedasticity. This assumption implies that the variance of residuals remains constant across estimated response variable values, to get the best and most efficient OLS estimator.

In our analysis, we set the categorical variable, *City*, as the index in our data frame. OLS automatically converts the categorical variable into binary dummy variables, capturing its impact on the dependent variable. The coefficients of these dummy variables show how much the dependent variable typically changes compared to a reference category. This helps our regression model consider different categories, making the results easier to understand.

4.3.3 Least Absolute Shrinkage and Selection Operator(Lasso)

For Lasso, we consider the same formula as the OLS predictor (Section 4.3.2), with the same assumption of $\tilde{Z}_{j,c,t}$, but without a city specific coefficient, $\gamma_j = [\gamma_{j,0}, \gamma_{j,t-1}, \gamma_{j,t-2}, \ldots, \gamma_{j,t-L}]$ (Section 4.3.2). However, we modify it to a minimisation problem. The objective function is given as (Chan-Lau, 2017):

$$\min_{\gamma_j} \left|\left| z_{j,c,t} - \gamma_j \tilde{Z}_{j,c,t} \right|\right|_2^2 \tag{4.3}$$

$$s.t.\ ||\gamma_{j,1:}||_1 \leq \phi \tag{4.4}$$

ϕ represents the size constraint on the parameters (Le, 2020). λ is a predefined free parameter that determines the degree of regularisation, and $||\gamma_{j,1:}||_1$ represents the L_1 norm of vector γ_j excluding the constant term $\gamma_{j,0}$. Lagrangian results in:

$$\min_{\gamma_j,\lambda} \left|\left| z_{j,c,t} - \gamma_j \tilde{Z}_{j,c,t} \right|\right|_2^2 + \lambda \left(||\gamma_{j,1:}||_1 - \phi\right) \tag{4.5}$$

We limit each individual L_1 norm separately for each economic factor. We define ϕ in our algorithm using cross-validation, λ is the penalty term.

- $\lambda = 0$: same coefficients as OLS regression.
- $\lambda = \infty$: all coefficients tend to go to zero while still satisfying the constraint, as the focus is on minimising the L_1 norm of the coefficients.
- $0 < \lambda < \infty$: coefficients are between 0 and that of the least-squares regression.

4.3.4 Ridge Regression (RR)

For RR, we consider the same formula as the OLS predictor (Section 4.3.2), with the same assumption of $\tilde{Z}_{j,c,t}$, but without a city specific coefficient, $\gamma_j = [\gamma_{j,0}, \gamma_{j,t-1}, \gamma_{j,t-2}, \ldots, \gamma_{j,t-L}]$. RR uses the L2 norm to analyse any data, it is a tuning method used to analyse data that contains multicollinearity (Chan-Lau, 2017). The objective function is:

$$\min_{\gamma_j} \left|\left| z_{j,c,t} - \gamma_j \tilde{Z}_{j,c,t} \right|\right|_2^2 \tag{4.6}$$

$$s.t.\ ||\gamma_{j,1:}||_2 \leq \phi \tag{4.7}$$

Lagrangian results in:

$$\min_{\gamma_j,\lambda} \left|\left| z_{j,c,t} - \gamma_j \tilde{Z}_{j,c,t} \right|\right|_2^2 + \lambda \left(||\gamma_{j,1:}||_2 - \phi\right) \tag{4.8}$$

here λ is the penalty term, just as in Lasso, and $||\gamma_{j,1:}||_2$ represents the L_2 norm of vector γ_j, excluding the constant term $\gamma_{j,0}$.

4.3.5 EXtreme Gradient Boosting (XGBoost)

XGBoost is a supervised algorithm applied in various fields (Yang et al., 2023). Implemented as gradient boosted decision trees, it combines multiple weak learners into a predictor. Through iterative improvement, this process improves predictive accuracy while optimising computational efficiency (Cohen & Aiche, 2023).

In practice, XGBoost creates prediction models in the form of decision trees, improving

their predictions through iterations with the aim of minimising a loss function. This optimisation process uses gradient descent, where new trees are gradually updated within the ensemble by leveraging the residuals of preceding trees to achieve more accurate predictions.

The objective function for the j-th dependent economic factor $z_{j,c}$ of XGBoost, $L_{j,c}^{(k)}$, integrates both the training loss $\sum_{t=1}^{T} l(z_{j,c,t}, \hat{z}_{j,c,t})$ and a regularisation term $\sum_{k=1}^{K} \Omega(f_{k,j,c})$. $L_{j,c}^{(k)}$ represents the objective function at the k-th stage of the boosting process of the j-th dependent variable for city c.

Here, $l(z_{j,c,t}, \hat{z}_{j,c,t})$ measures the difference between the actual $(z_{j,c,t})$ and predicted $(\hat{z}_{j,c,t})$ values for each sample $\tilde{Z}_{j,c,t}$. $\tilde{Z}_{j,c,t} = \left[z_{j,c,t-1}, z_{j,c,t-2}, \ldots, z_{j,c,t-L}\right]$ represents our lagged variables of $z_{j,c,t}$, with maximum lag L. The regularisation term $\Omega(f_{k,j,c})$ penalises the complexity of trees, mitigating overfitting and improving the model's generalisation ability. So s is the number of samples and k is the number of trees in the ensemble.

$$\min_{f_{k,j,c}} \left\{ \sum_{s=1}^{S} l(z_{j,c,t}, \hat{z}_{j,c,t}) + \sum_{k=1}^{K} \Omega(f_{k,j,c}) \right\} \tag{4.9}$$

Ultimately, the output of XGBoost is derived from the cumulative sum of the output values of all trees, represented as $\hat{z}_{j,c,t} = \sum_{k=1}^{K} f_{k,j,c}(\tilde{Z}_{j,c,t})$, where $f_{k,j,c}$ represents the individual trees within the model for city c. The purpose of this implementation of XGBoost's optimisation process is to effectively handle complex data sets while maintaining high predictive accuracy.

4.3.6 Super Vector Regression (SVR)

The primary goal of SVR is to find a function that accurately maps input features to the target variable. To find predicted values, this model uses support vectors, a subset of training data points. These support vectors help to define a hyperplane that maximises the margin around these predicted values. This method is particularly useful for handling complex high-dimensional data and excels at solving nonlinear regression problems.

SVR consists of two main components: the allocation of support vectors to address classification challenges and the regression of support vectors (Sarwar et al., 2023). The SVR optimisation objective is to minimise a combination of the squared weight norm (w) and a regularisation term (R) while satisfying certain constraints. For our economic factor model, we define $\tilde{Z}_{j,c,t}^{T} = [z_{j,c,t-1}, z_{j,c,t-2}, \ldots, z_{j,c,t-L}]$, where L are the lags to predict $z_{j,c,t}$, and $\gamma_j = \left[\gamma_{j,0}, \gamma_{j,1}, \ldots, \gamma_{j,t-L}\right]$. We define our predicted values $\hat{z}_{j,c,t} = \gamma_j \tilde{Z}_{j,c,t} + b = w^T \tilde{Z}_{j,c,t} + b$, where j represents a specific economic factor and c represents the city at time t.

$$\min_{w,b,\xi_{j,c,t},\zeta_{j,c,t}} \left\{ \frac{1}{2}||w||^2 + R \sum_{t=1}^{T} (\xi_{j,c,t} + \zeta_{j,c,t}) \right\} \tag{4.10}$$

$$s.t.: \tag{4.11}$$

$$\forall t : (w^T \tilde{Z}_{j,c,t} + b) - z_{j,c,t} \leq \varepsilon_t + \xi_{j,c,t} \tag{4.12}$$

$$\forall t : z_{j,c,t} - (w^T \tilde{Z}_{j,c,t} + b) \leq \varepsilon_t + \zeta_{j,c,t} \tag{4.13}$$

$$\forall t : \xi_{j,c,t},\ \zeta_{j,c,t} \geq 0 \tag{4.14}$$

These constraints ensure that the predicted values $(\hat{z}_{j,c,t})$ lie within a specified margin (ε_t) of the true target values $(z_{j,c,t})$, where t represents the time points. To deal with

the constraints effectively we introduce the Lagrangian multipliers ($\alpha_{j,c,t}$ and $\beta_{j,c,t}$) and include the Karush-Kuhn-Tucker (KKT) conditions (Balasundaram et al., 2014). Appendix G present the calculations of the Lagrangian multipliers.

$$\mathbf{w} = \sum_{t=1}^{T} (\alpha_{j,c,t} - \beta_{j,c,t}) \tilde{Z}_{j,c,t} \tag{4.15}$$

$$\alpha_{j,c,t}(\varepsilon_t + \xi_{j,c,t}) = 0 \tag{4.16}$$

$$\beta_{j,c,t}(\varepsilon_t + \zeta_{j,c,t}) = 0 \tag{4.17}$$

So, SVR is a regression technique that finds an optimal function for mapping input features to the target variable, making it well-suited for various regression tasks.

4.3.7 K-Nearest Neighbour (KNN)

KNN is a known algorithm used for classification and regression (Rigopoulos, 2022). For time series forecasting, the application of using KNN are limited; however, some papers show an increased accuracy compared to other models. We apply the forecasting technique based on the KNN regression method, and this forecasting method can be applied for macroeconomic variable forecasting as we are using lagged variables as input for forecasting. The nearest neighbour approach is a well-known technique for local approximation and is based on the concept that the evolution of the current state will be similar to the nearest neighbour evolution. So, the method examines the nearest neighbours in the data set, given a certain dimension that defines the pattern in the past to search. For our KNN model, we use the recursive strategy to model multi step forecasting, which trains a one-step model and then uses it recursively to return a multi-step forecast. Rigopoulos (2022) proposes a KNN time series forecasting process that we use as a foundation in our algorithm. We define a feature vector representation for each time point $z_{j,c,t}$ in city c as:

$$z_{c,t} = [z_{1,c,t}, z_{2,c,t}, \ldots, z_{m,c,t}] \tag{4.18}$$

where:

- $z_{j,c,t}$ represents the j-th economic factor for the t-th time point in city c,
- m is the number of economic factors.

The Euclidean distance $d(z_{j,c,t}, z_{c,t})$ between two instances $z_{j,c,t}$ and $z_{c,t}$ is calculated as:

$$d(z_{j,c,t}, z_{c,t}) = ||z_{j,c,t} - z_{c,t}||_2 \tag{4.19}$$

We can explain this with a two-dimensional feature space ($z_{j,c,1}$ and $z_{j,c,2}$) in Figure 4.2. Each point in this space represents these two economic factor observations. By drawing a straight line between the points and using Euclidean distance, we define the distance as $d(z_{j,c,t}, z_{c,t}) = \sqrt{(z_{1,c,1} - z_{2,c,1})^2 + (z_{1,c,2} - z_{2,c,2})^2}$. This distance measures the similarity or dissimilarity between the two data points, determining the nearest neighbours. Using KNN, we can then apply a regression.

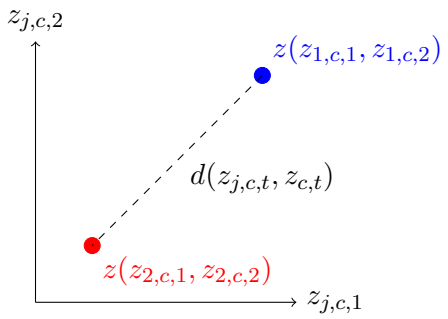


Figure 4.2: Principle of Euclidean distance for a two-dimensional feature space $z_{j,c,1}$ and $z_{j,c,2}$.

The weighted average ($\hat{z}_{j,c,t}$) for forecasting is computed as, where T is the set of T observations :

$$\hat{z}_{j,c,t} = \frac{\sum_{t=1}^{T} w_t \cdot z_{n_{j,c,t}}}{\sum_{d=1}^{T} w_t} \tag{4.20}$$

Here:

- w_t represents the weight assigned to the t-th nearest neighbour,
- $w_i = \frac{1}{d(z_q, z_{n_t})}$, where z_q is the query instance and z_{n_t} is the t-th nearest neighbour,
- $z_{n_{j,c,t,}}$ represents the target value for the t-th nearest neighbour of the j-th economic factor in city c.

4.4 Marketing - Variable selection

We discuss the methods and algorithms to select variables that have an effect on the dependent marketing variable, *Units Serviced*. In Section 2.2.2 we conclude that the best variable selection methods include FSS, BSS, PCA and, RF.

4.4.1 Forward Stepwise Selection (FSS)

FSS gradually incorporates variables into a method based on a model-fit criterion. The process begins with a null model, consisting of an intercept but no predictors (James et al., 2013). Subsequently, we perform p simple linear regressions, adding the variable that results in the lowest RSS to the null model. This procedure repeats, adding the variable with the lowest RSS to the current model at each step. The process concludes when a specified number of variables is added to the model. Appendix F.1 provides the pseudocode for implementing FSS.

4.4.2 Backward Stepwise Selection (BSS)

BSS starts with a pool of all variables as input data for prediction (Barak & Parvini, 2023). It iteratively excludes one variable at a time, evaluating the forecasts of the remaining variables. The variable to exclude in the next step is determined by the model's forecasting performance. This process continues until there is no improvement in forecasting performance or our stopping rule is satisfied, which involves a predetermined number of variables. Appendix F.2 details the pseudocode for BSS.

4.4.3 Principle Component Analysis (PCA)

PCA is a statistical technique used to simplify complex data sets while preserving most of their variability. By transforming the original variables into a new set of variables called principal components, we achieve this. PCA identifies linear combinations of variables along which the data varies the most. When there is a large set of correlated variables, principal components summarises them into representative features, collectively explaining the most variability in the data set. For a data set with n observations and p variables, the PCA process involves the following steps:

1. **Data centering.** Calculate the mean of each variable and subtract it from the original values. Centered variable: $x^*_{i,j} = x_{i,j} - \frac{1}{n}\sum_{i=1}^{n} x_{i,j} = x_{i,j} - \bar{x}_j$
2. **Calculating Covariance Matrix.** Construct the covariance matrix Σ for the centered data, where $X^* = [x^*_{i,j}]$: $\Sigma = \frac{1}{n-1} X^{*T} X^*$ where X^* is the centered data matrix.

3. **Eigenvalue Decomposition.** After obtaining the covariance matrix (Σ), we compute its eigenvalues λ_i and corresponding eigenvectors $\mathbf{v}_i$: $\Sigma\mathbf{v}_i = \lambda_i\mathbf{v}_i$

4. **Selecting Principal Components.** The principal components are formed from the eigenvectors. The first k principal components are chosen based on the highest explained variance or by setting a threshold for the cumulative explained variance.

The percentage of variance explained by each principal component can be calculated using the eigenvalues, allowing for insight into the significance of each component: Percentage variance explained by $\text{PC}_i = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j} \times 100\%$.

PCA allows variable selection by focussing on principal components that capture the most variance, enabling the reduction of dimensionality while retaining essential information. Appendix F.3 provides the pseudocode for implementing PCA.

4.4.4 Random Forest (RF)

RF, a supervised learning model that combines decision trees and the bagging method, involves resampling the training data set through "bootstrap". Each sample, which consists of a random subset of original columns, fits a decision tree. The number of models, columns, and hyperparameters contribute to prediction aggregation. Averaging the output of the trees reduces standard error and variance. For each tree, we calculate the importance of features by looking at how pure the leaves are, nodes or endpoints of decision trees where all the data points belong to the same class or have very similar values for regression tasks, measured by Gini impurity or entropy (Tangirala, 2020). These importance values are then combined so that they add up to 1 altogether.

The statistical framework of Genuer et al. (2010) underpins RF, offering estimators for the Bayes classifier or regression function. Let $L = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ represent a learning rate of n i.i.d. observations of a random vector (X, Y). $X = (X^1, \ldots, X^p)$ includes predictors or explanatory variables ($x \in \mathbb{R}^p$), and $y \in \mathcal{Y}$ denotes a numerical response. For our classification problems, a class t is a mapping $t : \mathbb{R}^p \to \mathcal{Y}$. RF provides estimators for either the Bayes classifier or regression function.

In our data set, most variable importance values are small. To ensure the capture of features with higher importance, we employ a robust feature selection method based on the 80th percentile of importance values. The formula for percentile-based thresholding is given by:

$$Threshold = Percentile(Importance\_Values, 80\%)$$

This approach guarantees the selection of the top 20% contributing features. Appendix F.4 contains the pseudocode of the RF algorithm, which includes an automated process to determine the optimal number of estimators and tree depth using random samples.

4.5 Marketing - Forecast

We explain the forecasting models for the dependent variable, mentioned in Table 2.2. This concerns the models Dynamic Bayesian, ARIMA, RF, SVR, AR, panel model, OLS and XGBoost. Section 4.3 already explains certain models that recur as prediction models in the marketing forecast. We modify these models slightly to create the right format for the marketing forecast to estimate *Units Serviced*. For all models, we start with the training data which includes marketing variables X and economic factors Z. We test all models with our testing data set.

4.5.1 AutoRegressive with eXogenous variables (ARX)

Section 4.3.1 explains the general AR model. For our marketing forecast, we need to adjust this model to include exogenous variables. We define the AR model with exogenous variables for forecasting the marketing dependent variable below, let $y_{c,t}$ denote the time series variable for city c at time t. We define $X_{c,t}^T = \left[1, X_{1,c,t}, X_{2,c,t}, \ldots, X_{n,c,t}\right]$, where n is the number of marketing variables in the model, and $Z_{c,t}^T = \left[Z_{1,c,t}, Z_{2,c,t}, \ldots, Z_{m,c,t}\right]$, where m is the number of economic factors in the model. Furthermore, $\beta = \left[\beta_0, \beta_1, \beta_2, \ldots, \beta_n\right]$ represents the coefficients for the marketing variables, where β_0 is the constant term, and $\gamma = \left[\gamma_1, \gamma_2, \ldots, \gamma_m\right]$ represents the coefficients for the economic factors. $Y_{c,t}^T = \left[y_{c,t-1}, y_{c,t-2}, \ldots, y_{c,t-L}\right]$, where L is the maximum lag and $\phi = \left[\phi_1, \phi_2, \ldots, \phi_L\right]$. Our modified expression is:

$$y_{c,t} = \phi Y_{c,t} + \beta X_{c,t} + \gamma Z_{c,t} + \varepsilon_{c,t}$$

$$\hat{\beta}, \hat{\gamma} = \underset{\beta,\gamma,\phi}{\arg\min} \, ||y_t - \phi Y_{c,t} - \beta X_{c,t} - \gamma Z_{c,t}||_2^2$$

4.5.2 Ordinary Least Squares (OLS)

Section 4.3.2 introduces the OLS equation for the economic factors with lags as exogenous variables. In the marketing forecast, we modify this equation to include both marketing variables and economic factors. Here, the marketing variables and economic factors serve as our exogenous variables, reflecting the influence of marketing efforts and economic conditions on the KPI. We exclude city-specific coefficients, aiming for a more generalised approach. Our dependent variable, $y_{c,t}$ represents *Units Serviced* at time t for city c, $X_{c,t}$ are the different marketing variables at time t for city c, and $Z_{c,t}$ illustrates the economic factors at time t for city c, aligning with the notations in Section 4.5.1.

$$y_{c,t} = \beta X_{c,t} + \gamma Z_{c,t} + \varepsilon_t \tag{4.21}$$

$$\hat{\beta}, \hat{\gamma} = \underset{\beta,\gamma}{\arg\min} \, ||y_{c,t} - \beta X_{c,t} - \gamma Z_{c,t}||_2^2 \tag{4.22}$$

4.5.3 Extreme gradient boosting (XGBoost)

The objective function for our dependent variable, *Units Serviced*, $y_{c,t}$ in XGBoost is $L_c^{(k)}$. Here, $\sum_{t=1}^{T} l(y_{c,t}, \hat{y}_{c,t})$ measures the difference between the actual ($y_{c,t}$) and predicted ($\hat{y}_{c,t}$) values for each sample in city c. $\tilde{X}_{i,c,t} = \left[X_{1,c,t}, X_{2,c,t}, \ldots, X_{n,c,t}, Z_{1,c,t}, Z_{2,c,t}, \ldots, Z_{m,c,t}\right]$ is a matrix containing the marketing variables and economic factors. The regularisation term $\sum_{k=1}^{K} \Omega(f_{k,i,c})$ penalises the complexity of trees, mitigating overfitting and improving the models generalisation ability. So T is the total number of samples and k is the number of trees in the ensemble.

$$\min_{f_{k,i,c}} \left\{ \sum_{t=1}^{T} l(y_{c,t}, \hat{y}_{c,t}) + \sum_{k=1}^{K} \Omega(f_{k,i,c}) \right\} \tag{4.23}$$

$$\hat{y}_{c,t} = \sum_{k=1}^{K} f_k(\tilde{X}_{i,c,t}) \tag{4.24}$$

4.5.4 Super Vector Regression (SVR)

For our marketing forecasting model, we have two independent variables instead of one. Therefore, we extend our SVR model in Section 4.3.6, and use the same notations for $X_{c,t}$, $Z_{c,t}$, β, and γ as in Section 4.5.1. We define $y_{c,t} = \beta X_{c,t} + \gamma Z_{c,t} + b = w^T \tilde{X}_{c,t} + b$, where

$w = \begin{bmatrix} \beta \\ \gamma \end{bmatrix}$ and $\tilde{X}_{c,t} = \begin{bmatrix} X_{c,t} \\ Z_{c,t} \end{bmatrix}$, with marketing variables $X_{c,t}$ and economic factors $Z_{c,t}$ at time t for city c. $y_{c,t}$ is our dependent marketing KPI.

$$\min_{w,b,\xi_{c,t},\zeta_{c,t}} \left\{ \frac{1}{2}||w||^2 + R\sum_{t=1}^{T}(\xi_{c,t} + \zeta_{c,t}) \right\} \tag{4.25}$$

$$s.t.: \tag{4.26}$$

$$\forall t : (w^T \tilde{X}_{c,t} + b) - y_{c,t} \leq \varepsilon_t + \xi_{c,t} \tag{4.27}$$

$$\forall t : y_{c,t} - (w^T \tilde{X}_{c,t} + b) \leq \varepsilon_t + \zeta_{c,t} \tag{4.28}$$

$$\forall t : \xi_{c,t},\ \zeta_{c,t} \geq 0 \tag{4.29}$$

These constraints ensure that the predicted values lie within a specified margin (ε_t) of the true target values ($y_{c,t}$), where t represents the time points. Lagrangian multipliers ($\alpha_{c,t}$ and $\theta_{c,t}$) are introduced to handle these constraints effectively. The calculations of the Lagrangian multipliers are denoted in Appendix H and include the Karush-Kuhn-Tucker (KKT) conditions (Balasundaram et al., 2014).

$$\mathbf{w} = \sum_{t=1}^{T}(\alpha_{c,t} - \theta_{c,t})\tilde{X}_{c,t} \tag{4.30}$$

$$\alpha_{c,t}(\varepsilon_t + \xi_{c,t}) = 0 \tag{4.31}$$

$$\theta_{c,t}(\varepsilon_t + \zeta_{c,t}) = 0 \tag{4.32}$$

So, $\begin{bmatrix} \beta \\ \gamma \end{bmatrix} = \sum_{t=1}^{T}(\alpha_{c,t} - \theta_{c,t}) \begin{bmatrix} X_{c,t} \\ Z_{c,t} \end{bmatrix}$. SVR is a regression technique that finds an optimal function for mapping input features to the target variable, making it well-suited for various regression tasks.

4.5.5 Bayesian

For marketing strategies and forecasting, continuous updates and adaptability are crucial (Tierney et al., 2023). Dynamic Bayesian forecasting uses the Bayesian framework into dynamic models that allow for the inclusion of new data. This method uses a variety of models and techniques that handles time-series data and adapts to changing conditions. It involves various models that incorporate Bayesian principles for dynamic forecasting and update beliefs about parameters or states as new data arrives. The Bayesian approach allows us to set priors for our coefficients.

The key difference between a Bayesian linear regression model and a standard OLS model lies in their approaches to parameter estimation and uncertainty quantification, with Bayesian regression providing posterior distributions for parameters to explicitly quantify uncertainty, whereas OLS regression produces point estimates without explicitly quantifying uncertainty.

We are interested in predicting outcomes y for city c at time t as normally distributed observations with expected value $\mu_{c,t}$, which is a linear function of our marketing variables $X_{i,c,t}$ and economic factors $Z_{j,c,t}$. In the Bayesian approach $\hat{\mu}$ represents the parameter we are estimating, which is the mean of the distribution of our dependent variable $y_{c,t}$, by

including the predictor variables with their corresponding coefficients.

$$y_{c,t} \sim \mathcal{N}(\hat{\mu}_{c,t}, \sigma^2) \tag{4.33}$$

$$\hat{\mu}_{c,t} = \beta_0 + \sum_{i=1}^{n} \beta_i X_{i,c,t} + \sum_{j=1}^{m} \gamma_j Z_{j,c,t} \tag{4.34}$$

We assign a prior distribution to the unknown coefficients in the model, a zero-mean normal prior with a variance of 100 for the intercept. We set the mean of the coefficients of the marketing variables and economic factors equal to the mean of a standard OLS estimation, and the variance equal to 100, which corresponds to weak information regarding the true parameter values. For the prior of σ, we assume homoscedasticity (Section 2.3.4), and a half-normal distribution (normal distribution bounded at zero).

$$\beta_0 \sim \mathcal{N}(0, 100) \tag{4.35}$$

$$\beta_i \sim \mathcal{N}(\mu_{OLS_{\beta_i}}, 100) \tag{4.36}$$

$$\gamma_j \sim \mathcal{N}(\mu_{OLS_{\gamma_j}}, 100) \tag{4.37}$$

$$\sigma \sim |\mathcal{N}(0, 100)| \tag{4.38}$$

4.5.6 AutoRegressive Integrated Moving Average (ARIMA)

ARIMA defines three order parameters: (p, d, q) (ArunKumar et al., 2022). AR(p) uses the interdependence between a current observation and past observations. I(d) involves differencing observations to achieve stationary by subtracting current values from previous values d times. MA(q) uses the relationship between an observation and residual errors from a moving average model applied to lagged observations. The moving average component represents the model error as a combination of previous error terms, with q denoting the number of terms in the model. Thus, the AR term p, MA term q, and I term d collectively form the ARIMA model. Uppercase letters P, D, and Q are similar as p, d, and q, but they are specifically applied to the seasonal component of the time series.

In our time series forecasting analysis, we use the ARIMA approach with exogenous variables (ARIMAX). Exogenous variables represent external factors that may impact time series dynamics. During modelling, we aim to develop an AutoForecaster that automatically generates models based on data. Therefore, we employ the `auto_arima` approach, a Python package that autonomously determines parameters. We predefine start and maximum values for both parameters p and q, and the model computes the best parameters based on the Akaike Information Criterion (AIC). During modelling we make several assumptions:

Assumption 1. We assume $d = 1$, $D = 1$, $max\_p = 5$, and $max\_q = 5$ in our time series modelling due to memory constraints.

For our mathematical model, let $y_{i,t}$ denote the time series variable for city c at time t. The ARIMA model is similar to the AR model from Section 4.5.1, we only add: $\epsilon_{c,t}^T = \left[\varepsilon_{c,t-1}, \varepsilon_{c,t-2}, \ldots, \varepsilon_{c,t-Q}\right]$, where Q is the maximum lag and $\theta = \left[\theta_1, \theta_2, \ldots, \theta_Q\right]$ are the moving average coefficients. Our modified expression is:

$$y_{c,t} = \phi Y_{c,t} + \theta \epsilon_{c,t} + \beta X_{c,t} + \gamma Z_{c,t} + \varepsilon_{c,t}$$

$$\hat{\beta}, \hat{\gamma} = \underset{\beta,\gamma,\phi,\theta}{\arg\min} ||y_t - \phi Y_{c,t} - \theta \epsilon_{c,t} - \beta X_{c,t} - \gamma Z_{c,t}||_2^2$$

We use the `auto_arima` function to integrate exogenous variables, which considers both statistical tests and algorithmic optimisation to determine the appropriate parameter

values, such as the order of differentiation. Exogenous variables, such as economic indicators or external events, capture additional influences on the time series. We include dummies in an attempt to decrease the RMSE value, when the prediction Root Mean Squeared Error (RMSE) is large.

4.5.7 Random Forest (RF)

RF trains on static data, whereas ARF adapts to sequential data streams, making it suitable for the continuous arrival of data (Yoon, 2021). In RF, methods such as Bagging and random feature selection are used to reduce correlations between base models. We implement the algorithm of Yoon (2021) in our AutoForecaster. We start with the training data which includes marketing variables $X_{c,t}$ for city c at time t, economic factors $Z_{c,t}$ for city c at time t, number of trees B, number of variables selected at each split m, and minimum node size $n_{\min}$. In the end, we have an ensemble of trees $\{T_b\}_{b=1}^{B}$. The steps involve for each $b = 1$ to B:

1. Z^* is a drawn bootstrap sample with training data size N.
2. T_b is a random-forest tree grown from to the bootstrapped data. The goal is to minimise the node size n_{min}. Repeat the following steps for each terminal node of the tree.
   (a) From p variables select m variables randomly.
   (b) Pick the best variable.
   (c) Split this into two daughter nodes.

To make a prediction at a new point x:

$$\hat{y}_{c,t} = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{4.39}$$

where:

- $\hat{y}_{c,t}$ is the predicted value at time t for city c,
- B is the number of trees in the RF,
- $T_b(x)$ is the prediction of the b-th tree for input x.

4.5.8 Panel model

In dynamic panel models, the independent variables and the dependent variable often simultaneously influence each other. We include past values of the dependent variable (lagged dependent variables) to capture the dynamic adjustment process over time. To our knowledge, in a marketing context, the dynamic panel model remains unexplored in the academic literature, making its application a novel technique in this field.

Our forecasting methodology uses a panel model tailored to predict *Units Serviced* from January 2023 onwards. In this approach, the *City* variable captures the cross-sectional nuances in the data set. Leveraging the `linearmodels` library and `PanelOLS`, we integrate dummy variables to augment the model's adaptability and precision.

In panel data analysis, we extend the OLS framework to implement both cross-sectional and time-series dimensions. We modify the panel model equation of Das (2019) such that it fits our data.

$$y_{c,t} = \beta X_{c,t} + \gamma Z_{c,t} + \alpha_c + \delta_t + \varepsilon_{c,t} \tag{4.40}$$

$$\hat{\beta}, \hat{\gamma} = \arg\min_{\beta,\gamma} ||y_{c,t} - \beta X_{c,t} - \gamma Z_{c,t} - \alpha_c - \delta_t||^2 \tag{4.41}$$

- α_c captures city-specific fixed effects, accounting for time-invariant heterogeneity across cities, $\alpha_c \overset{\text{i.i.d.}}{\sim} N(0,1)$.
- δ_t captures time-specific fixed effects, capturing time-specific factors affecting all cities, $\delta_t \overset{\text{i.i.d.}}{\sim} N(0,1)$.

The panel model systematically accounts for city-specific effects, ensuring a comprehensive understanding of diverse influences on *Units Serviced*. Using dummy variables, we identify and incorporate unique patterns associated with each city, this should contribute to a more nuanced and accurate forecasting process.

4.6 Marketing transformations

For our research, we consider the impact of advertising on *Units Serviced*. We apply the AdStock, and lag formulas outlined in Section 3.4.3 to our marketing data. However, as the hyperparameters are initially unknown, we conduct a grid search to optimise the least squares estimator under various assumptions.

Assumption 2. Using OLS, we determine the parameters for all variables, which remain consistent across different models, but differ per city.

This sampling technique allows us to estimate the optimal parameters by minimising the RSS:

$$y_{c,t} = \beta X_{c,t} + \varepsilon_t \tag{4.42}$$

$$\hat{\beta} = \arg\min_{\beta} ||y_{c,t} - \beta X_{c,t}||_2^2 \tag{4.43}$$

$$X_{i,c,t} = A_{i,c,t} + \lambda_{i,c} X_{i,c,t-1}, \quad t = 1, \ldots, T \tag{4.44}$$

$$0 \leq \lambda_{i,c} < 1, \quad i = 1, \ldots, n \tag{4.45}$$

$$X_{i,c,0} = A_{i,c,0}, \quad \text{for } t = 0 \tag{4.46}$$

Here, λ represents the decay rate for each marketing variable i in city c, with values between zero and one. We implement a cross-validation to minimise the OLS estimation to find the best λ. $A_{i,c,t}$ represents the raw data value for variable i, at time t in city c, and $X_{i,c,t}$ represents the AdStock value of the marketing variable i, at time t in city c. $y_{t,c}$ denotes the actual value of *Units Serviced* at time t for city c. The lag value (l) represents the number of lagged weeks.

Assumption 3. $0 < l < 8$

After applying both transformations, the method calculates the combined effect of AdStock and the lagged function by performing element-wise multiplication between the transformed data sets. This step integrates the cumulative impact of advertising exposure with the lagged effect, resulting in a final data set that represents the expected consumer response over time considering the prolonged influence of advertising.

4.7 Data analysis and evaluation

After implementing the variable selection methods and forecasting techniques for both economic factors and the marketing variables, the determination of the best model becomes necessary. This process involves three decision points: first, selecting the best model considering the baseline and dummy models; second, identifying the best model among all models for each variable selection method; and finally, determining the best model based on the variable selection methods. The selection of the best model is based on performance measures and weights, Section 4.7.1 explains this further. This section also addresses the challenges encountered in the algorithm, considerations for future values of marketing variables, and the use of dummy variables.

4.7.1 Performance measures

For our final model selection, we determine the best model using performance measures. These measures, outlined in Table 4.1, include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC). MSE calculates the average squared difference between estimated and actual values, indicating overall model quality. RMSE and MAE assess how accurately the model predicts the target value, with RMSE being more sensitive to outliers and MAE treating all errors equally. BIC and AIC score the model based on its log-likelihood and complexity.

Table 4.1: Performance measures and their definitions (Kao et al., 2021; Mohammed et al., 2015; James et al., 2013).

Metrics	**Calculations**
MSE	$\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$
RMSE	$\sqrt{\frac{\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}{T}}$
MAE	$\frac{\sum_{t=1}^{T}\lvert y_t-\hat{y}_t\rvert}{T}$
BIC	$-2log(L) + 2K$
AIC	$-2log(L) + klog(T)$

Note: y_t and $\hat{y}_t$ represent the actual and predicted value, T is the total number of observations to train the model, L is the likelihood, and K is the number of model parameters.
Abbreviations: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC).

The `best_model_determination` function in Appendix I.6, shows our implementation of the performance measures in our algorithm. First, we initialise equal weights for the evaluation metrics ($w_p = 0.3$, where w is the weight and p are the performance measures MSE, RMSE and MAE, $w_b = 0.05$ for b is BIC and AIC), calculate the total value of each metric across all models and compute the ratio of each model's value to the total for each metric. With the weights and the ratios we can combine the scores to find the best model; $ratio_p = \frac{metric\_value_p}{\sum_{p=1}^{2} metric\_value_p}$. We categorise our measurements into two groups: one including MSE, RMSE and MAE, and the other containing BIC and AIC. Instead of comparing the absolute values of these metrics, we evaluate their ratios across different models. This approach considers the relative performance within each measurement category.

For the evaluation of model performance, a lower value is preferable for MSE, RMSE, and MAE. For AIC and BIC we aim to minimise the values, thus we apply positive weights and take the absolute sum of the metric values to ensure the right ratios. These weights are applied to ensure that lower values of these measures, indicating a better fit of the model, contribute positively to the overall score. With this scoring function, we select the best model by minimising the metrics. This method is crucial for selecting the best model, changing the weight sign can cause a different model to be selected. We define the score function for the model m as $S_m = \sum_{p=1}^{N} ratio_{m,p} * w_p$. We have adjusted our approach to assigning weights to the AIC and BIC scores, favouring lower weights. This adjustment results from observations across multiple simulations where the AIC and BIC exerted a disproportionately strong influence on the overall performance measure. This tendency led to the selection of models that did not necessarily offer the best overall performance. The function identifies the best-performing model across all metrics, providing a comprehensive assessment of model performance by considering multiple criteria simultaneously.

4.7.2 Multicollinearity

While programming, we encounter multicollinearity issues with the marketing variables. Multicollinearity was identified when the Variance Inflation Factor (VIF) exceeded 2.5, as recommended by Johnston et al. (2018). To address this concern, we incorporated a multicollinearity check into our model to ensure that it did not negatively affect the analysis.

4.7.3 Future values marketing variables

We apply a simple AR(L) model to the raw data for simplicity reasons, to forecast the future values of our independent marketing variables selected by the variable selection method. By conducting a grid search, we find and implement the optimal order for the AR(L) model to forecast future values for the marketing variables. Subsequently, we apply AdStock and lag transformations using the predetermined parameters (λ and l) for the best variable selection method. This method enhances model flexibility by eliminating the need to incorporate AdStock and lag transformations during the forecasting process. Moreover, separating the forecast of the raw values from the adjustment of AdStock and lags contribute to a easier understanding of the forecast outcomes. Additionally, prioritising interpretability, we maintain the direct interpretability of the model by using forecast of the raw values. These forecasts are straightforward predictions of future marketing variables without additional transformations.

4.7.4 Dummy variables

In our forecast, we need to have future values for the dummy variables, when these variables are selected by the variable selection method and if they are present in the best model as exogenous variables. Therefore, we implement a method to forecast dummy variables with the `Prophet` method to find future values and trends of the dummy variables. For each dummy variable in the data set of the best model, we forecast the dummy values for the next year and implement the new values in the data frame to forecast *Units Serviced*.

4.8 Results interpretation

The objective of our algorithm is to generate forecast values for the dependent variable within a 30/60/90-day timeframe. To assess the impact of marketing variables on *Units Serviced*, we established a baseline model. This baseline model includes all necessary

steps excluding the marketing variables. By comparing the performance of our best model with this baseline, we can confirm the efficacy of incorporating marketing variables. Furthermore, we should interpret how the marketing variables affect *Units Serviced*. Moreover, we intend to visualise all models to give insights in the variable selection methods and their respective models. To facilitate this analysis, we will capture all results into a dashboard to provide a comprehensive overview of our findings. Chapter 5 will elaborate on these results and their interpretation within the dashboard. Appendix I includes the pseudocode of the AutoForecaster.

4.9 Conclusion

In this chapter, we explore various models, concerning economic factors and *Units Serviced*. We employ four different techniques for variable selection. Using one of these methods, we can identify if economic factors are selected. If so, we apply forecasting techniques, including AR, XGBoost, SVR, OLS, KNN, Lasso, and RR, as outlined in Chapter 2.

The introduction of a marketing AutoForecaster streamlines the process by automatically identifying important variables by the different variable selection methods, BSS, FSS, PCA, and RF. This shows us which variables affect *Units Serviced*. If specific channels, such as *TV* or *OOH*, are identified, the marketing team can allocate additional budgetary resources to enhance their effectiveness compared to other channels. Furthermore, identifying channels with minimal influence allows the team to reallocate resources effectively. With these insights, the team can make informed data-driven decisions to guide their strategic initiatives.

To identify the optimal model, we used five performance metrics: MSE, RMSE, MAE, BIC, and AIC. The preferred model is determined by the combined score, with the model exhibiting the lowest score being selected. Additionally, this preferred model identifies the most influential marketing variables.

Chapter 5

Results

> *"The goal of forecasting is not to predict the future but to tell you what you need to know to take meaningful action in the present."*
>
> — Paul Saffo

The quote above highlights the significance of determining the variables that have the most influence within the optimal prediction model. These insights facilitate in creating informed conclusions and adjustments to budgets. In our context, understanding the primary contributors among marketing channels is important for guiding investment decisions. Moreover, it is plausible that marketing's impact on the KPI may be minimal, with market dynamics playing a dominant role, offering novel insights. This chapter focusses on addressing our research question and presenting the findings obtained.

> *What are the findings and implications of the marketing AutoForecaster across the varying 30/60/90-day time intervals, and how do these findings contribute to understanding marketing strategy dynamics and formulations?*

We analyse diverse trends and patterns, and show our discoveries. Furthermore, we explore both raw and transformed media data to identify the channels making the most significant contributions. We also assess the impact of media on prediction accuracy by comparing a basic model, without any marketing variables, with our best media prediction model. Additionally, we examine the quantitative results of our algorithm and present forecasts generated by our best-performing prediction model. Appendix J visualises the detailed design of the results in a PowerBI dasboard.

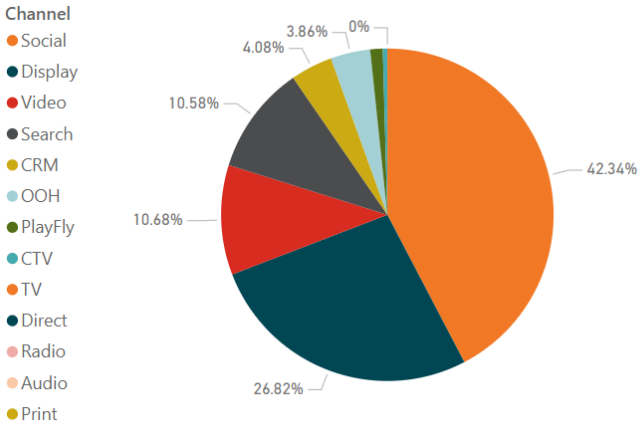
5.1 Media

We categorise our variables into different channel categories, each representing a unique medium for marketing communication. These include *Social* platforms like Facebook and Instagram, *Display* advertising on websites and apps, *Search* ads on the search engine Google, *Video* promotions on platforms like YouTube, Out-of-Home (*OOH*) displays such as billboards, and Customer Relationship Management (*CRM*) channels like email. Other marketing channels include *PlayFly*, advertising in gaming or mobile applications, Connected TV (*CTV*), advertising on Netflix and smart TVs, traditional *TV*, *Direct* marketing (telemarketing), *Radio* platforms, *Audio* platforms, and *Print* media (papers). These channels offer varied opportunities to reach and engage target audiences in different contexts and platforms.

Without applying any transformations, we can plot the different values measured in clicks, GRPs, impressions, and sent per channel to identify the channel with the highest values. In Figure 5.1b, we observe that the channel *Social* accounts for the largest share compared to all other channels, mainly due to the number of impressions. Each impression value indicates whether or not a customer saw the campaign. Taking a closer look into the *Social* channel, we find that it consists of 44.19% of the total impressions. Additionally, *Display*, *Search*, and *Video* channels also include a large proportion of the total values.

Channel	Clicks	GRPs	Impressions	Sent	**Total**
Audio			0.00%		**0.00%**
CRM				100.00%	**4.08%**
CTV			0.46%		**0.44%**
Direct			0.00%		**0.00%**
Display			27.99%		**26.82%**
OOH			4.03%		**3.86%**
PlayFly			1.27%		**1.22%**
Print			0.00%		**0.00%**
Radio		38.66%			**0.00%**
Search	100.00%		10.92%		**10.58%**
Social			44.19%		**42.34%**
TV		61.34%			**0.00%**
Video			11.15%		**10.68%**
Total	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**

(a) Distribution of clicks, GRPs, sent, and impressions across channels



(b) The contribution per channel, measured in clicks, GRPs, sent, and impressions

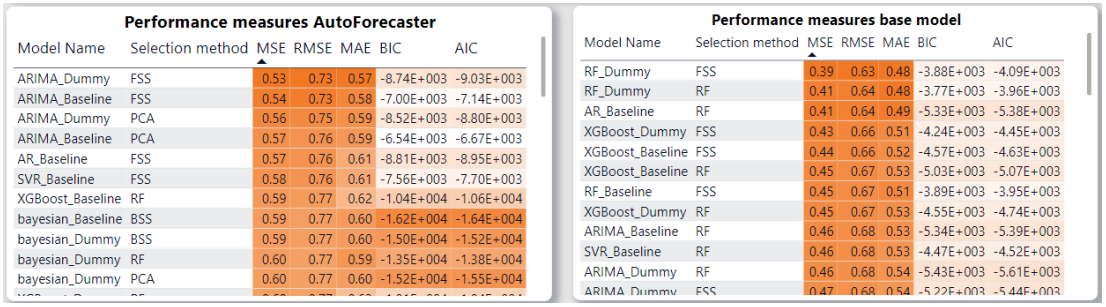
Figure 5.1: Both figures give information about the raw data before transformation.

This relative comparison across channels provides insights into their respective contributions based on clicks, GRPs, impressions, and sent. Figure 5.1a depicts that impressions are relatively important, given their substantial variance in values. The Search channel covers both clicks and impressions; however, our focus remains solely on clicks, given Google's nature as a click-based search engine.

5.2 Basic vs Marketing approach

To assess the impact of marketing variables on the predictive accuracy, we compare the performance of a baseline approach with that of a marketing-inclusive model. The baseline approach considers only economic factors and non-marketing variables. The methodology of both models is consistent. Figure 5.2 presents the performance metrics of all models with their respective variable selection techniques (BSS, FSS, PCA, RF), comparing both the baseline and the AutoForecaster.

The left-hand side of Figure 5.2 shows the performance metrics of the AutoForecaster, while the right-hand side shows those of the baseline model. We filter both figures based on the MSE (the highest MSE value is at the top). The visualisation uses conditional formatting, with a deeper orange colour for lower metric values, indicating better performance.



Performance measures AutoForecaster

Model Name	Selection method	MSE	RMSE	MAE	BIC	AIC
ARIMA_Dummy	FSS	0.53	0.73	0.57	-8.74E+003	-9.03E+003
ARIMA_Baseline	FSS	0.54	0.73	0.58	-7.00E+003	-7.14E+003
ARIMA_Dummy	PCA	0.56	0.75	0.59	-8.52E+003	-8.80E+003
ARIMA_Baseline	PCA	0.57	0.76	0.59	-6.54E+003	-6.67E+003
AR_Baseline	FSS	0.57	0.76	0.61	-8.81E+003	-8.95E+003
SVR_Baseline	FSS	0.58	0.76	0.61	-7.56E+003	-7.70E+003
XGBoost_Baseline	RF	0.59	0.77	0.62	-1.04E+004	-1.06E+004
bayesian_Baseline	BSS	0.59	0.77	0.60	-1.62E+004	-1.64E+004
bayesian_Dummy	BSS	0.59	0.77	0.60	-1.50E+004	-1.52E+004
bayesian_Dummy	RF	0.60	0.77	0.59	-1.35E+004	-1.38E+004
bayesian_Dummy	PCA	0.60	0.77	0.60	-1.52E+004	-1.55E+004

Performance measures base model

Model Name	Selection method	MSE	RMSE	MAE	BIC	AIC
RF_Dummy	FSS	0.39	0.63	0.48	-3.88E+003	-4.09E+003
RF_Dummy	RF	0.41	0.64	0.48	-3.77E+003	-3.96E+003
AR_Baseline	RF	0.41	0.64	0.49	-5.33E+003	-5.38E+003
XGBoost_Dummy	FSS	0.43	0.66	0.51	-4.24E+003	-4.45E+003
XGBoost_Baseline	FSS	0.44	0.66	0.52	-4.57E+003	-4.63E+003
XGBoost_Baseline	RF	0.45	0.67	0.53	-5.03E+003	-5.07E+003
RF_Baseline	FSS	0.45	0.67	0.51	-3.89E+003	-3.95E+003
XGBoost_Dummy	RF	0.45	0.67	0.53	-4.55E+003	-4.74E+003
ARIMA_Baseline	RF	0.46	0.68	0.53	-5.34E+003	-5.39E+003
SVR_Baseline	RF	0.46	0.68	0.53	-4.47E+003	-4.52E+003
ARIMA_Dummy	RF	0.46	0.68	0.54	-5.43E+003	-5.61E+003
ARIMA_Dummy	FSS	0.47	0.68	0.54	-5.22E+003	-5.44E+003

Figure 5.2: Comparison of performance metrics between the baseline and AutoForecaster models, with darker shades indicating better performance.

Comparing the models in Figure 5.2, we see that the MSE, RMSE, and MAE are lower for the baseline, implying better accuracy within this method. However, when considering the BIC and AIC values for both approaches, we conclude that the BIC and AIC values of the AutoForecaster are substantially lower. This discrepancy indicates that the AutoForecaster model aligns well with the data and that the model has a better trade-off between model fit and complexity, as a lower BIC and AIC suggest potential ease of model interpretation and generalisation to new data. So, even though the AutoForecaster has higher MSE, RMSE, and MAE compared to the baseline, it still has benefits because it is more straightforward and may work better with the data.

5.3 Best model

The Bayesian prediction model, coupled with BSS, emerges as our top performing model. This model selects eighteen variables, including two economic factors. Within this section we present the performance measures of the best-performing models, and elaborate on the visualisation of specific models and the forecast of our best model.

5.3.1 Performance measures

Figure 5.3 displays the performance metrics, including MSE, RMSE, MAE, BIC, and AIC, with BIC and AIC presented in a scientific format. Deeper shades of orange represent lower values, indicating better performance metrics. Chapter 4.7.1 explains the weighting technique used to identify the optimal model within the algorithm.

From Figure 5.3 we observe close similarities in the MSE, RMSE, and MAE values across the top models, and larger differences in the BIC and AIC values. Again, we filter based on the MSE values. Despite ARIMA models ranking among the top four, the Bayesian Baseline model holds the eighth position. Upon closer inspection, while the ARIMA model includes slightly better accuracy metrics, the Bayesian model outperforms in terms of BIC and AIC criteria. This suggests potential ease of model interpretation and generalisation to new data for the Bayesian model. The ARIMA model might be slightly overfitting the data, leading to lower BIC and AIC values. This shows the trade-off between model complexity and accuracy, with the Bayesian model achieving comparable accuracy while maintaining greater simplicity, as indicated by the weights mentioned in Chapter 4.7.1.

Model Name	Selection method	MSE	RMSE	MAE	BIC	AIC
ARIMA_Dummy	FSS	0.53	0.73	0.57	-8.74E+003	-9.03E+003
ARIMA_Baseline	FSS	0.54	0.73	0.58	-7.00E+003	-7.14E+003
ARIMA_Dummy	PCA	0.56	0.75	0.59	-8.52E+003	-8.80E+003
ARIMA_Baseline	PCA	0.57	0.76	0.59	-6.54E+003	-6.67E+003
AR_Baseline	FSS	0.57	0.76	0.61	-8.81E+003	-8.95E+003
SVR_Baseline	FSS	0.58	0.76	0.61	-7.56E+003	-7.70E+003
XGBoost_Baseline	RF	0.59	0.77	0.62	-1.04E+004	-1.06E+004
bayesian_Baseline	BSS	0.59	0.77	0.60	-1.62E+004	-1.64E+004
bayesian_Dummy	BSS	0.59	0.77	0.60	-1.50E+004	-1.52E+004
bayesian_Dummy	RF	0.60	0.77	0.59	-1.35E+004	-1.38E+004
bayesian_Dummy	PCA	0.60	0.77	0.60	-1.52E+004	-1.55E+004
XGBoost_Dummy	RF	0.60	0.77	0.63	-1.01E+004	-1.04E+004
bayesian_Baseline	RF	0.61	0.78	0.60	-1.52E+004	-1.53E+004
bayesian_Baseline	PCA	0.61	0.78	0.61	-1.55E+004	-1.56E+004
bayesian_Dummy	FSS	0.61	0.78	0.61	-1.27E+004	-1.30E+004
XGBoost_Baseline	FSS	0.61	0.78	0.63	-9.29E+003	-9.42E+003
XGBoost_Baseline	BSS	0.62	0.78	0.63	-1.07E+004	-1.08E+004
XGBoost_Dummy	FSS	0.62	0.79	0.64	-9.02E+003	-9.30E+003
AR_Baseline	PCA	0.62	0.79	0.63	-8.21E+003	-8.34E+003
SVR_Dummy	FSS	0.62	0.79	0.64	-8.93E+003	-9.22E+003
XGBoost_Dummy	BSS	0.63	0.79	0.64	-1.03E+004	-1.05E+004
bayesian_Baseline	FSS	0.63	0.79	0.62	-1.46E+004	-1.47E+004
SVR_Baseline	PCA	0.64	0.80	0.64	-7.79E+003	-7.92E+003
XGBoost_Baseline	PCA	0.64	0.80	0.65	-1.10E+004	-1.11E+004
AR_Dummy	FSS	0.65	0.81	0.65	-9.14E+003	-9.41E+003
XGBoost_Dummy	PCA	0.66	0.81	0.66	-1.05E+004	-1.08E+004

Figure 5.3: The performance measures of the top performing models with their variable selection methods, with a darker shade of orange indicates a better performance metric filtered on the MSE values.

5.3.2 Visualisation

We conduct a comparative analysis between the ARIMA and Bayesian models, using Figure 5.4, which shows the graphical representations. All models show deviations from the real values; however, due to confidentiality considerations, the y-axis values are not depicted, thereby presenting a somewhat distorted view. Nonetheless, the maximum difference we observe when knowing the values is approximately 0.4, suggesting minimal disparities.

Figure 5.4: The Bayesian and ARIMA model, both with and without dummy variables, for all variable selection techniques, with the green line as our testing set of the *Units Serviced*.

In Figure 5.4, a zoomed-in view shows the differences between the models. Upon closer examination, the ARIMA dummies model with FSS closely tracks the pattern of *Units Serviced*, with a noticeable gap between the two lines. In contrast, the Bayesian model demonstrates a more consistent trajectory, displaying fewer fluctuations while remaining close to the real values.

The variable *Units Serviced* displays an upward trend not reflected in the prediction models, as shown in the graph. This discrepancy could arise from the models' failure to capture the upward trend present in the dataset, as we did not include a trend variable in the first version of the AutoForecaster. Future versions of the model may include additional conditions to address this limitation.

5.3.3 Forecast

For our forecasting, we implement out-of-sample validation with a holdout period. This involves dividing the dataset into training and testing sets. The testing set, acting as out-of-sample data, also serves as the holdout period for evaluating the model's performance after training. Figure 5.5 shows the progression of forecast values over time. The orange line denotes the *Units Serviced*, while the blue line represents our optimal predictive model: the Bayesian model with BSS. We zoom in on the picture to highlight the distinctions between the forecasting values.

All forecasts display a more horizontal pattern with a slight downward trend, reflecting the increased uncertainty over a longer predicted time horizon. Both the 30-day and 60-day forecasts start from the same initial value. The endpoint of the 60-day forecast corresponds with the value predicted by the 90-day horizon at the same date.

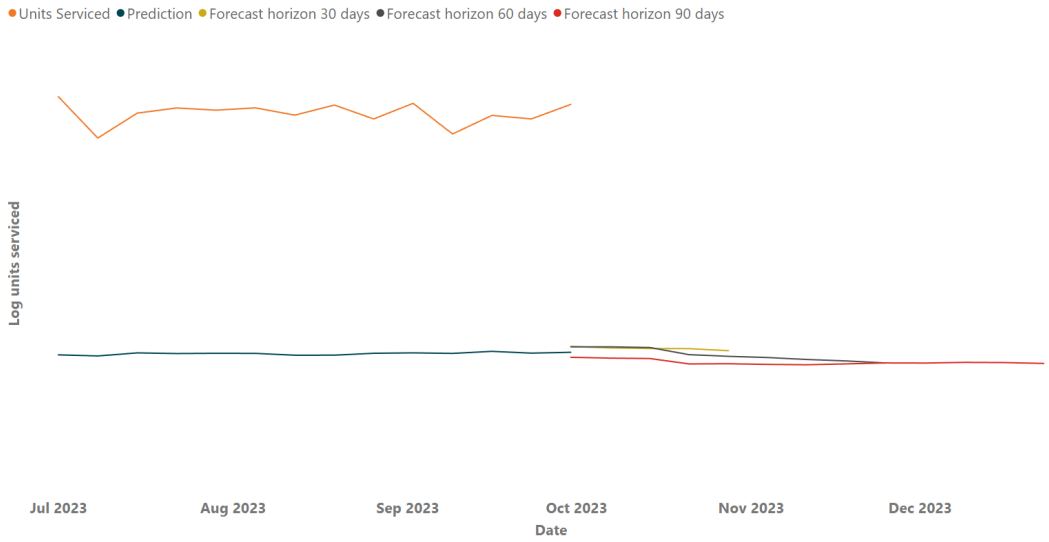


Figure 5.5: The Bayesian vs the actual values of *Units Serviced*, with the forecast values in a 30/60/90-days horizon. We zoom in on the picture to highlight the distinctions between the forecasting values.

We calculate the R squared as it explains the proportion of the variance in the *Units Serviced* that is predictable from the independent variables in our model, serving as a measure of goodness of fit (Seraj et al., 2024).

$$R^2 = 1 - \frac{\sum_{c=1}^{C} \sum_{t=1}^{T} (y_{c,t} - \hat{y}_{c,t})^2}{\sum_{c=1}^{C} \sum_{t=1}^{T} (y_{c,t} - \bar{y}_{c,t})^2} \tag{5.1}$$

Here, y_t represents the observed value of the dependent variable at time t for city c, $\hat{y}_t$ represents the predicted value of the dependent variable at time t for city c, $\bar{y}_t$ is the mean of the observed values of the dependent variable across all observations in the dataset, T represents the total number of time periods, and C is the total number of cities.

The performance measures of the model (Section 5.3.1) demonstrate accurate predictions and efficient model selection. However, despite these metrics, the R^2 value is 0.10, suggesting that the model explains only a small portion of the variance in the dependent variable. This value may result from omitted variable bias or inadequate capture of the underlying trend. Incorporating additional variables, such as tracking open stores, could improve the model's explanatory capacity and its ability to capture the trend.

5.4 Analysis of trends and patterns

In this section we analyse the trends and patterns emerging in the economic factors, the marketing variables and the different variable selection methods.

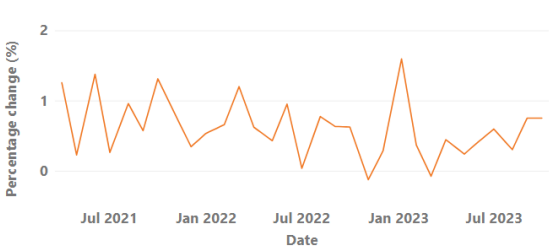
5.4.1 Economic factors

Figure 5.6 represents the economic factors over time. Upon closer inspection, we observe continuous data from the interpolation transformation. For visualisation purposes, we average our values across all cities; this enhances the interpretation of the factors. Given the previous mentioned constraints on using the real values for each city of Personal Consumption Expenditure (PCE) and unemployment, we use their percentages in the model, simplifying the allocation of real values per city. Furthermore, we assume the accuracy of the variables Vehicle Miles Travelled (VMT) and oil price, as these were given in the data set.

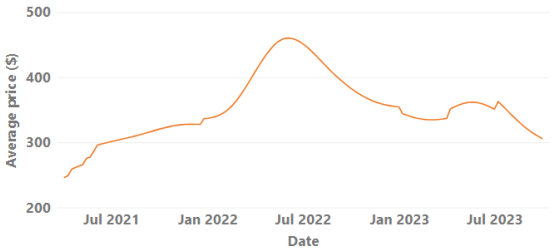
Our best model, the Bayesian method with BSS, identifies two economic factors as independent variables that impact *Units Serviced*. Figure 5.6 integrates the optimal forecast of these factors. The forecast models of the economic factors may differ from the one chosen for the marketing KPI, as they employ distinct methodologies to determine their best prediction model and forecasts.

Upon examining the unemployment rate, we see a downward trend, indicating several positive developments. A decreasing unemployment rate suggests increased job opportunities, improved job market conditions, and enhanced employer confidence in hiring. Hjazeen et al. (2020) confirms, there is a negative correlation between unemployment rate and economic growth. These trends reflect economic growth and optimism among consumers regarding future job prospects, encouraging higher expenditure and driving economic expansion.

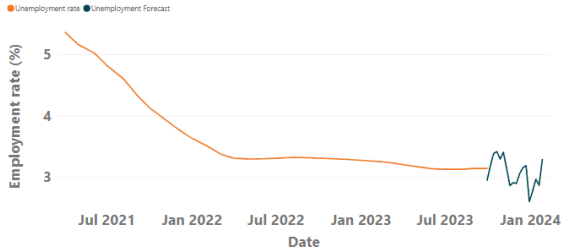
Regarding the percentage change in PCE, observations reveal a range from -0.13 to +1.59. Despite negative instances across two time periods, the predominant positive values suggest an increasing trend in the overall percentage PCE. This suggests a rise in consumer confidence and willingness to spend on goods and services, indicating economic growth and stimulating overall economic activity (Olusola et al., 2022).



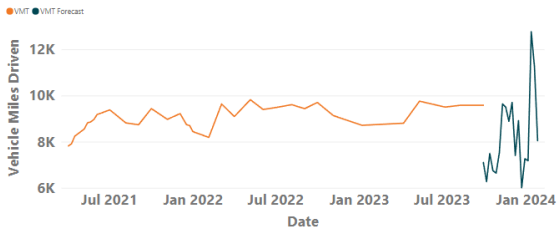
(a) The average percentage change per city over time for PCE.



(b) The average Oil Price in dollars ($) per city over time.



(c) The average unemployment rate (%) per city over time with forecasting values using the OLS method.



(d) The average Vehicle Miles Travelled (VMT) per city over time with forecasting values using the Lasso method.

Figure 5.6: Economic factors across time, averaged by city: Our top prediction model incorporates the unemployment rate and VMT, using their best forecast methods.

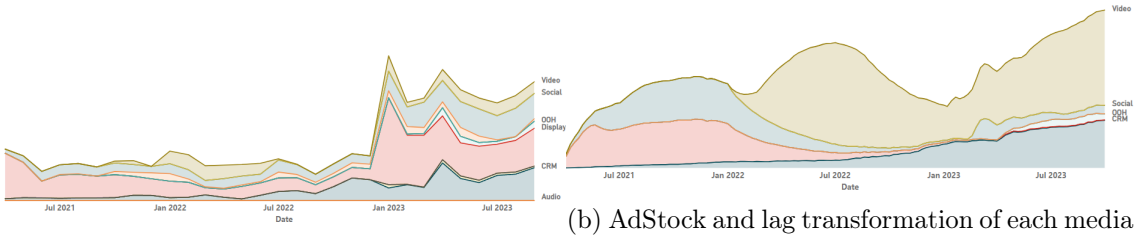
Furthermore, the analysis of VMT shows an upward trend, suggesting an increase in driving activities among individuals. Similarly, the trajectory of oil prices shows a slight increase, resulting in higher fuel costs. These trends collectively indicate increased economic activity and highlight the dynamic influence of market forces on consumer behaviour and spending patterns.

In Figure 5.6, we visualise the forecasts of economic factors in a three month horizon. BSS selects two economic factors, the unemployment rate and VMT, resulting in them influencing our KPI, *Units Serviced*. Given their selection, our forecast focusses solely on these two factors. The optimal model for forecasting the unemployment rate involves the OLS model with a 6 weeks lag as its exogenous variables. For forecasting VMT, the selected model is the Lasso model with a regularisation parameter (α) set to 0.999. These forecast values help the generation of reliable forecasts for our dependent marketing variable.

5.4.2 Marketing variables

We implement an AdStock and lag transformation for our marketing variables. Figure 5.8 displays the channel data before (Figure 5.7a) and after (Figure 5.7b) the transformation. The data shows smoothing over time, with an increase in values due to cumulative advertising expenditures. For the *CRM* channel, we find multiple spikes in Figure 5.7a, while after transformation, there is a consistent smoothed increase in values.

Display, represented by the green line starting at the beginning of the figure, maintains values beyond January 2023; however, these values fade compared to *Video*, *Social*, *OOH*, and *CRM*, making it almost undetectable. A similar scenario unfolds for *Radio*, *Search*, *TV*, and *Video* channels.



(a) Raw data of the media channels over time.

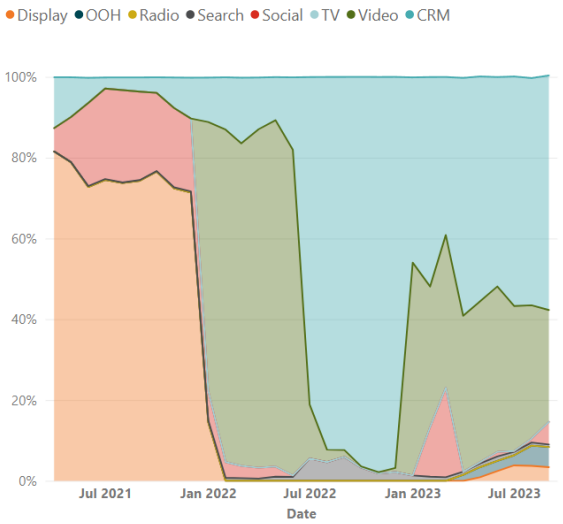
(b) AdStock and lag transformation of each media channel over time.

Figure 5.7: The AdStock and lag transformation of the marketing channels over time.

Figure 5.8a presents the coefficients per selected variable, extracted from our best prediction model. These coefficients show the relationship between each variable and *Units Serviced*. The constant term represents the expected $log(Units\ Serviced)$, when all variables are zero, while β_i and γ_i describe the percentage change in the expected value of *Units Serviced* for a one-unit change in the corresponding independent variable X_i and Z_i, holding other variables constant.

Variable	Coefficient
constant	12.96
CRM_Email_Total_Sent	-0.25
Discount_Perc	-0.37
Display_TTD_GRAND_OPENING_Impressions	0.46
Display_TTD_Promo_Impressions	0.37
HoursActual	0.28
OOH_DIGITAL_BULLETIN_GRAND_OPENING_Impressions	-0.33
Radio_TTWN_GRPs	-0.19
Search_Google_Brand_Clicks	0.34
Search_Google_NB_Clicks	-0.13
Search_Google_NB_Promo_Clicks	-0.56
Social_Facebook_Promo_Impressions	-0.04
Social_Tiktok_Other_Impressions	-0.05
TV_Cable_Campaign1_GRPs	-0.47
TV_Cable_Campaign6_GRPs	0.52
TV_Total_GRPs	-0.11
Unemployment_rate_%	0.00
Video_Youtube_Non-Promo_Impressions	-0.20
VMT	0.27

(a) Variable coefficients of our best model, the Bayesian model, selected by BSS.



(b) Media channel contributions over time as a percentage of the total media variables.

Figure 5.8: The coefficients of chosen variables and the media channels' percentage contribution to the total of media variables over time.

If β_i is positive, it means that a one-unit increase in the variable X_i correlates with a percentage increase in *Unit Serviced*. For example, consider the variable TV_Cable_Campaign6_GRPs with $\beta = 0.52$. A negative β means that one unit increase in the variable X causes a 0.52% decrease in *Units Serviced*. If we consider the coefficients in Figure 5.8a and the values of the raw data in Figure 5.7a.

Figure 5.8b illustrates the varying percentage contributions of media variables over time, reflecting a cumulative assessment of these variables. The calculation is expressed by the following formula:

$$Channel\ contribution_{k,t} = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{\sum_{i\in k}\beta_i X_{i,t,c}}{\sum_{i=1}^{n}\beta_i X_{i,t,c}}\right) * 100 \tag{5.2}$$

Here, $Channel\ contribution_{k,t}$ represents the contribution of a specific media channel k at time t, expressed as a percentage. In this formula, β_i is the coefficient of the i-th media variable, and $X_{i,t,c}$ represents the raw value of the i-th media variable at time t in city c. The summation $\sum_{i \in k}$ denotes the aggregation of all media variables within channel k, while n is the total number of the media variables, and C is the total number of cities.

In Figure 5.8b, we observe that *CRM* consistently contributes across all time periods. *Display* and *Social* have an inital presence followed by a decline. *Video* has a significant influence from January 2022 onwards. On the other hand, *Radio*, *OOH*, and *TV* have the least influence throughout the analysed time period, caused by the relatively small proportion in the overall metrics values, as depicted in Figure 5.1a and Figure 5.8a.

Table 5.1 displays the average coefficients across media channels. Our findings reveal that, on average, the coefficients associated with media channels tend to be negative, with the exception of the *Display* channel. Additionally, economic factors and non-media variables cannot be averaged due to their differing measurement metrics. However, unemployment demonstrates a slight negative coefficient of -0.0006. A 1% rise in the variable TV_Cable_Campaign6_GRPs results to a 0.52% increase in the *Units Serviced*. Thus, a 1% expansion in audience reach anticipates a 0.52% growth in *Units Serviced*.

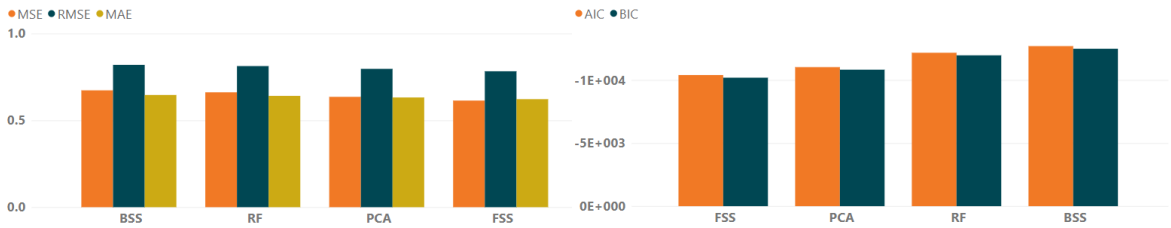
Table 5.1: The average coefficients per media channel, excluding non-media variables and economic factors due to their differing measurement metrics.

Metric	**Channel**	**Coefficient**
Clicks	Search	-0.12
GRPs	Radio	-0.19
	TV	-0.02
Impressions	Display	0.42
	OOH	-0.33
	Social	-0.05
	Video	-0.20
Sent	CRM	-0.25

The negative coefficients observed for certain channels suggest potential inefficiencies in their effectiveness. Allocating additional resources to these channels may not necessarily lead to an increase in *Units Serviced*. Such negative coefficients may arise due to various factors, including reaching saturation levels, diminished effectiveness of marketing efforts, poor execution of campaigns, or competition within the industry.

5.4.3 Variable selection methods

In our analysis of variable selection methods, we use two visualisations to determine if any method outperform the others in all models. To ensure a focused analysis, we exclude OLS and panel models, as their values were not comparable to the other models, resulting in odd trends and patterns in the visualisations. Figure 5.9 shows the average performance metrics per variable selection method. We split the MSE, RMSE, and MAE values from the AIC and BIC to enhance visual clarity, and invert the y-axis of Figure 5.9b.



(a) The average MSE, RMSE, and MAE values per variable selection method.

(b) The average BIC and AIC values per variable selection method, with an inverse y-axis.

Figure 5.9: The average performance measures MSE, RMSE, MAE, BIC, and AIC for each variable selection method, with an inverted y-axis for Figure 5.9b.

From these figures, we observe a consistent downward trend across Figure 5.9b. In both figures, the left side depicts the variable selection methods with higher values, and the right side those with lower values. In Figure 5.9a, there is a downward trend, however this is really small. FSS has a better performance in terms of MSE, RMSE, and MAE values, although the difference between the methods are relatively small, ranging from 0.03 to 0.06. There is bigger variation among the models in terms of BIC and AIC values. BSS outperforms the others, with the FSS method emerging as the least effective, having an average difference of around -2000 for both the AIC and BIC values. The close values of the MSE, RMSE, and MAE values among the methods, contrasted with the greater discrepancies in the BIC and AIC, clarifies the reasoning behind the selection of the BSS method.

5.5 Marketing strategy

From our best model, the Bayesian model, the channel *Display* positively influences the *Units Serviced*. We recommend allocating more resources to the *Display* channel, particularly focussing on the two variables selected by BSS. Additionally, we should monitor the selected economic factors (unemployment rate and VMT) closely, as they also influence the marketing model. A decrease in the VMT will likely correlate with a decrease in *Units Serviced*, as reduced travel activity can reduce consumer mobility and spending. An increase in the unemployment rate typically results in a decrease in *Units Serviced*, as higher unemployment leads to reduced purchasing power and consumer demand. While most channels influence the *Units Serviced* negatively, certain specific variables have a positive impact. Therefore, we suggest allocating more resources to these variables, TV_Cable_Campaign6_GRPs and Search_Google_Brand_Clicks, and reducing resources allocation to variables with negative influences on the *Units Serviced*.

5.6 Conclusion

In this chapter, our objective was to identify significant findings and implications, as well as finding emerging trends or patterns within the results to inform marketing strategies and shape future approaches. Our data set consists of *Display* and *Social* channels, which contribute the most to impressions. *Search* is a main contributor to clicks, while CRM contributes the most in sent, and *TV* to GRPs. Most of the variables selected from our best model originate in these four channels. Other channels influencing *Units Serviced* include *OOH*, *Radio*, *TV*, and *Video*. On average, all channels contribute negatively to *Units Serviced*, except for *Display*. Furthermore, we observe positive variables present in *Search* and *TV*. The implementation of the AdStock transformation ensures smoothness; however, the absence of diminishing returns could potentially enhance channel coefficients.

We apply different models to distinct variable sets based on the variable selection methods. Our analysis reveals that MSE, RMSE and MAE are closely aligned, whereas BIC and AIC have greater disparities. As the BIC and AIC are lower for BSS, this method results as our best variable selection method.

Our best model is a consideration between the ARIMA dummy model with FSS and the Bayesian baseline model with BSS. Despite the ARIMA model showing lower MSE, RMSE, and MAE values, the algorithm prefers the Bayesian baseline model due to its significantly lower BIC and AIC scores. Furthermore, the Bayesian model visualises a more consistent trajectory close to the actual values, whereas the ARIMA model demonstrates a noticeable gap between the real and predicted values.

Our best model selects two economic factors which influence *Units Serviced*. We forecast both VMT and unemployment rate so they can be implemented, with VMT forecast using the Lasso model and the unemployment rate using the OLS model. The VMT has a coefficient of 0.27, indicating that a 1% increase in the miles driven, will cause a 0.27% growth in *Units Serviced*. The unemployment rate has a negative coefficient, with a 1% increase resulting in a 0.0006% drop in *Units Serviced*.

While most channels influence the *Units Serviced* negatively, certain specific variables have a positive impact. Therefore, we suggest allocating more resources to these variables, TV_Cable_Campaign6_GRPs, Search_Google_Brand_Clicks and the *Display* channel, and reducing resources allocation to variables with negative influences on the *Units Serviced*.

Chapter 6

Conclusion, discussion, and recommendations

"Prediction is very difficult, especially about the future."

— Niels Bohr

In this chapter, we aim to address our main research question, while considering all preceding chapters. We also include discussions, recommendations, and suggestions for future research. Our research question is formulated as follows:

> *How can an efficient implementation of a marketing AutoForecaster optimise marketing strategies while simultaneously providing up-to-date financial insights and forecasting economic factors?*

The quote above acknowledges the challenges associated with prediction models, as they must anticipate upon independent variables and unforeseen market dynamics that could influence future values of the dependent variable. It suggests that despite our best efforts and advancements in prediction methods, accurately forecasting future events remains highly complex and subject to error. Therefore, it is essential to remain open to adapting our predictions as new information becomes available.

6.1 Conclusions

Our research demonstrates the efficacy of our AutoForecaster in optimising marketing strategies and providing valuable insights within a 30/60/90-day framework. By incorporating various methods from the literature and automatically selecting the best model with relevant variables among the different models, our AutoForecaster streamlines the process of model selection. This first version establishes a foundation, allowing for the future addition of more conditions and constraints to further enhance its capabilities and increase model predictability and accuracy.

Our analysis shows the contribution of marketing variables, non-marketing variables, and economic factors in driving marketing strategy decisions. The 30/60/90-day time horizon proves to be an effective approach for short-term predictions across different selected models, ensuring adaptability and responsiveness in decision-making. Adding marketing variables to our prediction model ensures better BIC and AIC values resulting in better model fit and complexity.

Expanding the dataset size becomes more manageable with our AutoForecaster as it automatically selects the most significant variables. Moreover, by integrating PowerBI with our Python algorithm, we improve interpretability and create user-friendly dashboards adaptable to client preferences. This integration facilitates real-time insights and can be extended to more comprehensive financial dashboards as additional data become available.

Our findings highlight that economic factors are critical into marketing models to enhance forecasting accuracy, as the variable selection models include these factors. By smoothing economic factor data through interpolation and predicting future values, our model becomes more reliable, enabling better allocation decisions and improving overall strategy effectiveness.

The AutoForecaster not only selects variables based on variable selection methods, but also provides corresponding coefficients, offering insights for optimising marketing strategies. For instance, our Bayesian model identifies the *Display* channel as positively influencing *Units Serviced*, suggesting a need for increased resource allocation to this channel. Similarly, monitoring economic factors such as the unemployment rate and Vehicle Miles Travelled (VMT) is essential as they significantly impact marketing performance.

In conclusion, our AutoForecaster represents a valuable tool for marketeers seeking to optimise their strategies while implementing the evolving market dynamics. By creating data-driven insights and integrating advanced analytics techniques, organisations can make more informed decisions and achieve greater success in their marketing accomplishments.

6.2 Discussion

We address the contributions to theory, practice, and the limitations of our research. This section ensures a comprehensive analysis of our AutoForecaster, as we not only acknowledge the strengths but also recognise the constraints and shortcomings of our study.

6.2.1 Contribution to theory

In the marketing setting, MMM models are commonly used. While financial models are used frequently outside the marketing setting, their application in a marketing setting remains relatively new. This study integrates various financial models within a marketing framework, using both traditional and novel approaches. By using variable selection methods and different forecasting techniques, we can compare the performances and identify key variables driving the KPI.

In literature, there is limited information about the process of automated techniques in a marketing context for selecting the optimal model among alternatives. We address this gap by introducing a novel methodology that implements diverse models into an automated system, enabling it to identify the most suitable model. Furthermore, we introduce an implementation of economic factor forecasting which is part of the data of the dependent marketing variable.

We apply the panel model within the marketing domain, a methodology that is unexplored in the existing literature to our knowledge. Additional we use KNN and RF techniques in time series modelling, and a marketing setting, which are not widely applied

yet.

The application of these techniques within the marketing domain holds theoretical significance by expanding the methodology. By incorporating diverse modelling approaches, our study helps in the understanding of the market dynamics, and efficacy of marketing strategies for the dependent variable. Although the individual techniques are not novel on their own, their adaptation and integration within the marketing framework could contribute to the improvement of theoretical frameworks and methodologies within marketing analytics, thereby increasing the novelty of this approach.

6.2.2 Contribution to practice

The model aligns with the interest of the company to compare their MMM model with other models. Currently, the company's software lacks the implementation of multiple variable selection techniques, relying solely on one method. By introducing various techniques and conducting comparative analyses, this application has the potential to uncover new insights and variables. The AutoForecaster helps with comparing the different influences of the variables on different prediction models. Doing so, it is possible to identify the most and least influential variables in the different models.

Additionally, the company currently relies on quarterly available data for economic factors. Integrating interpolation and forecasting models for these factors could enhance the models and the customer decision-making processes, as dashboards and model implementations are more up-to-date. The AutoForecaster automatically incorporates both economic factors and the dependent variable, simplifying the inclusion of economic factors. We also see whether economic factors indeed influence and affect the dependent variable by using variable selection methods for our dependent variable. Using an automated system, we reduce the time spent on finding out whether these factors influence the dependent variable and it is possible to forecast these factors to create accurate predictions.

Moreover, the automated nature of this model reduces time constraints, as it automatically selects the optimal model and predicts the dependent variable. This streamlined approach offers efficiency benefits through creating a more agile decision-making process. As such, modellers can spend their time refining the process and checking accuracy of the predictions. This first version of the AutoForecaster establishes a foundation upon which different constraints and additional features should be added to create an accurate automated forecasting system.

6.2.3 Limitations

In this study we used some assumptions and simplifications that may impact the validity of our findings. Initially, we conducted thorough examinations of the dataset and performed various tests to assess stationarity. However, based on visual inspection, we expect the data set to be non-stationary due to its increasing trend. Despite applying both the Levlin-Liu-Chu panel unit root test and the Augmented Dickey-Fuller (ADF) test across aggregated, time-series, panel, and raw data, all cases indicated that the data does not contain a unit root. Consequently, considering that it is easier for the clients to interpret and the implementation of marketing variables with lags, we assume a stationary data set. Nevertheless, this assumption may introduce biases and influence our results. The use of non-stationary time series data may yield false relationships between variables, so transformation to stationary data ensures consistent and reliable outcomes.

Moreover, due to time constraints, we only implemented two additional economic factors into our research. Integrating more economic factors could potentially enhance the accuracy of our model by capturing additional influences on the dependent variable, potentially increasing the prediction rate. Additionally, we also excluded diminishing returns, reflecting the reduced effectiveness of advertising exposure over time, due to time constraints. This could have caused the marketing coefficients to be negative as it provides a limit to how much you can spend in a channel before it reaches saturation.

Furthermore, we made certain assumptions regarding the ARIMA model due to memory constraints, which could affect its effectiveness and predictive capability. The limited memory resources resulted in relatively low model parameters. Enhancing computational resources could potentially address this limitation by facilitating increased parameters and improving the algorithm speed. Additionally, while we applied the RF regression model in our research, we did not implemented the ARF due to the unavailability of suitable implementation methods in Python, coupled with constraints related to computational resources. Moreover, the Bayesian method shows divergences, even when we set a very low acceptance rate (0.9999999) and use 6000 iterations for both drawing and tuning. This suggests that we should further explore the choice of our priors in our model and its restrictions.

Lastly, we use a grid search to fine-tune the least squares estimator of our OLS estimator to optimise marketing transformations. However, this estimation process may affect the reliability of our variables compared to other techniques. We also assumed consistency in transformation parameters across different models and cities, alongside a maximum lag of 8 weeks on the variables. These assumptions may impact the robustness and generalisation of our findings.

6.3 Recommendations and future research

Initially, we recommend comparing the results of the AutoForecaster with the client's current MMM model. It would be insightful to confirm whether the AutoForecaster creates comparable results. Specifically, it would be interesting to investigate whether the AutoForecaster recommends the same variables as the variables selection method in StrataQED. If there are differences in variable selection between the algorithms, we could incorporate these variables in one of the models to see if it enhances the results. Additionally, it is intriguing to know whether variables chosen by the client have significance in the AutoForecaster. We should also consider incorporating the diminishing returns, as excluding this transformation could have caused the coefficients to be negative.

Furthermore, we suggest incorporating the earlier mentioned economic factors in addition to conducting further research to confirm their influence on *Units Serviced*. Future research may involve identifying new economic factors not explored in this research, and finding city specific values for each factor.

Future research may also include extra steps for data pre-processing, exploring more detailed anomalies. While our primary focus remains on implementing the AutoForecaster methodology, enhancing pre-processing techniques could provide more insights. Exploring alternative approaches to addressing anomalies and assessing data stationarity would be beneficial.

Moreover, we advise to calculate the ROI and investment costs of different marketing campaigns. This analysis could inform clients in adjusting their investments based on the effectiveness of each campaign. For example, if the variable selection methods overlook high investment campaigns, there may be little rationale for investing in those specific campaigns. Perhaps reallocating resources to other campaigns that have greater impact on the number of *Units Serviced* would be more beneficial.

Examining our top model, the Bayesian model, reveals divergences in approximately half of all samples. For future research, we recommend to investigate the refinement of the model by specifying more dynamic priors and imposing constraints on marketing variables. These restrictions could ensure that all coefficients remain positive, reflecting the assumption that marketing activities have positive effects. Furthermore, investigating the effects of the marketing channels can give more insights. By establishing a general coefficient for channels and allowing variation to these coefficients for the underlying variables, a more nuanced understanding of channel and campaign performance could be attained. It is also possible to integrate AdStock, lag, and diminishing returns transformations into the Bayesian model, thereby selecting new parameters.

We evaluate our best model using five performance measures: MSE, RMSE, MAE, AIC, and BIC. However, the resulting R^2 value is low, suggesting that the independent variables explain only a small fraction of the variance in the dependent variable. Considering R^2 as part of our performance evaluation criteria in future iterations of the algorithm could be beneficial. Additionally, incorporating the number of open stores into the model may improve its ability to capture trends and patterns, potentially leading to an enhanced R^2 value. Trend variables can capture systematic changes or patterns over time, which are often crucial for understanding economic phenomena. However, researchers should exercise caution to prevent overfitting when incorporating such variables.

In future research, there is potential to implement dynamic models like the Adaptive Random Forest to handle continuous data arrival effectively. By adapting models to seamlessly incorporate ongoing data updates and dynamically adjust coefficients, valuable insights could be gained. Furthermore, it's worth exploring whether the same model consistently performs well when additional time periods are included. This could deepen our understanding of campaign dynamics over time.

References

Andersen, T. (2020). Managing in dynamic, complex and unpredictable business contexts. In T. Andersen & S. Torp (Eds.), *Adapting to Environmental Challenges: New Research in Strategy and International Business* (pp. 1-17). Emerald Publishing Limited.

Araujo, G., & Gaglianone, W. (2023). Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. *Latin American Journal of Central Banking*, *4*, 1-29.

ArunKumar, K., Kalaga, D., Kumar, C., Kawaji, M., & Brenza, T. (2022). Comparative analysis of Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) cells, Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) for forecasting COVID-19 trends. *Alexandria Engineering Journal*, *61*, 7585-7603.

Asad, M., Halim, Z., Waqas, M., & Tu, S. (2021). An in-ad contents-based viewability prediction framework using artificial intelligence for web ads. *Artificial Intelligence Review*, *54*, 5095-5125.

Balasundaram, S., Gupta, D., & Kapil. (2014). Lagrangian support vector regression via unconstrained convex minimization. *Neural Networks*, *51*, 67-79.

Bantis, E., Clements, M., & Urquhart, A. (2023). Forecasting GDP growth rates in the United States and Brazil using Google Trends. *International Journal of Forecasting*, *39*, 1909-1924.

Banton, C. (n.d.). *Market dynamics: Definition and examples.* Investopedia. Retrieved from `https://www.investopedia.com/terms/m/market-dynamics.asp`

Barak, S., & Parvini, N. (2023). Transfer-entropy-based dynamic feature selection for evaluating bitcoin price drivers. *Journal of Futures Markets*, *43*, 1695-1726.

Bayer, E., Srinivasan, S., Riedl, E. J., & Skiera, B. (2020). The impact of online display advertising and paid search advertising relative to offline advertising on firm performance and firm value. *International Journal of Research in Marketing*, *37*, 789-804.

Beltran-Royo, C., Escudero, L. F., & Zhang, H. (2016). Multiperiod multiproduct advertising budgeting: Stochastic optimization modeling. *Omega*, *59*, 26-39.

Billé, A., Tomelleri, A., & Ravazzolo, F. (2023). Forecasting regional GDPs: A comparison with spatial dynamic panel data models. *Spatial Economic Analysis*, *18*, 530-551.

Bureau of Economic Analysis (BEA). (2023). Personal consumption expenditures. In R. F. Douglas & S. H. McCulla (Eds.), *Concepts and Methods of the U.S. National Income and Product Accounts* (pp. 96-106).

Chan-Lau, J. (2017). *Lasso regressions and forecasting models in applied stress testing* (IMF Working Papers No. 2017/108). International Monetary Fund.

Cohen, G., & Aiche, A. (2023). Forecasting gold price using machine learning methodologies. *Chaos, Solitons Fractals*, *175*, 1-9.

Cui, G., Wong, M., & Zhang, G. (2010). Bayesian variable selection for binary response models and direct marketing forecasting. *Expert Systems with Applications*, *37*, 7656-7662.

Das, P. (2019). Panel data analysis: Static models. In *Econometrics in Theory and Practice: Analysis of Cross Section, Time Series and Panel Data with Stata 15.1* (pp. 457-497). Singapore: Springer Singapore.

Gan, M., Cheng, Y., Liu, K., & lin Zhang, G. (2014). Seasonal and trend time series forecasting based on a quasi-linear autoregressive model. *Applied Soft Computing*, *24*, 13-18.

Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*, 2225-2236.

Gijsenberg, M. J., & Nijs, V. R. (2019). Advertising spending patterns and competitor impact. *International Journal of Research in Marketing*, *36*, 232-250.

Gunjal, S. N., Kshirsagar, D. B., Dange, B., & Khodke, H. (2022). Fusing clustering and machine learning techniques for Big-Mart sales predication. In *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)* (pp. 1-6).

Guégan, D., & Rakotomarolahy, P. (2010). Alternative methods for forecasting GDP. In F. Jawadi & W. Barnett (Eds.), *Nonlinear Modeling of Economic and Financial Time-Series* (Vol. 20, pp. 161-185). Emerald Group Publishing.

Hartanto, C., Sofianti, T., & Budiarto, E. (2022). Multivariate sales forecast model towards trend shifting during COVID-19 pandemic: A case study in global beauty industry. In *Proceedings of the 2022 International Conference on Engineering and Information Technology for Sustainable Industry.* Association for Computing Machinery.

Hendry, D., & Clements, M. (2003). Economic forecasting: Some lessons from recent research. *Economic Modelling*, *20*, 301-329.

Hjazeen, H., Seraj, M., & Ozdeser, H. (2020). The Nexus between the economic growth and unemployment in Jordan. *Future Business Journal*.

Ishrat, I., Hasan, M., Farooq, A., & Khan, F. (2023). Modelling of consumer challenges and marketing strategies during crisis. *Qualitative Market Research: An International Journal*, *26*, 285-319.

Ismail, Z., Yahya, A., & Shabri, A. (2009). Forecasting gold prices using Multiple Linear Regression method. *American Journal of Applied Sciences*, *6*, 1509 – 1514.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning – with Applications in R* (2nd ed., Vol. 103). Springer.

Johnston, R., Jones, K., & Manley, D. (2018). Confounding and collinearity in regression analysis: A cautionary tale and an alternative procedure, illustrated by studies of British voting behaviour. *Quality & Quantity*, *52*, 1957–1976.

Joosten, R. A. M. G., Harmelink, R., & Sparrius, T. (2023). Short- and long-term optimality under sustainable threats in contest theory models of advertising and short-run competition.

Joseph, J. V. (2006). Understanding advertising adstock transformations. *University Library of Munich, Germany, MPRA Paper*(7683).

Kao, L.-J., Chiu, C.-C., Wang, H.-J., & Ko, C. (2021). Prediction of remaining time on site for E-Commerce users: A SOM and long short-term memory study. *Journal of Forecasting*, *40*, 1274-1290.

Le, C. (2020). How to choose tuning parameters in lasso and ridge regression? *Asian Journal of Economics and Banking*, *4*, 60-75.

Levin, A., Lin, C.-F., & James Chu, C.-S. (2002). Unit root tests in panel data: Asymptotic and finite-sample properties. *Journal of Econometrics*, *108*, 1-24.

Li, S., & Shi, W. (2023). Incorporating multiple textual factors into unbalanced financial distress prediction: A feature selection methods and ensemble classifiers combined approach. *International Journal of Computational Intelligence Systems*, *16*, 162-186.

Maccarrone, G., Morelli, G., & Spadaccini, S. (2021). GDP forecasting: Machine learning, linear or autoregression? *Frontiers in Artificial Intelligence*, *4*, 1-9.

Martin, G., Frazier, D., Maneesoonthorn, W., Loaiza-Maya, R., Huber, F., Koop, G., ... Panagiotelis, A. (2023). Bayesian forecasting in economics and finance: A modern review. *International Journal of Forecasting*, *40*, 811-839.

Migon, H., Alves, M., Menezes, A., & Pinheiro, A. (2023). A review of Bayesian dynamic forecasting models: Applications in marketing. *Applied Stochastic Models in Business and Industry*, *39*, 471-493.

Mohammed, E., Naugler, C., & Far, B. (2015). Emerging business intelligence framework for a clinical laboratory through big data analytics. In N. Quoc & A. Hamid (Eds.), *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology* (pp. 577-602). Kaufmann, M.

Morlotti, C., Mantin, B., Malighetti, P., & Redondi, R. (2024). Price volatility of revenue managed goods: Implications for demand and price elasticity. *European Journal of Operational Research*, *312*, 1039-1058.

Moshiri, S., Cameron, N., & Scuse, D. (1999). Static, dynamic, and hybrid neural networks in forecasting inflation. *Computational Economics*, *14*, 219–235.

Olusola, B. E., Chimezie, M. E., Shuuya, S. M., & Addeh, G. Y. A. (2022). The impact of inflation rate on private consumption expenditure and economic growth—evidence from Ghana. *Open Journal of Business and Management*, *10*, 1601-1646.

Pandey, S., Gupta, S., & Chhajed, S. (2021). Marketing Mix Modeling (MMM) - concepts and model interpretation. *International Journal of Engineering and Technical Research*, *10*, 784-793.

Peng, H., Bobade, S. U., Cotterell, M. E., & Miller, J. A. (2018). Forecasting traffic flow: Short term, long term, and when it rains. In F. Y. L. Chin, C. L. P. Chen, L. Khan, K. Lee, & L.-J. Zhang (Eds.), *BigData 2018* (pp. 57-71). Springer International Publishing.

Rigopoulos, G. (2022). Univariate time series forecasting using K-Nearest Neighbors algorithm: A case for GDP. *International Journal of Scientific Research and Management (IJSRM)*, *10*, 3807–3815.

Rosário, A., & Dias, J. (2023). How has data-driven marketing evolved: Challenges and opportunities with emerging technologies. *International Journal of Information Management Data Insights*, *4*, 1-15.

Rožanec, J., Kažič, B., Škrjanc, M., Fortuna, B., & Mladenić, D. (2021). Automotive OEM demand forecasting: A comparative study of forecasting algorithms and strategies. *Applied Sciences*, *11*, 1-19.

Sajawal, M., Usman, S., Alshaikh, H., Hayat, A., & Ashraf, M. (2023). Predictive analysis of retail sales forecasting using machine learning techniques. *Lahore Garrison University Research Journal of Computer Science and Information Technology*, *6*, 23-33.

Sarwar, S., Aziz, G., & Kumar Tiwari, A. (2023). Implication of machine learning techniques to forecast the electricity price and carbon emission: Evidence from a hot region. *Geoscience Frontiers*, *15*, 1-13.

ScanmarQED. (n.d.-a). *Fast and transparent insights into your marketing activities.* Retrieved from `https://www.scanmarqed.com/marketing-mix-modeling/strataqed`

ScanmarQED. (n.d.-b). *Scanmarqed certified modeling professional - course content.*

Seraj, H., Bahadori-Jahromi, A., & Amirkhani, S. (2024). Developing a data-driven AI model to enhance energy efficiency in UK residential buildings. *Sustainability*, *16*, 3151-3167.

Setiawan, B., Rukmana, D., & Mahyuddin. (2020). Strategies for strengthening various models of cocoa marketing partnerships among farmers in the Polewali Mandar regency, Western Sulawesi province. *IOP Conference Series: Earth and Environmental Science*, *473*, 1-5.

Singh, A., & Srivastava, S. (2020). Model Performance Evaluation: Sales Prediction. In U. Batra, N. Roy, & B. Panda (Eds.), *Data Science and Analytics* (pp. 24-37). Springer Singapore.

Tangirala, S. (2020). Evaluating the impact of gini index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, *11*, 612-619.

Thu, L., & Leon-Gonzalez, R. (2021). Forecasting macroeconomic variables in emerging economies. *Journal of Asian Economics*, *77*, 1-22.

Tierney, G., Hellmayr, C., Barkimer, G., & West, M. (2023). Multivariate Bayesian dynamic modeling for causal prediction.
(Submitted for publication. arXiv:2302.03200)

U.S. Bureau of Economic Analysis. (2024a). *Personal Consumption Expenditures [PCE].* Retrieved from `https://fred.stlouisfed.org/series/PCE`

U.S. Bureau of Labor Statistics. (2024b). *Unemployment rate in Kusilvak Census area, ak [laucn021580000000003].* Retrieved from `https://fred.stlouisfed.org/series/LAUCN021580000000003`

van Greunen, J., & Heymans, A. (2023). Determining the impact of different forms of stationarity on financial time series analysis. In P. W. Buys & M. Oberholzer (Eds.), *Business Research : An Illustrative Guide to Practical Methodological Applications in Selected Case Studies* (pp. 61-76). Singapore: Springer Nature Singapore.

Wang, C. H., & Gu, Y. W. (2022). Sales forecasting, market analysis, and performance assessment for us retail firms: A business analytics perspective. *Applied Sciences*, *12*, 8480-8498.

Wang, C. H., & Liu, C. C. (2022). Market competition, technology substitution, and collaborative forecasting for smartphone panels and supplier revenues. *Computers Industrial Engineering*, *169*, 1-9.

Wang, Z., Zhu, Z., & Yu, C. (2023). Variable selection in macroeconomic forecasting with many predictors. *Econometrics and Statistics*. (in press)

Wichitaksorn, N. (2022). Analyzing and forecasting Thai macroeconomic data using mixed-frequency approach. *Journal of Asian Economics*, *78*, 1-19.

Yang, Q., He, K., Zheng, L., Wu, C., Yu, Y., & Zou, Y. (2023). Forecasting crude oil futures prices using Extreme Gradient Boosting. *Procedia Computer Science*, *221*, 920-926.

Yankovoy, R., Kulish, D., Melnyk, V., Churkina, I., Shurpa, S., & Pidkaminny, I. (2023). Formation of the international marketing strategy of domestic enterprises under the conditions of increased financial risks. *Financial and Credit Activity Problems of Theory and Practice*, *4*, 466-479.

Yoon, J. (2021). Forecasting of real GDP growth using machine learning models: Gradient boosting and random forest approach. *Computational Economics*, *57*, 247–265.

Zhang, Q., Ni, H., & Xu, H. (2023). Nowcasting Chinese GDP in a data-rich environment: Lessons from machine learning algorithms. *Economic Modelling*, *122*, 1-15.

Zhao, L.-T., Zheng, Z.-Y., & Wei, Y.-M. (2023). Forecasting oil inventory changes with Google Trends: A hybrid wavelet decomposer and ARDL-SVR ensemble model. *Energy Economics*, *120*, 1-20.

Appendices

A Tools used

During this research, we used a variety of tools and services:

1. **Python:** We employ Python for algorithm development. The algorithm, divided into different classes and combined in the main script, uses various Python packages including Pandas, NumPy, and Statsmodels.

2. **Strata, RStudio:** These tools were instrumental in confirming the unit root of the dataset in Python. We implemented unit root tests for aggregated, time-series, panel, and raw data, employing the ADF and Levin-Lin-Chu panel unit root tests.

3. **Overleaf:** Overleaf was used for report creation.

4. **PowerBI:** PowerBI facilitated the creation of our dashboard, incorporating dependencies among tables to streamline the process.

5. **ChatGPT, Grammarly:** These tools were used for reviewing and suggesting improvements to academic writing language. Initially, we wrote the content ourselves, thereafter reviewing, and refining the text multiple times. Lastly, we used ChatGPT and Grammarly for improvements.

6. **ScienceDirect, Scopus, GoogleScholar:** These platforms were used for sourcing articles relevant to our research.

After using these tools and services, I thoroughly reviewed and edited the content as needed, taking full responsibility for the integrity of the work.

B Variable findings

1. We exclude two identical variables, $Direct\ Mail\_Postcard\_Impressions$, and $Direct\ Mail\_Postcard\_Total\_Impressions$, and retained $Direct\ Mail\_Total\_impressions$.

2. We calculate the total impressions for the *display* variable as the sum of three subcomponents: $display\_TTD\_grand\_opening\_impressions$, $display\_TTD\_promo\_impressions$ and $TTD\_non\_promo\_impressions$.

3. We derive the variable $search\_Google\_local\_total\_impressions$ by adding $search\_Google\_local\_impressions$ and $search\_Google\_local\_promo\_impressions$.

4. The variable $HoursActual$ is the sum of $HoursOT$ and $HoursRegular$.

5. We unify the variable $PlayFly\_In-stadium\_impressionsvariable$ for both 2022 and 2023 into a single variable for our research.

6. The variable $search\_Google\_brand\_total\_clicks$ originally split into two variables for 2022 and 2023; however, we decide to remove these split variables.

7. The variable $search\_Google\_conq\_total\_clicks$ was aggregated by summing $search\_Google\_conq\_clicks$ and $search\_Google\_conq\_promo\_clicks$.

8. The variables $Search\_Google\_discovery\_clicks$ and $Search\_Google\_discovery\_total\_clicks$ contain the same values, and we retained the former.

9. The variable $search\_Google\_NB\_total\_clicks$ was obtained by summing $search\_Google\_NB\_clicks$ and $search\_Google\_NB\_Promo\_clicks$.

10. The variable $search\_google\_total\_clicks$ was calculated as the sum of various Google search-related click variables.

11. The variable $print\_impressions$ was computed by summing impressions from different print sources in specific cities (City4, City1, City2, City3, City5).

12. The variable $Audio\_total\_impressions$ was deemed unnecessary, and we chose to exclude it, given that other impressions were all zero.

13. The variable $radio\_total\_GRPs$ was obtained by summing $radio\_TTWN\_GRPs$ and $radio\_terrestrial\_GRPs$.

14. Similar to the variables $search\_Google\_brand\_total\_clicks$ and $Social\_Facebook\_non - promo\_impressions$, we remove the split of $Social\_Facebook\_non - promo\_impressions$ into 2022 and 2023.

15. The variable $social\_FB\_promo\_n - promo\_only\_impressions$ was derived by summing $social\_Facebook\_non - promo\_impressions$ and $social\_Facebook\_promo\_impressions$.

16. The variable $social\_FBIG\_total\_impressions$ was calculated as the sum of various social media impression variables.

17. The variable $social\_Tiktok\_total\_impressions$ was obtained by summing impressions from different TikTok sources.

18. The variable $social\_total\_impressions$ was calculated as the sum of $social\_Tiktok\_total\_impressions$ and $social\_FBIG\_total\_impressions$.

19. Similar to $search\_Google\_brand\_total\_clicks$ and $Social\_Facebook\_non - promo\_impressions$, we decide to remove the split of 2022 and 2023 for $CTV\_TTD\_Non - Promo\_impressions$.

20. The variable $CTV\_other\_impressions$ contains zeros only, leading us to exclude this variable.

21. The variable $CTV\_total\_impressions$ was derived by summing $CTV\_TTD\_promo\_impressions$ and $CTV\_TTD\_non - promo\_impressions$.

22. The variable $TV\_cable\_total\_GRPs$ was calculated as the sum of various cable TV GRP components.

23. The variable $TV\_GSTV\_impressions$ consist solely of zeros, leading to its exclusion.

24. $TV\_linear\_total\_GRPs$ is obtained by summing various linear TV GRP components.

25. $TV\_Playfly\_total\_GRPs$ was calcualted as the sum of specific Playfly TV GRP components.

26. We observe that $Video\_TTD\_promo\_impressions$ and $Video\_TTD\_total\_impressions$ contain identical values, leading us to exclude the latter.

27. The variable $video\_YouTube\_total\_impressions$ was derived as the sum of various YouTube video impressions variables.

28. The variable $video\_total\_impressions$ was calculated as the sum of $video\_YouTube\_total\_impressions$ and $video\_TTD\_promo\_impressions$.

29. The variable $Search\_Google\_Brand\_Total\_Clicks$ was obtained as the sum of $Search\_Google\__Brand\_Total\_Clicks$ and $Search\_Google\__Brand\_Promo\_Clicks$.

30. The variable $OOH\_CO-DEV\_Impressions$ was calculated as the sum of two OOH CO- DEV impressions components.

31. The variable $OOH\_GRANDOPENING\_Impressions$ was derived by summing two OOH GRAND OPENING impressions components.

32. We remove the duplicate variable $TV\_Linear\_Campaign5\_GRPs$ as it is identical to $TV\_Linear\_Campaign7\_GRPs$.

33. We exclude two sales patterns of *Quadrant*.

C Correlated variables

Variable 1	**Variable 2**	**Correlation**
Display_TTD_Non-Promo _Impressions	Social_Facebook_Non-Promo _Impressions	0.61
Display_TTD_Non-Promo _Impressions	Video_Youtube_Non-Promo _Impressions	0.62
Display_TTD_Promo_Impressions	Audio_TTD_Promo_Impressions	0.61
Display_TTD_Promo_Impressions	Social_Facebook_Promo_Impressions	0.74
Display_TTD_Promo_Impressions	Video_TTD_Promo_Impressions	0.85
CRM_Email_Total_Sent	Search_Google_Brand_Clicks	0.72
CRM_Email_Total_Sent	Search_Google_NB_Clicks	0.63
CRM_Email_Total_Sent	HoursRegular	0.63
OOH_DIGITAL BULLETIN_CO-DEV_Impressions	OOH_STATIC BULLETIN_CO-DEV_Impressions	0.67
Search_Google_Brand_Clicks	Search_Google_NB_Clicks	0.88
Search_Google_Brand_Clicks	Social_Facebook_Non-Promo _Impressions	0.75
Search_Google_Brand_Clicks	Video_Youtube_Non-Promo _Impressions	0.71

Search_Google_Brand_Promo_Clicks	Search_Google_Conq_Promo_Clicks	0.80
Search_Google_Brand_Promo_Clicks	Search_Google_NB_Promo_Clicks	0.89
Search_Google_Conq_Clicks	Search_Google_NB_Clicks	0.69
Search_Google_Conq_Clicks	Social_Facebook_Non-Promo _Impressions	0.64
Search_Google_Conq_Promo_Clicks	Search_Google_NB_Promo_Clicks	0.83
Search_Google_NB_Clicks	Social_Facebook_Non-Promo _Impressions	0.76
Search_Google_NB_Clicks	Video_Youtube_Non-Promo _Impressions	0.69
Audio_TTD_Promo_Impressions	Video_TTD_Promo_Impressions	0.72
Social_Facebook_Non-Promo _Impressions	CTV_TTD_Non-Promo _Impressions	0.62
Social_Facebook_Non-Promo _Impressions	Video_Youtube_Non-Promo _Impressions	0.69
Social_Facebook_Other_Impressions	Social_Tiktok_Other_Impressions	0.60
Social_Facebook_Promo_Impressions	CTV_TTD_Promo_Impressions	0.64
Social_Facebook_Promo_Impressions	Video_TTD_Promo_Impressions	0.75
TV_Cable_Campaign1_GRPs	TV_Cable_Campaign5_GRPs	0.61
TV_Cable_Campaign1_GRPs	TV_Cable_Campaign6_GRPs	0.64
TV_Cable_Campaign1_RAMP _GRPs	TV_Cable_Campaign6_GRPs	0.68
TV_Cable_Campaign4_GRPs	TV_Cable_Campaign6_GRPs	0.77
TV_Cable_Other_GRPs	TV_Linear_GRPs	0.83
TV_Cable_Other_GRPs	TV_Linear_Campaign4_GRPs	0.71
HoursOT	HoursRegular	0.94

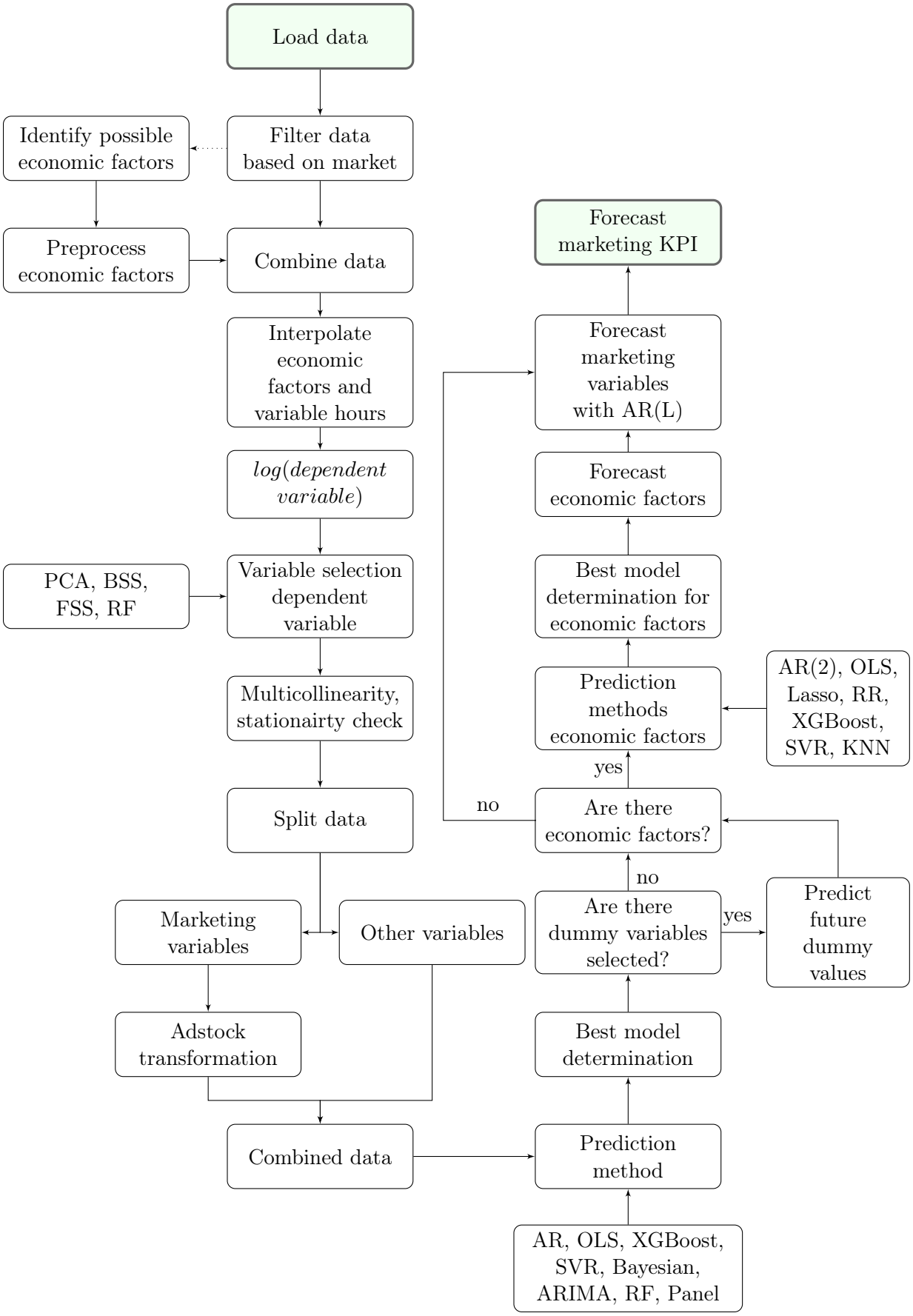
D Marketing variables

The marketing variables in our data set include:

1. Direct_Mail_Total_Impressions
2. Display_TTD_Promo_Impressions
3. Display_TTD_Non-Promo _Impressions
4. Display_TTD_GRAND_OPENING _Impressions
5. Search_Google_Local_Impressions
6. Search_Google_Local_Promo _Impressions
7. CRM_Email_Total_Sent
8. OOH_Impressions
9. OOH_DIGITAL_BULLETIN _GRAND_OPENING _Impressions
10. OOH_DIGITAL_GRAND_OPENING _Impressions
11. OOH_DIGITAL_BULLETIN_CO-DEV_Impressions
12. OOH_STATIC_BULLETIN_CO-DEV_Impressions
13. OOH_STATIC_BULLETIN_GRAND _OPENING_Impressions
14. OOH_STATIC_BULLETIN_FAST _OIL_CHANGES _RAMP_Impressions
15. OOH_RAMP_Impressions
16. OOH_Total_Impressions
17. PlayFly_In-stadium_Impressions

18. Search_Google_Brand_Clicks
19. Search_Google_Brand_Promo_Clicks
20. Search_Google_Conq_Clicks
21. Search_Google_Conq_Promo_Clicks
22. Search_Google_Discovery_Clicks
23. Search_Google_NB_Clicks
24. Search_Google_NB_Promo_Clicks
25. Print_City1_Impressions
26. Print_City2_Impressions
27. Print_City3_Impressions
28. Print_City4_Impressions
29. Print_City5_Impressions
30. Audio_TTD_Promo_Impressions
31. Radio_TTWN_GRPs
32. Radio_Terrestrial_GRPs
33. Social_Facebook_Promo_Impressions
34. Social_Facebook_Non-Promo _Impressions
35. Social_Facebook_Other_Impressions
36. Social_Instagram_Promo_Impressions
37. Social_FBIG_GRAND_OPENING _Impressions
38. Social_FBIG_Other_Impressions
39. Social_Tiktok_Non-Promo _Impressions
40. Social_Tiktok_Other_Impressions
41. Social_Tiktok_Promo_Impressions
42. CTV_TTD_Non-Promo_Impressions
43. CTV_TTD_Promo_Impressions
44. TV_Cable_Campaign1_GRPs
45. TV_Cable_Campaign1_RAMP_GRPs
46. TV_Cable_Campaign2_GRPs
47. TV_Cable_Campaign2 _RAMP_GRPs
48. TV_Cable_Campaign3_GRPs
49. TV_Cable_Campaign4_GRPs
50. TV_Cable_Campaign5_GRPs
51. TV_Cable_Campaign6_GRPs
52. TV_Cable_Campaign6_RAMP_GRPs
53. TV_Cable_Other_GRPs
54. TV_Cable_Campaign7_GRPs
55. TV_Linear_Campaign1_GRPs
56. TV_Linear_Campaign2_GRPs
57. TV_Linear_GRPs
58. TV_Linear_Campaign4_GRPs
59. TV_Linear_Campaign6_GRPs
60. TV_Linear_Campaign7_GRPs
61. TV_Playfly_GRPs
62. TV_Playfly_Campaign5_GRPs
63. TV_Total_GRPs
64. Video_TTD_Promo_Impressions
65. Video_Youtube_Non-Promo_Impressions
66. Video_Youtube_Other_Impressions
67. Video_Youtube_Promo_Impressions
68. Video_Youtube_GRAND_OPENING _Impressions

E Detailed process chart of the algorithm



F Variable Selection Algorithms

F.1 Forward stepwise selection

Algorithm 1 Pseudocode for Forward stepwise selection.

procedure FORWARD SELECTION($data\_X$, y)
    $X\_train\_forward, \_, y\_train\_forward, \_ \leftarrow train\_test\_split(data\_X, y, test\_size = 0.3, random\_state = 0)$
    $forward\_feature\_selection \leftarrow$ Initialise Sequential Feature Selector with Linear Regression
    $forward\_feature\_selection.fit(X\_train\_forward, y\_train\_forward)$
    $selected\_indices \leftarrow$ Get selected feature indices
    $forward\_variables \leftarrow$ Names of selected features
    return $forward\_variables$
end procedure

F.2 Backward stepwise selection

Algorithm 2 Pseudocode for Backward stepwise selection.

procedure BACKWARDSELECTION($data\_X$, y)
    $\_, X\_test\_backward, \_, y\_test\_backward \leftarrow train\_test\_split(data\_X, y, test\_size = 0.3, random\_state = 0)$
    $backward\_feature\_selection \leftarrow$ Initialise Sequential Feature Selector with Linear Regression
    $backward\_feature\_selection.fit(data\_X, y)$
    $selected\_indices \leftarrow$ Get selected feature indices
    $backward\_variables \leftarrow$ Names of selected features
    return $backward\_variables$
end procedure

F.3 PCA

Algorithm 3 Pseudocode for PCA variable selection.

procedure PCA SELECTION($data\_X$)
    $X\_scaled \leftarrow$ Standardise $data\_X$ using $preprocessing.scale$
    $pca \leftarrow$ Initialise PCA with $n\_components = 2$
    $pca.fit\_transform(X\_scaled)$
    $components\_df \leftarrow$ DataFrame of PCA components
    $variable\_importance \leftarrow$ Sum of squared loadings for each variable
    $sorted\_variables \leftarrow$ Sort $variable\_importance$ in descending order
    $pca\_variables \leftarrow$ Top 20 variables from $sorted\_variables$
    return $pca\_variables$
end procedure

F.4 Random Forest

Algorithm 4 Pseudocode for Random Forest variable selection.

procedure RANDOM FOREST SELECTION($data\_X$, y)
    $X\_train\_rf, \_, y\_train\_rf, \_ \leftarrow train\_test\_split(data\_X, y, test\_size = 0.3, random\_state = 0)$
    $scaler \leftarrow$ Initialise Standard Scaler
    $X\_train\_scaled \leftarrow scaler.fit\_transform(X\_train\_rf)$
    $rf \leftarrow$ Initialise Random Forest Regressor
    $param\_dist \leftarrow$ Define hyperparameter distributions
    $random\_search \leftarrow$ Initialise RandomisedSearchCV with Random Forest
    $random\_search.fit(X\_train\_scaled, y\_train\_rf)$
    $best\_params \leftarrow$ Get best hyperparameters from random search
    $best\_rf\_model \leftarrow$ Initialise Random Forest with best hyperparameters
    $best\_rf\_model.fit(X\_train\_scaled, y\_train\_rf)$
    $importance \leftarrow$ Get feature importances from the best model
    $threshold\_percentile \leftarrow$ Set percentile threshold
    $threshold \leftarrow$ Calculate threshold based on percentile
    $selected\_indices \leftarrow$ Get indices of features with importance higher than threshold
    $rf\_variables \leftarrow$ Names of selected features
    return $rf\_variables$
end procedure

G SVR calculations economic factor forecast

G.1 Lagrangian

We define the Lagrangian for SVR with economic factors $z_{j,c,t}$, the dependent variable, and $\tilde{Z}_{j,c,t}$, the independent variables, below. j is the specific economic factor, c represents the city, and t are the time points with T observations.

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{w}||^2 + R\sum_{t=1}^{T}(\xi_{j,c,t} + \zeta_{j,c,t}) \\ - \sum_{t=1}^{T}\alpha_{j,c,t}((\mathbf{w}^T\tilde{\mathbf{Z}}_{j,c,t} + b) - z_{j,c,t} - \varepsilon_t - \xi_{j,c,t}) \\ - \sum_{t=1}^{T}\beta_{j,c,t}(z_{j,c,t} - (\mathbf{w}^T\tilde{\mathbf{Z}}_{j,c,t} + b) - \varepsilon_t - \zeta_{j,c,t})\end{aligned}$$

G.2 Partial derivatives

Compute the partial derivatives of the Lagrangian with respect to the variables $\mathbf{w}$, and b, and set them equal to zero to obtain the conditions for optimality.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{t=1}^{T}(\alpha_{j,c,t} - \beta_{j,c,t})\tilde{\mathbf{Z}}_{j,c,t} = 0$$
$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{t=1}^{T}(\alpha_{j,c,t} - \beta_{j,c,t}) = 0$$

G.3 Express w in terms of Lagrange Multipliers

We rewrite the partial derivatives of $\mathbf{w}$ in terms of the Lagrange multipliers $\alpha_{j,c,t}$ and $\beta_{j,c,t}$:

$$\mathbf{w} = \sum_{t=1}^{T}(\alpha_{j,c,t} - \beta_{j,c,t})\tilde{\boldsymbol{Z}}_{j,c,t}$$

G.4 Apply KKT conditions

According to Balasundaram et al. (2014), the Karush-Kuhn-Tucker (KKT) conditions for optimality are:

1. Stationarity:
$$\frac{\partial L}{\partial w} = 0$$
$$\frac{\partial L}{\partial b} = 0$$
2. Primal feasibility:
$$z_{j,c,t} - (w\tilde{\mathbf{Z}}_{j,c,t} + b) \leq \varepsilon_t + \xi_{j,c,t}$$
$$(w\tilde{\mathbf{Z}}_{j,c,t} + b) - z_{j,c,t} \leq \varepsilon_t + \zeta_{j,c,t}$$
$$\xi_{j,c,t}, \zeta_{j,c,t} \geq 0$$
3. Dual feasibility:
$$\alpha_{j,c,t} \geq 0$$
$$\beta_{j,c,t} \geq 0$$
4. Complementary slackness:
$$\alpha_{j,c,t}(\varepsilon_t + \xi_{j,c,t}) = 0$$
$$\beta_{j,c,t}(\varepsilon_t + \zeta_{j,c,t}) = 0$$

If $\alpha_{j,c,t} > 0$, then $\xi_{j,c,t} = 0$ (since $\xi_{j,c,t} \geq 0$). Similarly, if $\beta_{j,c,t} > 0$, then $\zeta_{j,c,t} = 0$.

So, for $\alpha_{j,c,t} > 0$, $\xi_{j,c,t}$ must be zero, indicating that the corresponding data point lies on or within the margin. Similarly, for $\beta_{j,c,t} > 0$, $\zeta_{j,c,t}$ must be zero, indicating that the corresponding data point lies on or within the margin.

This completes the formulation of the SVR problem with complementary slackness conditions.

H SVR calculations marketing forecast

We define $y_{c,t} = \beta X_{c,t} + \gamma Z_{c,t} + b = w^T \tilde{X}_{c,t} + b$, where $w = \begin{bmatrix} \beta \\ \gamma \end{bmatrix}$ and $\tilde{X}_{c,t} = \begin{bmatrix} X_{c,t} \\ Z_{c,t} \end{bmatrix}$, with marketing variables $X_{c,t}$ (n variables) and economic factors $Z_{c,t}$ (m variables), for city c at time t. $y_{c,t}$ is our dependent marketing KPI.

H.1 Lagrangian

For the Lagrangian formula we define t as the time points with T observations.

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \frac{1}{2}||\mathbf{w}||^2 + R \sum_{t=1}^{T} (\xi_{c,t} + \zeta_{c,t}) \\ - \sum_{t=1}^{T} \alpha_{c,t}((\mathbf{w}^T \tilde{\mathbf{X}}_{c,t} + b) - y_{c,t} - \varepsilon_t - \xi_{c,t}) \\ - \sum_{t=1}^{T} \theta_{c,t}(y_{c,t} - (\mathbf{w}^T \tilde{\mathbf{X}}_{c,t} + b) - \varepsilon_t - \zeta_{c,t}) \end{aligned}$$

H.2 Partial derivatives

Compute the partial derivatives of the Lagrangian with respect to the variables $\mathbf{w}$, and b, and set them equal to zero to obtain the conditions for optimality.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{t=1}^{T} (\alpha_{c,t} - \theta_{c,t}) \tilde{\mathbf{X}}_{c,t} = 0 \\ \frac{\partial \mathcal{L}}{\partial b} &= - \sum_{t=1}^{T} (\alpha_{c,t} - \theta_{c,t}) = 0 \end{aligned}$$

H.3 Express w in terms of Lagrange Multipliers

We rewrite the partial derivatives of $\mathbf{w}$ in terms of the Lagrange multipliers $\alpha_{c,t}$ and $\theta_{c,t}$:

$$\mathbf{w} = \sum_{t=1}^{T} (\alpha_{c,t} - \theta_{c,t}) \tilde{\boldsymbol{X}}_{c,t}$$

So $\begin{bmatrix} \beta \\ \gamma \end{bmatrix} = \sum_{t=1}^{T} (\alpha_{c,t} - \theta_{c,t}) \begin{bmatrix} X_{c,t} \\ Z_{c,t} \end{bmatrix}$.

H.4 Apply KKT conditions

According to Balasundaram et al. (2014), the Karush-Kuhn-Tucker (KKT) conditions for optimality are:

1. Stationarity:

$$\begin{aligned} \frac{\partial L}{\partial w} &= 0 \\ \frac{\partial L}{\partial b} &= 0 \end{aligned}$$

2. Primal feasibility:

$$y_{c,t} - (w\tilde{\boldsymbol{X}}_{c,t} + b) \leq \varepsilon_t + \xi_{c,t}$$

$$(w\tilde{\boldsymbol{X}}_{c,t} + b) - y_{c,t} \leq \varepsilon_t + \zeta_{c,t}$$

$$\xi_{c,t}, \zeta_{c,t} \geq 0$$

3. Dual feasibility:

$$\alpha_{c,t} \geq 0$$

$$\theta_{c,t} \geq 0$$

4. Complementary slackness:

$$\alpha_{c,t}(\varepsilon_t + \xi_{c,t}) = 0$$

$$\theta_{c,t}(\varepsilon_t + \zeta_{c,t}) = 0$$

If $\alpha_{c,t} > 0$, then $\xi_{c,t} = 0$ (since $\xi_{c,t} \geq 0$). Similarly, if $\theta_{c,t} > 0$, then $\zeta_{c,t} = 0$.

So, for $\alpha_{c,t} > 0$, $\xi_{c,t}$ must be zero, indicating that the corresponding time point lies on or within the margin. Similarly, for $\theta_{c,t} > 0$, $\zeta_{c,t}$ must be zero, indicating that the corresponding time point lies on or within the margin.

This completes the formulation of the SVR problem with complementary slackness conditions.

I Pseudocode

Within in our code we have different classes to create the AutoForecaster, main is the main menu where we call the classes in the right order in order to come up with a forecasting system.

I.1 Main

Algorithm 5 Pseudocode for Main.

1: **procedure** MAIN(*data*, *economic_factors_data*, dependent variable)
2: **Input:** Main data set *data*, the data of economic factors *economic_factors_data*, and the name of the dependent variable.
3: **Output:** Forecast values of the dependent variables.
4: **Description:** This algorithm automatically forecasts the dependent variable based on the provided data set and the data set of the economic factors.

5: read *data*
6: filter on desired market
7: *eco_preprocessing* ← initialise `Economic factor preprocessing` ▷ Algorithm 6
8: *unemployment_data* ← `eco_preprocessing.state_monthly(eco_data_1, data_develop)`
9: *merged_data* ← `eco_preprocessing.monthly(eco_data_2, unemployment_data)`
10: *processor* ← initialise `DataPreprocessor` ▷ Algorithm 7
11: Exclude dummies, non essential variables
12: $y = dependent\_variable$ ▷ Number of units services
13: Log transformation: $log[y + 1]$

```
14:  for method in variable selection methods do
15:      feature_selector ← initialise FeatureSelector(X, y)          ▷ Algorithm 9
16:      if method == 'pca' then
17:          selected_variables ← feature_selector.pca_selection(num_variables=10)
18:      else if method == 'forward' then
19:          selected_variables ← feature_selector.forward_selection()
20:      else if method == 'backward' then
21:          selected_variables ← feature_selector.backward_selection()
22:      else if method == 'random_forest' then
23:          selected_variables ← feature_selector.random_forest_selection()
24:      else
25:          raise ValueError("Unknown variable selection method: " + method)
26:      end if

27:      multicol_vars ← ModelEvaluator.check_multicollinearity(
             data, selected_variables)                                ▷ Algorithm 10
28:      variable_data ← X[multicol_vars]
29:      ad_vars ← processor.create_marketing_variables_df(
             variable_data, method, economic_factors_names)
30:      ad_data ← processor.advertising_transformation(ad_vars)

31:      checker = DummyVariableChecker(data)                         ▷ Algorithm 11
32:      dummies = checker.process_dummy_variables()

33:      models ← Prediction_models(ad_data, data, dummies, dependent_variable,
             ad_vars)                                                 ▷ Algorithm 12
34:      results_OLS, OLS_model, OLS_data, OLS_all_data ← models.
             process_forecast(OLS)
35:      results_XGBoost, XGBoost_model, XGBoost_data, XGBoost_all_data ←
             models.process_forecast(XGBoost)
36:      results_ARIMA, ARIMA_model, ARIMA_data, ARIMA_all_data ←
             models.process_forecast(ARIMA)
37:      results_ARF, ARF_model, ARF_data, ARF_all_data ← models.
             process_forecast(ARF)
38:      results_panel, panel_model, panel_data, panel_all_data ← models.
             process_forecast(panel)
39:      results_SVR, SVR_model, SVR_data, SVR_all_data ← models.
             process_forecast(SVR)
40:      results_AR, AR_model, AR_data, AR_all_data ← models.
             process_forecast(AR)
41:      results_bayesian, bayesian_model, bayesian_data, bayesian_all_data ←
             models.process_forecast(bayesian)
42:      forecast ← Forecast(results_XGBoost, results_OLS, results_ARF,
             results_SVR)
43:      best_model, best_model_data, best_model_name ← forecast.main(
             OLS_model, XGBoost_model, ARF_model, SVR_model,
             ARIMA_model, panel_model, AR_model, bayesian_model,
             OLS_data, XGBoost_data, ARF_data, SVR_data,
             ARIMA_data, panel_data, AR_data, bayesian_data)
44:      non_marketing_variables ← [col for col in best_model_data.columns if col
             in economic_names]
```

```
        dummy_names ← checker.select_dummy_variables(best_model_data)
        if not len(dummy_names) == 0 then
            dummy_forecast_data ← checker.filter_dummies(dummy_data)
        end if

        forecast_horizon ← [4, 8, 12]

        for h in forecast_horizon do
            if economic_factors_names then
                non_marketing_df ← data[economic_factors_names] ▷ Algorithm 13
                economic_factors ← Economic_Factors_models(non_marketing_df)
                eco_data, eco_forecast_data, lower, upper ←
                economic_factors.forecast_economic(non_marketing_variables,
                non_marketing_df, horizon=h)
            end if
            for name in marketing_df do
                forecast_data ← forecast.forecast_data_variables(
                data=marketing_df, variable_name=name, h=h,
                 noise_scale=noise_scale)
            end for
            Merge marketing_df, dummy_forecast_data, non_marketing_df
            forecast_values ← forecast.predict_model(best_model_name,
              best_model, h, merged_df, check_order)
        end for
    end for
    return forecast values per method per horizon
end procedure
```

I.2 Economic Factor Pseudocode

We apply basic functions for our models, for example *AR_model* refers to the basic AR model. The general algorithm remains consistent across all models.

Algorithm 6 Pseudocode for Economic Factor.

```
procedure ECONOMIC_FACTORS_MODELS(data)
    Input: Economic factor data set data.
    Output: Forecast values of the economic factors.
    Description: This algorithm automatically forecasts the economic factors based
            on the provided data set and the best model selection.

    self.data ← data
end procedure

procedure CHECK_MODEL(variable_name, var_data, h)
    Input: Name of the economic factor, its data set, and forecast horizon.
    Output: Best prediction method.

    variable ← copy var_data
    data_AR, AR_model ← AR_model(variable, variable_name)
    data_OLS, OLS_model ← OLS(variable, variable_name)
    data_XGBoost, XGBoost_model ← XGBoost(variable, variable_name)
```

```
    data_SVR, SVR_model ← SVR(variable, variable_name)
    data_KNN, KNN_model ← KNN(variable, variable_name)
    data_RR, RR_model ← RR(variable, variable_name)
    data_Lasso, Lasso_model ← Lasso(variable, variable_name)

    data_models ← join data from models
    _, y_test, _, _ ← cutoff_test_train(variable, variable_name, weeks=52)

    model_evaluator ← ModelEvaluator(target_column=variable_name,
            prediction_column=prediction)

    for model_name in data_model.columns do
        MSE, RMSE, MAE, BIC, AIC ← model_evaluator.evaluate_model_
            performance_eco_factor(data_models[model_name, variable_name]
        results ← [MSE, RMSE, MAE, BIC, AIC]
    end for

    best_model_name ← model_evaluator.best_model_determination(results)
    prediction function contains all models
    best_model ← prediction_functions[best_model_name]
    forecast_data ← predict_model(variable, variable_name,
            best_model_name, best_model, h, noise_scale)

    return data_models, results_df, best_model_name, forecast_data
end procedure

procedure PREDICT_MODEL(data, variable_name, best_model_name, model, h)
    Input: Economic factor data set, name, best model name, the model, and horizon.
    Output: Predicted values of the economic factor.

    df ← dataframe of data
    last_value ← last date of df

    create new rows with random noise in forecast values that are lagged
    y_train, y_test, test, train ← cutoff_test_train(df, variable_name,
            weeks=52+h)
    lagged_X_train, lagged_X_test ← create a lagged dataframe with y_train,
            y_test, variable_name

    prediction ← model.predict()
    predicted_values[best_model_name] = prediction

    return predicted_values
end procedure

procedure CUTOFF_TEST_TRAIN(data, variable_name, weeks)
    Input: Economic factor data set, variable name and weeks for the cutoff date.
    Output: A training and testing set.

    cutoff_date ← max date - weeks
    model_evaluator ← ModelEvaluator(target_column=variable_name,
```

```
                prediction_column='prediction')
48:     _, _, y_train, y_test, test, train ← model_evaluator.train_test_split_model_panel(data,
                cutoff_date=cutoff_date)

49:     return y_train, y_test, test, train
50: end procedure

51: procedure FORECAST_ECONOMIC(economic_names, data_test, horizon)
52:     Input: Economic factor data set, variable name and horizon to forecast.
53:     Output: Foecasted values of the economic factor.

54:     forecast_eco = pd.DataFrame()
55:     for name in economic_names do
56:         eco_data, results, best_model, forecast_data ← check_model(name, data,
                horizon)
57:         if forecast_eco.empty then
58:             forecast_eco ← forecast_data
59:         else
60:             if forecast_data is not empty then
61:                 forecast_eco ← pd.merge(forecast_eco, forecast_data)
62:             end if
63:         end if
64:     end for

65:     return eco_data, forecast_eco
66: end procedure
```

I.3 DataPreprocessing Pseudocode

Algorithm 7 Pseudocode for DataPreprocessing.

```
1: procedure DataProcessor(data)
2:     Input: The original data set.
3:     Output: Transformed data and transformation of the marketing variables.
4:     Description: This algorithm transforms the data set by calling the
           Transformation algorithm and it transforms the marketing variables by
           apply adstock and lags.

5:     self.data ← data
6: end procedure

7: procedure PREPROCESS_DATA
8:     Input:
9:     Output: Transformed data, with interpolated values.

10:    data_transformer ← Transformation(self.data)                ▷ Algorithm 8
11:    self.data ← data_transformer.convert_string_columns()
12:    columns_to_drop ← columns to drop mentioned in Appendix B
13:    self.data ← drop columns_to_drop
14: end procedure

15: procedure INTERPOLATE_DATA(interpolated_values)
16:    Input: The variables that need interpolation.
```

```
    Output: Interpolated values.

    interpolator ← Transformation(self.data)                    ▷ Algorithm 8
    for value in interpolated_values do
        interpolator.add_interpolation_variable(value))
    end for
    interpolated_data ← interpolator.interpolate()
    for value in interpolated_values do
        self.data[value] ← interpolated_data[value]
    end for
end procedure

procedure ADVERTISING_TRANSFORMATION(data)
    Input: Marketing variable data.
    Output: Transformed marketing variables.

    cities ← unique cities
    adstock_dfs ← {}
    for city in cities do
        city_data ← select city data
        y ← data[dependent variable]
        X ← data.drop(columns=dependent variable)

        λ ← (i/9) * 0.95,   i = 0, 1, ..., 9    ▷ A value between 0 and 0.95 with steps of 0.1
        lags ← [0, ..., 8]                      ▷ A value between 0 and 8 with steps of 1
        city_adstock_dfs is a dictionary

        for col in X.columns do
            best_loss ← float(∞)
            best_params ← None
            for λ_adstock, lag_value in product (λ, lags) do
                X_values ← X[col].values
                X_adstock ← adstock(X_values, λ_adstock)
                X_transformed ← lag(X_adstock, lag_value)

                model ← LinearRegression()
                model.fit(X_transformed.reshape(-1, 1), y)

                y_pred ← model.predict(X_transformed.reshape(-1, 1))
                loss ← mean_squared_error(y, y_pred)

                if loss < best_loss then
                    best_loss ← loss
                    best_params ← (λ_adstock, lag_value)
                end if
            end for
            best_lam_adstock, best_lag ← best_params
            X_adstock ← adstock(X[col].values, best_lam_adstock)
            X_transformed ← lag(X_adstock, best_lag)
            city_adstock_dfs[col] = pd.DataFrame(X_transformed, columns=[col])
        end for
```

60: *adstock_dfs* ← combine adstock values, add city column
61: **end for**
62: *df* ← combine all adstock_dfs
63: **return** df
64: **end procedure**

65: **procedure** ADSTOCK(X, λ) ▷ Decay rate, λ
66: **Input:** Values and decay rate
67: **Output:** Adstock values.

68: $A \leftarrow \{\}$ ▷ Adstock, A
69: **for** t in $1, \ldots, T$ **do** ▷ Time t spanning from 1 till T
70: **if** $t = 0$ **then**
71: $A_t = X_t$
72: **else**
73: $A_t = X_t + \lambda * A_{t-1}$
74: **end if**
75: **end for**
76: **return** A
77: **end procedure**

78: **procedure** LAG(X, l) ▷ Lag, l
79: **Input:** Values and lag rate.
80: **Output:** Lagged values.

81: $$\text{X\_t} = \begin{cases} 0 & \text{for } t = 0, 1, \ldots, l-1 \\ X_{t-l} & \text{for } t \geq l \end{cases}$$
82: **return** X
83: **end procedure**

I.4 DataPreprocessing Tranformation

Algorithm 8 Pseudocode for Tranformation.

1: **procedure** TRANSFORMATION(*data*)
2: **Input:** Data that needs transformation.
3: **Output:** Interpolated data.
4: **Description:** This algorithm applies interpolation to the values.

5: *interpolation_variable* ← {}
6: *string_variable* ← {}
7: **end procedure**

8: **procedure** ADD_INTERPOLATION_VARIABLE(*variable*)
9: **Input:** Variable name that needs interpolation.
10: **Output:** List with variables that need interpolation.

11: `interpolation_variable.add(variable)` ▷ Fill set with interpolated values
12: **end procedure**

13: **procedure** INTERPOLATE
14: **Input:**

Output: Interpolated data.

interpolation_variables_list ← interpolation_variable
interpolated_data ← copy Date, City, interpolation_varibale_list
cities ← unique cities
for city in cities **do**
    city_data ← copy specific city data
    for variable in interpolation_variable **do**
        original_data, *df* ← select city data
        replace duplicates in df with NaN
        when value equal to zero replace it by NaN
        $y \leftarrow y_t + \frac{x - x_t}{x_{t+1} - x_t}(y_{t+1} - y_t)$ ▷ $t = 1, \ldots, T$

        nan_indices ← get indices of NaN values in df
        for index in nan_indices **do**
            original_value ← copy original value from original_data
            replace NaN values with the original value
        end for

        for index, row in df.iterrows() **do**
            data_row ← date
            mask ← find the row that corresponds to the date and city
            interpolated_data.loc[*mask*, *variable*] ← add the row of the variable
        end for
    end for
end for
return interpolated_data
end procedure

I.5 FeatureSelector Pseudocode

Algorithm 9 Pseudocode for FeatureSelector.

procedure FeatureSelector(data_X, y)
    Input: Data set *data_X* and target variable *y*
    Output: Selected features
    Description: This algorithm selects a subset of features from the input data set using PCA.
end procedure

procedure pca_selector(num_variables)
    Input: Number of components *num_components*
    Output: Selected features based on PCA

    Perform PCA on *data_X*
    Fit all variables in the PCA procedure
    Create the components
    Identify the variable importances
    pca_variables ← Select top *num_components* most important variables
    return *pca_variables*
end procedure

procedure forward_selector

Input:
Output: Selected features based on forward selection

Define forward selection with number of features to select 20
Fit all variables in the forward selection procedure
$forward\_variables \leftarrow$ Select top 20 variables most important variables
return $forward\_variables$
end procedure

procedure FORWARD_SELECTOR
Input:
Output: Selected features based on backward selection

Define backward selection with number of features to select 20
Fit all variables in the backward selection procedure
$backward\_variables \leftarrow$ Select top 20 variables most important variables
return $backward\_variables$
end procedure

procedure RANDOM_FOREST_SELECTION
Input:
Output: Selected features based on random forest selection

Define random forest feature selection
Fit all variables in the random forest feature selection procedure
$importance \leftarrow$ Extract feature importances
$threshold\_percentile \leftarrow 75$
$threshold \leftarrow$ np.percentile$(importance, threshold_percentile)$
$selected\_indices \leftarrow importance > threshold$
$rf\_variables \leftarrow X_{\text{train\_rf}}$.columns$[selected_indices]$
return $rf\_variables$
end procedure

I.6 ModelEvaluator Pseudocode

Algorithm 10 Pseudocode for ModelEvaluator.

procedure MODELEVALUATOR(target_column, prediction_column)
Input: Data set $target\_column$ and prediction data $prediction\_column$
Output: Performance measures and data split into train and test sets
Description: This algorithm calculates the performance measures to identify the best model, it also contributes to the splitting process to get a training and testing set.
end procedure

procedure EVALUATE_MODEL_PERFORMANCE($test$, $panel = False$)
Input: Data set called $test$ and panel determination, if true it is a panel data set
Output: Performance measures

$MSE \leftarrow$ `mean_squared_error(y_true=target_column, y_pred=prediction_column)`
$RMSE \leftarrow \sqrt{MSE}$
$MAE \leftarrow$ `mean_absolute_error(y_true=target_column,`

```
        y_pred=prediction_column)

    if panel then
        grouped_data ← group by city and date
    else
        grouped_data ← group by date
    end if

    n_samples, n_features ← test.shape
    k ← n_features
    n ← n_samples
    cov_matrix ← EmpiricalCovariance().fit(test[[self.target_column,
        self.prediction_column]]).covariance_
    log_likelihood ← −0.5 × n × log(det(cov_matrix))
    AIC ← 2 × k − 2 × log_likelihood                ▷ Akaike Information Criterion
    BIC ← k × log(n) − 2 × log_likelihood           ▷ Bayesian Information Criterion
    date_error ← Combine all performance metrics in one table
    return MSE, RMSE, MAE, MAPE, RMSPE, BIC, AIC, dates_error
end procedure

procedure TRAIN_TEST_SPLIT_MODEL(data, cutoff_date)
    Input: Data set called data and the cutoff date, to split the data set
    Output: A traning and testing set

    mask_train ← dates_array ≤ cutoff_date
    mask_test ← dates_array ≥ cutoff_date
    train ← copy the sliced data set from mask_train
    test ← copy the sliced data set from mask_test
    X_train, y_train ← copy train, divide dependent variable from train dataframe
    X_test, y_test ← copy train, divide dependent variable from test dataframe
    return X_train, X_test, y_train, y_test, test, train
end procedure

procedure BEST_MODEL_DETERMINATION(results)
    Input: Data set containing the results
    Output: Best model name

    weights ← {MSE : 0.3, RMSE : 0.3, MAE : 0.3, BIC : 0.05, AIC : 0.05}
    T ← {metric : ∑_{i=1}^{n} ∑_{j=1}^{m} results[metric_ij] ∀ metric ∈ results}
                                                ▷ Compute total metric values
    RR = {}                                     ▷ results ratios, RR
    for metric in results do
        MR = {}                                 ▷ model ratios, MR
        for m, v in metrics do                  ▷ model_name m, value v
            MR[m] ← v / T[metric]               ▷ total metric value T
        end for
        RR[metric] ← MR
    end for

    RC = {}                                     ▷ results combined, RC
    for model_name in RR[r_1] do                ▷ r_1, first key in RR
```

Initialise $S_m \leftarrow 0$
for metric in *weights* **do**
$S_m = \sum_{p=1}^{N} x_{m,p} * w_p$ ▷ model m, performance measure p
end for
RC[model_name] $\leftarrow S_m$
end for

$best\_model = \arg\max_{m \in RC} RC[m]$
return *best_model*
end procedure

I.7 DummyvariableChecker Pseudocode

Algorithm 11 Pseudocode for DummyvariableChecker.

procedure DummyVariableChecker(data)
Input: Data set *data*
Output: Performance measures and data split into train and test sets
Description: This algorithm returns the dummy variables, the forecast dummy variables and checks if there are dummy variables in a data set
end procedure

procedure process_dummy_variables
Input:
Output: Dataframe with dummy variables and its values.

Copy dummy variables from the original dataframe
Add additional features
return dummies
end procedure

procedure check_dummy_variables_panel(dates_error, dummies, time_col)
Input: Dates with a high error rate, dummy variables and time column.
Output: Dummy variables with high error rate.

Convert data indices to datetime objects
column_names_with_values ← initialise list to store column names
for column in dummies.columns **do**
dates_error[*is_in_dummies*] ← check if dates in date_error are in dummies for the current dummy varibales
if (dates_error['is_in_dummies'] & (dummies[column] != 0)).any() **then**
Add columns to dataframe *column_names_with_values*
end if
end for
return column_names_with_values
end procedure

procedure process_dummy_variables_panel
Input:
Output: Dataframe of dummy variables with high error rate.

Create a dummy variable dataframe: dummies
Add additional features to the dummy dataframe

```
    return dummies
end procedure

procedure DUMMY_FORECASTER(data)
    Input: Dummy data.
    Output: Dummy forecasts.

    dummy_names ← data.copy()
    forecast_combined ← create dataframe
    unique_cities ← data['City'].unique()

    for dummy_name in dummy_names do
        forecast_dummy ← create dataframe
        for city in unique_cities do
            city_data ← copy the data per city
            if dummy_names in data.columns then
                dummy_data ← city_data[columns_to_copy].copy()
                model ← Prophet()
                model.fit(dummy_data)
                future_dates ← model.make_future_dataframe(periods=365)
                forecast ← model.predict(future_dates)
                forecast_dummy ← add forecast to the forecast_dummy
            end if
        end for
        if forecast_combined is empty then
            forecast_combined ← forecast_dummy
        else
            forecast_combined ← merge forecast_combined and forecast_dummy
        end if
    end for
    return forecast_combined
end procedure

procedure SELECT_DUMMY_VARIABLES(data)
    Input: Dummy data.
    Output: The dummy variables that are in the data.

    dummy_variables ← list with dummy variable names
    dummy_variables_in_data ← create empty list
    for name in dummy_variables do
        if name in dummy.columns then
            add name in dummy_variables_in_data
        end if
    end for
    return dummy_variables_in_data
end procedure
```

I.8 Models Pseudocode

We apply basic functions for our models, for example *AR_model* refers to the basic AR model. However, the underlying algorithm remains consistent across all models.

Algorithm 12 Pseudocode for Models.

procedure PREDICTION_MODELS(*variable_data*, *data*, *dummies*, *dependent_variable*, *marketing_vars*)
    Input: Variable data, original data, dummy data, dependent variable and the marketing variables.
    Output: Best model, with its data, name and parameters.
    Description: This algorithm applies the predetermined models to the dependent variable to find the best prediction model. It selects the best model (baseline or with dummies) for each method.

    model_evaluator ← `ModelEvaluator(target_column=dependent_variable, prediction_column=prediction)`
    variable_data ← *variable_data*
    data ← *data*
    dummies ← *dummies*
    dependent_variable ← *dependent_variable*
    scaler_SVR ← *None*
    marketing_vars ← *marketing_vars*
end procedure

procedure PROCESS_FORECAST(*model_name*)
    Input: Name of the model (AR, OLS, XGBoost, SVR, ARIMA, ARF, panel, bayesian)
    Output: Baseline or dummy model with its name, data and parameter.

    $cutoff\_date$ ← max date - year
    method_name ← the function of the model name
    (*baseline_model*, *test*, *baseline_parameters*) ← `method_name(variable_data, cutoff_date)`

    (MSE_base, $RMSE\_base$, MAE_base, _, _, BIC_base, AIC_base, *error_day*, *dates_error*) ← `model_evaluator.evaluate_model_performance(test)`
    results[$X\_baseline$] ← make dataframe with performance measures

    checker ← `DummyVariableChecker(data)`
    result_dummies ← `checker.check_dummy_variables(dates_error, dummies)`

    data_with_dummy ← copy variable_data
    data_with_dummy ← add dummies from result_dummies

    (*model_dummy*, *updated_test_data*, *dummy_parameters*) ← `method_name(data_with_dummy, cutoff_date)`
    (MSE_dummy, $RMSE\_dummy$, MAE_dummy, _, _, BIC_dummy, AIC_dummy, *error_day*, *dates_error*) ← `model_evaluator.evaluate_model_performance(updated_test_data)`
    results[X_dummy] ← make dataframe with performance measures

    best_model_name ← model_evaluator.best_model_determination(results)

    if *best_model_name* == $X\_Baseline$ **then**

```
        X_model ← baseline_model
        parameters ← baseline_parameters
    else
        X_model ← model_dummy
        parameters ← dummy_parameters
    end if

    results_df ← results_df[results_df['index'] == best_model_name]
    merged_data ← merge test and updated_test_data

    return results_df, X_model, X_data, parameters
end procedure
```

I.9 Forecast Pseudocode

Algorithm 13 Pseudocode for best model determination.

```
procedure FORECAST(results_XGBoost, results_OLS, results_ARF, results_SVR,
results_panel, results_AR, results_ARIMA, results_bayesian)
    Input: Results of all models
    Output: Merged data sets and best model
    Description: This algorithm returns the merged results and the best model.
end procedure

procedure RESULTS_MERGE
    Input:
    Output: Merged dataframe with all models.

    results_df ← merge all resutls in one dataframe
    Set model names as indices
    return: results_df
end procedure

procedure MAIN(OLS_model, XGBoost_model, ARF_model, SVR_model,
panel_model, AR_model, ARIMA_model, bayesian_model, OLS_data,
XGBoost_data, ARF_data, SVR_data, panel_data, AR_data, ARIMA_data,
bayesian_data, OLS_parameter, XGBoost_parameter, ARF_parameter,
SVR_parameter, panel_parameter, AR_parameter, ARIMA_parameter,
bayesian_parameter)
    Input: The model, data and parameters of all prediction models.
    Output: Best prediction model name, the model itself, the data and parameters.

    model_evaluator ← ModelEvaluator(target_column=variable_name,
            prediction_column=prediction)
    best_model_name ← model_evaluator.best_model_determination(results_df)
    prediction_functions ← a dataframe containing all models
    data_selection_function ← a dataframe containing all data
    parameter_function ← a dataframe containing all parameters

    best_model ← prediction_functions[best_model_name]
    best_model_data ← data_selection_function[best_model_name]
    parameters ← parameter_function[best_model_name]
    return best_model, best_model_data, best_model_name, parameters
```

end procedure

Algorithm 14 Pseudocode for Forecast.

procedure FORECAST(best_model, best_model_data, best_model_name)
    Input: The data set, name and model of the best model.
    Output: Forecast values of best model
    Description: This algorithm returns the forecast values of the best model.
end procedure

procedure FORECAST_DATA_VARIABLES(data, variable_name, h)
    Input: Data set with the marketing variable name and horizon.
    Output: Predicted values for the marketing variables, we apply a simple AR model.

    data_var ← copy variable data from the data set data
    model ← `auto_arima(data_var[variable_name])`
    order ← model.order
    model_evaluator ← `ModelEvaluator(target_column=variable_name, prediction_column=prediction)`

    max_date ← max date of data_var
    min_date ← min date of data_var
    $cutoff\_date \leftarrow min\_date + \frac{max\_date - min\_date}{2}$
    $\_, \_, y\_train, y\_test, test, \_$ ← model_evaluator.train_test_split_model_panel (data_var, cutoff_date=cutoff_date)

    AR_model ← `sm.tsa.ARIMA(y_train[variable_name], order=order)`
    model_fit ← AR_model.fit()
    predicted_values ← add predicted new data in a data set.
    return predicted_values
end procedure

procedure PREDICT_MODEL(h, forecast_data, check_order)
    Input: Forecast horizon, forecast data and the order of the variables.
    Output: Forecast values of the dependent variable for horizon h.

    Add new dates to the dataframe
    if name == model_name **then**
        prediction ← self.model.predict(data)
    end if
    predicted_values ← prediction
    return predicted_values
end procedure

J Dashboard design

To provide financial insights, we develop a dashboard that incorporates raw data, predicted data, economic factors data, forecast data, and transformed data. The raw data offers insights into metrics like net sales and gross sales, while the transformed data gives information on the marketing variables, as it incorporates AdStock and lags. Additionally, we use economic factor data for interpolation and forecasting. The predicted data and forecast data offer insights in the models and variable selection techniques. After loading the data we create relationships between the different tables and create a calendar table.

Our audience for this dashboard includes data analysts and scientists at ScanmarQED. They can use the dashboard to visualise economic factors and different prediction models, enabling comparisons between their current models and our best model. Furthermore, they can analyse the impact of marketing campagins on units serviced and view forecasts. To present the dashboard effectively to clients, we aim to incorporate cost data in a later stage, enabling calculations of ROI, for now this aspects is outside the scope of this thesis.

J.1 Overview

For our starting page of our dashboard, we incorporate several key findings. Firstly, we present the units serviced over time, followed by the count of net sales, and customer satisfaction trends. While net sales show an upward trend, suggesting sales growth, units serviced do not follow the same pattern. This discrepancy suggests at other dynamics influencing net sales, such as pricing strategy changes or new product introductions not reflected in units serviced. The customer satisfaction chart reveals a decline from July 2023 onwards, dropping from approximately 9 to 7.5. Investigating the reasons behind this decline could be valuable for the client. For confidentiality, actual values have been transformed.

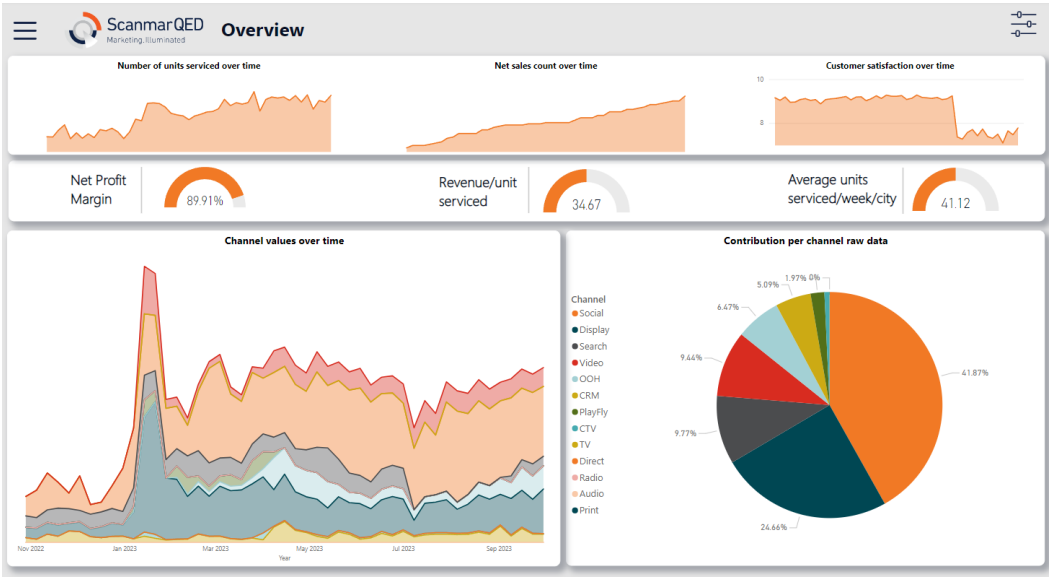


Figure 1: Dashboard overview.

Additionally, we present the net profit margin, revenue per unit serviced, and average serviced units per day per city. To address confidentiality concerns, we use fake values. When no value axis is visible, the data is real. We calculate the revenue per unit serviced

and the number of units serviced per week as:

$$Revenue\ per\ unit\ serviced = \frac{\sum_{t=1}^{T}\sum_{c=1}^{C} Net\_Sales_{t,c}}{\sum_{t=1}^{T}\sum_{c=1}^{C} Units\_seviced_{t,c}} \tag{1}$$

$$Number\ of\ units\ serviced\ per\ week = \frac{\sum_{t=1}^{T}\sum_{c=1}^{C} Units\_serviced_{t,c}}{T * C} \tag{2}$$

J.2 Baseline

Under the *"Baseline"* tab, the base model is depicted without any marketing variables. This page shows the prediction models over time, providing options to filter based on dummy or non-dummy models, the variable selection techniques, and different model names. Additionally, we have included the performance measures of the AutoForecaster models, to compare it with the performance metrics visualised for the base model.

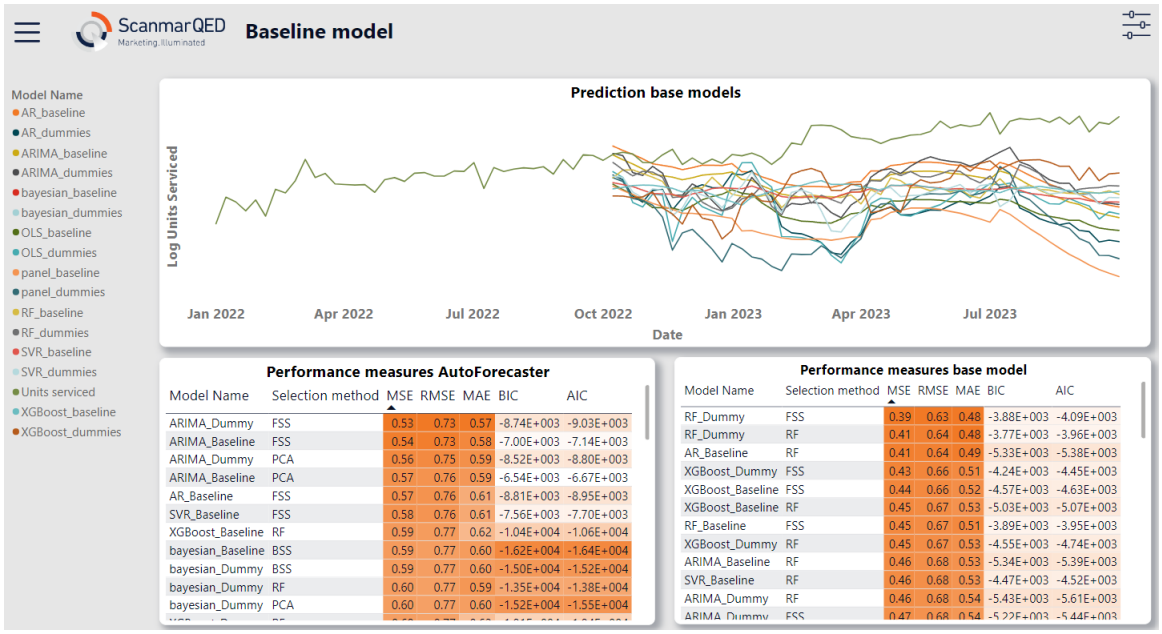


Figure 2: The Baseline tab.

J.3 Marketing effects

Under the marketing effects tab, we present the parameter values of our best model for each variable selected by the Backward selection method. Additionally, we compare raw data with AdStock data to illustrate the decay effect of after transformation. Furthermore, we display the channel percentages for all clicks, GRPs, and impressions. As mentioned prior, the constant value is changed to protect confidentiality.

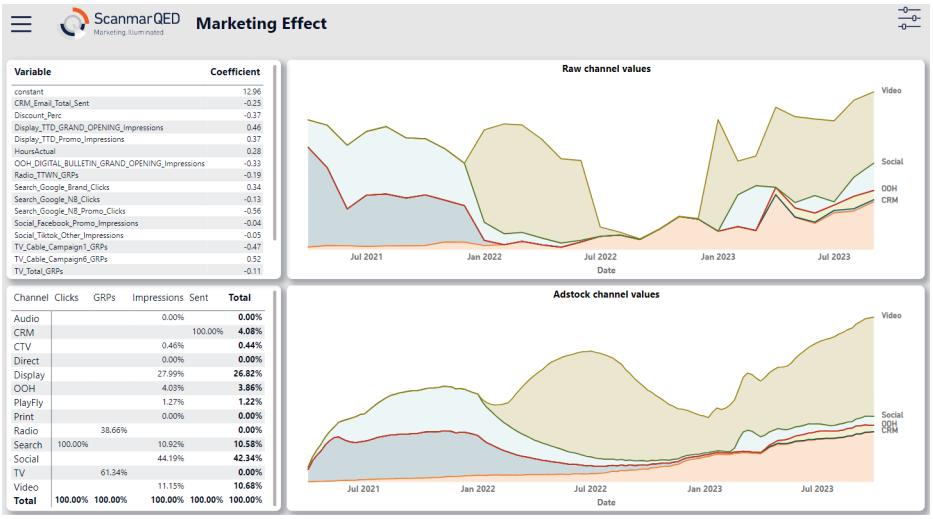


Figure 3: The marketing effect tab.

J.4 Efficiency Metrics

The efficiency metrics tab shows the performance metrics of all prediction models alongside the corresponding variable selection methods. A table on the left displays model names, variable selection methods, MSE, RMSE, MAE, BIC, and AIC values. Orange shading indicates better values. On the right, there are two stacked area charts, one that illustrates the BIC and AIC values, the other incorporates the MSE, RMSE, and MAE. These charts help to compare the variable selection methods.

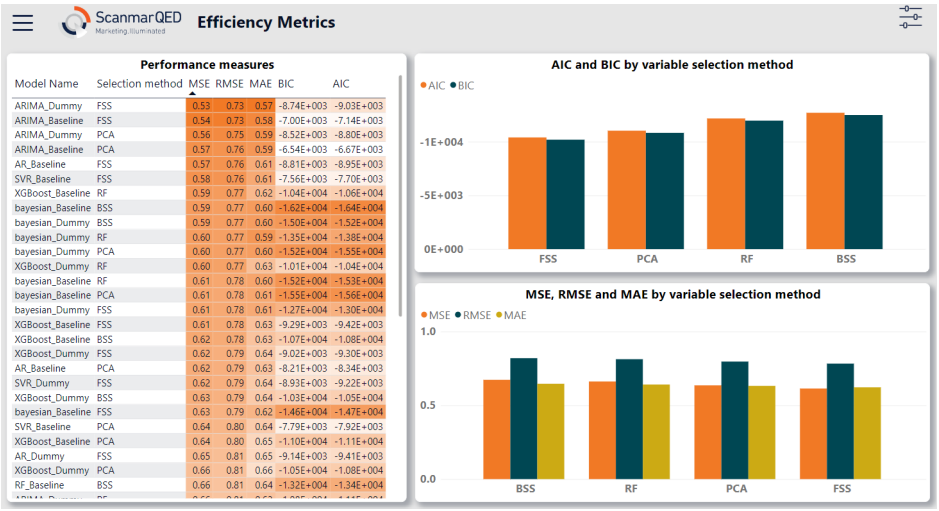


Figure 4: The efficiency metrics tab.

J.5 Prediction Models

The prediction models tab visualises various models per variable selection method through line graphs. Users can select dummy models and baseline models, compare different models across cities, and select specific models for comparison.

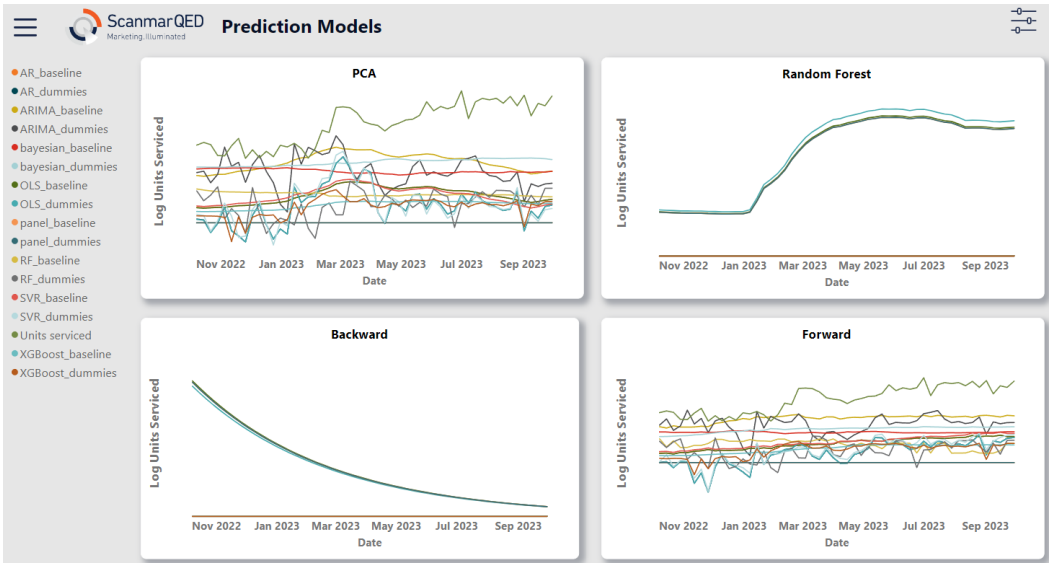


Figure 5: The prediction models tab.

J.6 Economic Factors

If economic factors are chosen by the variable selection method of the best model, we present their forecasts in this section. Additionally, we visualise economic factors over time, with the ability to view values per city.

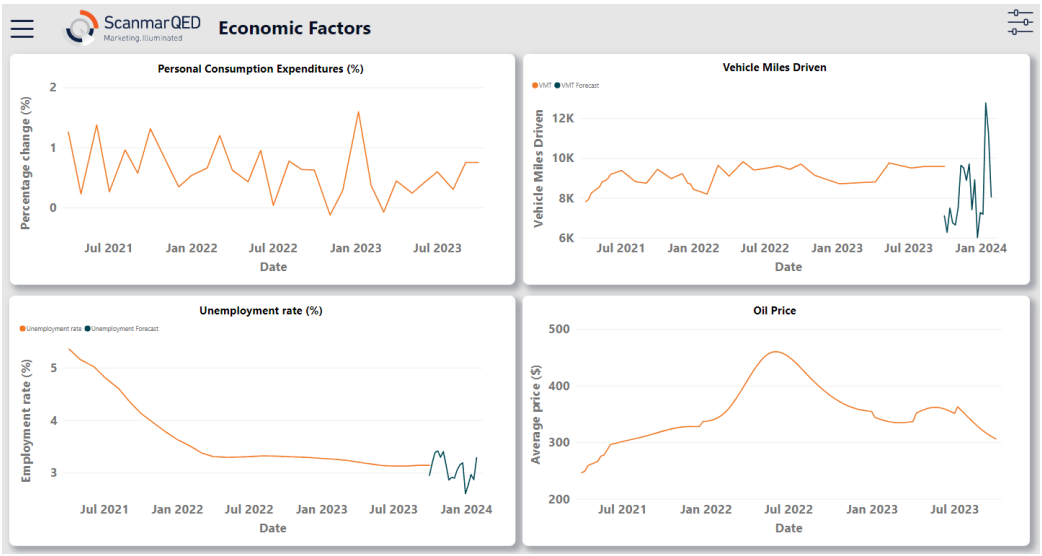


Figure 6: The economic factors tab.

J.7 Forecast

The forecast page shows predicted values alongside actual values to demonstrate differences.

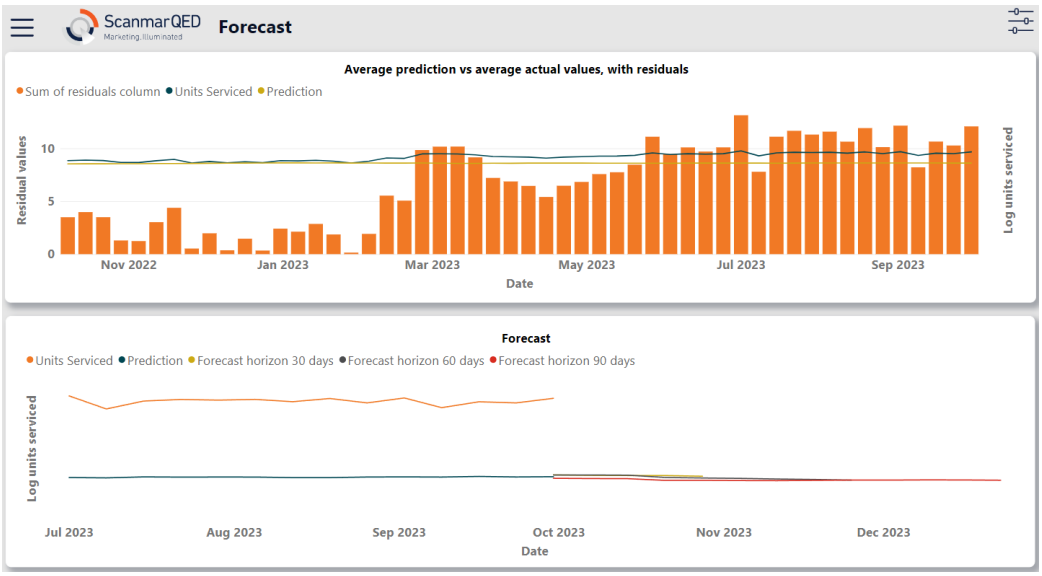


Figure 7: The forecast tab.