Unpacking the Costs

Predictive Analysis and Key Drivers for a Logistics Company's Cost Per Package

Isa Rethans

Technische Universiteit Delft

TU Delft

Unpacking the Costs

Predictive Analysis and Key Drivers for a Logistics Company's Cost Per Package

by

Isa Rethans

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday July 2, 2024 at 09:00 AM.

Student number:	4965809	
Project duration:	November 13, 2024 – July 2, 2024	
Thesis committee:	dr. N. Yorke-Smith,	TU Delft, supervisor
	ir. Marc van Geest,	The Company, daily supervisor
	dr. D.M.J. Tax,	TU Delft
	dr. rer. nat. F. Mies	TU Delft

Cover:	Icon from the Company's style library.
Style:	TU Delft report style modified by Daan Zwaneveld
Note:	At the request of the company, their name has been omitted and is instead referred to as 'The Company' throughout this document

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis marks the end of my Masters at TU Delft and, with it, the end of my student life. Reflecting on my journey, I realize that my time as a student at TU Delft has been enriching. I have grown and learned so much, and I am ready to start a new phase of my life. The past nine months have flown by, and it feels surreal that I am now at the finish line. This moment is exciting, and I look forward to my defence, which will truly be the completion of this chapter and will be followed by a well-deserved celebration.

This thesis has provided me with everything I had hoped for. My goal was to complete my thesis at a company, and although I faced some initial challenges, I was immediately enthusiastic when this opportunity with the company came by. Being part of the data science team was an incredible experience, offering me insights into their various projects, and shaping my understanding of what I want in a future job.

From a technical perspective, I have learned a lot about time series forecasting and analysis, as well as managing a long-term project. Beyond that, I found it interesting to learn about the entire The Company supply chain. A highlight was spending a day at the sorting centre near where I grew up, witnessing the scale and efficiency of the sorting process.

Throughout this journey, I encountered familiar personal challenges, but overall, I feel that everything went well. I could not have achieved this alone, and I want to express my gratitude to those who supported me.

First, I would like to thank Marc for providing me with the opportunity to do this thesis at The Company. Thank you for getting me started, for the enjoyable moments, and for being there whenever I was stuck. Thanks to Joachim for proposing this assignment and sharing your knowledge during our meetings. Thanks to Roderik for the biweekly meetings and your interest in the progress of my project. And, thanks to my direct colleagues at The Company for making me feel welcome. I enjoyed both the technical and informal conversations with all of you.

Also, special thanks to David Tax and Fabian Mies for making the effort to be part of my committee. Neil, thank you for being my supervisor. Your attentiveness, quick responses to my questions, and the time and effort you dedicated to our meetings were valuable.

Finally, and foremost, I want to thank the people closest to me. My friends, for their support throughout this process. My family, for always being there, not only during this project but far beyond it. Specifically, I want to thank my mom for always providing comfort and support. My dad, for the encouragement and the meaningful conversations during our Saturday morning coffee. My sister, for being attentive and caring. My brother, for simply being a great brother. And, of course, my boyfriend. Working on this at the same time and supporting each other, especially in the final stages, was very valuable. Your support made the process much more enjoyable.

Isa Rethans
Delft, June 2024

Abstract

This thesis investigates the drivers of the cost per package of a major Dutch parcel corporation and develops a forecasting model to predict these costs accurately. Despite its efficiency in delivering millions of packages daily, a comprehensive understanding of the cost per package is lacking due to inconsistent definitions, unknown key drivers, and incomplete data.

The primary objectives are to identify key cost drivers and develop a reliable forecasting model. Through the use of various statistical techniques and feature importance, volume and forecast realization ratio were identified as the most significant cost drivers. A comparative study of multiple forecasting models determined that the Least Squares model provides the best balance of accuracy and ease of implementation.

The analysis highlighted the impact of accurate volume forecasts and the need for more detailed data collection and documentation. The research findings offer actionable insights for the company to optimize its operations and reduce costs. Additionally, this thesis contributes to the broader field of time series analysis and forecasting in logistics, serving as a case study for future research.

Recommendations include improving volume forecasting accuracy, optimizing data collection processes, and establishing model monitoring. While the study's limitations include the dataset's size and granularity, future research could focus on creating more detailed datasets and exploring the impact of specific events on cost fluctuations and forecasting accuracy.

Contents

1 Introduction

From the canals of Delft to the heart of Amsterdam, a package delivery has become a familiar joy in our daily lives. Yet, beneath this simple act lies a complex combination of logistics, technology, and operational expertise. The Company efficiently manages this process, delivering a million packages each day and ensuring each parcel reaches its destination within 24 hours. While receiving a package is a common experience, the costs and processes associated with its journey often go unnoticed. Each delivery introduces various expenses, covering aspects such as fuel, vehicle maintenance, and employee wages. Understanding these costs and the factors that influence them is important for The Company's strategic decision-making and optimisation of their delivery process.

1.1. Problem Statement

Currently, a comprehensive understanding of the cost per package lacks due to several factors:

- **Inconsistent Definitions**: Costs are defined per stage in the supply chain, but varying definitions of volume lead to inconsistencies, making it difficult to accurately calculate the total cost per package.
- **Unknown Key Drivers**: While various factors are known to influence costs, there is no clear understanding of which factors have the most significant impact.
- **Incomplete Data**: The costs associated with delivery to Belgium are not included in the total cost calculation, resulting in skewed results and an incomplete understanding of the overall cost per package.

These issues highlight the need for a consistent and comprehensive approach to defining and analysing the cost per package. By addressing these inconsistencies and incorporating complete data, The Company can gain valuable insights into its cost structure and identify opportunities for optimisation.

1.2. Objectives

The primary objectives of this thesis are twofold: first, to identify the key drivers of cost per package; and second, to develop a reliable forecasting model using these key drivers to predict the cost per package. To achieve these objectives, the following research questions are formulated:

1. **What are the key factors influencing the cost per package at The Company?** This question will be answered through data analysis, employing various statistical techniques and feature importance assessments to identify the most influential drivers.
2. **What is the most effective forecasting model for The Company's cost per package six to eight weeks ahead?** This question will be addressed by comparing multiple forecasting models, considering their accuracy, complexity, and ease of implementation within The Company's existing workflows.

Ultimately, this thesis aims to develop a consistent approach to defining the cost per package, including the integration of volume and costs for Belgium, and to provide The Company with a comprehensive understanding of its cost per package and the tools to accurately forecast and manage these costs.

1.3. Contributions

By identifying key cost drivers and developing a forecasting model, this research seeks to provide The Company with actionable insights to reduce the cost per package. This deeper understanding of the relationship between various factors and costs will enable The Company to optimize operations, enhance cost management, and maintain a competitive advantage. Additionally, the research will result in a tool that can be integrated into The Company's existing workflows to inform strategic decisions and support cost reduction efforts.

Beyond its immediate relevance to The Company, this thesis contributes to the broader field of research. It serves as a case study on time series data analysis and forecasting in the logistics domain. The comparative study of different models and forecasting techniques, coupled with statistical methods to evaluate the key drivers, offers valuable insights that can inspire future research and practical applications in similar contexts.

1.4. Structure of the Thesis

The rest of this thesis is structured as follows. Chapter 2 provides an overview of the process of package delivery. Chapter 3 describes the potential key drivers and the data collection for the study. Chapter 4 presents the exploratory data analysis and initial efforts to identify the key drivers. Chapter 5 describes the methods used for cost forecasting. Chapter 6 provides a detailed analysis of the forecasting results, comparing the performance of different models and discussing feature importance. Chapter 7 interprets the forecasting results, addresses limitations, and integrates insights from the data analysis and feature importance to identify the key factors influencing the cost per package. Finally, Chapter 8 summarizes the key findings and offers recommendations for The Company.

2

The Delivery Process

This chapter provides an overview of The Company's delivery process, offering context for understanding the company's logistics and cost structure. By detailing each step of the parcel delivery process and the associated costs, this chapter sets the foundation for analysing the key drivers of cost per package and the forecasting efforts. Understanding these operationpal processes is useful for identifying areas of potential improvement and developing accurate predictive models.

2.1. Parcel Delivery Process

Originally, mail was the largest sector within The Company, but the volume of mail sent has been declining in recent years. In contrast, package delivery has been increasing, with The Company delivering an average of 1.1 million packages per working day in 2023. To manage the transportation of over a million packages daily, The Company must efficiently handle the entire package delivery process from the customer to the receiver. To facilitate this, The Company operates 37 depots in the Netherlands and 6 in Belgium. Of the 37 depots in the Netherlands, 30 are designated for packages, 5 for mail, 1 for international delivery, and 1 specifically for small packages. The 6 depots in Belgium are exclusively used for packages. These depots act as hubs where packages are sorted to ensure correct routing to their final destinations. They form part of an efficient delivery process, which is divided into five main steps: collection, first sorting, intertraffic, second sorting, and distribution. The whole process is depicted in Figure 2.1 and the next sections will provide a detailed explanation for each step.

Figure 2.1: Five key steps of the delivery process[1]

2.1.1. Collection

During the collection phase, all packages from The Company's customers are brought to one of the large depots. Typically, packages are taken to the depot nearest to their collection point. However, this is not always the case. Sometimes packages are redirected to another depot due to capacity issues at the nearest sorting centre or for various other operational reasons. Depending on the size and terms of the contract with the customer, collection may be performed by The Company or the customer may be required to bring the packages to a retailer or to a depot themselves.

[1] Illustration created by the author using icons from The Company

2.1.2. Send Sorting
After collection, packages undergo their first sorting process, known as send sorting. The objective of this step is to organize the packages according to the second depot they will be sent to. This second depot is located closer to the destination address, facilitating efficient distribution.

2.1.3. Intertraffic
During the intertraffic phase, packages are transported from the first depot to the second depot. This transportation process can follow one of two routes. The first route involves direct transportation of packages from the first depot to the second depot. The second route involves routing the packages through an intermediate location known as a cross-dock. In this scenario, the package's journey is divided into two segments. The first segment involves transportation from the first depot to the cross-dock. During this segment, packages destined for various depots are combined into single transport. Upon arrival at the cross-dock, the loads from multiple trucks are split and reorganized into groups of packages that are headed to specific destination depots. After this reorganisation, the packages are transported from the cross-dock to their respective destination depots. An exception exists for packages that are already at the correct depot after the first sorting. These packages bypass the intertraffic transport phase and proceed directly to the next phase.

2.1.4. Distribution Sorting
The second sorting process, known as distribution sorting, is the final step before the packages are delivered to the receiver. In this phase, packages are sorted into collections that correspond to the delivery area of one delivery person.

2.1.5. Distribution
The final step of the delivery process is the distribution of the packages to their final destination. Delivery personnel fill their buses with the packages on their route and drive out to deliver them. Ideally, the package is successfully delivered in the first attempt. However, in cases where delivery is not possible, such as when the receiver is not at home, or no package locker is available, the package is returned to the depot. It then either goes through the second sorting process again to be redelivered the next day, or it is handled differently.

2.2. Costs
At each stage in the delivery process, costs are introduced. The Company keeps track of these costs using weekly reports. These reports contain detailed information about the costs associated with specific steps of the supply chain. The weekly reports relevant to this thesis include those from depots, transport, and Belgium. The costs specified in the weekly depots include all expenses related to the depots, including collection and distribution. The transport reports, on the other hand, cover expenses introduced during the intertraffic phase. However, since intertraffic does not only apply to the package delivery supply chain, the correct costs had to be extracted from these general transport costs. The weekly reports from Belgium detail the costs relevant to the entire supply chain within Belgium. In addition to the costs detailed in the weekly reports, there is another cost component that contributes to the overall cost per package. This component is the cost of trailers. The Company uses two types of trailers: swap trailers and storage trailers. Swap trailers are filled by large customers with packages and swapped for empty ones when trucks arrive. Storage trailers provide flexible storage capacity, temporarily holding packages when depots are full until processing begins. These costs are billed monthly, so to get the weekly costs, the monthly costs are divided by the number of weeks in the month.

To determine the cost per package, the total costs are divided by the volume of packages. The term *volume* has various definitions within The Company, therefore a precise definition is necessary for clarity. In this thesis, the volume is defined as the total number of packages that had their first sorting in the Netherlands or Belgium. Since the costs are reported weekly, the volume is also specified every week. This is summarised into the following formula for the cost per package:

$$\text{Cost Per Package} = \frac{\text{Depot Costs} + \text{Transport Costs} + \text{Belgium Costs} + \text{Trailer Costs}}{\text{Volume}}$$

This definition of the cost per package is new for The Company and directly addresses two of the problems

identified in Section 1.1. First, it resolves the inconsistencies caused by varying definitions of volume across different stages of the supply chain. By establishing a definition, a standardized basis for calculating the cost per package is provided. Second, the inclusion of packages with their first sorting in Belgium in the volume calculation addresses the problem of incomplete data. This inclusion provides a more complete understanding of The Company's overall costs, making sure not to underestimate it.

2.3. Volume Forecasting

The forecasted volume is an important aspect of the The Company supply chain, as it informs the planning and resource allocation necessary to handle all package deliveries efficiently. The relevance of the volume forecast for this project lies in its important role in determining the cost per package. Accurate volume forecasts allow for efficient resource allocation and operational planning, which directly impacts the cost efficiency of parcel delivery. Therefore, understanding and anticipating the volume of packages will give insights into the cost per package. Besides, the volume forecast serves as an input for the cost-per-package forecasting model. If the volume forecast is inaccurate, this will add to the uncertainty of the cost per package forecast. Hence, the accuracy of the volume forecast is a foundational element in achieving reliable results in the forecasting of the cost per package.

The Company employs two distinct methods to create volume forecasts, which are then compared and combined to produce a final forecast known as the rolling forecast. This rolling forecast provides predictions for the next eight weeks and is updated weekly.

The first method relies primarily on forecasts provided by The Company's largest customers. These forecasts are based on the customers' own expectations and plans, including any scheduled promotions. By gathering and aggregating this information, The Company can predict the volume of packages expected from these major clients, allowing for adjustments in planning and operations based on anticipated demand.

The second method utilizes indices derived from historical data to predict future volumes. This approach involves several steps. An index is created based on the weekly volumes of the past years, where the index value of 100 represents the average volume over a specific reference period. Each week's volume is compared to the index value of 100 to understand deviations. These deviations help predict what the index will be for upcoming weeks. Knowing the volume that corresponds to an index value of 100 allows The Company to estimate the number of packages for each week. These estimates are further broken down into daily volumes based on historical daily distribution patterns.

Each week, a meeting is held to compare the forecasts generated by these two methods. The predicted costs from customer-based forecasts and index-based forecasts are discussed, and the most likely outcome is determined. The insights from these discussions are used to combine the two methods and create the final rolling forecast to be used for the planning of resources and employees. Since the planning is based on the rolling forecast, the forecast must be as accurate as possible. This ensures that the right amount of resources is allocated to handle the expected volume of packages.

3

Dataset

This chapter describes the creation of the dataset, beginning with the identification of potential factors that are of influence on the cost per package. For each of these factors, their potential impact on cost efficiency is described. Additionally, an overview is provided of the data sources and the methodology used for data collection and storage, forming the basis of subsequent analyses.

3.1. Factors Influencing Package Cost

To understand the various elements that influence fluctuations in the cost per package, first potential factors have to be identified. Through discussions with The Company employees across various departments, several factors were identified as potentially influencing the cost per package. These factors are categorized into four main groups: external factors, process characteristics, volume distribution, and package characteristics. An overview of these factors, along with brief explanations, is provided in Table 3.1. The rest of this section provides a more detailed description of each factor and its potential impact on the cost per package.

Category	**Factor**	**Explanation**
Package Characteristics	Average weight	Average package weight in grams.
	Average DM3	Average package volume in cm^3.
	Volume NMG ratio	Proportion of non-machineable goods.
Process Characteristics	Hit rate	Percentage of first-attempt deliveries.
	Avg distance	Average distance between the first and the second depot for sorting.
	Ratio forecast/realisation	Forecasted vs. actual volume ratio.
External Factors	Diesel price	Current diesel prices from CBS.
	Inflation	Inflation rate from CBS.
Volume Distribution	Ratio min/max volume	Ratio of the maximum to minimum volume observed within a week.
	Ratio Belgium	Share of packages destined for Belgium.
	Ratio Main Client	Share of packages processed at a specific sorting centre for the main client.
	Ratio international	Share of packages sorted for international delivery
	Total Volume	Quantity of packages that had their first sorting in a depot in the Netherlands of Belgium.

Table 3.1: Overview of the potential causes of fluctuation

3.1.1. Package Characteristics

Package characteristics, such as weight, volume, and the proportion of non-machineable goods (NMG), significantly influence operational costs. The size and weight of packages directly impact the resources needed for transportation and handling, thereby influencing overall costs. NMG packages are packages that cannot be processed automatically due to their size, shape, or weight. These packages require manual sorting and handling, which is more resource-intensive. Consequently, the presence of NMG packages can lead to increased depot costs.

3.1.2. Process Characteristics

Process characteristics include factors directly related to the logistics and operations involved in delivering a package, covering everything from sorting to delivery. The *hit rate*, which is the percentage of packages successfully delivered on the first attempt, influences labour and fuel costs, as well as sorting expenses. This effect occurs because unsuccessful deliveries require an additional delivery attempt and consequently an additional pass through the distribution sorting process.

Another considered factor is the average distance a package travels from the depot where the first sorting occurs to the depots where the second sorting is done. This average distance contributes to fuel consumption and personnel costs for the longer driving time.

Furthermore, the ratio of forecasted to actual volume delivered (*forecast realisation ratio*) is a way to assess the accuracy of operational planning. Differences between forecasted and actual volumes can lead to inefficiencies, either through resource underutilization or inability to meet demand. When actual volumes fail to meet the forecasted volumes, resources may be underutilized, leading to increased cost per package as the fixed costs are distributed across fewer packages. This effect is aggravated by the fact that costs are inflated initially due to the resource planning based on high forecast, and these elevated costs must then be spread across a smaller number of packages.

Conversely, exceeding forecasts can result in the overutilization of resources, driving up variable costs due to the need for additional labour and materials. The last-minute solutions that are required in these situations, often involve more expensive labour and resources, which can negatively impact the cost per package.

3.1.3. External Factors

External factors, such as diesel prices and inflation, are variables beyond The Company's control. Despite being outside The Company's influence, they can still impact the company's costs. For instance, changes in diesel prices directly affect transportation expenses, as increases in fuel costs lead to higher costs for shipping and delivery operations. Similarly, inflation affects the cost of goods and services which are important to The Company's package handling, leading to an overall increase in operational costs.

3.1.4. Volume Distribution

The last category of potential causes of fluctuations within the cost per package is volume distribution. This category includes factors related to the volume, including the total volume, the distribution of package volumes across different depots and how to volume is spread within the week. All these factors affect resource allocation and efficiency throughout the supply chain.

For instance, the total volume of packages, defined as the number of packages that had their first sorting in either the Netherlands or Belgium, is important for the total cost efficiency. Higher volumes generally lead to scaling advantages, reducing the cost per package.

Additionally, the volume distribution affects resource allocation. For instance, a busier day necessitates more trucks, as well as additional personnel in both the depots and distribution network. An imbalance in volume distribution, particularly the ratio of maximum to minimum volume within a week (min/max volume ratio), can cause inefficient resource allocation. A low min/max ratio reflects high variability in daily volumes. This inconsistency in handled volumes can make it difficult to plan and utilize resources efficiently, and thus could lead to increased costs per package.

Moreover, costs are influenced by factors such as the proportion of packages delivered within Belgium (Belgium ratio), and those processed at specific centres like the sorting centre specifically for the main client's packages (Main Client ratio), and for international packages (international ratio). Over the years, the volume of packages processed in Belgium has increased. However, the infrastructure of The Company in Belgium is

not as developed as in the Netherlands. Additionally, in Belgium, distribution is managed through contractors rather than in-house employees, as is the default in the Netherlands. Combined with the smaller processing volumes at Belgian centres, these factors contribute to less efficient handling of packages. Consequently, the cost per package in Belgium may be higher.

There is one sorting centre that specifically serves as the sorting centre of packages for both export and incoming imports. The timing of the arrival of imported packages, particularly from China, is very unpredictable; packages may arrive in large batches without warning, followed by periods with no arrivals. This introduces challenges for scheduling and necessitates expensive last-minute logistical adjustments.

On the other hand, the main client benefits from a strategic location near their warehouse, enhancing operational efficiency. This proximity allows for the quick and efficient processing of orders, likely contributing to lower operational costs per package.

3.2. Data Collection

With the features defined, a dataset can be constructed that incorporates these feature for analysis. To do this, multiple data sources were used, including internal databases, weekly reports, and external sources. This section explains the data collection process and describes how each feature was obtained.

3.2.1. Data Preparation

The Company works with Amazon AWS[1] to store and manage its data in a centralized data lake, allowing for the storage of both structured and unstructured data at any scale. The most relevant database for this thesis is one that contains detailed information about all packages. Each update to this database creates a new row, resulting in multiple rows for a single package.

Given that millions of packages are processed daily, each with many features and multiple rows, the database becomes very large. To manage this, a clean version of the database was created first by selecting only the relevant columns and aggregating the information from multiple rows into a single row per package. The initial goal was to match the number of rows per week in the clean dataset with the realisation volume reported in weekly reports. If the volume matches, it indicates that the correct packages are selected, making the dataset a reliable basis to extract other features. This task was challenging due to the presence of duplicates, missing data and irrelevant packages, requiring the application of correct filters. The limited documentation made it time-consuming to develop the correct query.

3.2.2. Definition of Features

Once the volume matching was successful, the dataset was extended to include the desired features. For each feature, one value per week was calculated. A week start is defined to be on Sunday as this is standard for The Company. The weekly data points were then combined into one dataset. A specific definition per feature is provided in the following sections.

Average Weight and Volume

For the weight and volume (DM3), the respective fields in the database were used. Since not every package had these values defined, the average weight and volume were calculated based on the packages that did have this information.

Volume NMG Ratio

Non-Machinable packages are classified based on their dimensions, weight, and shape. There are maximum limits for weight and size, as well as a minimum size to prevent small packages from getting lost. The ratio of Non-Machinable Goods was calculated by identifying packages falling outside the standard dimensions and dividing this count by the total number of packages.

Hit Rate

Packages delivered at the first attempt, either at home or to a pick-up point, are considered successful first-attempt deliveries. The database includes a field about the delivery status, which specifies these two options among others. This field is used to identify first-attempt deliveries. The hit rate is then calculated by dividing the number of first-attempt deliveries by the total number of packages.

[1] `https://aws.amazon.com/console/`

Average Distance
The distance between the first and second sorting depots is calculated using their latitudes and longitudes with the Haversine formula. This formula determines the shortest distance between two points (ϕ_1, λ_1) and (ϕ_2, λ_2) on a sphere. It is given by:

$$\text{distance} = 2 * r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

where r is the Earth's radius of approximately 6,371 km, $\Delta\lambda = \lambda_2 - \lambda_1$ and $\Delta\phi = \phi_2 - \phi_1$ [6]. The average distance is then calculated by taking the mean of all distances.

Ratio Min/Max Volume
The ratio of the minimum and maximum volume within a week is calculated by considering the daily volumes, excluding Sundays and holidays. This ratio is obtained by dividing the maximum volume by the minimum volume.

Ratio Belgium, Main Client, and International
Packages destined for Belgium were identified by the country code of the destination address, while main client and international packages were identified by the code of the first sorting centre. The ratios were calculated by dividing the number of packages for each destination by the total number of packages.

Forecast realisation Ratio
The forecasted volume is saved in a specific file. This file was used to extract the forecasted volume per week. The forecast realisation ratio was then calculated by dividing the forecasted volume by the total volume.

Diesel Price and Inflation
Data from CBS was used to incorporate Diesel Price[2] and Inflation[3]. The data files were saved and parsed to extract the relevant information.

[2] `https://www.cbs.nl/nl-nl/cijfers/detail/80416ned`
[3] `https://www.cbs.nl/nl-nl/cijfers/detail/70936ned`

4

Data Analysis

The objective of this chapter is to familiarize with the dataset and to gain insights into the factors influencing the cost per package. By examining individual features and their interactions, the aim is to identify patterns, trends, and relationships that can inform choices for methodology and provide initial insights into which factors might be of significant influence on the cost per package. The analysis begins by focusing on the cost per package, using descriptive statistics and time series analysis. Then, the interactions between these features are explored through various statistical methods.

4.1. Inspecting the Cost Per Package

Initially, a detailed analysis of the cost per package is performed to study the properties of the data. This is done by providing and overview of the cost per package over time and its basic statistics, followed by a deeper evaluation of its patterns and dependencies.

4.1.1. Descriptive Statistics

The analysis begins with a visualisation of the cost per package over time, as depicted in Figure 4.1. The histogram in the figure illustrates the distribution of costs, highlighting the spread of the data. Due to the sensitive nature of the costs for The Company, the absolute values are not displayed.

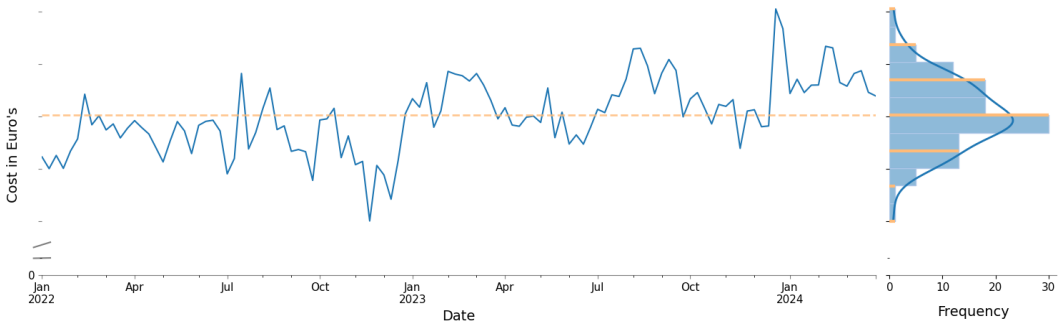

Figure 4.1: Data Visualisation of the Cost per Package[1]

The histogram reveals that the data is approximately normally distributed, with a slight skew towards lower values. This normal distribution is interesting for many statistical analyses as it is a frequently occurring assumption. However, the line plot shows that the costs are volatile, meaning that it exhibits many fluctuations over the observed period. This volatility is particularly interesting for forecasting purposes as it adds complexity to the forecasting task. It suggests that the cost per package is influenced by various factors that change over time. Understanding these fluctuations can help improve the accuracy of future cost predictions.

[1] The y-axis is not displayed for confidentiality reasons.

4.1.2. Auto Correlation Analysis

Auto-correlation is a standard method in time series analysis used to identify patterns and relationships within the data. This method involves calculating the correlation between the cost per package at different time lags. By analysing these correlations, it can be determined whether past values are informative of current values. If a time series has significant auto-correlation, it suggests that the past values have a meaningful relationship with the present values. This indicates temporal dependencies in the data, which is important for time series forecasting and analysis methods, as they rely on these dependencies to make accurate predictions. In contrast, if there is little to no auto-correlation, it implies that past values do not inform the current values. In such cases, regular analysis and forecasting methods, which do not account for past data, might be more appropriate [14].

Figure 4.2 shows the auto-correlation function (ACF) plot for the cost per package. Each bar in the plot represents the correlation coefficient for a specific lag. At lag 0, the correlation is always 1, as any variable is perfectly correlated with itself at the same time point. As the lags increase, the plot shows how the past values of the cost per package correlate with current values. The blue surrounding cone indicates the confidence interval set at 95%. Bars extending beyond the horizontal bounds of the cone indicate that the correlation at that lag is statistically significant. This means it can be asserted that, with 95% confidence, the observed correlation is not due to random chance.

The ACF plot shows that the cost-per-package autocorrelations are significant at lags 1, 2, 3, and 4. This indicates that the cost per package is influenced by its past values up to 4 time periods. Therefore, the cost per package is not a random process, and temporal dependencies should be considered in the analysis.

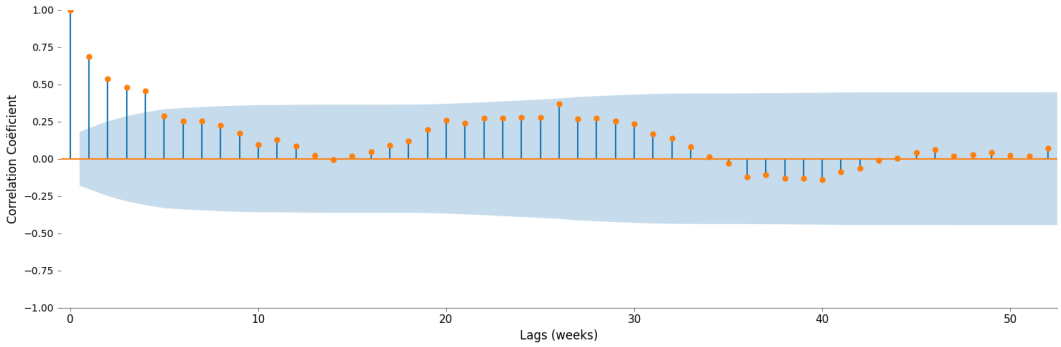

Figure 4.2: Auto Correlation Plot for the Cost per Package

4.1.3. Stationarity Analysis

Another important aspect of time series analysis is stationarity. A time series is said to be stationary if its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. Non-stationary time series often contain trends and seasonality, causing these statistics to change. Many time series forecasting and analysis methods require the data to be stationary to produce valid results. If the data is not stationary, the results may be unreliable as they can depend on the specific time at which the data is observed [25].

When a time series is non-stationary, it can be transformed into a stationary series using various techniques. One common transformation is differencing, where the value at time t is subtracted from the value at time $t-1$. Data in its original form is called level data, whereas data differenced n times is called nth difference data. By applying differencing, trends and seasonality can be removed from the data. However, to best preserve the inherent properties of the time series, differencing should be minimized. It should only be applied to achieve stationarity, as excessive differencing might lead to the loss of long-term information [25].

To test for stationarity, the most common methods are the Augmented Dickey-Fuller (ADF) test [15] and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [53]. The ADF test's null hypothesis is that the time series is non-stationary, while the KPSS test's null hypothesis is that the time series is stationary. For the ADF test, data is considered stationary if the null hypothesis is rejected, requiring a p-value below the significance level. Conversely, for the KPSS test, stationarity implies accepting the null hypothesis, necessitating a p-value above the significance level.

To test for stationarity of the cost per package, the ADF and KPSS tests were first performed on the level data. Their test statistics are shown in Table 4.1, with statistically significant results at the 5% level highlighted in blue. The ADF test accepts the null hypothesis, while the KPSS test rejects it, therefore it can be concluded that the level data is non-stationary. The tests provide opposing results for each the ADF and KPSS, which is expected and validates the outcomes' reliability. As a next step, differencing is applied to see if this transformation achieves stationarity. The differenced data was tested again using the ADF and KPSS tests. The results, as displayed in the second column of Table 4.1, indicate that the ADF test rejects the null hypothesis, while the KPSS test accepts it. Thus, it can be concluded that the first difference data is stationary.

Level Data		**First Difference**	
ADF	KPSS	ADF	KPSS
−2.773	1.098**	−9.137***	0.113

Table 4.1: Stationarity Test Results for the Cost per Package
*** statistically significant at the 1% level.
** statistically significant at the 5% level.

4.1.4. Time Series Decomposition

The observations that the data contains temporal dependencies and is non-stationary suggest the presence of underlying patterns such as trends and seasonality. To gain insights into these patterns, it is useful to decompose the time series into its fundamental components: trend, seasonality, and error. This can be done with a time series decomposition analysis, which breaks down the data into these components. The trend represents the general direction in which the data is moving over a longer period. Seasonality captures the repeating short-term cycle in the data, which helps understand periodic fluctuations that occur at regular intervals, such as monthly or yearly patterns in the cost per package. The error (residual) represents the random variation in the series, helping to identify the noise in the data that cannot be explained by the trend or seasonality [27].

The decomposition of the time series is performed using the seasonal_decompose function from the *statsmodels*[2] library as this is a popular and reliable implementation [24]. The results of the decomposition are visualized in Figure 4.3.

Figure 4.3: Trend, Seasonal and Error Decomposition of the Cost per Package.[3]

[2] https://www.statsmodels.org/stable/index.html

The trend component, depicted in the top plot, shows a gradual upward movement over time. This indicates that the average cost per package has been increasing steadily from January 2022 to early 2024.

The seasonal component in the middle plot, shows regular cycles of peaks and troughs, especially when comparing the two full years. For both years, the peaks and troughs occur at approximately the same time. Within the years, two main peaks and troughs are visible, one pair per half year. However, not with the same amplitude. Around the end of the year, the troughs are deepest, meaning that the costs are lowest. This can be explained by the fact that those are the busiest months for The Company, due to Black Friday, Sinterklaas and Christmas. Higher volume likely means scaling advantages and efficient use of the resources, which results in lower costs per package.

Finally, the residual component in the lowest plots shows small residuals which are centred around zero. This means that the decomposition effectively isolates the main components of the time series. However, the periods between July and October show irregular residuals, which indicate unexpected changes in costs. It could be that summer is a less predictable period in the year, where external factors such as weather and holidays influence customers' purchasing behaviour.

4.2. Identifying Significant Cost Drivers

After zooming in on the cost per package, the analysis shifts to its relationship to other features. The features were all chosen to be of potential influence on the cost per package, but the goal is to investigate which of the impacts are significant. As this is a challenging task with many possible approaches, a combination of methods is employed for a multi-sided analysis. This section focuses on the statistical methods used to identify significant relationships. These insights serve as the foundation for the upcoming chapter, which will focus on forecasting costs and assessing feature importance to further explore the interactions between the features and the cost per package.

4.2.1. Method Selection

The selection of statistical methods for finding the significant relationships between the features and the cost per package of is guided by multiple considerations.

First, the methods should vary in complexity, from simple pairwise dependencies to more complex approaches that consider the dataset as a whole. By applying simple methods, a basic understanding of the relationships between the features and the cost per package can be formed. Whereas, by using more complex methods that consider interactions between multiple features can provide deeper insights.

Second, given that the dataset consists of time series data, it inherently includes temporal dependencies. Therefore, the methods' ability to capture these temporal dependencies is included as a consideration. This makes causal discovery methods for time series particularly relevant. These methods aim to identify causal relationships in time series and estimate the effect of one variable on another [4]. There are four main classes of causal discovery approaches [10, 46]:

- **Granger causality based methods** are statistical methods that are used to determine whether one time series is useful in forecasting another [22].
- **Conditional independence based methods** determine causal relationships by assessing whether variables are independent of each other given the presence of other variables [50].
- **Structural Equation Models** involve representing causal relationships through a system of equations that describe how variables influence each other [57].
- **Deep Learning Approaches** use neural networks to find complex, non-linear relationships in data [47].

Due to the small size of the dataset, overly complex methods like the deep learning approaches are not suitable. The other classes all contain methods that could be of interest for this thesis, leaving many options.

To further refine the choices, it is necessary to specify the relevant definition of causality. Many causal discovery methods rely on temporal precedence, where a cause precedes its effect. However, for this thesis, it's more relevant to consider situations where cause and effect occur simultaneously, known as direct causality.

[3] The y-axis is not displayed for confidentiality reasons.

This is because the cost per package is likely influenced by features within the same week [4]. For example, the average weight of the packages will only affect the cost for that particular week. This concept is known as direct causality. This focuses attention on techniques that can capture these direct effects.

The final consideration involves latent cofounders, which are unobserved variables that influence both the independent and dependent variables, thereby potentially biasing the observed relationship between the [5]. While the dataset was chosen to capture factors influencing the cost per package comprehensively, external events, such as COVID-19 or the war in Ukraine, can also impact costs. Hence, a method that can account for latent cofounders could be valuable to include.

The methods chosen and the specific motivations for their selection will be detailed in the next sections.

4.2.2. Correlation

The first choice is correlation analysis, which is a common starting point to identify initial patterns and connections among variables [21]. This approach is chosen for its simplicity. Although correlation does not imply causation or directly confirm a relation between two variables, it helps identify initial patterns and connections among variables. Besides, it gives a possibility to verify if the direction and strength of the relationships between the variables are as expected. If so, this strengthens the belief that the data is correct and validates the reliability of subsequent analysis results.

The correlations are computed using the Pearson correlation coefficient. The Pearson correlation coefficient not only measures the linear relationship between two datasets but also involves a statistical test to evaluate the significance of this relationship. The significance is assessed through a p-value, derived from a beta distribution, which quantifies the probability that the observed correlation could occur by chance. The null hypothesis is that there is no true linear correlation between the two considered variables. Rejecting the null hypothesis suggests that the observed correlation is unlikely to have occurred by random chance, thereby indicating a statistically significant linear relationship between the two variables [49].

For this analysis, a significance level of 5% is used, and thus only those variable combinations with p-values under 0.05 is considered statistically significant. Accordingly, Figure 4.4 displays only those correlations that meet the 5% significance threshold. Exact numbers are omitted, but the colours indicate the relationships and their strengths.

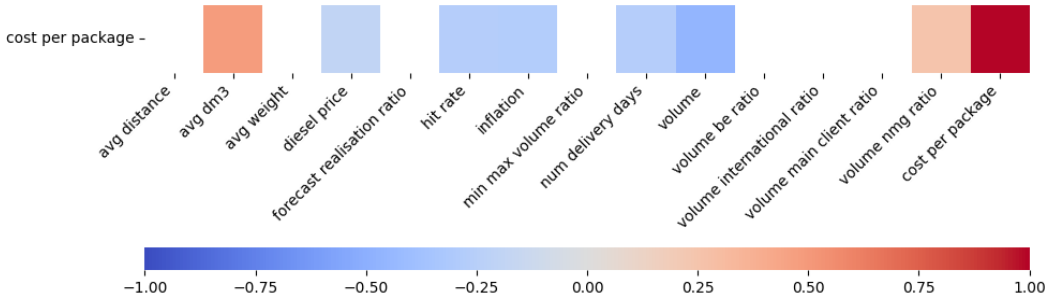

Figure 4.4: Correlations between the Cost per Package and Each of the Features

The correlation matrix reveals positive correlations between the cost per package and the *avg dm3* and *volume NMG ratio*. Whereas negative correlations are observed with the *diesel price*, *hit rate*, *inflation*, *num delivery days* and *volume*. These correlations are all in line with the reasoning when choosing the features, as described in Section 3.1, except for the correlation between the *inflation* and the *diesel price*. These correlations are negative, where a positive correlation is expected since higher diesel prices and inflation intuitively would lead to a higher cost per package. This might already be an indication that the relationship of the cost per package with both the *inflation* and the *diesel price* is not significant. This should be kept in mind during the evaluation of the validity of subsequent analytical methods.

The strongest positive and negative correlation for the cost per package is observed in combination with the *avg dm3* and *volume* respectively. This is not surprising due to their direct impact on the costs per package. Larger packages require more resources and are likely to increase the cost per package. Whereas higher volumes are likely to reduce the cost per package due to scale advantages.

4.2.3. Granger Causality

The second method applied is Granger causality, which is a popular method that has been widely used to infer the relationships from time series data [13, 54, 17]. It is a statistical method used to determine whether one time series is useful in forecasting another. This concept, known as predictive causation, measures whether past values of one series improve the prediction of another series [56]. Granger causality is chosen for its simplicity and intuitive approach to understanding relationships using the information in the predictor. This is particularly relevant for analysing the cost per package, as it provides insights into temporal relationships and predictive dependencies. However, Granger Causality does not account for latent confounders, thus the results require careful interpretation and validation [18].

The key idea of Granger causality is to use past values to predict future values and measure the variance of prediction errors. If including past values of a predictor variable X reduces the prediction error variance for a target variable Y, then X is said to Granger-cause Y [13].

Formally, a Vector Autoregressive (VAR) model is used to fit two models: one using only past values of the target variable Y, and another using past values of both Y and the predictor variable X. The VAR model is an autoregressive model which involves regressing each variable on its own lagged values as well as the lagged values of the other variables in the system. To statistically test if the additional predictor significantly improves prediction accuracy an F-test is used to compare the variance of prediction errors. The null hypothesis states that lagged values of X do not improve the prediction of Y. If the test rejects the null hypothesis, it is concluded that X Granger-causes Y [13].

Stationarity

In the definitions for Granger Causality, it is assumed that the time series are stationary. As mentioned in Section 4.1.3, a time series is stationary if its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. Non-stationary data can lead to spurious results because the outcomes depend on the specific time period analysed, and thus they may vary if a different time period is considered [22].

To test the stationarity of the features, the same approach used for testing the stationarity of the cost per package in Section 4.1.3 is applied. The results are shown in Table 4.2, with statistically significant results at the 5% level highlighted in blue. From the results, it can be concluded that the stationary time series are *forecast realisation ratio, hit rate, min max volume ratio, num delivery days, volume, volume BE ratio, volume international ratio* and that the remaining features are stationary in the first difference.

Feature	**Level Data**		**First Difference**	
	ADF	KPSS	ADF	KPSS
avg distance	−0.754	0.760***	−3.904***	0.252
avg DM3	−2.471	1.412***	−6.817***	0.032
avg weight	−2.574	0.965***	−5.651***	0.052
cost per package	−2.773	1.098**	−9.137***	0.113
diesel price	−2.113	0.568**	−6.455***	0.122
forecast realisation ratio	−6.937***	0.115	−6.78***	0.147
hit rate	−3.051**	0.407	−11.682***	0.359
inflation	−0.979	1.158***	−10.595***	0.175
min max volume ratio	−5.751***	0.315	−6.535***	0.225
num delivery days	−8.817***	0.042	−7.331***	0.079
volume	−4.491***	0.176	−6.221***	0.084
volume be ratio	−3.846***	0.228	−6.380***	0.063
volume main client ratio	−2.608	1.109***	−19.893***	0.033
volume export ratio	−5.991***	0.162	−6.442***	0.066
volume international ratio	−2.138	1.118***	−9.689***	0.091

Table 4.2: Unit Root Test Results on level data

*** statistically significant at the 1% level.
** statistically significant at the 5% level.

To ensure valid results, the tests are performed using level data for the features that are stationary in their level form, and differenced data is used for the features that are not stationary in their level form.

Cointegration
If the data is non-stationary, there is an additional assumption that must be met for the Granger causality test to be valid. This assumption is that the time series are not cointegrated. Cointegration is a statistical property of a collection of time series variables. Two or more time series are cointegrated if they share a common trend, meaning that they form a stationary time series together. [1]. In cases of cointegration, the autoregressive part of the Granger causality test should be performed using Vector Error Correction Models (VECM) instead of a VAR model. The VECM accounts for the long-run equilibrium relationship between the series, ensuring validity of the results [13].

The Johansen Trace Test is commonly used to test for cointegration among time series. The null hypothesis states that there are at most r cointegrating vectors, while the alternative hypothesis suggests there are more. By iteratively evaluating different values of r, the test identifies the number of cointegrating vectors. It does so by assessing the eigenvalues of a matrix derived from the data. Each eigenvalue significantly larger than zero corresponds to one cointegrating vector. If no significant eigenvalues are found, it indicates that there is no cointegration among the time series, and the Granger causality test is valid [29].

The Johansen Test is applied to each combination of the cost per package and a non-stationary feature. Since the evaluation is on pairs, $r = 0$ and $r = 1$ are relevant. The results, presented in Table 4.3, show that almost all test results are significant. The only exception is *inflation* at $r = 1$. This suggests that each non-stationary feature has at least one cointegrated feature with the cost per packag and thus a VECM should be used for valid Granger causality testing.

Feature	$r = 0$	$r = 1$
volume main client ratio	34.363**	9.43**
volume NMG ratio	23.733**	7.102**
avg DM3	47.866**	19.882**
avg weight	32.458**	12.146**
avg distance	34.237**	12.257**
inflation	27.762**	4.821
diesel price	20.628**	6.441**

Table 4.3: Cointegration test results
** statistically significant at the 5% level.

Granger Causality Test Results
After addressing stationarity and cointegration, Granger causality can now be applied. To do this, one value has to be specified which is the lag order of the autoregressive models. This value determines the number of lagged values of the regressors included in the model. The Akaike Information Criterion (AIC) is most commonly used to determine the optimal lag value by evaluating various lag lengths and selecting the one that minimizes the AIC value [13]. This criterion balances model fit and complexity, penalizing the number of parameters to prevent overfitting. By comparing AIC values across different lag lengths, the lag with the lowest AIC value is chosen as it represents the best trade-off between goodness of fit and simplicity [61]. With the specification of the lag order, the autoregressive models can be fitted and the Granger Causality test can be applied. The results of the Granger causality test and the selected lag values are indicated in Table 4.4. The features that show are statistically significant result are *avg dm3, avg weight, forecast realisation ratio, hit rate* and *volume*.

Whiteness
The final criteria for validity states that the residuals of the regression are white noise. White noise is a sequence of uncorrelated random variables with a mean of zero and a constant variance [43]. The residuals are the difference between the actual value and the predicted value. The residuals are tested for whiteness using the Portmanteau test as is often done [41]. The null hypothesis of the Portmanteau test is that the residuals are white noise, so for validity of the test this hypothesis should be accepted. The results of the whiteness test are shown in the "Whiteness" column in Table 4.4. None of the features have significant results, which means that the residuals are white noise and the Granger causality test is valid [13].

Feature	Causality	Lag	Whiteness
avg distance	0.776	5	15.293
avg DM3	2.374**	5	11.399
avg weight	2.723**	4	35.777
diesel price	1.99	3	35.22
forecast realisation ratio	2.784**	5	12.12
hit rate	3.268***	4	44.329
inflation	0.247	1	47.288
min max volume ratio	1.926	3	23.106
num delivery days	0.389	1	26.907
volume	2.906**	4	19.658
volume main client ratio	0.417	2	56.919
volume international ratio	0.046	1	24.501
volume NMG ratio	1.89	5	18.185

Table 4.4: Granger Causality test results Cost per Package

*** statistically significant at the 1% level.
** statistically significant at the 5% level.

4.2.4. LPCMCI

As a final method, an approach that addresses all considerations mentioned in Section 4.2.1 is sought. This implies that latent confounders must be accounted for, time dependencies captured, and only direct causation evaluated. Unlike correlation and Granger Causality which only consider pairwise relationships, a multivariate method that captures the entire dataset is needed. Given these requirements, LPCMCI was chosen. LPCMCI is effective in identifying direct causation and accounting for latent confounders [18]. Despite being more sophisticated, LPCMCI has high detection power in small datasets and can handle the time-dependent structure of the data effectively [19]. Furthermore, it has a reliable and well-documented implementation in the Tigramite[4] package, adding to its practicality for this analysis. All these considerations make it a suitable option for finding more nuanced relationships beyond simple correlations and predictive causation.

Method

LPCMCI falls into the conditional independence based models of causal discovery. It is a graphical model, which means it uses a graph-based representation to show the conditional dependencies between variables. The algorithm begins by creating a fully connected graph. This graph considers all potential links between variables at different time lags. The number of time lags to be considered is a parameter that needs to be specified. In the case of direct causation, this parameter is set to 0 [18].

After establishing these initial links, the algorithm iteratively refines the graph using conditional independence tests of two variables while controlling for the influence of other variables [19]. If two variables are found to be conditionally independent, the corresponding edge is removed. This process continues until no more edges can be removed. The final output is a graph with only the significant causal links, represented by an adjacency matrix. The values in this matrix correspond to the test results the conditional independence test, indicating the strength and direction of causal relationships. Positive values indicate a positive relationship, while negative values indicate a negative relationship [51].

LPCMCI Results

The LPCMCI algorithm assumes stationarity. From Table 4.2, it is observed that all features are stationary in the first difference. Therefore, the algorithm is applied data in the first difference and the outcomes are shown in Figure 4.5.

Two relationships with the cost per package are established. A positive relationship is observed between the cost per package and the *forecast realisation ratio* and a negative relation is found for volume and cost per package.

When inspecting the data for the *forecast realisation ratio*, it was found that the value is often above 1, indicating that the forecasted volume is higher than the realised volume. In those cases, planning is based on

[4] https://github.com/jakobrunge/tigramite

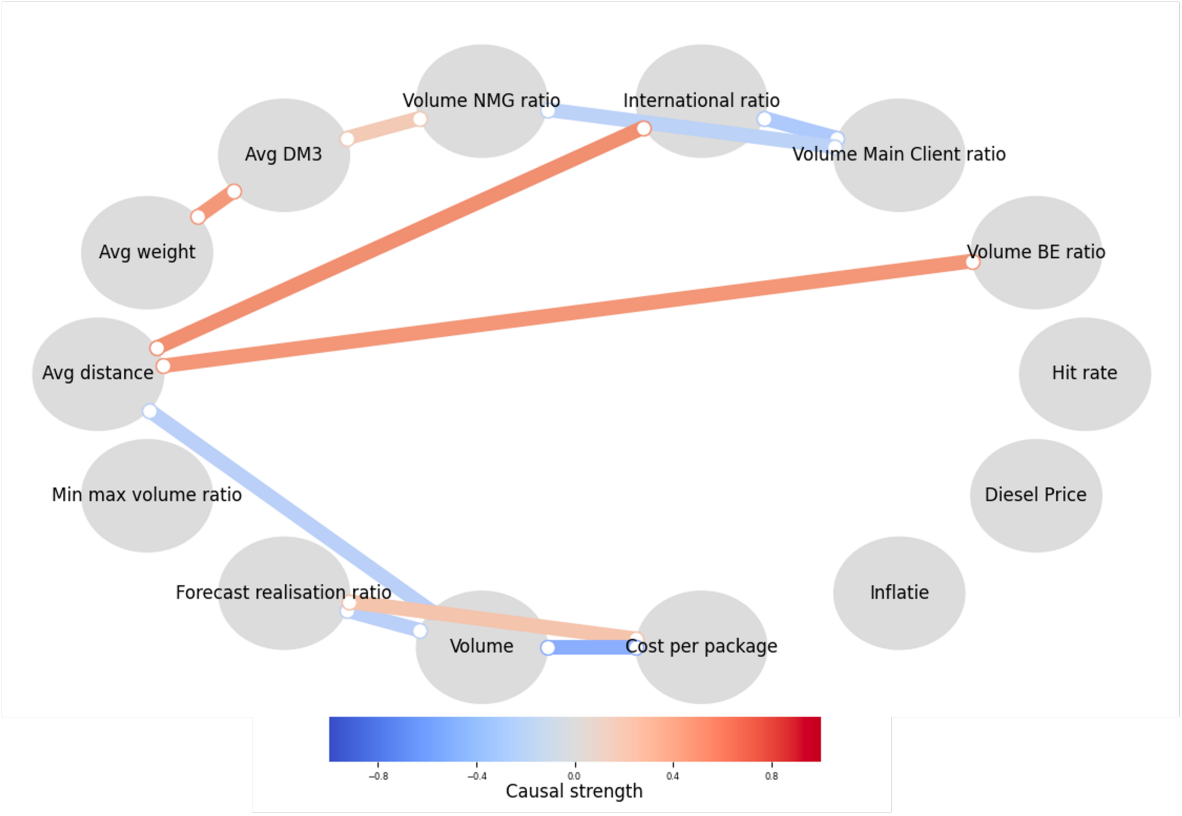

Figure 4.5: LPCMCI Results

these inflated forecasts and the related costs need to be divided over less volume, leading to higher costs per package. This explains the positive relationship as observed in the results of LPCMCI. The negative relation between volume is explained by the scaling advantages as was already discussed in Section 4.2.2.

To ensure the validity of the results it is also interesting to look at the relationship between the features. As logical relationships among features increase confidence in the reliability of the test results.

The positive connection between *average weight*, *average dm3*, and *ratio NMG* is an expected outcome. Heavier packages are likely to correspond with larger packages, and larger packages are more likely to fall outside the specifications for a machinable good. Whereas, the negative relationship between non-machinable goods and the packages sorted in the main client sorting centre is can be attributed to the average size of their packages. Generally, packages are smaller than those processed in other sorting centres, making them less likely to be non-machinable. In contrast, packages from the main client are mostly sent to Belgium and the Netherlands, explaining the negative relation with share of packages that is sorted at the international sorting centre.

The average distance has significant positive relationships with *International ratio* and *BE ratio*. The is only one sorting centre for international shipping. Therefore, every package originating from anywhere outside the Netherlands should pass through there, increasing the distance. Similarly, packages destined for Belgium must all be shipped to Belgium. The only negative relation for *average distance* is found for *volume*. As volume increases, the spread of volume will expand in proportion to the usual spread. This means that in areas like the Randstad, the volume will see a larger increase. As a result, packages become more concentrated, leading to a decrease in the average distance travelled.

After critical evaluation, it appears that no significant relationships between the features is missing, which further strengthens the reliability of the findings.

4.3. Summary of Key Findings

The data analysis performed in this chapter provides valuable insights into the factors influencing the cost per package. Initially, through descriptive statistics and time series analysis, it was observed that the cost per package exhibits volatility, shows a slight upward trend, and follows seasonal patterns. These findings suggest that the cost per package is not constant and is influenced by other factors. Additionally, the autocorrelation analyses revealed significant temporal dependencies, indicating that past values of the cost per package are informative of future values up to a lag of four periods. This validates the choice to interpret the data as a time series rather than as independent data points.

By applying a combination of statistical methods, valuable insights into the factors influencing the cost per package were found. The correlation analysis provided an initial orientation into the dependencies in the data. Two notable variables, *avg dm3* and *volume*, showed the strongest correlations with the cost per package, which is expected given their direct influence on costs.

The correlations were generally consistent with the hypothesized effects, confirming the correctness of the data. The only unexpected results were for both *inflation* and *diesel price*, where a positive correlation with the cost per package was expected, but the results showed negative correlations. This suggests that, if these variables are found to be significantly relevant, their interpretation requires caution.

The Granger causality tests were employed to check for predictive causation for each pair of features with the cost per package. Significant results were found for *average size*, *average weight*, *forecast realisation ratio*, *hit rate*, and *volume*. However, this method's limitation lies in its inability to account for latent confounders and its focus on pairwise relationships.

The LPCMCI analysis addressed these limitations by identifying direct causal relationships while accounting for latent variables and time dependencies. Additionally, this method provides a measure for causal strength, offering extra information on the direction and strength of the relationships found. LPCMCI confirmed the positive relationship between the cost per package and the *forecast realisation ratio*, as well as the negative relationship with *volume*. These findings align with earlier results.

Comparing the results, *forecast realisation ratio* and *volume* form the set of overlapping findings from both Granger causality and LPCMCI analyses. The strong correlation and significance in multiple tests confirm that volume significantly impacts the cost per package, an expected but important validation. The *forecast realisation ratio* similarly shows consistent significance, highlighting its importance.

Additionally, *avg dm3* showed a strong correlation and significance in the Granger causality test. However, more evidence is needed to conclusively determine its influence.

In conclusion, this analysis has explored the key factors influencing the cost per package. So far, it seems that the precision of the forecast and volume have the most influence on the cost per package. These findings will be further compared in the next chapters, which will focus on forecasting costs and assessing feature importance to further explore the interactions between the features and the cost per package.

5

Methodology for Cost Prediction

This chapter focuses on the development and evaluation of various predictive models to achieve this goal. The process begins with the selection of appropriate models, taking into consideration the characteristics of the dataset, as discussed in the model selection section. Following this, the dataset used for training and evaluating these models is detailed. The model optimisation process, including hyperparameter tuning and feature selection, is then outlined. Finally, the methods used for evaluating the models are presented.

5.1. Model Selection

The selection of models for this comparative analysis was informed by the inherent characteristics of the dataset. Consisting of only 118 data points, the dataset is relatively small. This is an important consideration since, in line with the curse of dimensionality, the use of overly complex models can prove to be counterproductive. These complex models are prone to overfitting when applied to small datasets, which means they might perform well on the training data but fail to generalise to new, unseen data [35]. Hence, simpler models are likely to be more effective in this context.

Additionally, the data for the cost per package exhibits volatility and contains outliers as was already shown in Figure 4.1. These characteristics can skew the results of predictive models, leading to poor performance if the models are not able to handle such variations [11]. Therefore, selecting models that can mitigate the effects of these fluctuations will probably increase forecasting accuracy.

Another important thing to consider is the temporal aspects of the dataset, as it introduces additional complexities such as trends, seasonality, and potential cyclic behaviour. The models should be capable of accounting for such time dependencies inherent in the data to make accurate predictions. One approach to achieve this is by employing time series models, which are specifically designed to handle data indexed in time order. Alternatively, non-time series models can be modified to account for time dependencies. Both options are explored in this thesis to determine the most effective approach for forecasting the cost per package. This will be elaborated on in the sections 5.1.1 and 5.1.2.

In addition to considering dataset characteristics, the selection of models was also influenced by their interpretability. Although at the point of time, the models were selected it was still uncertain to what extent the interpretability would be elaborated on in this thesis, the option was kept open. This was done because it could be valuable to The Company to understand the basis of the model's predictions and to understand the drivers behind the forecasted values of the cost per package. Especially in cases when the predictions deviate from the expected values of the cost per package. By offering clarity on which variables most significantly impact the predictions, the analysis becomes more transparent, facilitating a more informed decision-making process. Consequently, models equipped with mechanisms for assessing feature importance were prioritized.

5.1.1. Time Series Models

Time series models are specifically designed to manage data indexed in time order, making them well-suited for capturing temporal dependencies and patterns. Traditionally, many time series models are univariate,

focusing solely on historical values of the target variable to make predictions. However, in this study, more data is available. Potentially, this data can be used to improve the accuracy of the predictions. Therefore, a choice was made to use multivariate models to predict the cost per package.

SARIMAX
The first model selected for this thesis is SARIMAX model which stands for Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors. The SARIMAX model is a classic time series model that extends the Autoregressive Moving Average (ARMA) model by incorporating integration, seasonality, and exogenous variables. The ARMA model itself is a combination of two simpler models: the Autoregressive (AR) model and the Moving Average (MA) model. The AR model assumes that the current value of the series can be explained as a function of p past values. The MA model, on the other hand, assumes that the current value of the series can be explained as a function of the error terms from q past values. The ARMA model combines these two models to capture the temporal aspects of the data. However, it assumes the data to be stationary and as seen in Section 4.2.3, this condition is not met by the data for the cost per package. The SARIMAX model addresses this through the integration feature, which differences the data to create a stationary series. Furthermore, the cost per package follows seasonality which is captured with the seasonal extension in SARIMAX, and finally, the addition of exogenous variables aligns with the requirement of a multivariate model [55].

Prophet and Neural Prophet
Included in the comparative analysis are two more modern time series models: Prophet and Neural Prophet. Prophet is designed to handle small to medium-sized datasets effectively. It is particularly useful when dealing with datasets with irregular patterns and missing data, and it works well for time series data that exhibit seasonal variations and trends [59]. These characteristics make it well-suited for the cost per package dataset.

Neural Prophet, an extension of Prophet, integrates neural networks with the aim to better capture complex patterns and relationships within the data [62]. Although in this thesis, there is a preference for simpler models due to the small dataset size, the inclusion of the Neural Prophet model in this analysis is intentional. The aim is to validate if it is true that extra complexity translates into decreased forecasting performance in this case. Furthermore, one consideration at the start of this project was to perform data augmentation to increase the size of the dataset. Part of this evaluation would be to see if a complex model like Neural Prophet would benefit from a larger dataset. This was an additional reason to include Neural Prophet in the analysis. Finally, both models have features to evaluate model interpretation and both allow for additional regressors to be included in the model which is desired.

5.1.2. Non-Time Series Models
Non-time series models are not explicitly designed to handle time series data. However, through appropriate modifications and feature engineering, these models can be adapted to account for the temporal structure of the data. This adaptation involves introducing features such as lagged variables, rolling averages, and time-specific indicators like day of the week, month, or cyclically encoded time variables. Lagged variables are essentially values of the target variable from previous time steps, providing insights into historical trends and patterns. Time-specific features help capture the periodic nature of certain variables, accommodating regular patterns that occur over specified time intervals. Additionally, rolling and moving windows can help smooth out short-term fluctuations and reveal underlying trends. These techniques average the values within a specified moving window across the dataset, thereby reducing the impact of anomalies and short-term irregularities on the model's performance. Together, these adaptations enable non-time series models to effectively capture and use the temporal dependencies, to improve their forecasting accuracy in time series data [27].

Linear Models
The group of models representing the non-time series approaches selected for this thesis includes linear models. These type of models assume a linear relationship between the input features and the target variable, offering a simple yet powerful means to understand how changes in the inputs affect the output. The simplest among these, Ordinary Least Squares (OLS) estimates the coefficients of the linear equation by minimizing the sum of the squared differences between the observed and predicted values [48]. The inclusion of this model serves as the baseline for comparison with more sophisticated models.

To enhance robustness, three additional linear regression models were introduced: Theil-Sen, RANSAC, and Huber regressions. Theil-Sen regression is resistant to outliers because it calculates slope estimates based on medians rather than means [60]. Random Sample Consensus (RANSAC) is an iterative algorithm designed to fit a model to the inliers within the data set, effectively disregarding the outliers [16]. Huber regression implements a Hybrid approach by combining squared error loss for data points close to the model's predictions and absolute error loss for outliers, thus mitigating their influence on the overall model fit [26].

Gradient Boosting Models
Finally, two advanced gradient boosting models, XGBoost and LightGBM, are included in the analysis. Gradient boosting is a machine learning technique that constructs a predictive model as an ensemble of simpler models, typically decision trees. The idea is that the simple models, known as weak learners, on their own have poor predictive performance, but when combined, they create a strong predictive model. The main principle behind all boosting algorithms, including gradient boosting, involves iteratively training these weak learners. Each subsequent model focuses on correcting the errors made by its predecessors. In gradient boosting, this correction is achieved by fitting new models to the residual errors of the previous models, thereby improving the model's accuracy over time. This technique uses gradient descent optimisation to minimize a predefined loss function, resulting in an iterative process that progressively enhances the model's predictive performance [12].

XGBoost and LightGBM are two popular implementations of gradient boosting algorithms. The difference between the two models lies in their implementation details and performance characteristics. XGBoost employs a more regularized model formulation to prevent overfitting, while LightGBM uses a histogram-based approach to split nodes in the decision tree, resulting in faster training times [33]. The models were chosen because they are known for their effectiveness in handling complex data relationships [34] and are commonly used in time series forecasting [64, 65]. Additionally, they offer mechanisms to assess feature importance, and they have been successfully used for other projects within The Company, which makes them a natural choice for this study.

5.1.3. Model Implementations
Each of the selected models was implemented using the Python programming language, leveraging popular libraries to enhance the project's efficiency. The implementation of SARIMAX was done using the `pmdarima`[1] package which provides an interface known as auto ARIMA. This feature enables the automatic selection of optimal model parameters through a stepwise search algorithm that is adaptable to ARIMA, SARIMA, and SARIMAX models depending on the supplied arguments. Specifically for the SARIMAX model, the seasonal parameter was set to $True$ during class initialisation, and exogenous variables were incorporated during the model fitting and evaluation phases.

The Prophet and Neural Prophet models were implemented by their respective original `Prophet`[2] and `Neural Prophet`[3] package. The linear models were implemented using the `Scikit Learn`[4] package, which is a well-known library for machine learning in Python. The gradient boosting models, XGBoost and LightGBM, were implemented using the `XGBoost`[5] and `LightGBM`[6] packages, respectively. These packages are widely used for gradient boosting and provide efficient implementations of the algorithms.

5.2. Dataset
After selecting the models, the next step involved preparing the dataset for training and evaluation. This dataset builds upon the one developed in the previous chapter, which identified potential causes of fluctuation in the data. These identified causes now serve as potential features for further analysis. However, some adjustments were made to the dataset to ensure it was suitable for time series forecasting. The adjustments include feature engineering and preprocessing steps, which are detailed in the following sections.

[1] `https://pypi.org/project/pmdarima/`
[2] `https://facebook.github.io/prophet/`
[3] `https://neuralprophet.com/`
[4] `https://scikit-learn.org/stable/`
[5] `https://xgboost.readthedocs.io/en/stable/`
[6] `https://lightgbm.readthedocs.io/en/latest/`

5.2.1. Feature Engineering

As detailed in Section 5.1.2, additional modifications were necessary to adapt the dataset for time series forecasting using non-time series models. To achieve this, lagged features, window features, and time-related features were included in the dataset.

Lagged Features

The inclusion of lagged features involves adding four additional columns to the dataset, each representing the cost per package at different lagged time points ranging from one to four weeks in the past. The decision to use a lag of up to four weeks was informed by the autocorrelation analysis as detailed in Section 4.1.2. For the cost per package, the ACF plot in Figure 4.2 reveals significant correlations at lags 1, 2, 3, and 4. This observation validates the choice to include these specific lagged features in our dataset, as these past values appear to have a statistically significant impact on the current cost per package, thereby providing valuable predictors for forecasting future costs [27].

Rolling Window Features

The window features consist of the mean and standard deviation of the cost per package over a rolling window. Again, the choice of a window size of four weeks was informed by the results of the autocorrelation analysis. The aim of these two features is to smooth out short-term fluctuations and highlight underlying trends in the data. By averaging the values over a four-week period, the model can focus on the broader patterns and tendencies in the cost per package, reducing the impact of short-term anomalies on the predictions. The standard deviation feature provides insights into the variability of the cost per package over the same window. This information can be valuable for understanding the stability and consistency of the cost per package over time [9].

Time-Related Features

The time-related features include encodings of the week number and year each encoded in ways that enhance model interpretation and performance. The week number is encoded using sin/cos encoding. In this encoding, the week number w is transformed into two separate variables x and y, calculated as follows:

$$x = \sin(\frac{2\pi w}{52}) \ , \ y = \cos(\frac{2\pi w}{52})$$

In this formula, the denominator 52 represents the total number of weeks in a year, denoting the length of a full cycle. This sin/cos transformation allows the model to more effectively capture the cyclic nature of weeks throughout the year [42].

Finally, the year is encoded as a centred variable, which is calculated by subtracting 2022 from the actual year. This transformation makes sure the year starts at zero since 2022 is the first year in the dataset. Such transformation makes sure that potential scaling issues are avoided and that the model can capture long-term trends in the data [9]. By including these time-related features, the aim is that the model can account for seasonal patterns and trends that may influence the cost per package over time.

Number of Delivery Days

In addition to the extra features added to the dataset to adapt to non-time series models, one other feature was introduced: the number of delivery days within a week. The number of delivery days can be different per week due to various factors. For instance, during national holidays, fewer delivery days are scheduled. Conversely, during peak periods such as Black Friday, Sinterklaas, and Christmas, an extra delivery day may sometimes be required. This could influence the cost per package in two ways. Firstly, more delivery days in a week could distribute the fixed costs over more packages, potentially lowering the cost per package. Second, the number of delivery days can impact the volume forecast. An additional delivery day would increase the forecasted volume for that week, while a day less would decrease it. Given the expectation that the volume forecast significantly impacts the cost per package, the number of delivery days could provide additional context to the models. For example, in a scenario where the volume forecast is low, but the number of delivery days is also low, the cost per package might be similar to a situation with a higher volume and more delivery days. Therefore, it is hypothesized that this feature could help the models make more accurate predictions.

5.2.2. Preprocessing

The next step is to preprocess the data to ensure that it is in a suitable format for training the models. The only preprocessing step that is applied to the dataset is normalisation. The goal of normalisation is to ensure that all features are on a comparable scale, preventing any one feature from dominating the model's predictions. Two normalisation methods were considered: scaling and standardisation. Scaling involves transforming the data to a specific range, typically between 0 and 1, while standardisation involves transforming the data to have a mean of 0 and a standard deviation of 1 [3]. For this thesis Min-Max scaling was chosen because it only scales the data and therefore preserves the shape of the original data. This is important because the interpretation of the model's predictions remains consistent with the actual values of the cost [52]. The normalisation was done using the `MinMaxScaler` function from the `Scikit Learn` package and was applied to all features except for the time-related features described in section 5.2.1, which were already in a suitable range.

5.2.3. Train Validation Test Split

Following the preprocessing steps, the dataset was divided into training, validation, and testing sets. The validation set is included, because it is used for hyperparameter optimisation and feature selection processes included in the pipeline. Details on both hyperparameter tuning and feature selection will be elaborated upon in sections 5.3.2 and 5.3.3 respectively.

Determining the size of each dataset portion was not trivial due to the limited size of the dataset. Ideally, the training set should be as large as possible to make sure the model can capture the patterns in the data, during training. However, the testing set must also be large enough to ensure a reliable evaluation of the models on unseen data. Furthermore, the validation set needs to be adequately sized to make sure that the tuning of hyperparameters and the feature selection process are as generalizable as possible.

To balance these requirements, it was decided to use a full year (52 weeks) of data for training to capture the seasonal impacts fully. Given the total dataset size of 116 weeks, this leaves 64 weeks to be distributed between the validation and testing sets. The test set is assigned 40 weeks, leaving 24 weeks for the validation set. This division of data ensures there is enough data for effective training and evaluation, aiming to optimize the models' accuracy and generalizability. This split of 52 weeks for training, 24 weeks for validation, and 40 weeks for testing is visualized in Figure 5.1. This figure illustrates the distribution and timeline of the data across the different sets.

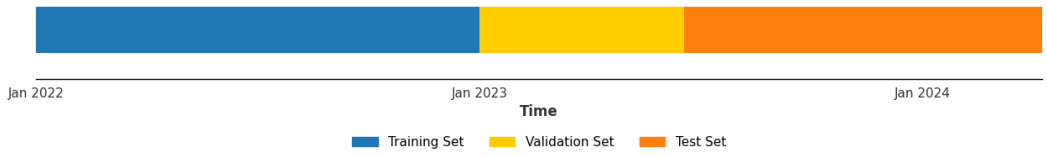

Figure 5.1: Train Validation Test Split

5.3. Model Optimisation

To optimize the performance of the models, both hyperparameter tuning and feature selection were conducted. As was already mentioned in the previous section, cross-validation was employed in both of these processes. This section will first provide a explanation of the cross-validation procedure, followed by a detailed explanation of its application in hyperparameter tuning and feature selection.

5.3.1. Cross validation

Cross-validation is a technique used to evaluate the performance of a predictive model. Typically, cross-validation involves partitioning the data into multiple subsets, training the model on a portion of the data, and then evaluating it on the remaining data. This process is repeated multiple times, with each subset serving as the test set once. The results are then averaged to provide a good estimate of the model's performance [28].

In addition to model evaluation, cross-validation is also used in model optimisation, as is done in this thesis.

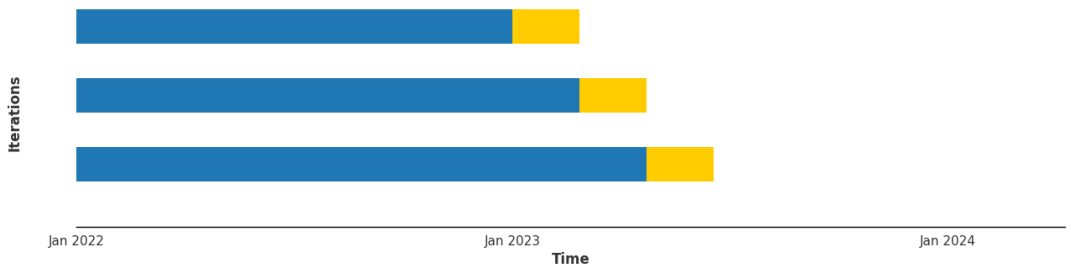

Figure 5.2: Cross Validation

During this phase, the training and validation sets are used to assess the model's performance under various settings. The configurations that yield the best average performance across each fold are selected for the final model setup. In this way, the parameters are adjusted to enhance model performance without overfitting the training data. This ensures that the model generalises well to new, unseen data. After determining the best configurations using cross-validation, the model can be retrained on the full training dataset. The final evaluation can be performed on the test set that was not used during the cross-validation process [7].

Time series data presents a challenge for cross-validation due to its temporal nature. In traditional cross-validation, data points are randomly shuffled and divided into folds. However, in time series data, the order of the data points is important, as the past values of a variable are used to predict future values. Shuffling the data would break this temporal structure, leading to inaccurate model evaluation [8]. To address this issue, a special form of cross-validation, adapted for time series, is used. In this method, the data is sequentially split into folds, ensuring that the order of the data points is preserved [27]. Figure 5.2 illustrates this approach, showing the distribution of data into three folds while adhering to the temporal structure of the data.

The three folds illustrated correspond to an evaluation subset of 8 weeks each, fitting into the validation period of 24 weeks. The project at hand requires a forecast window of 6 to 8 weeks. This makes 24 strategically chosen as it is divisible by both 6 and 8. This allows for three folds of cross-validation with an 8-week forecast window and four folds with a 6-week forecast window. Initially, an 8-week forecast window was selected to provide the broadest analysis scope.

5.3.2. Hyperparameter Tuning

The hyperparameters of the models were tuned using a grid search approach. This method involves defining a grid of hyperparameters and evaluating the model's performance for each combination of hyperparameters. The performance evaluation is done using cross-validation, as described in the previous section. The hyperparameters that were tuned for XGBoost and LightGBM, Prophet, Huber Regressor and Ransac Regressor. The Ordinary Least Squares Regressor and the Theil-Sen regressor do not require hyperparameter tuning. Sarimax does have hyperparameters that are very important for accurate model performance, however with the choice for the auto Arima implementation, the hyperparameters are automatically tuned.

The parameter grids that were used for each model are overviewed in Appendix C. The grid was chosen based on the model's characteristics and the hyperparameters that are most likely to impact the model's performance. The small dataset size was also an important factor in defining the grid. Multiple values for each hyperparameter were selected to cover a wide range of values for each hyperparameter, allowing for a comprehensive search of the hyperparameter

5.3.3. Feature Selection

Feature selection refines machine learning models by identifying the most relevant subset of features within set of available features. It reduces the dimensionality of the data, which can decrease the computational cost and improve the performance by removing irrelevant or redundant features. Moreover, it enhances model interpretability and helps in avoiding the curse of dimensionality, potentially leading to more robust and accurate predictions [31].

The methods for feature selection can be divided into three categories: Filter, wrapper and embedded meth-

ods. Filter methods assess features based on statistical measures without involving any machine learning algorithms, making them fast but less tuned to specific model aspects. In contrast, wrapper methods evaluate feature subsets based on the performance of a specific predictive model. They treat the model as a black box and use its performance as the basis for selecting features. Wrappers typically provide better performance than filters because they evaluate features in the context of the model. However, they are computationally intensive and can be prone to overfitting. Embedded methods incorporate feature selection Directly into the model training process and are usually specific to given learning machines [23].

Additionally, Hybrid methods can be employed that combine the strengths of filter and wrapper methods [31]. That is the approach chosen for this thesis. This Hybrid approach involves applying a filter method to rank all feature subsets and then using a wrapper method on the top 25 feature sets to further refine the feature selection. In this way, it is ensured that the chosen features are tailored to the specific model while also being efficiently selected.

Correlation-based Feature Selection
The Correlation-based Feature Selection (CFS) algorithm serves as the filter method. It is a commonly used method with an intuitive interpretation. The underlying principle of CFS is that an ideal feature subset should contain features that are highly correlated with the output variable but not correlated with each other [30]. For a given feature subset S, the definition of CFS translates into the following formula [36]:

$$\text{Merits}_S = \frac{k \cdot \bar{r}_{cf}}{\sqrt{k + k \cdot (k-1) \cdot \bar{r}_{ff}}}$$

where k is the number of features in S, $\bar{r}_{cf}$ is the average correlation between each feature f in S and the output variable c, and $\bar{r}_{ff}$ is the average pairwise correlation between the features in S.

In the dataset at hand, there are a total of 21 features, including the lagged variables and the time-related features. This gives rise to a total of 2^{21} possible feature subsets. That is a large number of possible subsets to evaluate, and therefore an exhaustive search is not feasible. Instead, a search algorithm is used to identify the optimal feature subset.

Forward Selection
While there are many search strategies, there are four usual starting points for feature subset generation: forward selection, backward elimination, bidirectional selection, and heuristic feature subset selection. Forward selection typically starts with an empty feature set and then considers adding one or more features to the set. Backward elimination typically starts with the whole feature set and considers removing one or more features from the set. Bidirectional search starts from both sides - from an empty set and the whole set, simultaneously considering larger and smaller feature subsets. Heuristic selection generates a starting subset based on a heuristic (e.g. a genetic algorithm), and then explores it further [30].

In this thesis, a forward selection algorithm is employed to identify the top 25 optimal feature subsets. The approach is inspired by other research where CFS is also successfully combined with a forward selection [36, 32]. The forward selection process begins with an empty subset of features. It then iteratively evaluates each of the available features by adding them to the subset and calculating the merit score using the CFS merit score. The features are ranked based on their scores, and the top 25 are retained. For each of these top 25 features, the algorithm considers the addition of all other features. So in this case, for each of the top 25 features, the algorithm considers 20 subsets (21 features minus the one already in the top 25). For each subset, the merit score is calculated. Based on that score, all the considered subsets are then sorted, and the top 25 are kept. This process is repeated until all features have been considered. The result is a list of the top 25 feature subsets according to the CFS merit score. This approach ensures that the selected features are not only individually predictive but also complementary to each other. To make the algorithm more efficient it is ensured that the same feature subset is not considered twice. This is done by keeping track of the visited feature subsets. The detailed implementation steps of the algorithm are outlined in Algorithm 1.

The forward selection algorithm is run twice, once for the features of time series models and once for the features of non-time-series models. The difference is in the lagged features and time-related features that are included for the non-time-series models but not for the time series models. The results of both runs are then used as input for the wrapper method.

Algorithm 1 Forward Selection for Feature Subset Selection

```
Input: X (feature set), y (target variable),
Output: top_25_features (list of top 25 feature subsets)
available_features ← X.columns
top_25_merits ← [(0, [])] * 25                    ▷ Initialize with 25 entries of (0, [])
candidates ← [[]]                                 ▷ Start with an empty feature subset
visited ← []
while candidates is not empty do
    new_candidates ← []
    for each feature_set in candidates do
        for each f in available_features do
            if f not in feature_set then
                new_candidate ← feature_set + [f]
                if set(new_candidate) in visited then
                    continue
                end if
                merit ← calculate_merit(new_candidate, X, y)
                if merit > top_25_merits[−1][0] then
                    new_candidates.append(new_candidate)
                    top_25_merits.append((merit, new_candidate))
                    Sort top_25_merits in descending order by merit
                    top_25_merits ← top_25_merits[: 25]
                end if
                visited.append(set(new_candidate))
            end if
        end for
    end for
    candidates ← new_candidates
end while
top_25_features ← [features for (_, features) in top_25_merits]
return top_25_features
```

Wrapper Method

The final stage of the feature selection comprises the evaluation of the top 25 feature subsets using a wrapper method. The wrapper method involves training and evaluating the models using each of the top 25 feature subsets. The evaluation is done using cross-validation, where the results over the various folds are averaged. The performance of each candidate feature subset is then compared, and the subset that yields the best performance is chosen. This process is done for each model, resulting in an optimal feature subset per model.

5.4. Multivariate Forecasting approaches

Having established the dataset and the models, along with their optimisation, the next step is to define how to effectively use them to predict the cost per package. This task is characterized as a multivariate time series forecasting problem, where the target variable is predicted using a set of related variables.

5.4.1. Known and Unknown Features

Predicting future values of the target requires knowing the future values of these related variables. This poses a challenge, as not all future values of the available variables are known at the time of prediction. Therefore, the features are split into two categories: known features and unknown features at the time of prediction. The known features include the lagged values of the target variable, the time-related features, and the volume forecast. Although including the volume forecast may initially seem surprising, it is justified within the context of this project. The project description explicitly states that the objective is to predict the cost per package given the volume forecast. Moreover, The Company has established methods for accurately predicting the volume of packages, making the volume forecast a reliable and known value for this analysis. The remaining features are considered unknown at the time of prediction. An overview of the known and unknown features is provided in Table 5.1.

Known Features	Unknown Features
Lag 1 cost per package	Average weight
Lag 2 cost per package	Average DM3
Lag 3 cost per package	Volume NMG ratio
Lag 4 cost per package	Hit rate
Number of delivery days	Avg distance
Week number (X coordinate)	Ratio forecast / realisation
Week number (Y coordinate)	Diesel price
Year	Inflation
Volume forecast	Ratio min / max volume
	Ratio Belgium
	Ratio main client
	Ratio international
	Ratio export

Table 5.1: Overview of known and unknown features at the time of prediction.

5.4.2. Considered approaches

Based on the categorisation of known and unknown features, three different approaches to forecasting the cost per package are considered:

1. **Direct Forecasting with Multiple Variables**: This approach involves forecasting the target variable directly using only the known variables as specified in Table 5.1. This method is straightforward and does not require any additional steps to forecast the cost per package. The question is whether the known features are sufficient to make accurate predictions of the target variable.
2. **Lagged Feature Modeling**: In this method, lagged versions of the features are used, meaning the features used are historical data points relative to the time of prediction. For example with a window of 8 weeks, the features are adjusted by shifting them back by 8 weeks. Consequently, for a forecast of the window spanning from $t = 0$ to $t = 8$, the features will range from $t = -8$ to $t = -1$. By using lagged versions of the features, this technique allows for the use of all available features to predict the target variable, without the need to forecast any of the features.
3. **Hybrid Forecasting with Feature Estimations**: This approach, uses both known and unknown features to predict the target variable. To overcome the problem of unknown features, forecasts or educated estimates are created for the unknown features before they are used in making predictions. The model is then trained using both these forecasted values and known historical values. However, the quality of the feature estimations is crucial for the accuracy of the predictions.

Each of these methods was evaluated to determine their effectiveness in predicting the cost per package. During some initial tests, it was found that the lagged feature modelling approach underperformed compared to the other two methods. This is likely due to the short-term fluctuations in the data, which made it challenging to capture the underlying trends when relying on feature information that is delayed by 8 weeks. As a result, the lagged feature modelling approach was excluded from the final analysis, leaving the Direct Forecasting and Hybrid Forecasting methods for further evaluation.

5.4.3. Hybrid Forecasting using Historical Replay

The Hybrid Forecasting approach requires future values of predictive features to be estimated first before they can be used for predictions. This could be a time-consuming process, as it involves creating forecasts for each of the features. To evaluate the effectiveness of this approach without the time investment required to create feature estimations, a historical replay technique could be employed first. This technique involves treating a past day as if it were today, allowing forecasts to be made and evaluated from that point. Since the 'future' data for this past day is already known, these values can serve as stand-ins for future estimations.

However, the historical replay technique does not provide an entirely accurate assessment of the hybrid approach because the true accuracy of future estimations also influences prediction quality. Nonetheless, it establishes a maximum expected performance level for the Hybrid model., which can be compared with the Direct Forecasting method to determine if further exploration of the Hybrid approach is useful.

Additionally, the pipeline incorporates feature selection. If the Hybrid model shows strong performance with a specific subset of features during the historical replay, only those features would need to be estimated for the actual implementation. This reduces the number of features that require estimation, and thus the data preparation process would be simplified.

5.5. Model Evaluation

The final step in the process is to evaluate the performance of the models. This section first defines the performance metrics and then explains the walk-forward validation technique that is used to evaluate the models. Finally, the method to gain insight into the performance of the models across different forecast offsets is described.

5.5.1. Performance Metrics

Before proceeding with model evaluation, it is important to define the metrics that will be used to assess their performance. Given that this project's findings need to be explained to The Company employees, both with and without technical expertise, only the most intuitive evaluation metrics were considered. To maintain consistency with other forecasting projects within The Company, where the error is typically expressed in percentages, the same approach is adopted here. Therefore, the main metric employed in this project is the Mean Absolute Percentage Error (MAPE). MAPE is a measure of the average difference between the predicted and actual values, expressed as a percentage of the actual value [58]. It is calculated as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

where n is the number of observations, y_i is the actual value of the target variable, and $\hat{y}_i$ is the predicted value of the target variable. Another metric included is the Mean Absolute Error (MAE). MAE is the average of the absolute differences between the predicted and actual values [58]. The MAE for the cost per package represents the error amount in euros. It is calculated as follows:

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

5.5.2. Walk-Forward Validation

In the process of model evaluation, a technique known as walk-forward validation is employed. This method is a specialized form of cross-validation, specifically adapted for time series data. It involves training the model on a rolling basis, and updating the model with new data points as they become available. The process is illustrated in Figure 5.3.

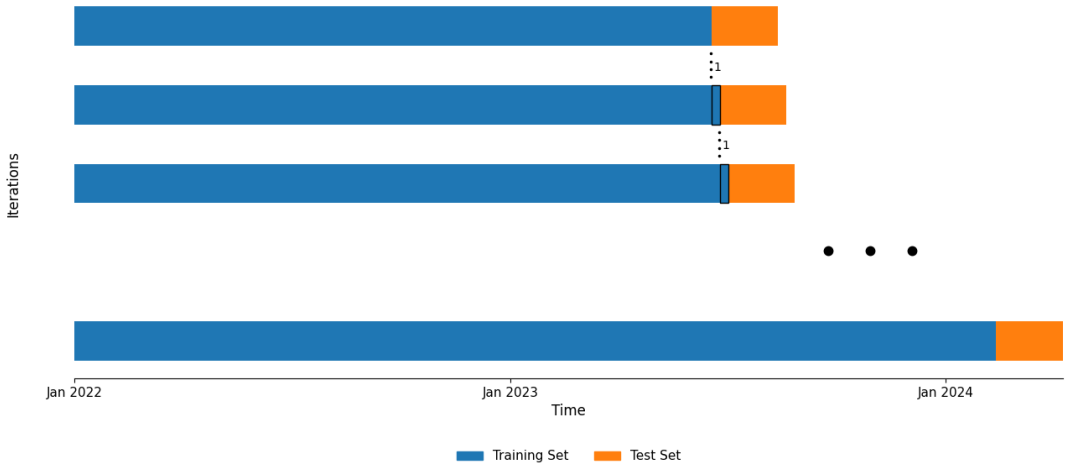

Figure 5.3: Walk-Forward Validation with an expanding window.

In the first iteration, the model is trained on the full training set as defined in Figure 5.1. Evaluation is done on the subsequent 8 weeks which complies with the 8-week forecast window. In the next iteration, the training set is extended by one data point, and the model is retrained on the updated training set. Again, evaluation is done on the subsequent 8 weeks. This process is repeated until all data points in the test set have been used for testing. The final result is the average score over all iterations.

By using this approach look-ahead bias is avoided [63]. Look-ahead bias occurs when information that was not available at the time of prediction is used to evaluate the model's performance. This leads to an overestimation of the model's accuracy, as it is being evaluated on data that was not known at the time of prediction. Additionally, this method provides a more realistic evaluation of the model's performance, as it simulates how the model would be used in a real-world scenario where new data points are continuously becoming available [37].

5.5.3. Standard Error

In addition to evaluating the average performance of the models using metrics like MAPE and MAE, it's important to understand the uncertainty associated with these performance metrics. This uncertainty is quantified using the standard error. The standard error provides a measure of the spread of sample means around the population mean.

In the context of this project, the standard error helps evaluate the reliability of the average MAPE across different folds of the walk-forward validation process. A smaller standard error indicates less variability in the performance metrics, suggesting greater confidence in the reported average performance. Conversely, a larger standard error implies more variability, which decreases the reliability of the reported performance metrics [2].

The standard error of the mean MAPE is calculated as follows:

$$\text{SE} = \frac{\sigma}{\sqrt{n}}$$

Where σ is the standard deviation of the MAPE values across the folds, and n is the number of folds. This formula shows that the standard error decreases as the number of folds increases, assuming the standard deviation remains constant. This is because a larger number of folds provide more data points for the calculation of the average MAPE, leading to a more accurate estimate of the true performance [2].

5.5.4. Significance Testing

Significance testing is used to determine whether differences in model performance are statistically significant. As explained in Section 5.5.2, model performance is expressed as the average score across the folds of walk-forward validation. This means that comparing the results of two models means checking if the means of two groups are significantly different. Two common methods for this purpose are the paired t-test and the Wilcoxon signed-rank test. Both tests are appropriate when dealing with paired measurements. Paired measurements occur when the same subjects are measured under different conditions, or when two related measurements are taken from the same subjects [44]. In the context of walk-forward validation, the first scenario applies. The folds of the walk-forward validation serve as the subjects, and the performance scores of the models on these folds are the measurements being compared. The specifics and implementations of each test are further detailed in the following sections.

Paired t-test

Besides the assumption of paired samples, the paired t-test requires the differences between the paired observations to be normally distributed and the observations to have equal variances [39]. To check for the normality of the differences, the differences between the scores of the two models are calculated for each fold. Then, the Shapiro-Wilk test, a well-known test for normality, is applied to these differences. The test has the null hypothesis that the data is normally distributed and the alternative hypothesis that the data is not normally distributed [20]. A common test to check for equal variances is Levene's test. This test has the null hypothesis that the variances are equal, while the alternative hypothesis is that the variances are not equal [40]. In case the assumptions are met, then the paired t-test can be used to compare the models. The calculation is as follows:

$$t = \frac{\bar{d}}{s_d / \sqrt{n}}$$

where $\bar{d}$ is the mean of the differences between paired observations, s_d is the standard deviation of the differences, and n is the number of pairs. The test statistic t is then compared to a critical value from the t-distribution to determine whether the difference in performance is statistically significant [39].

Wilcoxon Signed-Rank Test
If the assumptions for the paired t-test are violated, the Wilcoxon signed-rank test is a suitable alternative. It has the same purpose as the paired t-test but has less strict assumptions. The assumptions include an ordinal scale and symmetry of the differences.

The ordinal scale assumption requires that the data be measured at an ordinal level, meaning it can be ranked. Given that the average MAPE is a numerical value, which is ordinal by nature, this requirement is satisfied. Symmetry implies that the differences between the paired observations are symmetrically distributed around the median. While this is less strict than normality, it still requires the distribution of differences to be roughly symmetric. Visual inspection using a histogram and a box plot can be used to check for symmetry. Additionally, the skewness of the differences can be calculated to verify symmetry. A skewness near zero indicates symmetry. The latter method is used in this project to be able to express the symmetry in a numerical value.

The Wilcoxon signed-rank test is calculated through a series of steps. First, the differences between the paired observations are computed, and the absolute values of the differences are ranked based on their absolute values. Next, the signs are reintroduced to the ranks according to the sign of difference. So if the difference originally was negative, the rank will be assigned a negative sign. Then, the W-statistic is calculated by summing the ranks for positive differences and for negative differences separately. Finally, the W-statistic is compared to a critical value from the Wilcoxon signed-rank distribution to determine if the difference in performance is statistically significant [38].

Testing Procedure
Each time significance testing is performed, the assumptions of the paired t-test are first checked. If these assumptions are met, the paired t-test is used. If the assumptions are violated, the Wilcoxon signed-rank test is used, with the skewness value also reported ensuring validity. This approach ensures that the most appropriate test is used for each comparison and that the results are reliable.

The `Scipy`[1] package was used to implement the assumptions and the tests. `Scipy` provides reliable and well-documented functions that are easy to implement and is widely used in scientific communities. The Shapiro-Wilk test, Levene's test, and skewness are calculated using the `shapiro`, `levene`, and `skew` functions, respectively. The paired t-test and the Wilcoxon signed-rank test are performed using the `ttest_rel` and `wilcoxon` functions. These functions output a test statistic and a p-value, which is compared to a significance level of 0.05 to determine if the difference in performance is statistically significant.

For both tests, the null hypothesis is that there is no difference in performance between the models. For the alternative hypothesis, there are two options. Either the models perform differently, which requires a two-sided test, or one model performs better than the other, which requires a one-sided test. Since it is more relevant in this context to determine if one model outperforms the other, a one-sided test is used. Therefore, the p-value should be less than 0.05 to conclude that one model performs better than the other.

5.5.5. Forecasting Window and Offset Analysis

Initially, the models are assessed using an average score over a forecasting window of eight weeks. However, it is interesting to understand the models' performance across various forecasting offsets within those eight weeks. This understanding is necessary to find out if there is a big difference between various offsets and if a different forecasting window could yield better results. To do this the model's performance is evaluated across different forecast offsets. This implies that when a prediction is made at $t = 0$ for the next eight weeks, i.e., for $t = 1$ through $t = 8$, the performance of the model is evaluated individually for each forecast offset.

[1] `https://docs.scipy.org/doc/scipy/`

This was done for each fold in the walk-forward validation process. Then, the average performance score is computed, resulting in an average score per forecasting offset.

To ensure a fair and reliable evaluation, the first and last seven data points of the test set were excluded from this calculation. This was necessary because these data points do not have performance measurements across all forecast offsets. For instance, the first data point in the dataset is only evaluated in the first fold of the walk-forward validation before it transitions into the training set for subsequent folds.

This is illustrated in Table 5.2, which represents the case for a forecasting window of 4 weeks with a walk-forward validation of 7 folds. In the table, the letters a to j represent the data points in the test set, the rows represent the folds in the walk-forward validation, and the columns represent the forecast offsets.

	Offset 1	**Offset 2**	**Offset 3**	**Offset 4**
Fold 1	a	b	c	d
Fold 2	b	c	d	e
Fold 3	c	d	e	f
Fold 4	d	e	f	g
Fold 5	e	f	g	h
Fold 6	f	g	h	i
Fold 7	g	h	i	j

Table 5.2: Data Points Per Fold With a Forecasting Window of 4

It can be observed that only the data points d, e, f, and g are evaluated for all forecast offsets. Including the other data points in the evaluation would make the comparison between the various offsets unfair, as different data points would be included in the calculation of the average score for each offset. Moreover, it could introduce bias, as some data points might inherently be easier or more difficult to predict. Therefore, the first and last three data points (a, b, c, h, i, and j) would be excluded from the calculation of the average score per forecasting offset in this example with a forecasting window of 4 weeks. To translate this to the eight-week forecasting window, the first and last seven data points would be excluded from the calculation.

6

Results of Forecasting

This chapter outlines the performance results of the forecasting models for both Direct Forecasting and Hybrid Forecasting. The analysis begins with results of a naive method, establishing a baseline for comparison. Next, the models are trained and evaluated in detail, providing insights into their performance across various aspects, thereby improving the understanding of their effectiveness.

6.1. Naive Method

To establish a reference point for the models' performance and provide a benchmark for comparison, a Naive method is defined. Naive methods are simple and do not require any training, helping to determine the minimum performance that the models should achieve to be considered effective. Three methods were considered as the Naive Method.

- **Total Average**: The average value of the cost per package of the training set as the prediction for the next 8 weeks.
- **Last Value**: The last known value of the cost per packages as the prediction for the next 8 weeks.
- **Moving Average**: The average of the last four weeks as the prediction for the next 8 weeks, informed by the autocorrelation analysis in Section 4.1.2.

The results of these naive methods are presented in Figure 6.1. The MAPE for each method is calculated, and the average score across all folds in the walk-forward validation is displayed. The results show that the moving average method has to lowest MAPE and the smallest standard error. Therefore, this method is used as the baseline for the models and is referred to as the Naive Method from now on.

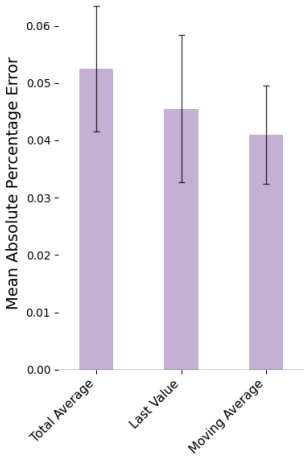

Figure 6.1: Naive Method Results

6.2. Results without Optimisation

Following the establishment of the Naive Method, the models were trained using default hyperparameters and all available features. These results provide a reference point, allowing to monitor progress and assess the impact of subsequent optimisation steps on the models' performance. Again, the models were evaluated using the walk-forward validation with a forecasting window of 8 weeks. The Mean Absolute Percentage Error for each model is calculated, and the average score across all folds in the walk-forward validation is displayed in Figure 6.2. The error bars represent the standard error of the mean of the folds. The figure provides an overview of the relative performance of the various models for both Direct and Hybrid Forecasting, intending to compare models within each method. Comparison between the methods will be addressed in Section 6.5.

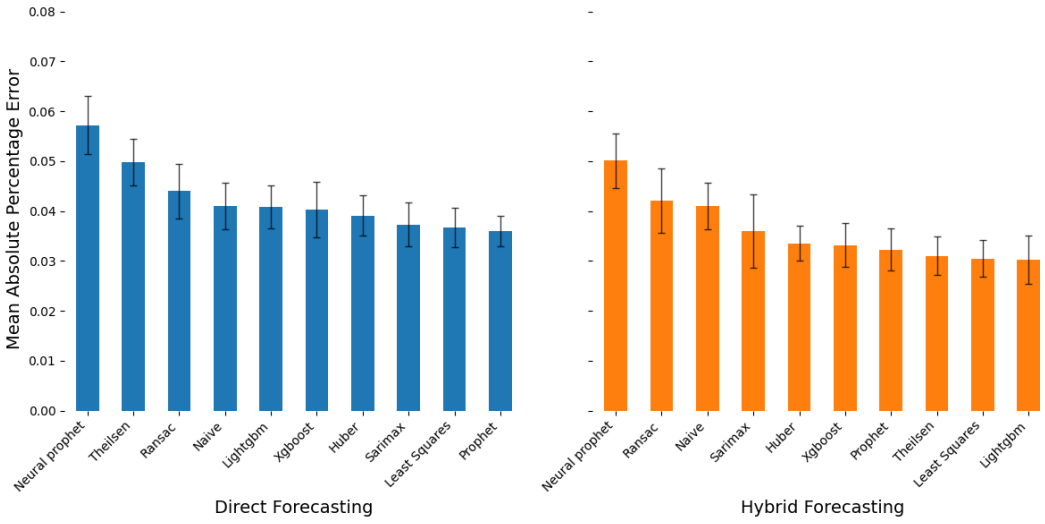

Figure 6.2: Unoptimized Results

The results show that the Prophet model performs best in Direct Forecasting, with an average MAPE of 0.0360. Whereas, LightGBM performs best in the Hybrid Forecasting with an MAPE of 0.0304. In contrast, Neural Prophet performs the worst with a score of 0.0572 and 0.0501 respectively. This is expected as the Neural Prophet model is more complex and requires more data to perform well. Besides Neural Prophet, the Theil-Sen and Ransac regressors have a higher average MAPE than the Naive method in Direct Forecasting. In Hybrid Forecasting, only Neural Prophet and Ransac underperform in comparison to the Naive method.

To assess the significance of these results, significance testing was performed using the scores per fold of the walk forward validation for each model. The null hypothesis is that the average MAPE is equal to the average MAPE of the Naive method, while the alternative hypothesis is that the average MAPE is lower. The results of the significance testing for both Direct and Hybrid Forecasting can be found in Appendix A.1. It turns out that for both Direct and Hybrid forecasting, the Huber regressor is the first model that performs significantly better ($p < 0.05$) than the Naive method. This means that so far for Direct Forecasting only four models are significantly better than the Naive method, and for Hybrid Forecasting six models are significantly better than the Naive method.

6.3. Results after Optimisation

Hyperparameter tuning and feature selection were applied for model optimisation. Both these processes require cross-validation, as described in Section 5.3.1. This also means that both processes are interdependent: hyperparameter tuning is performed using the optimal feature subset identified by the feature selection process, and vice versa. To overcome this circular dependency, the optimisation process was split into two iterations:

- **Iteration 1:** Feature selection using models with default parameters, followed by hyperparameter tuning using the optimal feature subset identified.
- **Iteration 2:** Repeating feature selection with the optimized models from the first iteration, followed by another round of hyperparameter tuning.

The results of feature selection and hyperparameter tuning did not change much between the first and second iteration. To avoid overfitting on the validation set, it was chosen not to perform a third iteration. The optimal parameters and features identified in the second iteration can be found in Appendix B and Appendix C respectively. These parameters and features were then used to evaluate the models using the walk-forward validation technique. The results are shown in Figure 6.3. The average MAPE across all folds in the walk-forward validation is displayed for each model. To maintain consistency with the unoptimized results, the y-axis is scaled to match the range of the unoptimized results.

The results show a shift in the best-performing models as compared to the unoptimized models. The Least

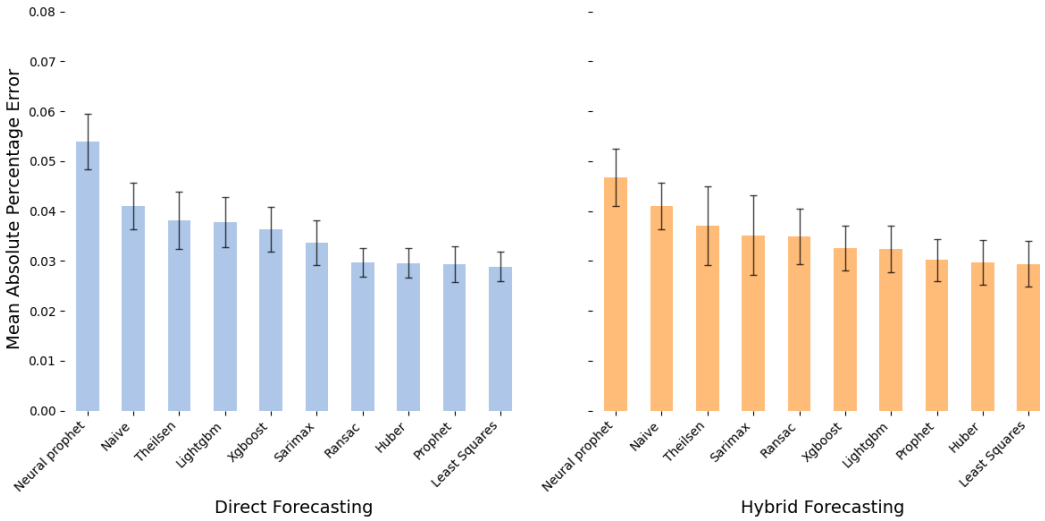

Figure 6.3: Optimized Results

Squares model now performs best in both Direct Forecasting and Hybrid Forecasting with an average MAPE of 0.0289 and 0.0294 respectively. For Direct Forecasting, this reflects an improvement of 0.071 percentage point difference of the best-scoring model compared to the unoptimized results. For Hybrid Forecasting, the improvement is 0.010 percentage points. Neural Prophet still performs the worst in both Direct and Hybrid Forecasting, with an average MAPE of 0.0539 and 0.0467 respectively.

6.3.1. Evaluating Model Improvements Over the Naive Method

The Naive method is second to last in both Direct and Hybrid Forecasting. Similar to the unoptimized models, significance testing was performed to determine which models significantly outperformed the Naive method. Detailed results can be found in Appendix A.2.

The results indicate that for both Direct and Hybrid forecasting, the XGBoost model is the first model that performs significantly ($p < 0.05$) better than the Naive method. In the context of Direct Forecasting, this implies that alongside Neural Prophet, Theil-Sen and LightGBM also fail to show a significant improvement over the Naive method. Similarly, in Hybrid Forecasting, this holds for the Theil-Sen, Ransac, and Sarimax models. Although the average MAPE for each of these models is lower compared to the Naive Method, the error bars reveal larger standard errors. This indicates higher uncertainty in the results, which explains why the significance tests do not show significance ($p > 0.05$).

However, still more methods showing significant results compared to the unoptimized models. Now, for Direct Forecasting six instead of four models are significantly better than the Naive method. For Hybrid Forecasting, six models remain significantly better than the Naive method.

6.3.2. Comparative Analysis of Best Performing Models

Additionally, significance testing was conducted to compare the performance of the best-performing models, as multiple models have similar average MAPE. Least Squares is the best-performing model, therefore the scores of this model form the basis of comparison. For each model, the null hypothesis of an equal average MAPE with Least Squares is tested against the alternative hypothesis that the average MAPE of Least Squares is lower. Details of this testing can be found in Appendix A.3.

For Direct Forecasting, the results show that the performance of the Least Squares model is not significantly ($p > 0.05$) different from that of the Prophet, Ransac, and Huber models. In Hybrid Forecasting, even more models are not significantly different from the Least Squares model, including XGBoost, LightGBM, Prophet, and Huber.

6.4. Effectiveness of Model Optimisations

To gain insight into the effectiveness of the model optimisations, the results of the optimized models are compared to the baseline results in Figure 6.4. This is shown in Figure 6.4 where for each model the blue bars are for the direct forecasting method and the orange bars for the hybrid forecasting method. The plot on the bottom shows the difference to provide an easy way to see the difference. This figure reveals that for almost all models, the MAPE was at least slightly reduced after optimisation. The only two exceptions are Theil-Sen and LightGBM in Hybrid Forecasting.

Significance testing was used to determine whether the optimisations led to a statistically significant improvement. The null hypothesis states that the optimized and unoptimized models have equal performance, while the alternative hypothesis states that the average MAPE of the optimized models is lower. Detailed results are presented in Appendix A.4. No significant improvement ($p > 0.05$) was found for any model in Hybrid Forecasting. However, in direct forecasting the effect was larger as the top four performing models plus Theil-Sen showed significant improvement ($p < 0.05$).

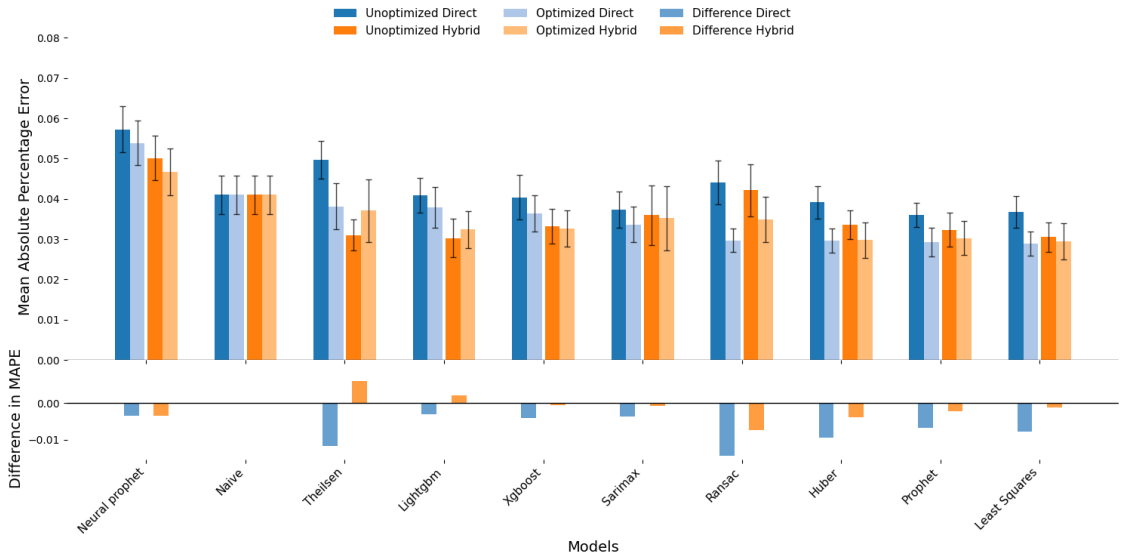

Figure 6.4: Unoptimized vs Optimized Results

6.5. Direct vs Hybrid Forecasting

The last comparison to be made is between the two different forecasting approaches: Direct and Hybrid. This analysis is supported by Figure 6.5, which compares the two approaches for each model and displays their differences. At first glance, the differences seems small, especially for the best-performing models.

Again, significance testing was conducted to determine if there is a significant difference between the approaches. This time, the null hypothesis of equal MAPE for models trained using the hybrid and direct approaches, was tested against the alternative that the hybrid forecasting method is significantly better. The results in Appendix A.5 show that only for Neural Prophet and LightGBM, the hybrid forecasting is significantly better ($p < 0.05$). However, for most models, particularly the best-performing ones, the differences are minimal with Direct Forecasting sometimes performing better.

As explained in Section 5.4, in the current evaluation, the features are known since a historical replay is performed. When forecasting into the future, the unknown features in the optimal feature subset would need to be estimated, adding an extra margin of error. Given that the differences are minimal, the hybrid forecasting methods will likely decrease in performance due to this additional error. This suggests that these models do not benefit from the additional features provided in the Hybrid setup. Therefore, the Direct Forecasting method is the preferred approach.

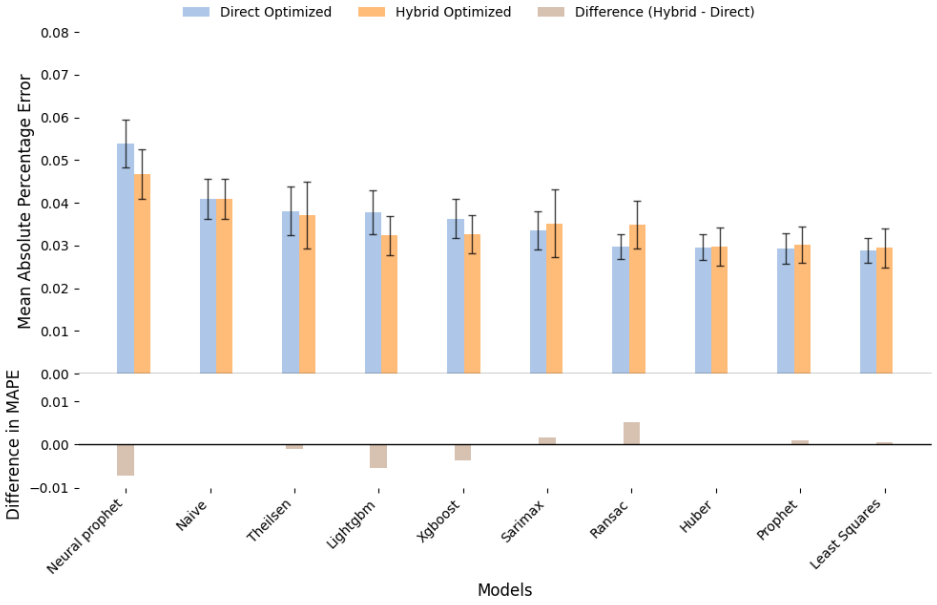

Figure 6.5: Comparison of the Direct and Hybrid Forecasting Approaches after Optimisation

6.6. Results Per Offset

In the analysis performed so far, an eight-week forecasting window was used. To gain insight into the performance of the models across different forecast offsets, the performance of the models was evaluated for each forecast offset. This is done using the methodology described in Section 5.5.5. Since Direct Forecasting was determined to be the preferred method in the previous section, this analysis focuses only on this approach. The results are shown in Figure 6.6.

Most models show an increasing trend in MAPE as the forecasting window extends, which is expected since forecasts further away are intuitively harder to make. This increase is most apparent in the first five models from left to right. However, the last four models, which have the highest average scores, display a different pattern. They show similar performance across all offsets and, in some cases, even a decrease in MAPE.

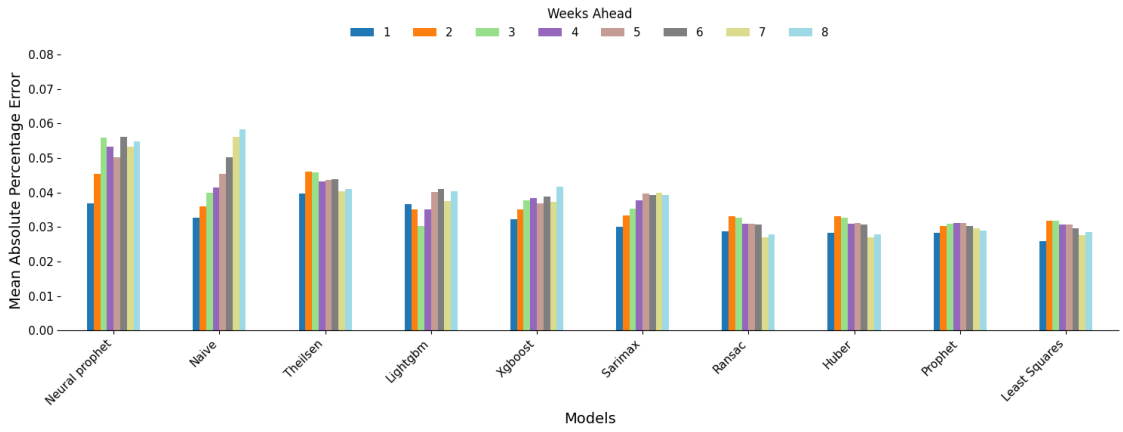

Figure 6.6: Average Score per Forecast Offset For Direct Forecasting

6.7. Analysis of High Forecast Errors

To inspect the quality of forecasts in a detailed way, it is useful to look at cases where the errors are the highest and understand why this occurs. In Appendix D a full overview is provided of the predictions per iteration in the walk forward validation of the Least Squares model in Direct Forecasting. Figure 6.7 shows the forecasts for four iterations specifically chosen because they had a peak in MAPE for one week.

Week 52 of 2023 is consistently forecasted with a high error across these iterations. The actual value even falls outside the 95% prediction interval, which is the only instance of this happening. Upon inspecting the data, the only unusual finding was a very low volume, which was the minimum volume recorded in the entire dataset. This low volume can be explained by the holidays during that week, as both Christmas and New Year's where then.

This should ideally be captured by the feature *num delivery days*, but since this is a very specific time period occurring only once a year, it was probably not adequately accounted for.

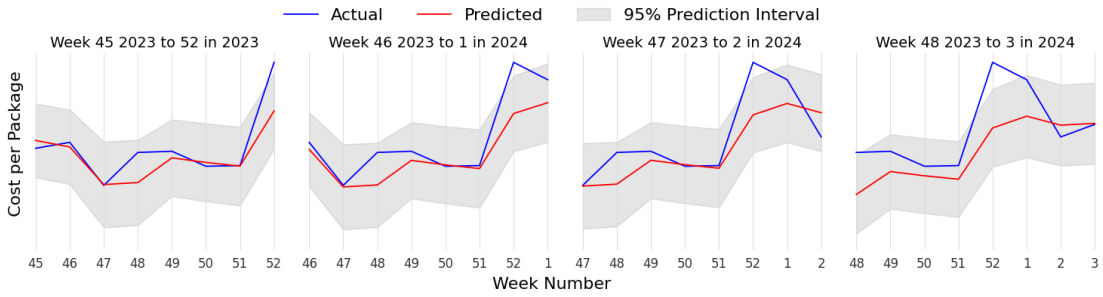

Figure 6.7: Four Iterations of Walk Forward Validation with a High Error

6.8. Feature Importance

To build on the analysis in Chapter 4, feature importance can be used to evaluate the impact of various features on the predicted cost per package. Feature importance refers to assigning scores to input features based on their contribution to predicting the target variable [45]. This measure indicates how significant each feature's influence is on the cost per package, with higher importance linked to a stronger relationship.

As the features of interest are only used in the hybrid forecasting method, the best-performing models in that approach will be evaluated for feature importance. In hybrid forecasting, the top five performing models are Least Squares, Huber Regressor, Prophet, LightGBM, and XGBoost. All these models, except for Prophet, have built-in methods to assess feature importance. Therefore, only the importance of the models with such a method is evaluated, as comparing four models should be sufficiently informative.

For the linear models Least Squares and Huber Regressor, feature importance is expressed as the absolute value of their coefficients. Since linear regression learns a linear function, the coefficient indicates how much a specific feature contributes to the final result [45]. As each feature is preprocessed to be scaled between zero and one, the comparison between the coefficients is fair.

Additionally, the two boosting models, XGBoost and LightGBM, have their own mechanisms for calculating feature importance. XGBoost's feature importance is based on the frequency and quality of splits that each feature contributes across all trees in the model [12]. Similarly, LightGBM assesses feature importance by evaluating the number of times a feature is used in splits and the associated information gain, effectively capturing the feature's contribution to improving model accuracy [34].

When evaluating feature importance in this thesis, an important consideration is that the models are retrained in each iteration of the walk-forward validation. This implies that feature importance will vary over the iterations. To evaluate the feature importance measured over the iterations, a score per iteration was obtained, and the results are shown as a line plot. The plot highlights the changes in feature importance over the iterations and provides a relative comparison between the models. The relative comparison is more relevant in this case than the actual values of the importance.

Figure 6.8 presents the feature importance scores for all the models. Note that the optimized models are used,

and thus feature selection was already incorporated, resulting in different features per model in the plot. The importances of *forecast realisation ratio* and *volume forecast* are highlighted as these two features were identified as having significant relationships in Chapter 4. For the linear models Least Squares and Huber Regressor, these two features are the top features. For LightGBM, the volume forecast has an average score compared to other features, while the *forecast realisation ratio* has the highest importance. For XGBoost, both features rank a bit above average.

For all models, the feature ranking based on importance remains approximately constant over time, indicating that the relative importance compared to other features remains consistent. Thus, it can be concluded that feature importance confirms the relevance of *forecast realisation ratio* and *volume forecast* for the cost per package.

Regarding the other features, no consistent conclusion can be drawn about their impact. Many features have scores close to zero or vary in their importance between the various models.

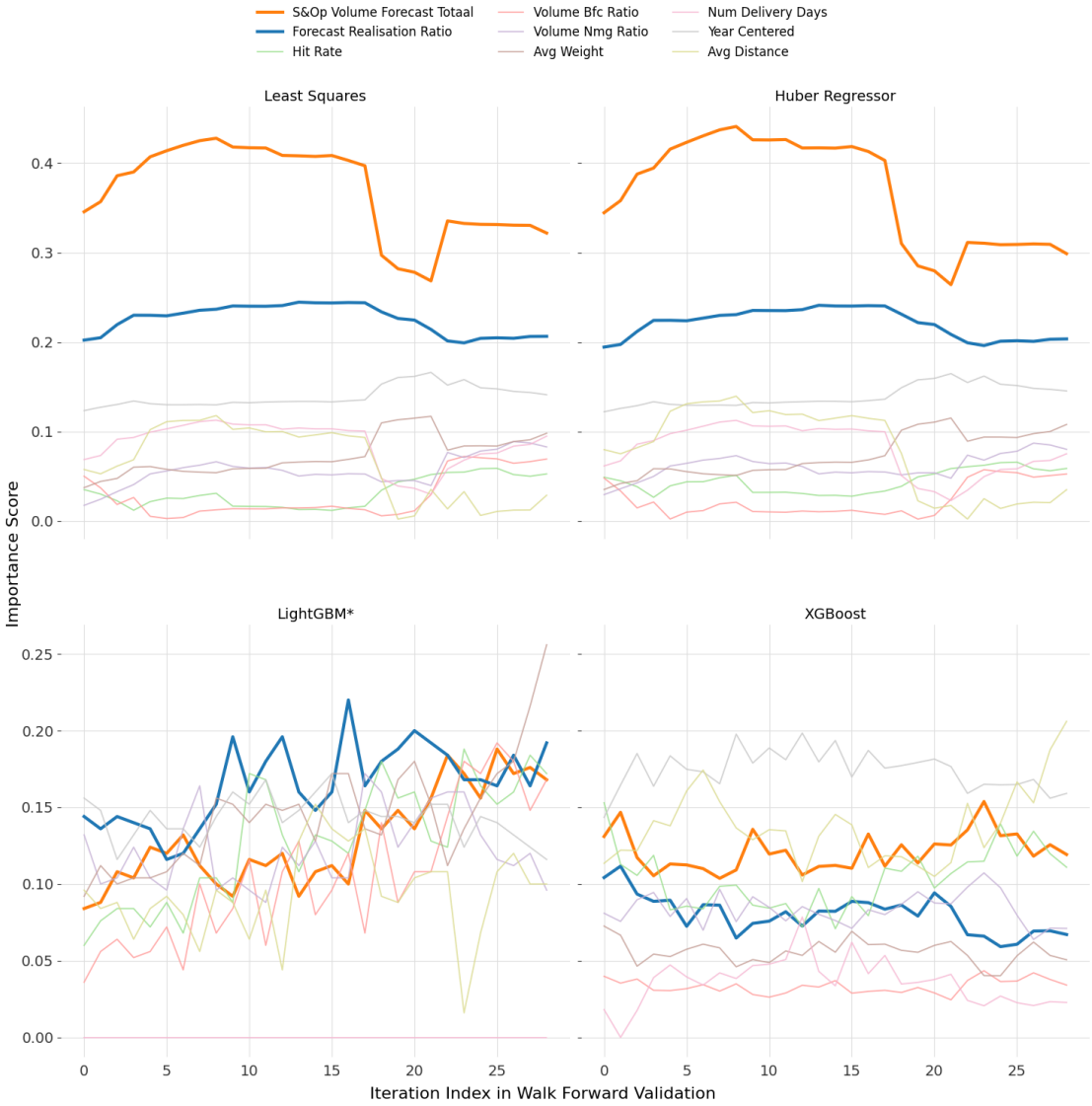

Figure 6.8: Feature importance for Each Iteration of the Walk Forward Validation
* The importance scores of LightGBM were adjusted by a factor of 1/250 to align them with the range of XGBoost. Since the focus is on the relative comparison of the models rather than the actual values, this approach is deemed appropriate.

7

Discussion and Integrated Analysis

This chapter interprets the forecasting results, addresses limitations, and integrates insights from data analysis and feature importance to identify the key factors influencing the cost per package at The Company.

7.1. Interpretation of Forecasting Results

First an interpretation of the performance of the eight selected forecasting models is provided, comparing their accuracy and highlighting key observations.

7.1.1. Summary of Key Findings

The Least Squares model had the lowest scores after optimization with an average MAPE of 0.0289 and 0.0294 for Direct and Hybrid Forecasting respectively, This score is comparable to error margins for other forecasts within The Company. Moreover, with the volatility of the cost per package and the fact that it is prone to be influenced by one time events that are hard to capture in the model, this is considered a good result.

The Least Squares model was not the only one with good results; multiple models performed similarly in both forecasting approaches. Interestingly, many of these models are linear. Their strong performance is likely due to the simplicity of linear models, which makes them less prone to overfitting on small datasets. Besides the linear models, Prophet also performed well in both approaches, proving its usefulness in time series forecasting on small datasets.

In the comparison for Direct and Hybrid Forecast in Section 6.5, direct forecasting was chosen as the preferred method, leaving four similarly performing models are now to be chosen from: RANSAC Regressor, Huber Regressor, Prophet, and Least Squares. This similarity in performance requires additional considerations beyond model scores to choose a best model. This could be an advantage, as factors such as ease of implementation can also be considered when selecting the most suitable model for The Company. Given that The Company uses Excel extensively, there is an incentive to choose Linear Regression. As this model also has the lowest average error, it seems an appropriate choice that balances model accuracy and practicality of implementation.

7.1.2. Significance Testing

Significance testing was used in every comparison for performance improvement. For example, in Section 6.3 it was shown that most models outperform the Naive methods, and thus it can be concluded that the forecasting efforts provide useful additional information over naive methods.

However, when evaluating the effect of optimisation in Section A.4 it was observed that it was not always as effective. Especially for hybrid forecasting none of the models displayed significant improvement compared to their unoptimized versions. A possible explanation could be attributed to the feature selection process. This process first selects 25 subsets based on the heuristic for Correlated Feature Selection, Then, cross-validation is performed to determine which of these feature subsets works best. For hybrid forecasting, many features are available as both the known and unknown features are included, resulting many possible feature subsets.

The preselection with CFS only selects 25 subsets, potentially excluding a very good subset. This makes the optimization less effective. For direct forecasting, this problem does not occur as fewer features are available and thus fewer feature subsets can be chosen from. The majority of the possible subsets are evaluated using cross-validation.

The observation in Section 6.5 that hybrid forecasting generally does not outperform direct forecasting, might be related to this issue. As the full set of features available for Direct Forecasting is a subset of the available features for Hybrid forecasting, It should be possible for the Hybrid Approach to at least perform as good since the same settings applied for Direct Forecasting could be selected. However again the preprocessing step in the feature selection causes that this is not possible, explaining why in some case Hybrid Forecasting underperforms as compared to Direct Forecasting.

Another feature selection approach could have resulted in better performance of the Hybrid method. However, even if this leads to higher accuracy on the test set it must be noted that hybrid forecasting uses features that are unknown in the future and require an estimation. As this analysis was done on historical data, the estimations did not have to be made yet. This implies that the extra error margin that will be introduced by the estimations is not included yet. So, even if another feature selection approach would lead to better accuracy on this historical data, it is unlikely that Hybrid Forecasting will have higher accuracy when it includes the estimated features. Therefore, the Direct Forecasting is reliably chosen as the preferred approach and the effect of the limitations of feature selection is not expected to have changed the outcome of this research.

7.1.3. Error Analysis

Analysis was also done into the errors in the forecasting models to understand their performance over varying forecasting offsets and their response to anomalies.

Generalisation and Forecasting Offset

In Section 6.6 it was observed that for the most models the average MAPE increases with the forecasting offset. However, the four best performing models this pattern was not observed. Their average MAPE remained consistent or even decreased for the forecast offset that are further away.

One potential reason for this could be related to the inclusion of lag features. When looking at the optimal feature sets in Table B.1 in Appendix B all these models have lagged values of the cost per package in their optimal feature set. During the autocorrelation analysis in Section 4.1.2, it was observed that lag values up to 4 weeks are informative to the cost per package. After four weeks, these lag values are not informative, introducing extra randomness. This additional randomness during training might help the models generalise better to unseen data, resulting in relatively stable or even improved performance in the last few offsets for the last four models. However, this is an educated guess and is not confirmed.

What is sure, it that for the best performing models the forecast window of eight works well and there is no reason to decrease it. Increasing the forecast window is also unnecessary, as The Company specifically requested a maximum forecasting window of eight weeks.

Anomalies

Finally, one case was highlighted with high error is Section 6.7, which was attributed to low volume due to a special week with Christmas and New Year. This finding suggests that the models struggle to capture such rare events, indicating a potential area for improvement in handling anomalies.

7.1.4. Data Limitations

The current analysis, based on 118 weekly data points, has offered valuable insights. However, due to the weekly data points, averages have to be taken over multiple packages and days. This averaging may limit the informativeness of individual features and potentially discard more subtle relationships. By adding more granularity, a more nuanced understanding of the factors influencing the cost per package could be obtained.

Additionally, a larger and more granular dataset could offer multiple advantages in forecasting. First, it would likely improve model accuracy by providing a larger training and validation set. This would allow models to better capture underlying patterns and relationships within the data. Second, a larger dataset could enable the use of more complex models that are less prone to overfitting. Such models might be better equipped

to capture data volatility and potentially even account for anomalies. Furthermore, the models that are currently best-performing could likely benefit from additional data to refine their predictions.

Finally, a larger sample size would enhance the statistical power of comparisons between models and methods. As statistical tests are sensitive to sample size, a larger dataset could reveal significant differences that were not detectable in the current analysis. This would provide more evidence for model selection and optimization decisions, ultimately leading to more accurate and reliable cost forecasts.

7.2. Model Implementation within The Company

To bring the findings of this thesis into practice, the forecasting model should be implemented and put into production for effective use by The Company.

In the previous experiments it was concluded that the Least Squares model is the preferred choice over the other models. This is beneficial since a linear regression model can be conveniently implemented in Excel and given that Excel is often used within The Company for dashboards and data storage, implementing the model in Excel aligns with existing workflows. The model will be integrated into an existing dashboard where, for example, the volume forecast is also presented. Detailed instructions were provided on how to add the required data, and one person was made responsible for keeping the data up to date and monitoring the quality of the forecast.

The practical implementation is similar to the example iterations of the walk-forward validation shown in Figure 6.7, including the forecast and the 95% prediction interval. To provide insights into past predictions and compare them to actual values, users can select a specific period to display. If the selected period is in the past, the actual values will also be provided.

The model will be retrained for every new forecast created. To ensure this process is user-friendly, automatic training is triggered once all the required data is added. Additionally, documentation was created to provide users with a reference in case of questions. With this implementation in Excel, the goal is to make the forecast easy to access and use, increasing the chances of adoption by users familiar with Excel and the existing dashboard.

7.3. Integrated Analysis of Feature Importance and Key Drivers

This section integrates the findings from the key driver analysis in Chapter 4 with the feature importance assessment in Section 6.8 to combine their insights on the key factors influencing the cost per package.

The data analysis consisting of Correlation Analysis, Granger Causality, and LPCMCI, identified three potential key drivers: average DM3, volume, and forecast realization ratio. The strongest evidence was found for volume, as all three tests highlighted its importance. Similarly, the *forecast realisation*, was also consistently identified as important through Granger Causality and LPCMCI. Although average DM3 had a significant result in Granger Causality and a strong correlation, its significance was considered slightly weaker due to the exploratory nature of the correlation analysis.

To complement the data analysis, feature importance was assessed in the best performing models of hybrid forecasting, being Least Squares, Huber Regressor, XGBoost, and LightGBM. The feature selection was evaluated on the models after optimization, thus only the optimal feature subsets have results in this case. Average DM3 was not part of the optimal feature subsets for any of these models, suggesting it might not be a key driver for forecasting. However, considering the critical evaluation of feature selection methods in Section 7.1.2, a definitive conclusion about average DM3 cannot be drawn based on this analysis.

The feature importance analysis confirmed the importance of *forecast realisation ratio* and *volume forecast* as key drivers. In the linear models (Least Squares and Huber Regressor), these features achieved the highest importance scores, validating their significance in predicting the cost per package. In the boosting models (LightGBM and XGBoost), the difference was less obvious but both features consistently ranked high in importance.

Integrating the findings from the data analysis and forecasting models reveals an alignment between the key drivers identified. The *forecast realisation ratio* and *volume forecast* consistently appeared as key factors across all analytical methods.

The relationship of *forecast realisation ratio* with the cost per package is positive, meaning it occurs when the forecast volume exceeds the actual volume, leading to higher costs per package. This is relevant as the analysed data mainly consists of overestimated forecasts. Conversely, the impact of underestimated forecasts is less apparent in the data. However, underestimating volume can result into expensive last-minute solutions which also impacts the cost negatively, again highlighting the importance of accurate forecasts.

The relationship with *volume* is negative suggests the presence of economies of scale. This implies that as volume increases, the cost per package decreases due to operational efficiencies and reduced fixed costs per packages.

It is important to acknowledge that this thesis primarily focused on factors with a consistent impact on cost and those measurable by the selected methods. Other factors like specific events might also contribute to cost fluctuations, but these are not accounted for. Additionally, alternative analytical approaches could have potentially highlighted different features. However, the combination of three data analysis methods and feature importance assessment aims to mitigate model-specific biases and offer a comprehensive analysis.

8

Conclusion

This thesis investigated the drivers of cost per package at The Company and developed a forecasting model to predict these costs accurately. By creating a consistent definition of the cost per package and combining data analysis with machine learning techniques, this research provided actionable insights and a practical tool for cost forecasting while also contributing to the broader field of time series analysis and forecasting in logistics through its comparative study of various models and techniques.

8.1. Summary of Key Findings

The study identified volume and forecast realisation ratio as the most significant drivers of cost per package. These findings were consistent across multiple analytical methods, including correlation analysis, Granger causality tests, and LPCMCI, as well as feature importance assessments in the best-performing forecasting models. The alignment of these results hightlights the importance of the impact of these factors. Additionally, the Least Squares model was found to be the most suitable forecasting model for The Company, balancing accuracy with ease of implementation into an existing dashboard in Excel.

8.2. Recommendations for The Company

Based on the findings of this thesis, the following recommendations are made:

- **Improve Volume Forecasting Accuracy** Given the influence of the quatlity of the volume forcast on costs, improving the accuracy of volume forecasts should be a primary focus.
- **Optimize Data Collection and Documentation** Gathering daily data instead of weekly averages could provide insights into more subtle relationships and enhance model performance. Additionally, proper documentation of the data will help future analyses and reduce the time required for data preparation.
- **Establish Model Monitoring** Continuous monitoring of the implemented model is required to evaluate its performance in production and identify any potential issues. To ensure the model remains relevant and maximize the value of the insights it provides, data must be updated weekly.

By implementing these recommendations, The Company will be able to make more informed decisions regarding operational planning, ultimately improving cost efficiency.

8.3. Limitations and Future Research

The main limitation of this thesis is in the size and granularity of the dataset. A larger, more granular dataset could potentially lead to additional insights, improve forecasting accuracy, and increase the statistical power of comparisons. Future research should look into creating and analyzing such a dataset to further refine the understanding of cost drivers and forecasting models.

Additionally, the current analysis focusses on factors that are of general impact. Specific or one-time events were not explicitly accounted for, but were found to influence cost. Future studies could explore these factors to better understand their impact on costs and the quality of the forecast.

References

[1] R. F. Aghdam et al. “On the relationship between energy and development: a comprehensive note on causation and correlation”. In: *Energy Strategy Reviews* 46 (2023), p. 101034. DOI: 10.1016/j.esr.2022.101034.

[2] D. G. Altman and J. M. Bland. “Standard deviations and standard errors”. In: *BMJ* 331 (7521 2005), p. 903. DOI: 10.1136/bmj.331.7521.903.

[3] A. Asesh. “Normalisation and bias in time series data”. In: *Digital Interaction and Machine Intelligence* (2022), pp. 88–97. DOI: 10.1007/978-3-031-11432-8_8.

[4] C. K. Assaad, É. Devijver, and É. Gaussier. “Survey and evaluation of causal discovery methods for time series”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 767–819. DOI: 10.1613/jair.1.13428.

[5] C. K. Assaad, É. Devijver, and É. Gaussier. “Survey and evaluation of causal discovery methods for time series”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 767–819. DOI: 10.1613/jair.1.13428.

[6] R. A. Azdy and F. Darnis. “Use of haversine formula in finding distance between temporary shelter and waste end processing sites”. In: *Journal of Physics: Conference Series* 1500 (1 2020), p. 012104. DOI: 10.1088/1742-6596/1500/1/012104.

[7] T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann. “Tuning: methodology”. In: *Hyperparameter Tuning for Machine and Deep Learning With R* (2023), pp. 7–26. DOI: 10.1007/978-981-19-5170-1_2.

[8] C. Bergmeir and J. M. Benítez. “On the use of cross-validation for time series predictor evaluation”. In: *Information Sciences* 191 (2012), pp. 192–213. DOI: 10.1016/j.ins.2011.12.028.

[9] J. Brownlee. *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*. Machine Learning Mastery, 2017. URL: https://books.google.nl/books?id=-AiqDwAAQBAJ.

[10] M. Castro et al. “Time series causal relationships discovery through feature importance and ensemble models”. In: *Scientific Reports* 13 (1 2023). DOI: 10.1038/s41598-023-37929-w.

[11] J. Chen, L. Kuhn, and S. Raschka. “Techniques for developing reliable machine learning classifiers applied to understanding and predicting protein:protein interaction hot spots”. In: (2022). DOI: 10.1101/2022.12.26.521948.

[12] T. Chen and C. Guestrin. “Xgboost”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016). DOI: 10.1145/2939672.2939785.

[13] Song Zan Chiou-Wei, Ching Fu Chen, and Zhen Zhu. “Economic growth and energy consumption revisited - Evidence from linear and nonlinear Granger causality”. In: *Energy Economics* 30 (6 Nov. 2008), pp. 3063–3076. ISSN: 01409883. DOI: 10.1016/j.eneco.2008.02.002.

[14] P. S. Cowpertwait and A. Metcalfe. “Introductory time series with r”. In: (2009). DOI: 10.1007/978-0-387-88698-5.

[15] D. Dickey and Wayne Fuller. “Distribution of the Estimators for Autoregressive Time Series With a Unit Root”. In: *JASA. Journal of the American Statistical Association* 74 (June 1979). DOI: 10.2307/2286348.

[16] M. A. Fischler and R. C. Bolles. “Random sample consensus”. In: *Communications of the ACM* 24 (6 1981), pp. 381–395. DOI: 10.1145/358669.358692.

[17] John R Freeman. “Granger causality and the times series analysis of political relationships”. In: *American Journal of Political Science* (1983), pp. 327–358.

[18] A. Gerhardus and J. Runge. “Lpcmci: causal discovery in time series with latent confounders”. In: (2021). DOI: 10.5194/egusphere-egu21-8259.

[19] Andreas Gerhardus and Jakob Runge. “High-recall causal discovery for autocorrelated time series with latent confounders”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 12615–12625. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/94e70705efae423efda1088614128d0b-Paper.pdf.

[20] A. Ghasemi and S. Zahediasl. “Normality tests for statistical analysis: a guide for non-statisticians”. In: *International Journal of Endocrinology and Metabolism* 10 (2 2012), pp. 486–489. DOI: 10.5812/ijem.3505.

[21] Nithya J Gogtay and Urmila M Thatte. “Principles of correlation analysis”. In: *Journal of the Association of Physicians of India* 65.3 (2017), pp. 78–81.

[22] C W J Granger. *Investigating Causal Relations by Econometric Models and Cross-spectral Methods*. 1969.

[23] Isabelle Guyon and Andre@tuebingen Mpg De. *An Introduction to Variable and Feature Selection André Elisseeff*. 2003.

[24] U. R. Hodeghatta and U. Nayak. “Time series: forecasting”. In: *Practical Business Analytics Using R and Python* (2023), pp. 443–484. DOI: 10.1007/978-1-4842-8754-5_12.

[25] C. Huang and A. Petukhina. “Applied time series analysis and forecasting with python”. In: *Statistics and Computing* (2022). DOI: 10.1007/978-3-031-13584-2.

[26] P. J. Huber. “Robust statistics”. In: *Wiley Series in Probability and Statistics* (1981). DOI: 10.1002/0471725250.

[27] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.

[28] Gareth James et al. “Resampling Methods”. In: *An Introduction to Statistical Learning: with Applications in R*. New York, NY: Springer US, 2021, pp. 197–223. ISBN: 978-1-0716-1418-1. DOI: 10.1007/978-1-0716-1418-1_5. URL: https://doi.org/10.1007/978-1-0716-1418-1_5.

[29] S. Johansen. “Statistical analysis of cointegration vectors”. In: *Journal of Economic Dynamics and Control* 12 (2-3 1988), pp. 231–254. DOI: 10.1016/0165-1889(88)90041-3.

[30] A. Jović, K. Brkić, and N. Bogunović. “A review of feature selection methods with applications”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015, pp. 1200–1205. DOI: 10.1109/MIPRO.2015.7160458.

[31] Bahavathy Kathirgamanathan and Pádraig Cunningham. “A Feature Selection Method for Multi-dimension Time-Series Data”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 220–231. ISBN: 9783030657420. DOI: 10.1007/978-3-030-65742-0_15. URL: http://dx.doi.org/10.1007/978-3-030-65742-0_15.

[32] Bahavathy Kathirgamanathan and Padraig Cunningham. *Correlation Based Feature Subset Selection for Multivariate Time-Series Data*. 2021. arXiv: 2112.03705 [cs.LG].

[33] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[34] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[35] E. Keogh and A. Mueen. “Curse of dimensionality”. In: *Encyclopedia of Machine Learning and Data Mining* (2017), pp. 314–315. DOI: 10.1007/978-1-4899-7687-1_192.

[36] Irena Koprinska, Mashud Rana, and Vassilios G. Agelidis. “Correlation and instance based feature selection for electricity load forecasting”. In: *Knowledge-Based Systems* 82 (2015), pp. 29–40. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2015.02.017. URL: https://www.sciencedirect.com/science/article/pii/S0950705115000714.

[37] P. Ładyżyński, K. Żbikowski, and P. Grzegorzewski. “Stock trading with random forests, trend detection tests and force index volume indicators”. In: *Artificial Intelligence and Soft Computing* (2013), pp. 441–452. DOI: 10.1007/978-3-642-38610-7_41.

[38] E. L. Lehmann and J. P. Romano. “Invariance”. In: *Springer Texts in Statistics* (2022), pp. 241–314. DOI: 10.1007/978-3-030-70578-7_6.

[39] E. L. Lehmann and J. P. Romano. “Unbiasedness: applications to normal distributions; confidence intervals”. In: *Springer Texts in Statistics* (2022), pp. 171–240. DOI: 10.1007/978-3-030-70578-7_5.

[40] Howard Levene. “Robust tests for equality of variances”. In: *Contributions to probability and statistics*. Vol. 2. Stanford Studies in Mathematics and Statistics. Stanford Univ. Press, Stanford, CA, 1960, pp. 278–292.

[41] Helmut Lütkepohl. “Specification of VECMs”. In: *New Introduction to Multiple Time Series Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 325–352. ISBN: 978-3-540-27752-1. DOI: 10.1007/978-3-540-27752-1_8. URL: https://doi.org/10.1007/978-3-540-27752-1_8.

[42] Tanisha Mahajan, Gaurav Singh, and Glenn Bruns. *An Experimental Assessment of Treatments for Cyclical Data*. 2021.

[43] J. R. McCrorie and M. J. Chambers. “Granger causality and the sampling of economic processes”. In: *Journal of Econometrics* 132 (2 2006), pp. 311–336. DOI: 10.1016/j.jeconom.2005.02.002.

[44] P. Mishra et al. “Application of student’s t-test analysis of variance, and covariance”. In: *Annals of Cardiac Anaesthesia* 22 (4 2019), p. 407. DOI: 10.4103/aca.aca_94_19.

[45] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[46] R. Moraffah et al. “Causal inference for time series analysis: problems, methods and evaluation”. In: *Knowledge and Information Systems* 63 (12 2021), pp. 3041–3085. DOI: 10.1007/s10115-021-01621-0.

[47] M. Nauta, D. Bucur, and C. Seifert. “Causal discovery with attention-based convolutional neural networks”. In: *Machine Learning and Knowledge Extraction* 1 (1 2019), pp. 312–340. DOI: 10.3390/make1010019.

[48] D. J. Olive. “Linear regression”. In: (2017). DOI: 10.1007/978-3-319-55252-1.

[49] J. L. Rodgers and W. A. Nicewander. “Thirteen ways to look at the correlation coefficient”. In: *The American Statistician* 42 (1 1988), p. 59. DOI: 10.2307/2685263.

[50] J. Runge. “Causal network reconstruction from time series: from theoretical assumptions to practical estimation”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28 (7 2018). DOI: 10.1063/1.5025050.

[51] Jakob Runge, Dino Sejdinovic, and Seth Flaxman. “Detecting causal associations in large nonlinear time series datasets”. In: *Science Advances* 5 (Feb. 2017). DOI: 10.1126/sciadv.aau4996.

[52] Zezhi Shao et al. *Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis*. 2023. arXiv: 2310.06119 [cs.LG].

[53] Y. Shin and P. Schmidt. “The kpss stationarity test as a unit root test”. In: *Economics Letters* 38 (4 1992), pp. 387–392. DOI: 10.1016/0165-1765(92)90023-r.

[54] A. Shojaie and E. B. Fox. “Granger causality: a review and recent advances”. In: *Annual Review of Statistics and Its Application* 9 (1 2022), pp. 289–319. DOI: 10.1146/annurev-statistics-040120-010930.

[55] R. H. Shumway and D. S. Stoffer. “Arima models”. In: *Springer Texts in Statistics* (2017), pp. 75–163. DOI: 10.1007/978-3-319-52452-8_3.

[56] Christopher A Sims. *Macroeconomics and Reality*. 1980. URL: https://about.jstor.org/terms.

[57] P. Spirtes and K. Zhang. “Causal discovery and inference: concepts and recent methodological advances”. In: *Applied Informatics* 3 (1 2016). DOI: 10.1186/s40535-016-0018-x.

[58] M. Steurer, R. J. Hill, and N. Pfeifer. “Metrics for evaluating the performance of machine learning based automated valuation models”. In: *Journal of Property Research* 38 (2 2021), pp. 99–129. DOI: 10.1080/09599916.2020.1858937.

[59] S. J. Taylor and B. Letham. “Forecasting at scale”. In: (2017). DOI: 10.7287/peerj.preprints.3190v2.

[60] Luis Tedeschi and Michael Galyean. “A practical method to account for outliers in simple linear regression using the median of slopes”. In: *Scientia Agricola* 81 (Jan. 2024). DOI: 10.1590/1678-992x-2022-0209.

[61] Daniel L. Thornton and Dallas S. Batten. “Lag-Length Selection and Tests of Granger Causality Between Money and Income”. In: *Journal of Money, Credit and Banking* 17.2 (1985), pp. 164–178. ISSN: 00222879, 15384616. URL: http://www.jstor.org/stable/1992331 (visited on 06/15/2024).

[62] Oskar Triebe et al. *NeuralProphet: Explainable Forecasting at Scale*. 2021. arXiv: 2111.15397 [cs.LG].

[63] K. Żbikowski. “Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy”. In: *Expert Systems With Applications* 42 (4 2015), pp. 1797–1805. DOI: 10.1016/j.eswa.2014.10.001.

[64] Lingyu Zhang et al. “Time series forecast of sales volume based on XGBoost”. In: *Journal of Physics: Conference Series*. Vol. 1873. 1. 2021, p. 012067.

[65] Youyang Zhang, Changfeng Zhu, and Qingrong Wang. “LightGBM-based model for metro passenger volume forecasting”. In: *IET Intelligent Transport Systems* 14.13 (2020), pp. 1815–1823.

A

Significance Testing

A.1. Unoptimized Models and the Naive Model

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.095	0.238	T	-	1.000
Theilsen	0.481	0.797	T	-	0.103
Ransac	0.206	0.549	T	-	0.778
Lightgbm	0.406	0.520	T	-	0.084
Xgboost	0.132	0.295	T	-	0.422
Huber	0.984	0.473	T	-	0.067
Sarimax	0.512	0.696	T	-	0.058
Least Squares	0.871	0.245	T	-	0.018**
Prophet	0.108	0.048**	W	-0.274	0.004**

Table A.1: Direct Forecasting

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.174	0.316	T	-	0.966
Ransac	0.027**	0.243	W	0.896	0.391
Sarimax	0.017**	0.047**	W	0.542	0.179
Huber	0.680	0.146	T	-	0.013**
Xgboost	0.569	0.673	T	-	0.000**
Prophet	0.057	0.535	T	-	0.003**
Theilsen	0.551	0.507	T	-	0.009**
Least Squares	0.592	0.384	T	-	0.005**
Lightgbm	0.472	0.964	T	-	0.000**

Table A.2: Hybrid Forecasting

** test statistic is significant at the 5% level. Condition is met Significant improvement over the Naive Method.

The table shows the results of the significance tests for significant improve of the MAPE of the *unoptimized* models as compared to the naive methods. First Normality and Variance are tested. If the conditions are met, a T-test (T) is performed. Otherwise, a Wilcoxon (W) test is performed and skew is reported. The models are ordered by average MAPE.

A.2. Optimized Models and the Naive Model

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.119	0.692	T	-	0.999
Theilsen	0.035**	0.360	W	0.449	0.215
Lightgbm	0.310	0.985	T	-	0.110
Xgboost	0.887	0.808	T	-	0.018**
Sarimax	0.568	0.582	T	-	0.040**
Ransac	0.482	0.015**	W	-0.384	0.000**
Huber	0.634	0.019**	W	-0.342	0.000**
Prophet	0.701	0.135	T	-	0.001**
Least Squares	0.148	0.009**	W	-0.673	0.000**

Table A.3: Direct Forecasting

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.085	0.434	T	-	0.880
Theilsen	0.055	0.066	T	-	0.208
Sarimax	0.059	0.070	T	-	0.159
Ransac	0.138	0.447	T	-	0.049**
Xgboost	0.017**	0.744	W	-0.531	0.000**
Lightgbm	0.221	0.725	T	-	0.006**
Prophet	0.217	0.581	T	-	0.000**
Huber	0.040**	0.615	W	-0.060	0.004**
Least Squares	0.026**	0.733	W	-0.074	0.004**

Table A.4: Hybrid Forecasting

** test statistic is significant at the 5% level. Condition is met Significant improvement over the Naive Method.

The table shows the results of the significance tests for significant improve of the MAPE of the *optimized* models as compared to the naive methods. First Normality and Variance are tested. If the conditions are met, a T-test (T) is performed. Otherwise, a Wilcoxon (W) test is performed and skew is reported. The models are ordered by average MAPE.

A.3. Optimized Models and Least Squares

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.363	0.011**	W	-0.521	0.000**
Naive	0.148	0.009**	W	-0.673	0.000**
Theilsen	0.009**	0.002**	W	-0.575	0.000**
Lightgbm	0.016**	0.036**	W	-1.179	0.001**
Xgboost	0.202	0.034**	W	-0.699	0.002**
Sarimax	0.114	0.047**	W	-0.294	0.038**
Ransac	0.571	0.686	T	-	0.075
Huber	0.404	0.611	T	-	0.101
Prophet	0.384	0.184	T	-	0.403

Table A.5: Direct Forecasting

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.266	0.291	T	-	0.000**
Naive	0.026**	0.733	W	-0.074	0.004**
Theilsen	0.011**	0.043**	W	-0.541	0.010**
Sarimax	0.388	0.046**	W	-0.357	0.048**
Ransac	0.237	0.291	T	-	0.000**
Xgboost	0.167	0.990	T	-	0.068
Lightgbm	0.421	0.988	T	-	0.099
Prophet	0.124	0.860	T	-	0.368
Huber	0.057	0.875	T	-	0.063

Table A.6: Hybrid Forecasting

** test statistic is significant at the 5% level. Condition is met Least Squares performs significantly better.

The table shows the results of the significance tests for significant improvement of the MAPE of the Least Squares models as compared to the other optimized models. First Normality and Variance are tested. If the conditions are met, a T-test (T) is performed. Otherwise, a Wilcoxon (W) test is performed and skew is reported. The models are ordered by average MAPE of the unoptimized models Direct Forecasting.

A.4. Optimized and Unptimized Models

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.287	0.631	T	-	0.405
Theilsen	0.058	0.350	T	-	0.049**
Lightgbm	0.487	0.457	T	-	0.924
Xgboost	0.847	0.335	T	-	0.204
Sarimax	0.000**	0.938	W	1.998	0.992
Ransac	0.190	0.017**	W	-0.611	0.000**
Huber	0.332	0.187	T	-	0.003**
Prophet	0.391	0.466	T	-	0.028**
Least Squares	0.000**	0.303	W	-0.569	0.000**

Table A.7: Direct Forecasting

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.557	0.950	T	-	0.391
Theilsen	0.039**	0.016**	W	0.362	0.889
Sarimax	0.000**	0.822	W	-2.339	0.960
Ransac	0.316	0.711	T	-	0.093
Xgboost	0.307	0.770	T	-	0.703
Lightgbm	0.025**	0.967	W	-0.961	0.950
Prophet	0.507	0.701	T	-	0.391
Huber	0.254	0.252	T	-	0.208
Least Squares	0.519	0.466	T	-	0.565

Table A.8: Hybrid Forecasting

** test statistic is significant at the 5% level. Condition is met Optimized results are significantly better.

The table shows the results of the significance tests for significant improvement of the MAPE of the optimized as compared to unoptimized models. First Normality and Variance are tested. If the conditions are met, a T-test (T) is performed. Otherwise, a Wilcoxon (W) test is performed and skew is reported. The models are ordered by average MAPE.

A.5. Direct and Hybrid Forecasting

Model	Normality	Variance	Test	Skew	Result
Neural Prophet	0.501	0.730	T	-	0.016**
Theilsen	0.055	0.255	T	-	0.349
Lightgbm	0.024**	0.937	W	-0.277	0.031**
Xgboost	0.051	0.758	T	-	0.066
Sarimax	0.969	0.035**	W	0.230	0.657
Ransac	0.296	0.004**	W	-0.010	0.954
Huber	0.052	0.076	T	-	0.525
Prophet	0.037**	0.299	W	0.932	0.391
Least Squares	0.022**	0.026**	W	-0.120	0.576

Table A.9: Direct Forecasting

** test statistic is significant at the 5% level. Condition is met Hybrid forecasting performs significantly better.

The table shows the results of the significance tests for significant improvement of the MAPE of hybrid forecasting as compared to direct forecasting. First Normality and Variance are tested. If the conditions are met, a T-test (T) is performed. Otherwise, a Wilcoxon (W) test is performed and skew is reported. The models are ordered by average MAPE.

B

Feature Selection

Method	Selected Features	Method	Selected Features
Least Squares	cost per package lag 1, volume forecast, year centred	RANSAC	cost per package lag 1, num delivery days, volume forecast, week number x, year centred
Theil-Sen	cost per package lag 1, num delivery days, volume forecast, week number x, year centred	Huber Regressor	cost per package lag 1, volume forecast, year centred
LightGBM	cost per package lag 1, cost per package lag 2, volume forecast,	XGBoost	cost per package lag 1, cost per package lag 4, volume forecast, year centred
Neural Prophet	num delivery days	Prophet	volume forecast
SARIMAX	volume forecast		

Table B.1: Results of Feature Selection for Direct Forecasting

Method	Selected Features	Method	Selected Features
Least Squares	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio, year centred	**RANSAC**	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio, year centred
Theil-Sen	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio, year centred	**Huber**	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio, year centred
LightGBM	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio,	**XGBoost**	avg distance, avg weight, forecast realisation ratio, hit rate, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio, year centred
Neural Prophet	avg weight, forecast realisation ratio, hit rate, volume forecast, volume Main Client ratio, volume export ratio, volume NMG ratio	**Prophet**	avg weight, forecast realisation ratio, hit rate, min max volume ratio, num delivery days, volume forecast, volume Main Client ratio, volume export ratio, volume NMG ratio
SARIMAX	avg weight, forecast realisation ratio, hit rate, min max volume ratio, num delivery days, volume forecast, volume Main Client ratio, volume NMG ratio		

Table B.2: Results of Feature Selection for Hybrid Forecasting

C

Hyperparameter tuning

	Grid Specification		Results	
Model	**Parameter**	**Options**	**Hybrid Method**	**Direct Method**
RANSAC	Max Trials	50, 100, 200	50	100
	Min Samples	0.1, 0.2, 0.5	0.1	0.5
	Loss	Absolute Error, Squared Error	Squared Error	Absolute Error
	Residual Threshold	0.1, 0.3 ,0.5, 1.0	0.1	0.1
Huber Regressor	Epsilon	1.35, 1.5, 1.75, 2.0	2.0	2.0
	Max Iterations	100, 200, 500	100	100
	Alpha	0.0001, 0.001, 0.01	0.0001	0.0001
	Tol	1e-05, 1e-04, 1e-03	1e-05	1e-05
LightGBM	Max Depth	3, 4, 5	3	3
	N Estimators	50, 100, 200	50	50
	Learning Rate	0.01, 0.05	0.05	0.01
	Num Leaves	15, 31	15	15
	Subsample	0.8, 0.9	0.9	0.9
	Subsample Freq	1, 3, 5	5	3
XGBoost	Max Depth	3, 4, 5	3	3
	Learning Rate	0.01, 0.05, 0.1	0.1	0.1
	N Estimators	50, 100, 200	50	50
	Colsample Bytree	0.3, 0.5, 0.7	0.3	0.7
	Alpha	0, 5, 10	0	0
Prophet	Changepoint Prior Scale	0.001, 0.01, 0.1, 0.5	0.01	0.01
	Seasonality Prior Scale	0.01, 0.1, 1.0, 10.0	0.01	0.01
Neural Prophet	Learning rates	0.001, 0.01, 0.1	0.01	0.1
	Batch Size	4, 8, 16	8	8
	Epochs	100, 200, 300	200	200

Table C.1: Parameter Grid and for Hyperparameter Optimisation

D

Predictions

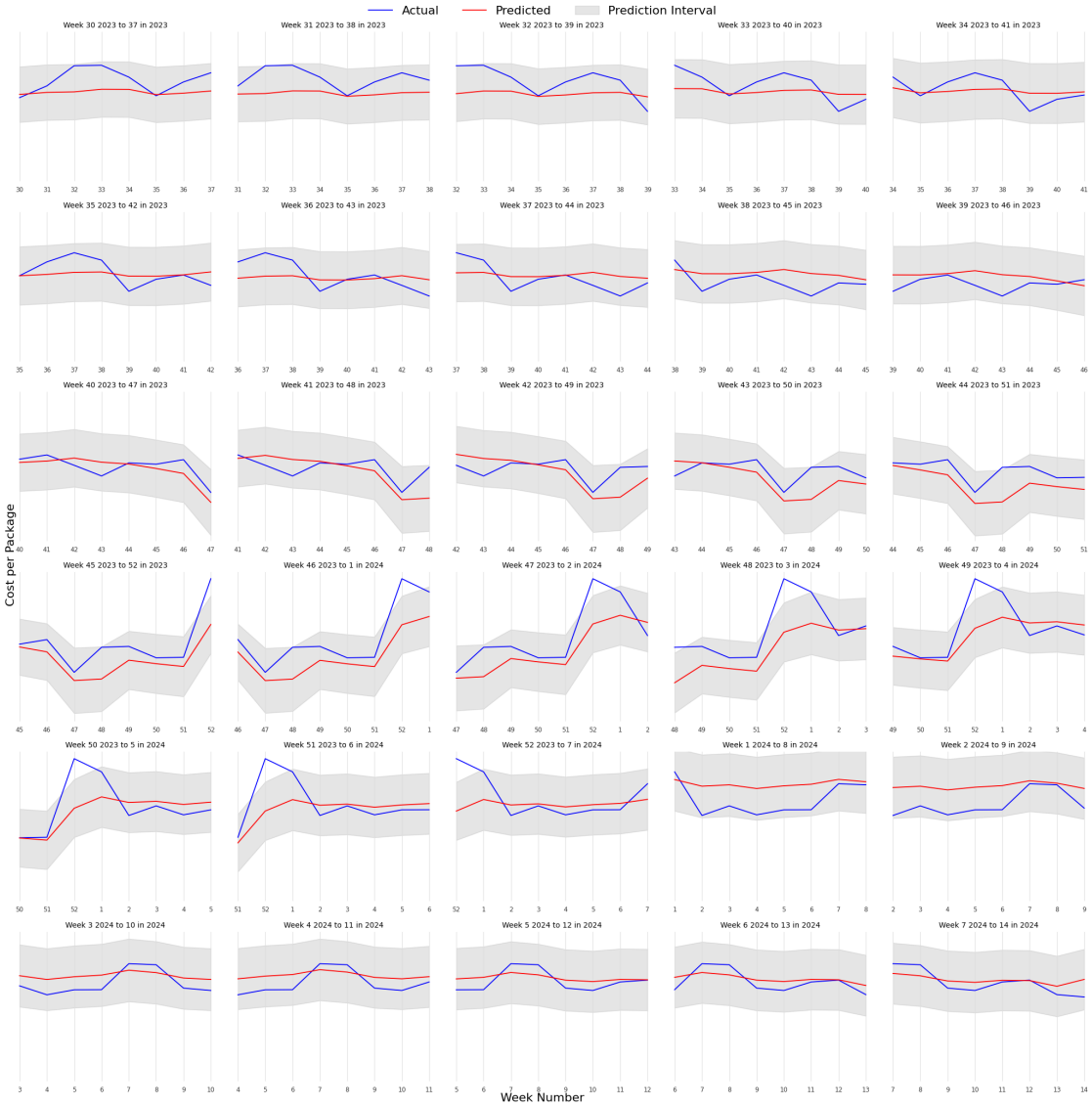

Figure D.1: Comparison of predicted cost with the actual cost for each iteration in walk forward validation. The y-axis is not displayed for confidality reasons.