Understanding and Predicting Labor Productivity of Different Goodsflows in an Omnichannel Warehouse

de Bruijn, J.E.M. (Emma)

Award date:
2023

Department of Industrial Engineering & Innovation Sciences
Operations, Planning, Accounting & Control

Understanding and Predicting Labor Productivity of Different Goodsflows in an Omnichannel Warehouse

J.E.M. de Bruijn
0954321

Supervisors Eindhoven University of Technology:
N.R. Mutlu
C. Drent
W.L.V. Jaarsveld

Supervisor MediaMarkt:
Robert Kemp

Eindhoven, September 6th, 2023

Eindhoven University of Technology
School of Industrial Engineering
Operations Management and Logistics
2022-2023

Keywords: Omnichannel Warehousing, Labor Productivity, Gradient-Boosting Decision Trees

Abstract

This research aims to find factors influencing labor productivity and leverage these to more accurately predict labor productivity, thereby improving workforce control in an omnichannel warehouse setting. The research uses advanced machine learning models XGBoost and LightGBM to determine feature importance and predict labor productivity based on these features. Model agnostic techniques helped determine bounds for which productivity displayed particular behavior, i.e., increasing, decreasing, or constant. The increasing and decreasing patterns can be leveraged to boost labor productivity and thereby control the workforce. It was observed that an increase in the average ordersize and average quantity per orderline could improve productivity. Moreover, the thresholds at which economies of scale manifest and, conversely, where diminishing returns impact productivity were discerned. Although most features' importance scores aligned with expectations, the advanced models exhibited some inherent randomness, occasionally producing less intuitive results. A significant limitation of this study is the absence of consideration for the daily effort exerted by employees. Labor productivity is assessed on an aggregated daily level and does not provide insights into the warehouse employee's individual productivity. Based on the feature evaluation, recommendations are formulated for the organization. The current research contributed significantly to academic research by introducing novel features to the warehouse employee productivity framework. Yet, substantial room exists for future research on essential features affecting labor productivity in different warehouse contexts with different types of research.

Executive Summary

In a highly competitive business environment such as today's retailing landscape, warehouses are constantly under pressure to increase productivity and accuracy while reducing costs and improving customer service (Karim et al., 2020). However, challenges in the omnichannel context add to the uncertainty in predicting productivity due to differences in demand profiles, the wide range of SKUs and product flows, and their characteristics. This is especially true in the consumer electronics sector (Kembro et al., 2018; Kembro and Norrman, 2019). Analyzing and predicting productivity performance can assist warehouse managers in various ways, including identifying areas of improvement, analyzing the performance levels of individuals and departments within the warehouse, and developing better labor resource plans (Rahman et al., 2021). This research has explored labor productivity and its influencing factors to gain insight into performance and workforce control within MediaMarkt's consumer electronics warehouse in the Netherlands.

Problem Statement

Poor workforce control results from the inaccurate prediction of labor productivity. Labor productivity is defined as *"the ratio of the total number of items managed to the amount of item-handling working hours"* (Karim et al., 2020). Currently, the *expected* labor productivity, defined by MediaMarkt as the Open Warehouse Rate (OWR), is determined by averaging labor productivity over the past four to six weeks and manually adapting it based on the knowledge and experience of the warehouse manager. No statistical or prediction methods are used to predict the OWR. In general, it is seen that labor productivity varies hugely over time and fluctuates highly around the performance target set by MediaMarkt. As labor productivity varies over time, prediction based on subjective methods results in inaccurate productivity estimations. This, again, leads to wrong predictions of the workforce with costly consequences due to over or under-capacity. Over-capacity leads to unnecessary logistical expenses for MediaMarkt. Lastly, under-capacity leads to low service levels, low customer satisfaction, and (potential) loss of sales. Therefore, it is essential to control the workforce. Moreover, the current forecast of labor productivity does not account for any factors influencing it. MediaMarkt knows from practical experience that features such as volume, order frequency, and composition affect labor productivity. However, the effect of these factors on labor productivity has not been tested or quantified. Therefore, differences in productivity performance cannot be quantitatively explained. MediaMarkt desires to understand what influences labor productivity, as this essential measurement determines the needed workforce. Understanding the underlying patterns

of the OWR would lead to better predictions of the expected workforce per goodsflow. A better prediction would result in a workforce more aligned with the capacity needed, minimizing over- and under-capacity and decreasing costs while increasing service levels. Therefore, the main research question was formulated as follows:

"What factors influence labor productivity in an omnichannel warehouse setting, and how can these factors be leveraged to control the workforce, particularly in the context of different goodsflows?"

Employed Method

Labor productivity at MediaMarkt seems to exhibit significant fluctuations from week to week, and it has become apparent from theory and practice that these fluctuations are caused due to numerous variables influencing labor productivity. The current prediction method was deemed deficient in accounting for these influencing variables. Therefore, an effort was made to employ advanced models to capture the underlying relationship between labor productivity and the influencing features and more accurately predict it. Gradient Boosting Decision Trees (GBDT) were identified as a suitable model choice due to their accurate, understandable, and interpretable characters. Additionally, no large datasets were retrieved; thus, the computational time remained accessible. Specifically, the XGBoost and LightGBM models were implemented because of their robustness, understandability, accuracy, and flexibility. The most influential features per model per goodsflow were determined using permutation-based feature importance. The best model was determined based on the lowest RMSE score, and the corresponding features were evaluated with model-agnostic methods. The GBDT models were efficiently tuned for speed and accuracy by implementing Bayesian optimization, while their robustness was ensured by applying k-fold cross-validation. A comparison was made to a simple baseline model using a moving average to assess the models' performance. Due to a lack of data, the outbound B2C and 2MH were excluded from the research. Table 1 provides an overview of the essential features resulting from the best-performing model per goodsflow.

Results

For inbound MDA, both GBDT models underperformed compared to the baseline model, and the coefficient of variation indicated that the current features only minimally explained the variability in the dependent variable. Therefore, it is excluded from the results and further evaluation. Notably, the LightGBM model outperformed all other models for outbound B2S and BBXD goodsflows in terms of performance and speed. However, the XGBoost, although slower, demonstrated superior performance for regular inbound. For outbound B2S, the features predict labor productivity well with a coefficient of variation of 0.9. However, for the inbound and BBXD goodsflow, the current features do not convincingly explain labor productivity behavior.

Conclusion

Model agnostic techniques helped determine bounds for which productivity displayed particular behavior, i.e., increasing, decreasing, or constant. These findings are organized in tables. These bounds should be considered approximations and interpreted with caution as the values are estimates and may change with model improvements. However, the increasing and decreasing

Table 1: Overview of the most important features from the best performing model per goodsflow

	Inbound	**B2S**	**BBXD**
Important features	quantity full pallets CD/DVD mixed pallets week number of orders mezzanine month console high value cage racking average orderline quantity computer bulk Master carton adherence browngoods	quantity high value cage ordersize bulk week general average orderline quantity total weight browngoods Wednesday foto racking Master carton adherence month bulk computer	computer average orderline quantity
Best Model	XGBoost	LightGBM	LightGBM
RMSE	43.05	6.03	39.01
R^2	0.62	0.90	0.38

patterns can be leveraged to boost labor productivity and control the workforce. It was observed that an increase in the average ordersize and average quantity per orderline for inbound, outbound B2S and BBXD could improve productivity. Moreover, the study determined the quantities at which economies of scale can be achieved and for which warehouse location or product categories diminishing returns on productivity occurred when the quantities (or the number of orderlines) became too high. Although most features' importance scores aligned with expectations, some surprising findings emerged. Particularly regarding the smaller product categories or warehouse locations being important features, as previously no pattern was found in the analysis, and practical insights could not explain the behavior. It's worth noting that, despite being white-box models, GBDT models can exhibit inherent randomness, occasionally producing less intuitive results. Moreover, the permutation-based feature importance might also report lower importance scores for highly correlated variables, thereby influencing the final feature selection. A significant limitation of this study pertains to the utilization of hours used to determine productivity. These hours, by definition, do not account for the truly *"effective"* hours. Although productivity was computed with the direct hours allocated to various activities, it is essential to note that these hours contain unnecessary hours. Situations may arise where employees cannot be sent home, resulting in low productivity due to an excessive workforce for the available tasks. Conversely, during periods of high workload, employees may demonstrate higher productivity compared to days with lighter workloads, where they might intentionally slow down work to fill their hours. Currently, the level of effort employees exert on a given day is not considered. Labor productivity is assessed on an aggregated daily level and does not provide insights into the productivity of individual activities.

Recommendations

Based on the findings in this study, several recommendations are formulated for the organization.

- It is recommended to continue using advanced models for labor productivity prediction, as the advanced GBDT models, XGBoost and LightGBM, have demonstrated the availability to predict and effectively capture complex relationships in the data more accurately. However, continuous improvements are needed to provide new insights into the additional data. Moreover, the results must be carefully interpreted due to the random behavior inherent to the GBDT models.
- Enhance model performance by emphasis on feature engineering. Enhance data availability and quality and create new features that may improve predictions. Additional features, such as workforce size or equipment availability, trained personnel availability, function shift, or exerted effort by employees, should be added to better predict productivity for all goodsflows.
- Univariate methods should be applied to predict productivity if additional data is unavailable accurately.
- Increase the quantity per orderline and the ordersize. Increasing these two features will lead to less diverse orders, reducing travel time and lowering the need for sorting and ventilation, thereby increasing efficiency and labor productivity over the different goodsflows. The current bounds can serve as guidelines.
- Do not surpass the different quantities' thresholds to avoid diminishing returns. The current bounds can serve as guidelines.
- Analyze inter-dependence between goodsflows within the warehouse. One can explore how bottlenecks in one goodsflow effect impact another. Subsequently, one can implement strategies to balance the workforce effectively. Information on employee shifting functions could provide a holistic view of warehouse productivity across different goods flows

Acknowledgments

This Master's Thesis represents the end of my academic career as a student. It has been a long journey which now has regrettably come to an end. I have had a wonderful time being a student at the University of Technology Eindhoven, mainly because of my friends.

First and foremost, I would like to thank my company supervisor, Robert Kemp for his continuous support during my Thesis. Although the project took longer than expected, and your responsibilities increased considerably, time and effort to guide me through the process never wandered. Moreover, your extensive knowledge of MediaMarkt's supply chain helped me better understand retail industry operations. Moreover, I want to express my great appreciation for all my colleagues at MediaMarkt. Working alongside you has been a great pleasure.

Additionally, I would like to thank Nevin Mutlu for being my university mentor. You were always available, whether it would be online or offline. Your critical yet fair feedback helped me to considerably improve my research. I would also like to thank Collin Drent and Willem van Jaarsveld for participating in the final assessment committee for this master's Thesis.

Finally, I thank my family and partner for their unconditional support throughout my academic career. It was long and hard, but you have always been there when needed. A special thanks to my eldest sister and father for guiding me during the last days when finalizing the report.

Emma de Bruijn
Eindhoven, September 2023

Contents

List of Figures

List of Tables

List of Abbreviations

1MH	One-man-handling
2MH	Two-man-handling
ALE	Accumulated Local Effec
ANN	Artificial Neural Networks
B2C	Business-to-Consumer (i.e. direct online selling to the consumer)
B2S	Business-to-Store (i.e. from MediaMarkt's NDC to the MediaMarkt stores)
BBXD	Breaking Bulk Cross Docking
EFB	Gradient-Based One-Side Sampling
FTE	Full Time Equivalent
GBDT	Gradient Boosting Decision Trees
GOSS	Gradient-Based One-Side Sampling
IDL	ID Logistics - a service logistics provider
LSP	Logistics Service Provider
MC	Master Carton
MDA	Major Domestic Appliances
NDC	National Distribution Center
NDPU	Next-Day-Pick-Up
OWR	Open Warehouse Rate
PDP	Partial Dependence Plot
RMSE	Root Mean Squared Error
SKU	Stock Keeping Unit
SMA	Moving Average
TPE	Tree-Parzen Estimator

Chapter 1

Introduction

1.1. General Introduction

Back-end logistics is critical to today's omnichannel retailing strategy, especially warehousing. Retail warehouses are facilities where operations such as receiving, put-away, storing, picking, shipping, return handling, and cross-docking flows are performed (Kembro and Norrman, 2019). These facilities are considered a strategic component of omnichannel retailing (Kembro et al., 2018). As warehouses become more crucial, they are expected to operate efficiently and effectively (Gu et al., 2007). Resources, such as labor, must be allocated among the different warehouse functions. Each function must be carefully implemented, operated, and coordinated to achieve system requirements in capacity, throughput, and service at minimum cost (Gu et al., 2007). Therefore, many challenges result in designing and operating an omnichannel warehouse to meet all these requirements. Retail warehouses often have labor as their most crucial resource and cost. The employees are essential for the productivity of the warehouse as they are the closest to the operations. Any inconsistencies in the (manual) operations may lead to delays, additional costs, and loss of reputation (Glock et al., 2017).

In a highly competitive business environment such as today's retailing landscape, warehouses are constantly under pressure to increase productivity and accuracy while reducing costs and improving customer service (Karim et al., 2020). Therefore, productivity measurement is the most significant dimension for warehousing to monitor the output from the input provided in warehouse operations (Karim et al., 2020; Rahman et al., 2021). However, due to the stochastic demand-driven environment of retailing, there is much uncertainty in predictable phenomena and unpredictable random uncertainty (De Leeuw and Wiers, 2015). Moreover, challenges in the omnichannel context add to the uncertainty in predicting productivity and workload due to differences in demand profiles, the wide range of SKUs and product flows, and their variety of characteristics, which lead to the need for increased handling, integration, and coordination of in-going and outgoing flows. This is especially true in sectors such as consumer electronics (Kembro et al., 2018; Kembro and Norrman, 2019). Analyzing and predicting productivity performance can assist warehouse managers in various ways, including identifying areas of improvement, analyzing the performance levels of individuals and departments within the

warehouse, and developing better labor resource plans (Rahman et al., 2021). This research will explore and predict warehouse productivity to gain insight into performance and workforce control within a consumer electronics warehouse in the Netherlands.

1.2. Company Description

The research pertains to the warehouse operated by MediaMarkt. MediaMarkt Saturn Holding is Europe's leading commerce company for Consumer Electronics. Their product portfolio consists mainly of electronics and electronic-related products, ranging from small to large products. In 2021 the company had 1,018 stores across 13 different countries with about 52,000 employees. The Netherlands has 49 "brick-and-mortar" MediaMarkt stores and an online presence. For these 49 stores, nearly half of the demand is replenished via the National Distribution Center (NDC) in Etten-Leur, Noord-Brabant. Furthermore, the NDC fully covers the online consumer demand. The NDC is managed by the external service logistic provider ID Logistics Benelux (IDL). IDL is a contract logistics specialist in the area of warehousing, with a focus on Food, Retail, Fast-Moving consumer goods & e-commerce. It operates nine warehouses within the Benelux, including the warehouse in Etten-Leur.

MediaMarkt, here-forth used to describe the Dutch segment of MediaMarkt Saturn Holding, operates in a highly competitive landscape with big competitors, such as Coolblue and Bol.com. To maintain market leadership, MediaMarkt has decided to implement an omnichannel strategy with a strong focus on online sales. MediaMarkt aims to be the customer's first choice in Consumer Electronics in stores and digital platforms. In 2019, MediaMarkt switched from a store-central supply model to a centralized supply chain model. In the former, suppliers would directly deliver to the 49 stores in the Netherlands, and each store was responsible for its inventory management. Logistics provider Fiege serviced the online segment. The current supply chain model replenishes the stores and online segment via the new NDC in Etten-Leur. This NDC holds the inventory of both the Business-to-Customer (B2C) and the Business-to-Store (B2S) streams. Since the beginning of 2022, the B2C and B2S goodsflows have been consolidated, with both streams being served by the same inventory. The incoming inventory is subdivided into regular inbound and inbound Major Domestic Appliances (MDA). Moreover, a break-bulk cross-dock (BBXD) flow was set up in 2020 for store replenishments. All Stock Keeping Units (SKUs) sold online are available at the NDC. B2C orders are segmented into parcels, one-man-handling (1MH), two-man-handling (2MH), and Next-Day-Pick-Up (NDPU). For the B2S segment, about 70 to 80% of the value of all SKUs is available at the NDC, which consists of approximately half the number of SKUs. The other half of the SKUs are shipped directly from the supplier to the stores. IDL is responsible for all warehouse operations and the B2S outbound logistics.

1.3. Problem Definition

Many challenges have arisen since the centralization of the NDC to accommodate the omnichannel strategy. The challenge focused on in this report is poor workforce control. Poor workforce control results in the constant need for re-balancing and over- or under-capacity of

labor. Constantly re-balancing employees across goodsflows leads to inefficiencies Vanheusden et al. (2020). Moreover, over-capacity leads to unnecessary logistical expenses for MediaMarkt. Lastly, under-capacity leads to low service levels, low customer satisfaction, and (potential) loss of sales. Therefore, it is essential to control the workforce.

Poor workforce control results from the inaccurate prediction of labor productivity. The unit-based forecast is divided by the expected labor productivity to determine the needed workforce. Labor productivity itself is defined as *"the ratio of the total number of items managed to the amount of item-handling working hours"* (Karim et al., 2020). The division of the forecast by labor productivity results in the number of working hours needed, which is translated into Full-Time-Equivalents (FTE). The *expected* labor productivity is defined by MediaMarkt as the Open Warehouse Rate (OWR). Currently, the OWR is determined by averaging labor productivity over the past four to six weeks and manually adapting it based on the knowledge and experience of the warehouse manager. No statistical or prediction methods are used to predict the OWR.

In general, it is seen that labor productivity varies hugely over time and fluctuates highly around the performance target set by MediaMarkt with IDL, see Appendix A. As labor productivity varies over time, prediction based on subjective methods results in inaccurate productivity estimations. This, again, leads to wrong predictions of the workforce with costly consequences. Moreover, the current forecast of labor productivity does not account for any factors influencing it. MediaMarkt knows from practical experience that order characteristics, such as volume, frequency, and composition, affect labor productivity. However, the effect of these factors on labor productivity has not been tested or quantified. Therefore, differences in productivity performance cannot be quantitatively explained. Moreover, the workload forecast cannot adequately translate into the needed workforce. Even when MediaMarkt knows precisely what the daily inbound, outbound, and BBXD orders and their characteristics are, they still do not know the expected productivity and, thus, the needed workforce to process them in time.

1.4. Research Goal

MediaMarkt desires to understand what influences labor productivity, as this essential measurement is used to determine the needed workforce and assess performance. Specifically, they want to understand the large fluctuations and variations in the OWR performance and what factors influence it. Understanding the underlying patterns of the OWR would lead to better predictions of the expected workforce per goodsflow. A better prediction would result in a workforce more aligned with the capacity needed, minimizing over- and under-capacity and decreasing costs while increasing service levels. The objective is to identify the factors that affect labor productivity and use this information to more accurately predict labor productivity, thereby improving workforce control over all goodsflow in the NDC: outbound B2S, outbound B2C, outbound 2MH, inbound, inbound MDA and BBXD. Specifically, MediaMarkt wants to focus on the order characteristics that influence labor productivity. Therefore, the main research question is formulated as follows:

"What factors influence labor productivity in an omnichannel warehouse setting, and how can these factors be leveraged to control the workforce, particularly in the context of different goodsflows?"

To answer the main research questions, the following sub-questions have been formulated:

1. *What are the key characteristics affecting the labor productivity of the different goodsflows (B2C, B2S, 2MH, Inbound, MDA, BBXD)?*
2. *How do these characteristics affect labor productivity, and how do they differ across different goodsflows?*
3. *What are the most effective methods for measuring and predicting productivity in a warehouse setting, based on the influencing factors and underlying key patterns?*
4. *How can the established influential factors be leveraged to improve labor productivity and workforce control in a warehouse setting?*

In chapter 2, relevant literature to the research is discussed. Then, chapter 3 presents the deployed research methodology. Chapter 4 analyzes and diagnoses influencing factors on labor productivity at MediaMarkt's NDC. Chapter 5 presents the solution design to predict productivity based on relevant factors. Followed by implementing the model in chapter 6. The results are evaluated in 7. Chapter 8 discusses the performed research. Finally, in chapter 9, the study's overall conclusion is drawn, and the recommendations and future research are stated.

Chapter 2

Theoretical Background

2.1. Labor Productivity and Influencing Factors

Labor productivity is a highly employed partial productivity metric to measure the efficient use of labor. It can provide better performance analysis for warehouse management to increase or decrease the number of resources (i.e., FTE) needed. Labor productivity can be impacted by multiple factors from one of three sources: the performed activity, the employee performing the activity, and the environment in which it is operated (Nasirzadeh et al., 2020). The literature review by Basahal et al. (2022) suggests a few universal factors influencing employee productivity in any context. These include employee motivation, monetary rewards, employee recognition, work flexibility, management communication, and labor experience and skills. The current research body on predicting labor productivity in an operation context mainly focuses on Manufacturing, Production, or Construction. An overview of labor productivity's influencing factors in operation literature can be found in Table 2.1.

Table 2.1: Overview of labor productivity's influencing factors in operation's research literature

authors	Field	factors identified
Goel et al. (2017)	Manufacturing	(1) working conditions, (2) pay, (3) work environment, (4) organization structure, and culture, (5) training, learning, and development, (6) HR policies of the organization, (7) technology adoption level (8) focus on clear business goals, (9) conscious focus on improving productivity (10) physical and mental well-being of employees, (11) motivation and enthusiasm, (12) employee's education, (13) country's & worlds macroeconomics (14) number of competitors in the industry, (15) regulatory body's presence in the industry (16) employee's attitude, belief, values, and skills, (17) government regulation environment (18) cross-country skilled labor migration developments, (19) evolution of best practices
Sreekumar et al. (2018)	Manufacturing	(1) product design characteristics, (2) correctness of process plans (3) availability of machinery, material, tools, and other equipment (4) scheduling, (5) technology advancements, (6) working environment (7) standard time estimation, (8) inspection delays (9) assembly problems, (10) workers-related issues, (11) lack of supervisory support (12) working hours, (13) HR & IR related matters.
Naoum (2016)	Construction	(1) Individual worker characteristics: skills, experience, motivation, and health (2) Project-related factors: project complexity, design changes, site conditions, resource availability (3) Organizational factors: management practices, communication, and teamwork.
Hamza et al. (2022)	Construction	(1) project characteristics: project size, complexity, and type (2) project management practices, such as planning, scheduling, and coordination (3) resource availability, such as availability and allocation of equipment, materials (4) site conditions, such as weather, access, and safety (5) individual-related factors: worker skills, experience, motivation, health, and work organization
Thomas and Sakarcan (1994)	Construction Production	(1) work environment: congestion, sequencing, weather, supervision, plant status, information, equipment, tools, materials, and rework (2) execution environment: size of components, specification & quality requirements, work content, design features, and work scope
Sonmez and Rowings (1998)	Construction	(1) quantities completed, (2) job type, (3) crew size, (4) percent overtime, (5) percent laborer, (6) equipment type, (7) temperature, (8) humidity, and (9) precipitation.

Labor productivity has long been recognized as an important measurement in the labor-intensive manufacturing sector, as it does not only reduce costs, it also leads to improved quality, and customer and employee satisfaction, growth, and competitiveness (Goel et al., 2017; Sreekumar et al., 2018). Goel et al. (2017) identifies several factors that are internal to labor productivity after an extensive literature review and categorizes the factors into seven dimensions in the *"FLOPACE"* model, presented in Appendix B. Sreekumar et al. (2018) identify the various factors influencing labor productivity through personal interviews with 100 employees and conclude that delay in ensuring the availability of the right material at the right time is the most significant factor affecting labor productivity in manufacturing industries (Sreekumar et al., 2018). Labor productivity is also an important, well-known concept in the construction environment, as understanding labor productivity's influencing factors helps to effectively measure, analyze, and improve construction performance (Hamza et al., 2022). Both Naoum (2016) and Hamza et al. (2022) provide a comprehensive review of labor productivity's influencing factors in the construction environment. The factors include both project, organization, and individual-related factors. The authors highlight the complex nature of labor productivity and the interrelationships among different factors. Due to the construction industry's complex and multifaceted nature of labor productivity, advanced machine learning models are used to predict productivity performance. Various forms of artificial neural networks (ANN) are often applied. ANN models can handle the complex non-linear relationships of the influencing factors and have an adequate ability to learn from historical data to make accurate predictions. Kavya et al. (2022) provides a comprehensive review of the applications, methodologies, advantages, and limitations of ANN models in predicting construction productivity. The paper by Falkenberg and Spinler (2022) confirms that ANN models are suitable models to include influencing factors to predict labor productivity. However, they remark that ANN models often have low computational efficiency and a black-box nature, complicating the analysis of important variables. The authors show that Extreme Gradient Boosting models can also adequately capture the complex relationship between multiple variables, with improved interpretation over ANN (Falkenberg and Spinler, 2022).

Falkenberg and Spinler (2022) is the first paper to address the lack of overview of potential factors impacting warehouse employees' productivity. The authors devise a framework to identify the main categories that impact warehouse employees' productivity. The main categories are the warehouse in which the operation takes place, the operator, the shift in which the operator is placed, and the product. This framework is based on the work by Goel et al. (2017) on productivity factors and the case-specific factors mentioned by Thomas and Sakarcan (1994) and Sonmez and Rowings (1998). From the research by Goel et al. (2017), only the first two dimensions are considered: employee and organization. Thomas and Sakarcan (1994) use a factor model to predict labor productivity. The authors find that, broadly, labor productivity is affected by the organization and execution continuity. In other words, the work environment and the work to be done. The paper by Sonmez and Rowings (1998) lists several factors which affect construction labor productivity. Falkenberg and Spinler (2022) translates the factors from the manufacturing, production, or construction environment into relevant features influencing

warehouse employees' productivity. Moreover, other factors are added based on interviews with several logistic service providers (LSPs). The relevant factors are summarized in Table A.1 in Appendix B, together with examples provided by Falkenberg and Spinler (2022).

2.2. Predictive Models

Overall, studies predicting productivity either use statistical models or machine learning models. Whereas statistical models handle only a few variables, machine learning methods can integrate multiple variables to consider complex relationships (Falkenberg and Spinler, 2022). As multiple variables influence productivity, machine learning methods can incorporate these to more accurately predict the outcome, whereas univariate models can only use one factor to predict outcomes (Nasirzadeh et al., 2020). Machine learning models learn from the given data and predict more accurately based on continuous learning and improvement over time (Bell, 2022). A popular non-parametric supervised machine learning method is Decision Trees, which is applied in many real-world applications in various industries, including the operational environment. Decision Tree models are generally seen as white-box models and provide higher interpretability. Advanced Decision Tree models show exceptionally high performance and are computationally less expensive than ANN models.

Decision Trees predict a target output by examining the dataset's features and finding the features that present the best possible performance by splitting the data into sub-groups until a final prediction value is found. The splitting decision is based on minimizing the target variable's variance in the subset (James et al., 2013). Single Decision Trees are known for overfitting the training data. Ensemble Tree methods are an extension based on the combination of multiple single Decision Trees, thereby preventing overfitting. Ensemble methods are often preferred as they are highly interpretable, require little data pre-processing, and simultaneously allow for both categorical and numerical variables. Moreover, the methods can handle large amounts of data with many features and predict accurately (James et al., 2013). Ensemble methods combine single trees into an ensemble with bagging or boosting.

Random Forests is a bagging algorithm that builds multiple single trees by simultaneously repeatedly resampling training data with another set of training data. Single trees, i.e. weak learners, are combined to make a collective prediction. The consensus is achieved by averaging the performance of all weak learners. Random Forest algorithms can improve prediction accuracy, and they are particularly useful when complex datasets involve a large number of variables with many interactions between them. However, the combination of multiple decision trees does reduce interpretability. Gradient Boosting Decision Trees (GBDTs) is a boosting algorithm that sequentially builds trees. The algorithm improves the subset of the previous tree with the highest prediction error. The Gradient Boosting method often outperforms Random Forest due to its sequential and self-learning nature. However, the algorithm is more computationally expensive. In general, GBDTs are highly customizable to the application's particular needs, like being learned concerning different loss functions (Natekin and Knoll, 2013). Other key strengths of GBDTs in modeling tasks include their robustness against irrelevant features and scale independence. Two well-known algorithms are associated with

gradient boosting: Extreme Gradient Boosting (XGBoost) and LightGBM.

Firstly, XGBoost is a scalable ensemble technique that has been demonstrated to be a reliable and efficient machine-learning challenge solver (Chen and Guestrin, 2016). The main difference between XGBoost and other boosting techniques is that it uses a new regularization technique, which adds a new term to the loss function to control over-fitting. Therefore, it is faster and more robust (Al Daoud, 2019). Secondly, LightGBM is an accurate model that provides extremely fast training performance using selective sampling of high-gradient instances (Ke et al., 2017). The implementation time is reduced by growing the decision trees leaf-wise instead of evaluating all the previous leaves for each new leaf (Al Daoud, 2019). The main advantages are higher accuracy, faster training speed, handling of large-scale data, independence of data pre-processing and transformations, and a relatively small number of hyperparameters to tune (Makridakis et al., 2022).

2.3. Research Gap

The workforce is determined by dividing the workload by labor productivity (units handled per time period). Currently, whenever this process occurs, the stochasticity in labor productivity is often not considered and seen as a given (Ernst et al., 2004; Van den Bergh et al., 2013). Translating workload into the workforce without any form of stochasticity in productivity ignores any form of uncertainty (Van den Bergh et al., 2013). However, due to inherent uncertainty in demand and the challenges arising from omnichannel strategy, warehouse operations are subject to variability and uncertainty caused by influencing factors (Ishfaq et al., 2016; Van Gils et al., 2017). It is important to consider influences on productivity when determining and balancing the workforce (Vanheusden et al., 2020). Falkenberg and Spinler (2022) are the first authors to address the influencing characteristics on labor productivity in the warehouse context. Building upon this research, the following report aims to analyze how multiple characteristics within and outside of the formulated framework affect labor productivity. Several GBDTs models are explored to find the best prediction method to best understand underlying patterns influencing labor productivity. Thereby providing further evidence of the success of machine learning techniques in prediction.

Chapter 3

Research Design

3.1. Methodology

The 'problem-solving cycle' by Van Aken et al. (2012) serves as the foundation for the research methodology; see Figure 3.1. This methodology is consistent with the requirements of the Master Thesis.

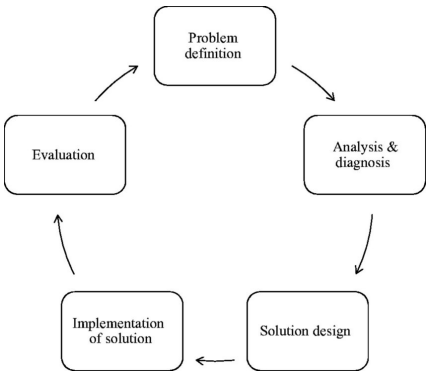


Figure 3.1: The 'problem-solving cycle' by Van Aken et al. (2012)

The first step is identifying and defining the problem. The introduction provided the grounds for the problem identification by stating the problem definition and the main research question. The problem has been identified through multiple semi-structured interviews with stakeholders, such as the warehouse manager from IDL and MediaMarkt, supply planners, IDL business analysts, transport planners, and return logistics managers. Furthermore, the relevant theoretical background is given in chapter 2. Both these aspects motivate the proposed research.

The second step consists of the Analysis and Diagnosis of the problem. This phase also answers sub-questions 1 and 2. Sub-question 1: *What are the key characteristics affecting the labor productivity of the different goodsflows (B2C, B2S, 2MH, inbound, MDA, BBXD)?*. The key features affecting labor productivity are identified by analyzing current literature on labor productivity and influencing factors. Moreover, additional characteristics are determined through stakeholder interviews. To answer sub-question 2: *How do these characteristics affect labor productivity, and how do they differ across different goodsflows?* data is collected on

factors influencing labor productivity according to the critical stakeholders and literature. Based on this data, the relationship between the characteristics and labor productivity is analyzed using exploratory data analysis techniques. Various statistical and visualization techniques are used to identify the driving characteristics of labor productivity per goodsflow. Furthermore, patterns and relationships among features are determined to understand how these characteristics interact and affect labor productivity.

After the preliminary Analysis and Diagnosis phase, the third and fourth steps in the problem-solving cycle are addressed in the Solution Design phase and Solution Implementation, which will also answer sub-question 3: *"What are the most effective methods for measuring and predicting productivity in a warehouse setting, based on the influencing factors and underlying key patterns?"*. MediaMarkt seeks to predict productivity better to translate known workload into the expected number of hours needed. Furthermore, the company wants to understand what (order) characteristics influence labor productivity most to consider these factors when determining the hours required to process quantities on certain days. The prediction model should thus be accurate in predicting labor productivity and should be able to encompass multiple variables for prediction. Generally, the goal is to provide insights into the underlying behavior of labor productivity based on prediction with numerous variables.

There is a trade-off between interpretability and flexibility to consider when choosing the model, and the trade-off is furthermore based on the research goal. When the goal is to understand the association between the dependent variable and the independent variable(s), also known as inference, interpretability becomes more critical, and therefore, more restrictive parametric methods would suit best. When the research goal is to accurately predict based on multiple factors, more flexible and complex models are needed (James et al., 2013). Similarly, in this research, the trade-off holds. On the one hand, MediaMarkt is interested in understanding what characteristics influence labor productivity. On the other hand, the company wants to be able to predict the expected productivity to determine the workforce as accurately as possible. The method must be accurate, understandable, and easy to interpret. Gradient Boosting Decision Trees (GBDTs) are popular non-parametric supervised learning methods applied in many real-world applications in various industries, including the operational environment. Their white-box approach provides higher interpretability, requires little data pre-processing, and reduces overfitting. Advanced decision tree models are often preferred as they show exceptionally high performance but are also computationally less expensive than, for example, artificial neural networks (James et al., 2013). Two well-known algorithms are associated with gradient boosting: XGBoost and LightGBM. Due to the strengths of these models in accuracy, feature selection, and interpretability, these methods are explored in the current research.

The performance of the GBDT models will be compared among each other and to the current prediction method (i.e., the baseline model). The performance of each model is evaluated based on several essential performance measures, including the root mean squared error (RMSE) and the coefficient of determination (R^2). Overall, selecting the most effective method for measuring and predicting productivity and workforce will depend on the data's specific characteristics and the analysis's goal. Identifying the most accurate and robust approach may require

experimentation and iterative testing.

The final step evaluates the solution design and implementation. In this step, the final sub-question can be answered: *"How can the established influential factors be leveraged to improve labor productivity and workforce control in a warehouse setting?"*. Based on the insights gained from sub-questions 1 and 2 and the insights from the prediction model analysis (sub-question 3), recommendations and effective strategies can be formulated to improve workforce control. By analyzing productivity through different characteristics, it may be possible to identify patterns and trends to inform decisions. Leveraging the characteristics to improve labor productivity and workforce control in a warehouse setting will require data analysis, process optimization, and strategic decision-making. By using insights from the data to inform decisions, it may be possible to achieve significant improvements in efficiency and productivity while reducing the workforce and ensuring high customer service levels.

Chapter 4

Analysis and Diagnosis

4.1. Warehouse Labor Productivity

The research will analyze and predict warehouse labor productivity based on several influencing factors in an omnichannel environment. The goal is to gain insight into (productivity) performance and workforce control within a consumer electronics warehouse in the Netherlands. The *actual* labor productivity is calculated by dividing the quantity processed per goodsflow in period t by the total number of hours registered on that goodsflow in period t, see equation 4.1.

$$labor\ productivity\ (t) = \frac{total\ quantity\ processed\ (t)}{total\ number\ of\ hours\ (t)} \tag{4.1}$$

The *expected* labor productivity, i.e., OWR, is currently determined by taking the average labor productivity over the past four to six weeks and manually adapting it based on the knowledge and experience of the warehouse manager. The prediction is judgmentally created and not based on statistical or prediction methods, leading to inaccurate estimations. Furthermore, the current prediction of labor productivity does not account for any factors influencing it. The theoretical background highlights that multiple factors exert influence over labor productivity. It is not constant over time, contrary to what is often assumed in workforce predictions (Van den Bergh et al., 2013).

4.2. Evaluation of Impact Factors on Labor Productivity

Falkenberg and Spinler (2022), are the first authors to devise a framework to identify the main categories that impact warehouse employees' productivity. The authors do not claim it is fully exhaustive. Therefore, the current research builds on the established framework. Based on several stakeholder interviews, additional factors are added to the existing framework. Once the important factors were identified, an effort was made to gather relevant data on the features whenever feasible. The method of data collection and mutations is described in Appendix C. The extracted features from MediaMarkt's and IDL's information systems used in the current research are presented in Table 4.1.

Table 4.1: Extracted features from MediaMarkt placed in the warehouse employee's productivity framework by Falkenberg and Spinler (2022)

Impact category	Impact factor	Extracted Feature MediaMarkt
Warehouse	Location	-
	Design	Warehouse Location
	Size	Warehouse Location
	Process maturity	-
	Degree of automation	-
Operator	Job role	-
	salary	-
	Age	-
	Experience	-
	Training	-
	Days off	-
	Sick days	-
	Past performance	-
Shift	Date	Month of the year
		Week of the year
		Day of the week
	Shift type	Goodsflow type
	Work monotony	-
	Supervisors	-
	Extra payments	-
	Workforce size	-
Product	Quantity	Quantity
		Number of orders (average ordersize)
		Number of orderlines (average orderline quantity)
	Volume	Daily volume processed
	Weight	Daily weight processed
	Special Handling Requirements	Product category
		Master Carton value adherence
		Mixed pallet ratio
		Inbound issue score

The first category is the **warehouse**, i.e., the working environment. The features used in the research are the total quantity and number of orderlines per warehouse location. The locations have been categorized under bulk, racking, mezzanine, high-value cage, and smartbar. Differences between SKUs from separate warehouse locations are expected to have an impact on warehouse operations, thereby impacting labor productivity. For example, larger SKUs stored in bulk are more unwieldy to handle, needing additional equipment or machinery. Moreover, it leads to faster exhaustion of order pickers (Falkenberg and Spinler, 2022), causing lower productivity. In contrast, products stored in the mezzanine location, consisting of smaller items frequently packaged together, are expected to have a faster storage process. These discrepancies between warehouse locations may have an impact on labor productivity. For the second category, **operator** (i.e., employee), employees' data could not be gathered due to privacy regulations at IDL. Therefore, no features are included. The third impact category is **shift**, the environment in which the operator is active. The impact factors *Date* and *Shift type* are included in the research. The *Date* feature includes the weekday, month, and week. The workload might differ across days of the week, weeks, or months, impacting labor productivity. The shift type distinguishes between the different goods flows: inbound regular and MDA, outbound B2S and

BBXD. The goodsflow type provides additional information on the type of work the employees perform.

The final impact category, **product**, represents the actual product handled by the employee. The efficiency with which a product can be handled depends on the quantity, number of orders, number of orderlines, volume, and weight. Larger values of these features create economies of scale. However, when these features' values cross a certain threshold, productivity growth is expected to diminish, stagnate, or even decrease. For example, when the number of orderlines increases, the SKU diversity also increases. At outbound, this leads to higher travel time to order-pick the SKUs from each location. At inbound, this leads to a higher need for sorting of SKUs and increased travel time to store the items at different locations. Both negatively impact labor productivity. Another example is the total weight handled per day. For outbound operations, larger products are expected to be more unwieldy to handle and could lead to faster exhaustion of order pickers Falkenberg and Spinler (2022). Conversely, picking many small items could lead to inefficiencies. The average order size and average quantity per orderline are also included. It is expected that when the average ordersize and/ or the average quantity per orderline decreases, inefficiencies in handling occur, negatively impacting labor productivity.

Additional features from stakeholder interviews are placed under special handling requirements. The quantity and number of orderlines per product category are added. The different product categories are general, foto, CD/DVD, computer, console, browngoods, and whitegoods. SKUs belonging to the same product category share similar attributes, which might lead to similar handling. Conversely, SKUs from different product groups might differ, leading to different handling. Therefore, it is expected that the product category mix influences labor productivity. Furthermore, the Master Carton (MC) value adherence is added. The MC adherence is the daily ratio of orderlines adhering to the MC value (i.e., case pack size). Low adherence to the MC value is expected to lead to lower efficiency. When the MC value is more adhered to at inbound, the sorting and put-away of these items are more efficient. Similarly, handling efficiency increases for picking activities if the MC value is adhered to. Therefore, it is expected that the MC adherence has an effect on labor productivity. The mixed pallet ratio represents the number of mixed pallets on the total number of pallets received on a day. Mixed pallets contain many different SKUs, which must be sorted before further processing. Therefore, a higher mixed pallet ratio is expected to decrease efficiency and negatively influence inbound labor productivity. The inbound issue score is a weighted score, which represents the relative amount of issues a supplier is expected to create based on the total number of issues and the total quantity supplied by the supplier. When the inbound issue score is high, this means that the probability of inbound issues is higher. It is expected that a higher inbound issue score negatively impacts labor productivity.

The framework provides an overview of the features included and excluded from the research. Several features were excluded due to a lack of available data or access to the relevant data. The collected data is subjected to an evaluation against general expectations. This assessment aims to determine to what extent the observed data aligns with MediaMarkt's beliefs.

4.3. Analysis Labor Productivity Features

The available data does not provide evidence that changes in labor productivity within a given goodsflow are influenced by positive or negative fluctuations in the productivity of other goodsflows. Therefore, the relationships between the features and labor productivity are analyzed independently per goodsflow. The relationship between the features and labor productivity is analyzed using exploratory data analysis techniques. Various statistical and visualization techniques are used to identify the driving features of labor productivity. An overview of the relationships between each feature and labor productivity per goodsflow is presented in Table 4.2 for regular inbound and MDA and in Table 4.3 for outbound B2S and BBXD. The outbound B2C and 2MH goodsflows have been excluded from further research as insufficient data on features could be extracted from the SAP system. Furthermore, BBXD features and labor productivity are analyzed on a weekly level because the BBXD activities are divided over the week, but the features' values are only registered on one day of the week.

A Spearman Rank coefficient correlation test is used to test the strength and direction of the relationships. This test is used as it does not assume normality or a linear relationship between the variables. It is less sensitive to outliers, making it a more robust measure. Note that the extreme outliers have been removed from the dataset, using the rule of thumb: $[y - 3 \cdot \sigma(y), y + 3 \cdot \sigma(y)]$. A significance level of $p < .01$ is chosen. The significant correlations are classified as follows:

- $correlation_{spearman} < 0.2$: extremely weak
- $correlation_{spearman} < 0.3$: weak
- $0.3 \leq correlation_{spearman} < 0.5$: moderate
- $0.5 \leq correlation_{spearman} < 0.7$: strong
- $correlation_{spearman} \geq 0.7$: extremely strong

Note that a negative correlation is indicated with a minus sign. Moreover, the relationships between each goodsflows' features and corresponding labor productivity are visualized with scatterplots. Linear and non-linear trendlines are plotted to find the relationship type. Based on the visualization, the relationships between the features and productivity are either linear or represent diminishing returns. Diminishing returns in labor productivity describes the effect when increasing one feature yields progressively smaller increases in labor productivity. The more this feature increases, the more the rate of increase in labor productivity begins to decline. At a certain point, the growth can even lead to a negligible difference or negative gain in productivity. When the table indicates *"no discernible trendline*, this indicates that the data is either (almost) vertical or horizontal, and no trend was discerned, although a significant correlation was found.

Table 4.2: Overview relationship labor productivity and features Inbound Regular and MDA

Characteristics	Inbound	Inbound MDA
quantity	moderate positive linear trendline (r=0.38, p<.01)	moderate positive linear trendline (r=0.33, p<.01)
mixed pallet ratio	no significant correlation	-
number full pallets	no discernible trendline (r=0.14,p<.01)	-
number mixed pallets	no discernible trendline (r=0.18,p<.01)	-
inbound issue score	strong positive linear trendline (r=0.55,p<.01)	no discernible trendline (r=0.28,p<.01)
number of orderlines	strong positive linear trendline (r=0.62,p<.01)	moderate positive linear trendline (r=0.45,p<.01)
number of orders	strong positive linear trendline (r=0.64,p<.01)	weak positive linear trendline (r=0.28,p<.01)
average quantity per orderline	moderate positive linear trendline (r=0.34,p<.01) *	no significant correlation
average ordersize	moderate positive linear trendline (r=0.33,p<.01) *	weak positive linear trendline (r=0.20,p<.01)
master carton adherence	extremely weak negative linear trendline (r=-.16,p<.01) *	-
warehouse location: bulk	quantity: no discernible trendline (r=0.25,p<.01)	quantity: moderate positive linear trend (r=0.36,p<.01)
	orderlines: no discernible trendline (r=0.25,p<.01)	-
warehouse location: racking	quantity: strong positive linear trendline (r=0.56,p<.01)	quantity: moderate positive (linear) trendline (r=0.31,p<.01)**
	orderlines: strong positive linear trendline (r=0.51,p<.01)	-
warehouse location: mezzanine	quantity: strong positive linear trendline (r=0.61,p<.01)	no significant correlation**
	orderlines: strong positive linear trendline (r=0.52,p<.01)	-
warehouse location: high value cage	quantity: strong positive linear trendline (r=0.65,p<.01)	-
	orderlines: strong positive linear trendline (r=0.61 ,p<.01)	-
warehouse location: SmartBar	quantity: no significant correlation	-
	orderlines: no significant correlation	-
product category: general	quantity: no discernible trendline (r=0.25,p<.01)**	-
	orderlines: no discernible trendline (r=0.25,p<.01)**	-
product category: browngoods	quantity: strong positive linear trendline (r=0.53,p<.01)	-
	orderlines: moderate positive linear trendline (r=0.47,p<.01)	-
product category: CD/DVD	quantity: no discernible trendline (r=0.39,p<.01)**	-
	orderlines: no discernible trendline (r=0.39,p<.01)**	-
product category: Computer	quantity: strong positive linear trendline (r=0.67,p<.01)	-
	orderlines: strong positive linear trendline (r=0.58,p<.01)	-
product category: Foto	quantity: no discernible trendline (r= 0.31,p<.01)**	-
	orderlines: no discernible trendline (r=0.23,p<.01)**	-
product category: Console	quantity: strong positive (linear) trendline (r=0.51,p<.01)**	-
	orderlines: moderate positive (linear) trendline (r=0.48,p<.01)**	-
product category quantity: Whitegoods	quantity: moderate positive linear trendline (r= 0.44,p<.01)	-
	orderline: moderate positive linear trendline (r= 0.42,p<.01)	-

* clustered data with (some) outliers
** data contains many zero values
***potentially a pattern of diminishing returns

Table 4.3: Overview relationship labor productivity and features Outbound B2S and BBXD

Characteristics	Outbound B2S	BBXD
quantity	extremely strong positive linear trendline (r=0.79,p<.01)	strong positive trendline diminishing returns (r=0.52,p<.01)
number of orderlines	strong positive linear trendline (r=0.67,p<.01)	strong positive trendline diminishing returns (r=0.54,p<.01)
number of orders	strong positive trendline diminishing returns (r=0.57,p<.01)	moderate positive trendline diminishing returns (r=0.49,p<.01)
average quantity per orderline	no significant correlation	no significant correlation
average ordersize	no significant correlation	no significant correlation
master carton adherence	weak negative linear trendline (r=-0.27,p<.01)	moderate positive linear trendline (r= 0.34,p<.01)
total weight	strong positive linear trendline (r= 0.53,p<.01)	no significant correlation
total volume	data excluded[1]	data excluded[1]
warehouse location: bulk	quantity: moderate positive linear trend (r= 0.30, p<.01)	no significant correlation
	orderlines: moderate positive linear trend (r=0.46, p<.01)	no significant correlation
warehouse location: racking	quantity: strong positive linear trendline (r=0.66,p<.01)	quantity: weak positive linear trendline (r=0.27,p<.01)
	orderlines: strong positive trendline (r=0.61,p<.01)	orderlines: moderate positive linear trendline (r= 0.45,p<.01)
warehouse location: mezzanine	quantity: strong positive linear trendline (r=0.56,p<.01)	quantity: strong positive linear trendline (r= 0.58,p<.01)
	orderlines: strong positive linear trendline (r=0.57,p<.01)	orderlines: strong positive linear trendline (r= 0.56,p<.01)
warehouse location: high value cage	quantity: strong positive trendline diminishing returns (r=0.68,p<.01)	quantity: moderate positive trendline diminishing returns (r=0.44,p<.01)
	orderlines: strong positive trendline diminishing returns (r=0.61,p<.01)	orderlines: moderate positive trendline diminishing returns (r=0.35,p<.01)
warehouse location: SmartBar	-	quantity: moderate positive (linear) trendline (r=0.47,p<.01)**
	-	orderlines: moderate positive (linear) trendline (r=0.45,p<.01)**
product category: general	quantity: moderate positive (linear) trendline (r= 0.32,p<.01)**	no significant correlation
	orderlines: moderate positive (linear) trendline (r=0.31,p<.01)**	no significant correlation
product category: browngoods	quantity: strong positive trendline (r= 0.63,p<.01)***	no significant correlation
	orderlines: strong positive linear trendline (r=0.59,p<.01)	no significant correlation
product category: CD/DVD	quantity: no discernible trendline (r=0.37,p<.01)**	no significant correlation
	orderlines: no discernible trendline (r=0.36,p<.01)**	no significant correlation
product category: Computer	quantity: extremely strong positive trendline diminishing returns (r=0.72,p<.01)	orderlines: strong positive trendline diminishing returns (r=0.53,p<.01)
	orderlines: strong positive trendline (r= 0.63,p<.01)***	quantity: strong positive trendline diminishing returns (r=0.58,p<.01)
product category: Foto	quantity: moderate positive linear trendline (r=0.38,p<.01)	no significant correlation
	orderlines: moderate positive linear trendline (r=0.35,p<.01)	no significant correlation
product category: Console	quantity: moderate positive (linear) trendline (r= 0.48,p<.01)**	no significant correlation
	orderlines: moderate positive (linear) trendline (r=0.46,p<.01)	no significant correlation
product category quantity: Whitegoods	quantity: strong positive linear trendline (r= 0.55,p<.01)	no significant correlation
	orderlines: strong positive trendline (r= 0.57,p<.01)***	no significant correlation

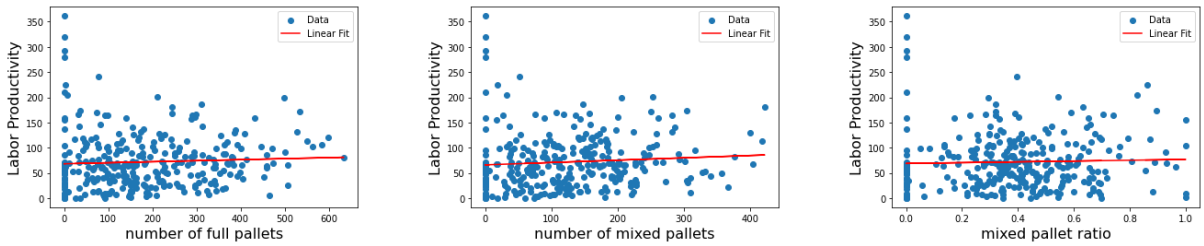
* clustered data with (some) outliers
** data contains many zero values
***potentially a pattern of diminishing returns
[1] data is excluded as it contained predominantly inaccuracies in entries

4.4. Interesting Findings

The detailed analysis of the features per goodsflows is available in Appendix D. In this section, the remarkable findings from the analysis are discussed.

4.4.1. Mixed Pallet Ratio

The relationships between the number of full pallets, mixed pallets, and mixed pallet ratio and labor productivity are displayed in Figure 4.1. The number of full and mixed pallets have an extremely weak positive relationship with labor productivity ($r = 0.14$ and $r = 0.18$, $p < .01$, respectively). Moreover, the relationship between the mixed pallet ratio and labor productivity is insignificant ($p = .14$). The number of full pallets was expected to correlate with labor productivity positively. The rationale is that full pallets simplify operations by reducing the need for sorting and enabling easier receiving and storage than mixed pallets. Relatively more full pallets indicate higher quantities processed more efficiently, creating economies of scale. A pattern of diminishing returns was expected for the number of mixed pallets and mixed pallet ratio. However, this anticipation was not substantiated. Therefore, the relationships are further analyzed.



(a) Labor Productivity vs. number of full pallets

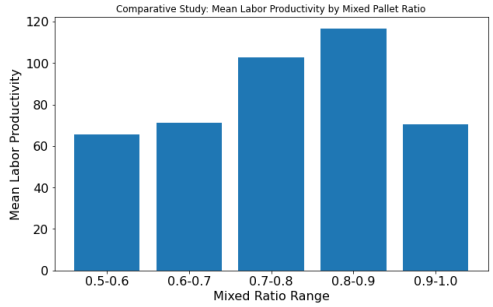
(b) Labor Productivity vs. number of mixed pallets

(c) Labor Productivity vs. mixed pallet ratio

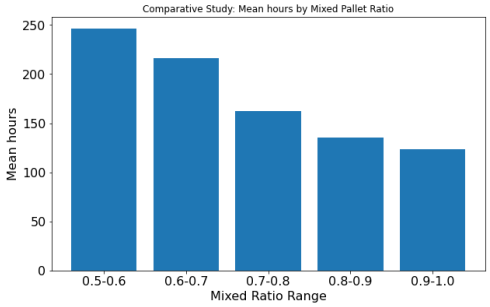
Figure 4.1: Relationship of full pallets, mixed pallets, and mixed pallet ratio vs. labor productivity

A comparative analysis is performed, which compares the mean labor productivity over different ranges of the mixed pallet ratio; see Figure 4.2a. The mean labor productivity seems to increase as the mixed pallet ratio rises. However, a Kruskal-Wallis test indicated no significant difference in mean labor productivity across the different mixed ratio ranges (test statistic = 6.87, $p = .14$). A comparative study with the mean number of hours across the mixed ratio ranges shows a clear decreasing pattern; see Figure 4.2b. There is a significant difference between the mean hours across the mixed pallet ratio ranges (Kruskal-Wallis test statistic 19.35,$p < .01$). Thus, as the mixed pallet ratio increases, i.e., relatively more mixed pallets arrive, fewer hours are used to process these pallets. No significant differences were found between the mean quantities across the mixed ratio ranges (Kruskal-Wallis test statistic = 5.59, $p = .23$). Thus, the mean quantity remains constant across the mixed pallet ratio ranges. Still, a decrease in the number of hours is confirmed. Therefore, labor productivity increases as the relative amount of mixed pallets increases.

According to the comparative study findings, relatively more mixed pallets lead to higher



(a) Mean Labor Productivity vs. mixed pallet ratio



(b) Mean number of hours vs. mixed pallet ratio

Figure 4.2: Relationship of full pallets, mixed pallets, and mixed pallet ratio vs. Labor Productivity

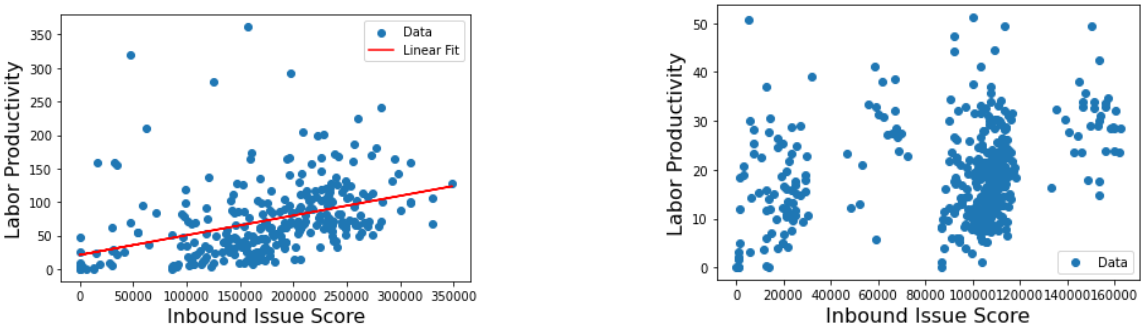
productivity, contrary to MediaMarkt's beliefs. Processing mixed pallets requires more competency from inbound employees, as the pallets must be more carefully received and sorted compared to full pallets. It could be that when the mixed pallet ratio is higher, inbound employees with more training and skills are scheduled. Another reason could be that higher task complexity and variety associated with an increased mixed pallet ratio may lead to higher productivity. The theory has shown that task complexity is an essential determinant of human behavior and task performance (Liu and Li, 2011). Additional data on employees' skills, training, level of experience, or motivation could provide further insights into the observed patterns in the data. However, this data is currently unavailable.

4.4.2. Inbound Issue Score

Suppliers may have inaccuracies when delivering orders, such as missing information or damaged, missing, or surplus items. These inaccuracies lead to lower productivity as more work must be performed to process these items than when no inaccuracies occur. It was expected that the inbound issue score would negatively impact labor productivity. However, for regular inbound, a strong positive linear trend is found; see Figure 4.3a. Furthermore, the inbound issue score for inbound MDA displays an unusual pattern; see Figure 4.3b. Only a limited amount of suppliers supply MDA inbound. Therefore, the score falls within certain bounds, and there is no discernible pattern between the inbound MDA issue score and labor productivity. The difference between the data and expectation is probably due to the several assumptions underlying the inbound issue score. The inbound issue score is too highly dependent on the total daily quantity. Another method should be applied to address the effect of the number of inbound issues on inbound productivity. The current data does not allow for more straightforward implementation due to the many inconsistencies in the manually entered data. Due to the data inconsistency, poor data quality and high dependence on the total daily quantity, the inbound issue score is excluded from the prediction model.

4.4.3. Average Quantity per Orderline

An increase in the average quantity per orderline is expected to lead to higher efficiency in warehouse operations due to consolidated processing and handling, thereby increasing

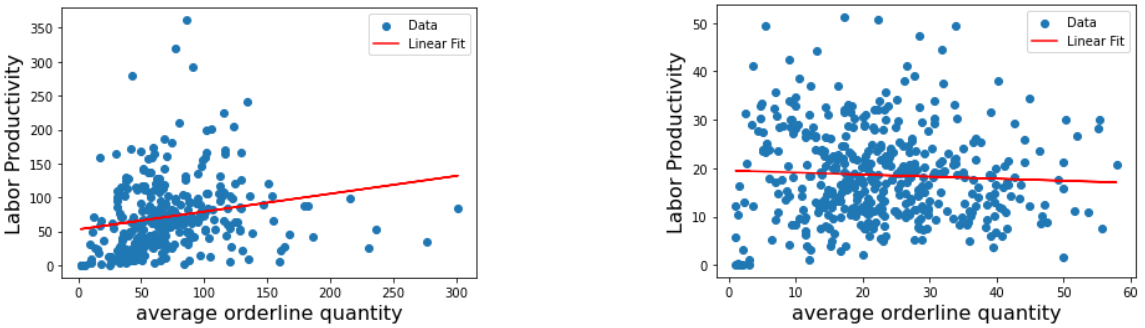


(a) Regular Inbound

(b) Inbound MDA

Figure 4.3: Relationship between Labor Productivity and the inbound issue score

productivity. Conversely, decreasing the average quantity per orderline can lead to lower efficiency because it creates more diverse orderlines with smaller quantities, increasing the overall processing time and reducing productivity. The relationship between the average quantity per orderline and labor productivity is displayed in Figures 4.4a, 4.4b, 4.5a and 4.5b, for regular inbound, inbound MDA, outbound B2S, and BBXD, respectively.



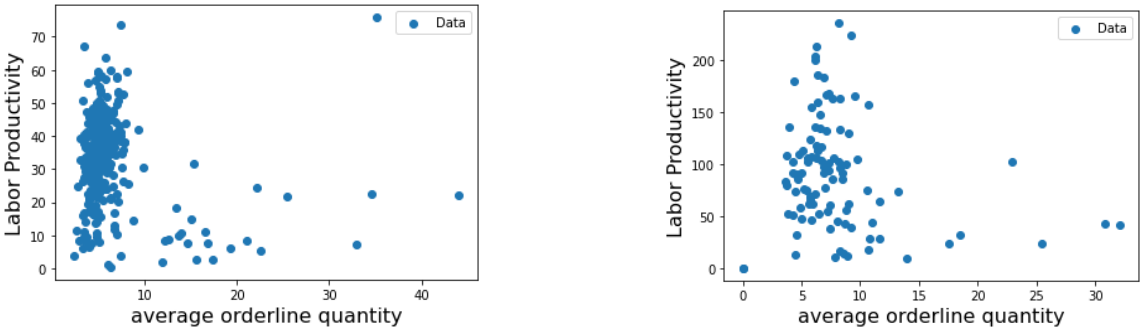
(a) Regular Inbound

(b) Inbound MDA

Figure 4.4: Relationship between Labor Productivity and the average quantity per orderline

A moderate positive trendline is discovered for regular inbound between the average quantity per orderline and labor productivity. No discernible trendline is established between the average orderline quantity and labor productivity for outbound B2S, inbound MDA, and BBXD. The data is scattered for inbound MDA, indicating no clear pattern between labor productivity and average orderline quantity. For outbound B2S and BBXD, the average orderline quantity is relatively stable and falls between narrow bounds, with some extremely high averages as exceptions. Thus, the average quantity per orderline remains relatively constant, independent of labor productivity. Overall, from the visualization, the average orderline quantity does not seem to impact labor productivity.

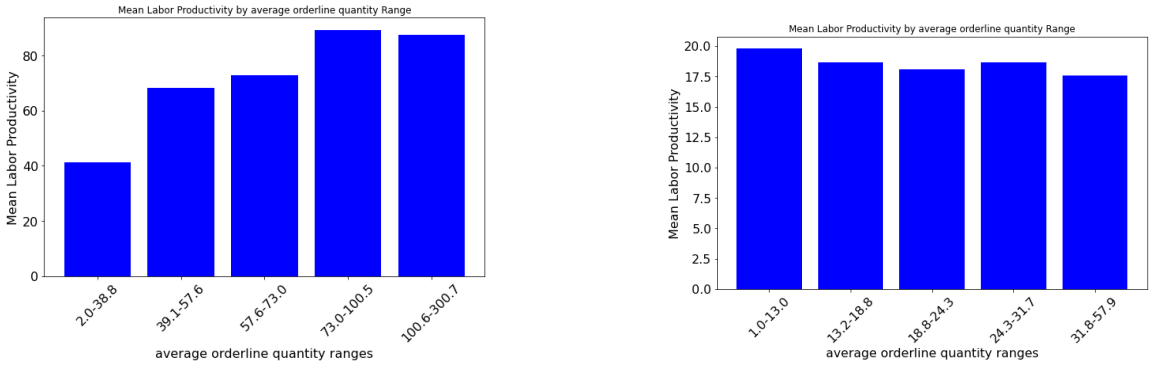
A comparative analysis is performed because, contrary to expectation, no positive trend was discerned for inbound MDA, outbound B2S, and BBXD. For each goodsflow, the mean labor productivity over different ranges of the average orderline quantity is calculated. For regular inbound, the comparative analysis showed similar findings as scatterplot 4.4a: the mean labor productivity increases significantly as the average orderline quantity increases (Kruskal-Wallis



(a) Outbound B2S (b) BBXD

Figure 4.5: Relationship between Labor Productivity and the average quantity per orderline

test statistic = 39.50, $p < .01$). Moreover, it showed that the mean labor productivity remains quite constant after a threshold average quantity per orderline is reached, possibly indicating diminishing returns; see Figure 4.6a. For inbound MDA, the Kruskal-Wallis test indicated no significant differences between mean labor productivity and the average orderline quantity (test statistic = 3.20, $p = 0.52$). The mean labor productivity is quite similar for all ranges of the average quantity per orderline; see Figure 4.6b. Thus, the average orderline quantity does not indicate labor productivity well for inbound MDA.

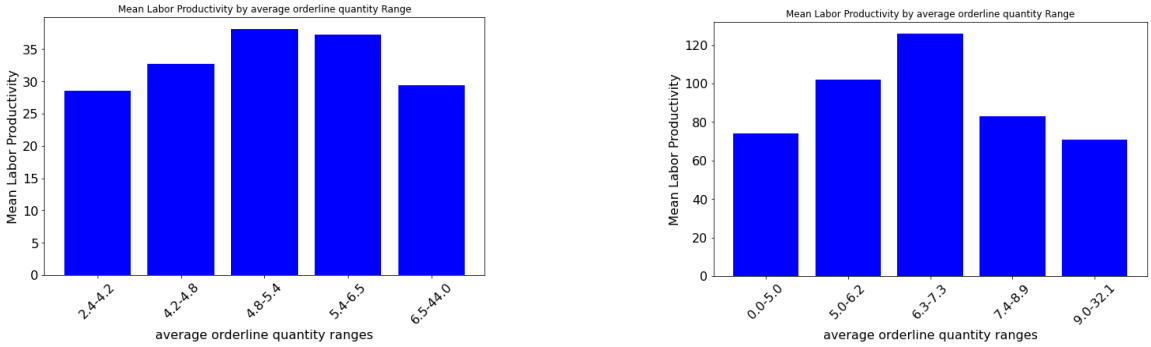


(a) Regular inbound (b) Inbound MDA

Figure 4.6: Comparative analysis: mean labor productivity vs. average orderline quantity

For outbound B2S, according to the scatterplot, the average quantity per orderline remains relatively constant, independent of labor productivity. However, Figure 4.7a shows a slight increase in the mean productivity as the average orderline quantity increases. After a certain average quantity per orderline is achieved, the mean productivity decreases again. The difference between the mean productivity across the average orderline quantity range is deemed significant (Kruskal-Wallis test statistic = 23.37, $p < .01$). A similar significant pattern is found for BBXD (Kruskal-Wallis test statistic = 18.17, $p < .01$); see Figure 4.7b. Thus, although the average orderline quantity for outbound B2S and BBXD initially seemed entirely independent of labor productivity, some effect is visible in the comparative analysis. However, it is unknown if this threshold is created due to days/weeks where the quantities were extremely high, causing an overload of work for the available capacity, or if the average itself influences the efficiency in handling. The behavior of the average orderline quantity will be further analyzed with the

prediction model.



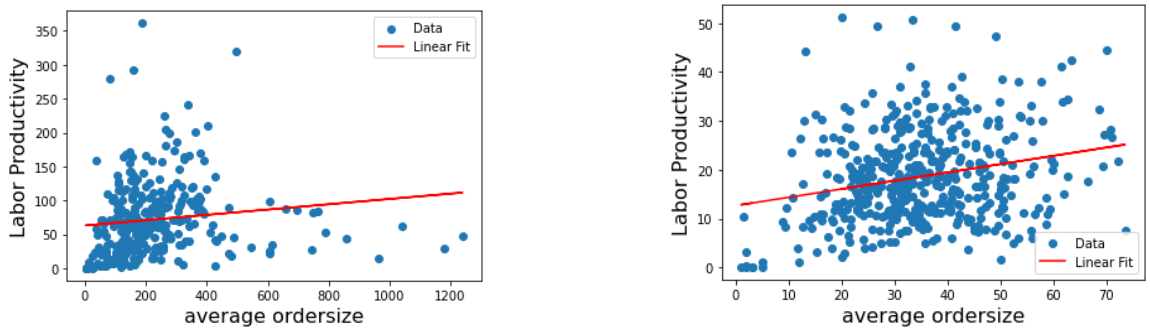
(a) Outbound B2S

(b) BBXD

Figure 4.7: Comparative analysis: mean labor productivity vs. average orderline quantity

4.4.4. Average Ordersize

Similar to the average orderline quantity, it is expected that when the average ordersize increases, this leads to higher efficiency due to consolidated processing and handling, thereby increasing productivity. Conversely, if the average ordersize decreases, the handling efficiency will diminish and negatively impact labor productivity. The relationship between the average ordersize and labor productivity is displayed in Figures 4.8a, 4.8b, 4.9a and 4.9b, for regular inbound, inbound MDA, outbound B2S, and BBXD, respectively.

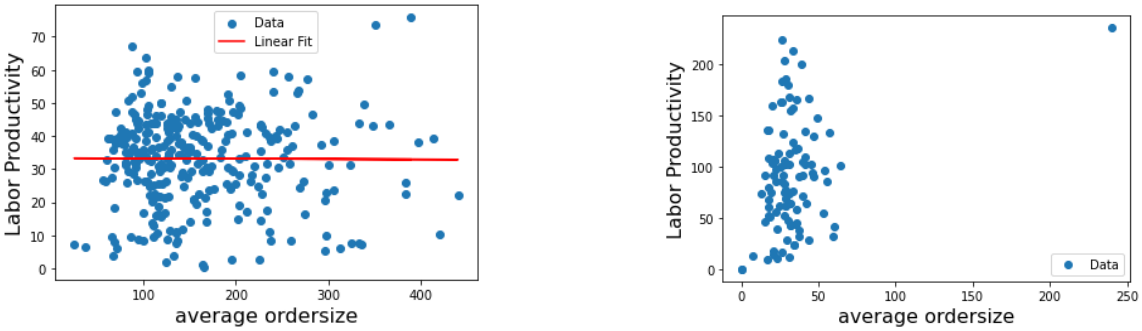


(a) Regular Inbound

(b) Inbound MDA

Figure 4.8: Relationship between Labor Productivity and the average ordersize

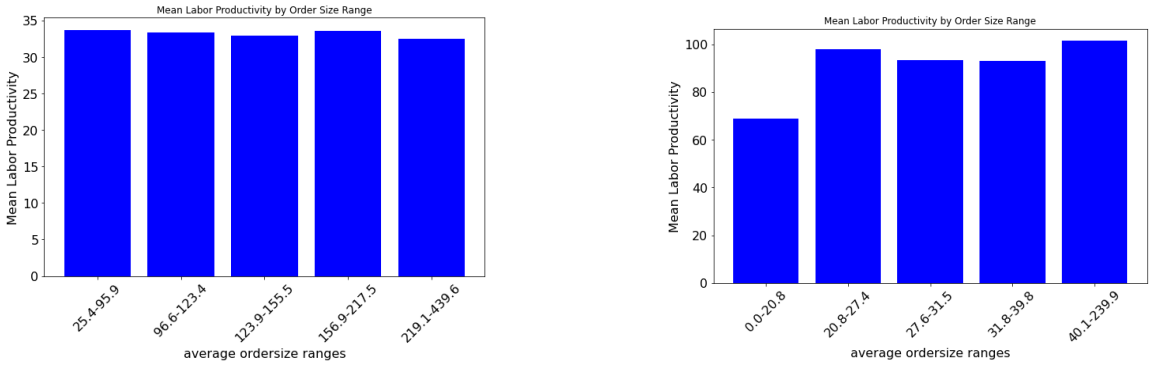
Positive linear trendlines are discovered between the average ordersize and labor productivity for regular inbound and inbound MDA. However, no significant pattern exists for outbound B2S and BBXD. The outbound B2S data is scattered, and the average ordersize and labor productivity move independently. For BBXD, it seems that the average ordersize remains relatively constant independent of the changes in productivity. A positive relationship was expected between the average ordersize and labor productivity for all goodsflows. Therefore, an additional comparative analysis is performed in a similar fashion as with the average orderline quantity. As expected for regular inbound and inbound MDA, the mean productivity increases as the ordersize increases for these goodsflow. The increasing differences between the mean labor productivity and the average ordersize range are deemed significant with a Kruskal Wallis test statistic of 32.68 and 20.52, $p < .01$, for inbound regular and MDA, respectively.



(a) Outbound B2S (b) BBXD

Figure 4.9: Relationship between Labor Productivity and the average ordersize

For outbound B2S, the Kruskal-Wallis test indicated no significant difference in the mean labor productivity across the average ordersize ranges (test statistic = 0.61, $p = 0.96$). The average ordersize does not indicate labor productivity at outbound B2S, and other factors are better predictors of labor productivity. Possibly, average ordersize has little influence as orders are consolidated for picking purposes. The picking operation is the largest outbound warehouse manual operation and, therefore, the most influential on labor productivity. For BBXD, a very small average ordersize does seem to lead to lower average labor productivity, as seen in Figure 4.10b. When a certain threshold is reached, the mean productivity seems relatively stable. This would be as expected, indicating that smaller orders at BBXD would lead to more ventilation efforts as more orders with smaller amounts must be distributed over the stores. However, the Kruskal-Wallis test indicates no significant difference (test statistic = 4.61, $p = 0.33$). Thus, this conclusion cannot be drawn based on the current data. The prediction model further tests the behavior of the average ordersize and possible interaction effect with other essential features.

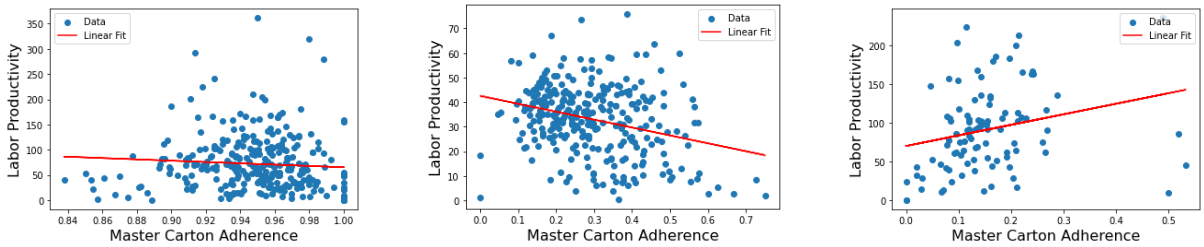


(a) Outbound B2S (b) BBXD

Figure 4.10: Comparative analysis: mean labor productivity vs. average ordersize

4.4.5. Master Carton Adherence

The relationship between the Master Carton (MC) adherence and labor productivity has been displayed in Figure 4.11a, Figure 4.11b, and 4.11c, for inbound, outbound B2S and BBXD, respectively. For BBXD, a positive relationship exists between MC adherence and labor productivity, which aligns with expectations. When the orderlines adhere to the master carton

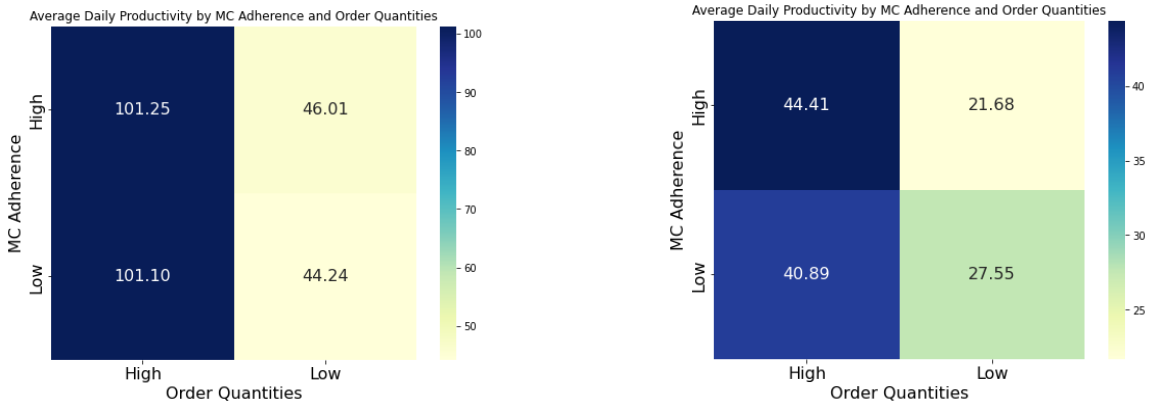


(a) Inbound (b) Outbound B2S (c) BBXD

Figure 4.11: Relationship between Labor Productivity and Master Carton Adherence

value, the handling time is expected to be lower, thereby increasing productivity. Higher adherence would thus indeed result in higher productivity and vice versa. For outbound B2S and regular inbound, the relationship between the master carton value adherence and productivity is negative. Contrary to the general expectation, labor productivity seems to decrease (increase) when the master carton adherence increases (decreases), and vice versa. An underlying negative relationship between the master carton adherence and the total daily quantity causes this unexpected relationship.

Further analysis is performed to understand the difference in expectations. A 2x2 matrix has been created that categorizes the data into four different combinations: low quantity - low MC adherence, low quantities - high MC adherence, high quantities - high MC adherence, and high quantities - low MC adherence. High and low are determined based on whether the data is higher or lower than the mean. The average labor productivity per category is then computed. Now, one can identify how the different scenarios of MC adherence and daily quantities relate to daily labor productivity. The color-coding and annotations in the matrix visually represent each combination's average daily labor productivity values. Darker colors indicate higher average productivity values, while lighter colors indicate lower values. Furthermore, Tukey's HSD test is used to identify which specific combinations show statistically significant differences in average labor productivity.



(a) Inbound (b) Outbound B2S

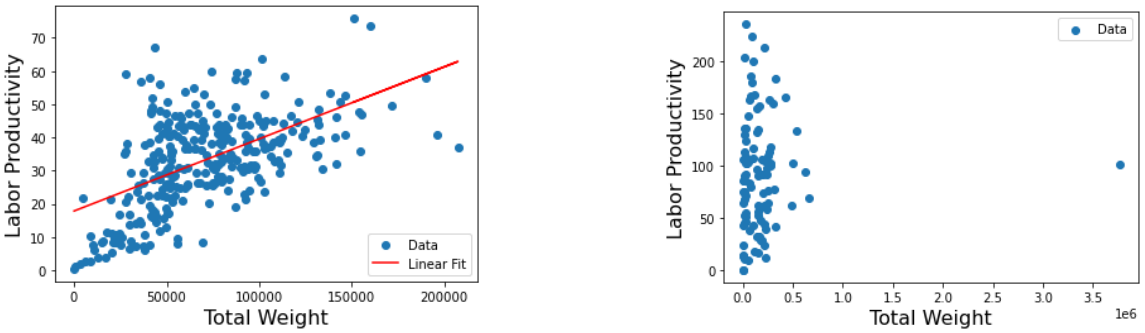
Figure 4.12: Average labor productivity by Master Carton adherence and quantity

For the inbound, the productivity is independent of the master carton adherence. No significant

differences were found between mean labor productivity for the scenarios where the quantity is high and the master carton adherence is either high or low ($p < .01$). Moreover, no significant difference was found between mean labor productivity for the scenarios where the quantity is low and the master carton adherence is either high or low. This indicates when the quantity is high, independent of the master carton adherence, labor productivity is high on average. Conversely, when the quantity is low, independent of whether the master carton adherence is high or low, labor productivity is lower on average. The master carton adherence is not a good indicator of productivity, which explains the very weak correlation found. For outbound B2S, no significant differences were found between mean labor productivity for the scenarios where the quantity is high, and the master carton adherence is either high or low ($p < .01$). Indicating that when the quantity is high, the master carton adherence has no influence. Moreover, when the quantities are lower on average, the average labor productivity is also lower, as expected. Still, it is also noticed that the average productivity is slightly lower when the master carton value is more adhered to. The master carton value has been shown to impact distribution logistics efficiency significantly. Defining the optimal master carton value for each SKU is a vital planning problem that affects warehouse operations Wensing et al. (2018). Adhering to master carton requirements (which might be non-optimal) can introduce complexity in the overall outbound process. Thereby negatively impacting labor productivity. The behavior will be analyzed further with the prediction model.

4.4.6. Total Weight

The relationships between the total daily weight and labor productivity are displayed in Figure 4.13a and Figure 4.13b, for B2S and BBXD, respectively. No significant relationship was found between the total weight and productivity for BBXD. A strong positive significant relationship is found between the total weight and productivity for outbound B2S. As the weight increases, the quantity increases, which increases the efficiency of handling and, therefore, increases productivity. However, a pattern of diminishing returns was expected as larger products are more unwieldy to handle, needing additional equipment or machinery. Moreover, larger items could lead to faster exhaustion of order pickers, negatively impacting productivity (Falkenberg and Spinler, 2022). The findings are not aligned with MediaMarkt's beliefs. Therefore, an additional analysis is performed.

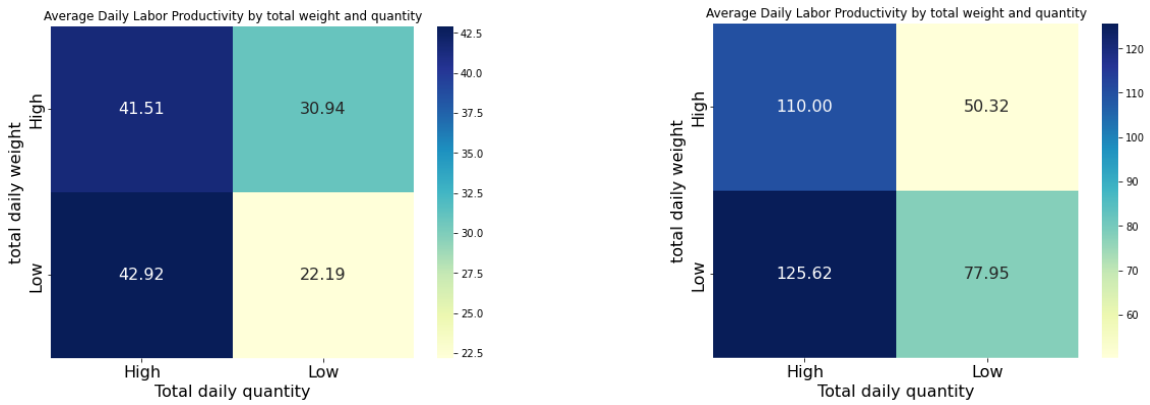


(a) Outbound B2S

(b) BBXD

Figure 4.13: Relationship between Labor Productivity and the total weight

Again, a 2x2 matrix is created to categorize the four different combinations in a similar fashion as for the master carton adherence; see Figure 4.14. For outbound B2S, no significant difference was found between the average labor productivity in the scenario where the quantity and total weight are high and the scenario where quantity is high and total weight is low. All other combinations were significantly different from each other. This indicates that when the total daily quantity is high, independent of the total weight being high or low, the average productivity is higher than when the total quantity is low. In the scenarios where the quantity is lower than on average, it seems that productivity is higher when the total weight is high than when the total weight is low. So if the total weight handled is high relative to the total quantity, i.e., predominantly larger items are handled on days where the quantity is low, productivity is higher. When the total weight is low relative to the total quantity, i.e., predominantly smaller items are handled on days where the quantity is lower than average, productivity is low. This could be caused by the inefficiency of picking a few small items. When quantities are low, and many small items have to be picked, orders cannot be adequately consolidated to overcome the inefficiencies of smaller quantities. This is most likely caused by lower average ordersize and quantity per orderline. This relationship will be further investigated with the prediction model.



(a) Outbound B2S

(b) BBXD

Figure 4.14: Average labor productivity by total weight and quantity

For BBXD, no significant differences were found between mean labor productivity for the scenarios where the quantity is high and the total weight is either high or low. Moreover, no significant difference was found between mean labor productivity for the scenarios where the quantity is low, and the total weight is either high or low. This indicates that labor productivity is higher on average when the quantity is high, independent of the total daily weight. Conversely, labor productivity is lower on average when the quantity is low, independent of whether the total daily weight is high or low. The total weight is not a good indicator of productivity, which explains why no correlation is found. The interaction effect of the total weight with other features on labor productivity will be further explored with the prediction model.

The analysis has yielded an overview of the various features associated with each goodsflow and the type, strength, and direction of the features' relationship with labor productivity. Moreover, some valuable insights on certain features were generated. The features are utilized in a predictive model described in the next chapter

Chapter 5

Solution Design

It has been established that multiple features influence labor productivity. However, the current prediction approach relies solely on past weeks' historical labor productivity average, excluding influential features. The aim is to construct a prediction model that predicts more accurately by incorporating influential features. Additionally, the model will provide valuable insights into the significance of the factors and how they interact. This section provides an overview of the models' development.

5.1. Choice of Prediction Model

The aim of the prediction model is to provide insights into the underlying behavior of labor productivity based on accurate prediction with multiple variables to improve workforce control. There is a trade-off between interpretability and flexibility to consider when choosing the model based on the research aim. Suppose the aim is to understand the association between the dependent and independent variable(s), also known as inference. Then, interpretability becomes more critical, so more restrictive parametric methods would suit best. When the research goal is to accurately predict based on multiple factors, more flexible and complex models are needed (James et al., 2013). Similarly, in this research, the trade-off holds. On the one hand, MediaMarkt is interested in understanding what characteristics influence labor productivity. Conversely, they want to be able to accurately as possible predict the expected productivity to determine the workforce. The method must be accurate, understandable, and easy to interpret.

Advanced decision tree models are popular non-parametric supervised learning methods applied in many real-world applications in the operational environment. They are seen as white-box models with high interpretability and low computational time compared to artificial neural networks (James et al., 2013). Ensemble tree methods are highly interpretable, require little data preprocessing, reduce overfitting, and simultaneously allow for categorical and numerical variables. Moreover, the methods can handle large amounts of data with many features and predict very accurately (James et al., 2013). Ensemble methods combine single trees into an ensemble with bagging or boosting. Random Forests is a bagging algorithm that builds multiple single trees simultaneously. The algorithm can improve prediction accuracy. However,

the combination of multiple decision trees does reduce interpretability. Conversely, Gradient Boosting algorithms sequentially build trees. Gradient Boosting Decision Tree (GBDT) method often outperforms Random Forest due to its sequential and self-learning nature. However, the algorithm is more computationally expensive. In general, GBDTs are highly customizable, simple to implement, robust against irrelevant features, scale-independent, interpretable, and accurate (Natekin and Knoll, 2013). Two well-known algorithms are Extreme Gradient Boosting (XGBoost) and LightGBM. Due to the strengths of GBDT models in accuracy, feature selection, and interpretability, these methods are used in the current research.

5.2. Gradient Boosting Decision Trees

Gradient Boosting Decision Trees (GBDTs) is a learning procedure that consecutively fits new models to provide more accurate output predictions. The new base-learner (single decision tree) is constructed to maximally correlate with the negative gradient of the loss function associated with the whole ensemble. The general form of the gradient boosting algorithm as originally proposed by Friedman (2001) can be found in Table 5.1.

Table 5.1: Retrieved from Natekin and Knoll (2013)

Gradient Boosting Algorithm by Friedman (2001)	
inputs:	- input data $(x, y)_{i=1}^{N}$
	- number of iterations M
	- choice of the loss function $\psi(y, f)$
	- choice of the base-learner model $h(x, \theta)$
Algorithm	1. initialize $\hat{f}_0$ with a constant
	2. for t = 1 to M do:
	3. compute the negative gradient $g_t(x)$
	4. fit a new base-learner function $h(x, \theta_t)$
	5. find the best gradient descent step-size ρ_t
	$\rho_t = argmin_\rho \sum_{i=1}^{N} \psi[y_i, \hat{f}_{t-1}(x_i) + (x_i, \theta_t)]$
	6. update the function estimate: $\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$
	7. end for: stop criteria

5.2.1. Extreme Gradient Boosting (XGBoost)

XGBoost is a highly scalable GBDT model widely used in machine learning implementations. The following theory is based on the paper by Al Daoud (2019). XGBoost introduces a balance between fitting the training data and controlling model complexity by adding a regularization term to the loss function. This helps prevent overfitting and improves the model's ability to generalize well to unseen data.

$$\psi(y, f(x)) = \sum_{i=1}^{N} \psi(y_i, f(x_i)) + \sum_{m=1}^{M} \Omega(\delta_m) \tag{5.1}$$

with

$$\Omega(\delta) = \alpha|\delta| + 0.5\beta||w||^2$$

Here α and β are hyperparameters that control the strength of regularization. The term δ represents the number of branches, w represents the value of each leaf, and Ω is the regularization function. Moreover, XGBoost has optimized the gain function to make the algorithm more efficient and effective. The gain is approximated using the division of the squared sum of gradients (loss function's partial derivative w.r.t. predicted value) by the sum of Hessians (i.e., loss function's second derivative w.r.t. predicted value) minus a regularization term α. The latter controls the complexity of the trees added to the ensemble.

5.2.2. LightGBM

LightGBM was created by a team from Microsoft to overcome previous limits of GBDTs in the efficiency and scalability of the implementation when features dimension is high, and data sizes are large (Ke et al., 2017). In comparison to XGBoost, LightGBM does not apply a level-wise tree growth but a leaf-wise growth. Thus, instead of checking all the previous leaves for each new leaf, the decision trees are grown leaf-wise Al Daoud (2019). There are two novel techniques used: *Gradient-Based One-Side Sampling* (GOSS) and *Exclusive Feature Building* (EFB). The algorithms as defined by Ke et al. (2017) can be found in Figure 5.1. The GBDT with the GOSS and EFB algorithm implementation is referred to as LightGBM algorithm (Ke et al., 2017).

Algorithm 2: Gradient-based One-Side Sampling

Input: I: training data, d: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: $loss$: loss function, L: weak learner
models ← {}, fact ← $\frac{1-a}{b}$
topN ← a × len(I) , randN ← b × len(I)
for $i = 1$ ***to*** d **do**
  preds ← models.predict(I)
  g ← $loss$(I, preds), w ← {1,1,...}
  sorted ← GetSortedIndices(abs(g))
  topSet ← sorted[1:topN]
  randSet ← RandomPick(sorted[topN:len(I)], randN)
  usedSet ← topSet + randSet
  w[randSet] × = fact ▷ Assign weight $fact$ to the small gradient data.
  newModel ← L(I[usedSet], − g[usedSet], w[usedSet])
  models.append(newModel)

(a) Gradient-Based One-Side Sampling

Algorithm 4: Merge Exclusive Features

Input: $numData$: number of data
Input: F: One bundle of exclusive features
binRanges ← {0}, totalBin ← 0
for f ***in*** F **do**
  totalBin += f.numBin
  binRanges.append(totalBin)
newBin ← new Bin(numData)
for $i = 1$ ***to*** $numData$ **do**
  newBin[i] ← 0
  for $j = 1$ ***to*** *len(F)* **do**
    if $F[j].bin[i] \neq 0$ **then**
      newBin[i] ← F[j].bin[i] + binRanges[j]
Output: $newBin$, $binRanges$

(b) Exclusive Feature Building algorithm

Figure 5.1: GOSS and EFB algorithm for LightGBM model by Ke et al. (2017)

The GOSS algorithm keeps the instances with large gradients (i.e., under-trained instances) and randomly drops the instances with small gradients. In doing so, the GOSS algorithm ensures that important instances that contribute significantly to the model's improvement are retained, leading to more accurate gain estimation for tree construction than uniform random sampling. Furthermore, EFB applies a greedy algorithm to efficiently group sparse features together, reducing feature dimensionality while maintaining the most critical information. Additional theoretical background on GBDTs and the XGBoost and LightGBM models can be found in appendix E

5.3. Data Preparation

Before implementing the models, the relevant data is gathered, adapted, and cleaned. Further details for data collection are described in Appendix C. An overview of the variables (i.e., features) used in the research is presented in Table 5.2.

Table 5.2: Available variables for prediction model

Impact factor	Variable Explanation	Variable name	variable type[shape]
Warehouse	Quantity per warehouse location	[Bulk_x, High value cage_x, Mezzanine_x, Racking_x, Smartbar_x]	Continuous [>0]
	Orderlines per warehouse location	[Bulk_y, High value cage_y, Mezzanine_y, Racking_y, Smartbar_y]	Continuous [>0]
	The number of the month	month	integer[1:12]
	The number of the week	week	integer[1:52]
Shift	Day of the week	[Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]	Categorical variable
	Goodsflow type	[inbound, inbound MDA, Outbound B2S,BBXD]	Categorical variable
Product	Quantity	Quantity	Continuous [>0]
	Number of orders	#orders	Continuous [>0]
	Number of orderlines	#orderlines	Continuous [>0]
	Average orderline quantity	averageOLqty	Continuous [>0]
	Average ordersize	ordersize	Continuous [>0]
	Daily weight processed	Weight	Continuous [>0]
	Quantity per product category	[General_x, Browngoods_x, CD/DVD_x, Computer_x, Console_x, Foto_x, Whitegoods_x]	Continuous [>0]
	Orderlines per product category	[General_y, Browngoods_y, CD/DVD_y, Computer_y, Console_y, Foto_y, Whitegoods_y]	Continuous [>0]
	Master Carton value adherence	MC_adherence	Continuous [>0]
	Number of full pallets	full	Continuous [>0]
	Number of mixed pallets	mixed	Continuous [>0]
	Mixed pallet ratio	mixed ratio	Continuous [>0]

The day of the week is a categorical feature and must be transformed into numerical form. As this variable has no ordinal relationships, one-hot encoding can be used. The days of the weeks are added to the data set, and a binary variable represents whether or not the row in the data set is a particular day. As mentioned before, the extreme outliers are removed from the dataset, using the rule of thumb that observation is considered an outlier if it is outside the interval: $[y - 3 \cdot \sigma(y), y + 3 \cdot \sigma(y)]$. Next, any rows with missing values are removed. Finally, the dataset is split into training, testing, and independent validation sets. First, a split of 80/20 is made for the training and validation set, and then an 80/20 split on the training set is made, which results in a training and test set.

5.4. Model Development

Figure 5.2 presents a schematic overview of the model. The first box represents the feature selection based on permutation-based feature importance. In general, permutation feature importance measures the change in the model error after permuting one feature's value, i.e., randomly shuffling the feature's value. The drop in the model score indicates the model's dependence on a particular feature. The general algorithm is explained in Appendix E (Breiman, 2001). An XGBoost and LightGBM regression models are built to select the important features. First, the hyperparameters are slightly optimized using Bayesian optimization with the *hyperopt* python package. The Bayesian optimization is run for ten iterations with a thousand trials. The best hyperparameter combination of each iteration is evaluated on the test set. The resulting best-fitted model with the lowest RMSE is then used for the permutation-based feature importance. This approach chooses features using an adequate model, avoiding excessive computational load or over-optimization. Permutation scores are computed across the entire dataset using this sound model, and the average importance is determined through numerous iterations. Features with scores exceeding 0.01 are then incorporated into the final model.

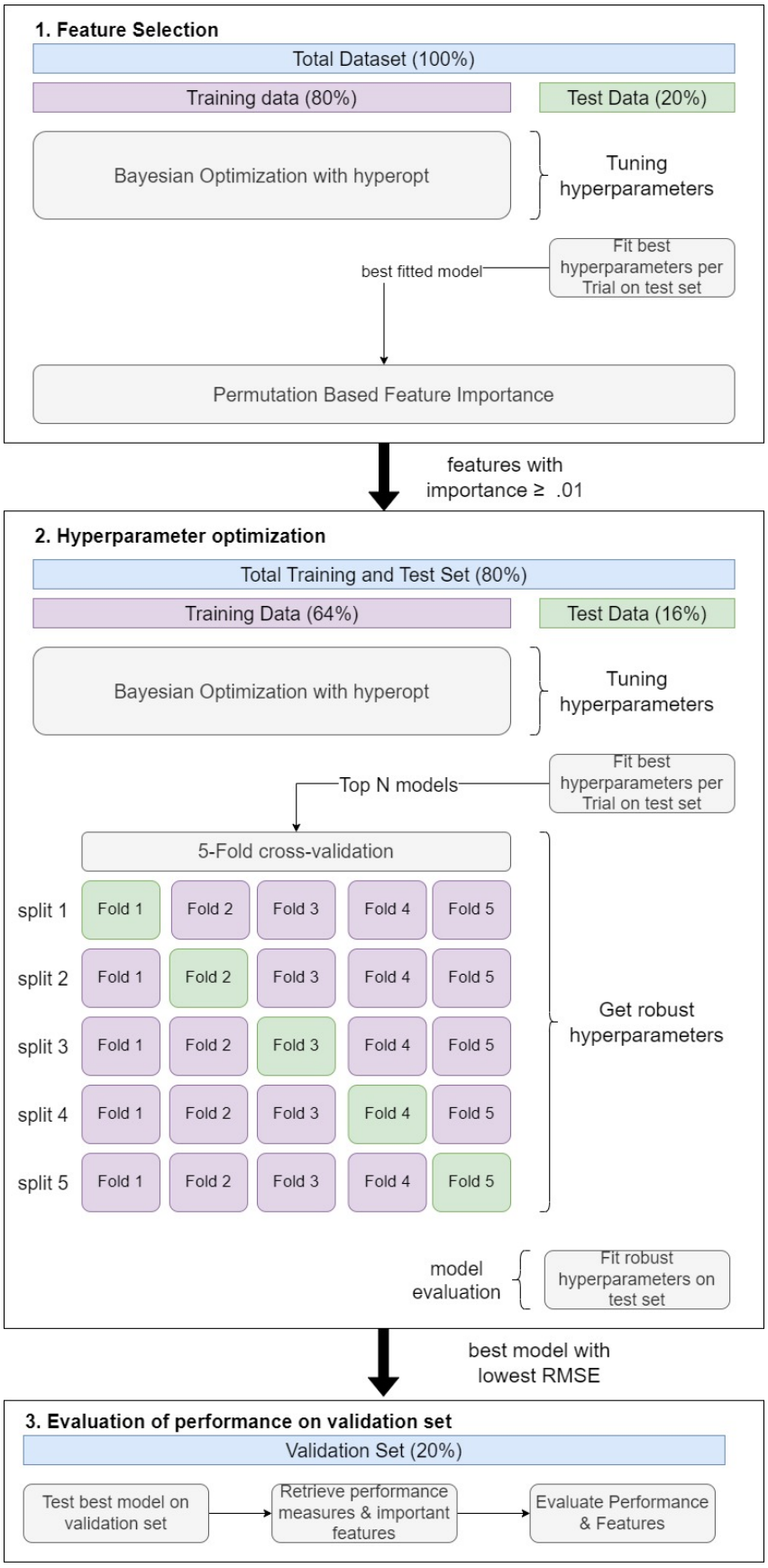


Figure 5.2: Model Development Schematic Overview

The hyperparameters are optimized after selecting the relevant features per model for each goodsflow. This process is represented in the second box in overview 5.2. Bayesian Optimization is used as it requires fewer iterations, is computationally less exhaustive, and yields better test set performance than other techniques. The optimization uses a surrogate model to approximate the objective function. Instead of exhaustively exploring the entire search space, the algorithms use a combination of exploration and exploitation strategies to select hyperparameters that are expected to perform well based on information provided by the surrogate model. By iteratively evaluating and updating the surrogate model, Bayesian optimization algorithms guide the search process toward promising regions of the hyperparameter space. This approach efficiently explores the space, gradually narrowing down to the optimal hyperparameters and minimizing the number of evaluations required on the actual objective function. Two well-known surrogate models are employed, Gaussian Processes and Tree-structured Parzen Estimator (TPE). TPE model has shown effectiveness in high-dimensional and discrete search spaces, as opposed to Gaussian-based methods (Bergstra et al., 2011, 2013). TPE applies Bayes' rule instead of directly representing the surrogate model. It models the distribution of hyperparameters conditioned on the scores achieved by evaluating those hyperparameters. The distribution is divided into lower and higher scores based on a threshold. The TPE method samples new hyperparameters from the lower score distribution. Thereby, TPE takes advantage of knowledge gained from previous evaluations to draw hyperparameter samples more likely to improve performance. By prioritizing exploration in regions associated with better scores, the search space is effectively explored to find (near) optimal solutions (Bergstra et al., 2011, 2013).

Since Bayesian optimization uses a combination of exploration and exploitation strategies, it may converge to different optimal points in different runs due to the random initialization of the search process, the number of evaluations performed, and the random search space. Therefore, cross-validation is used to generalize the results of the hyperparameter tuning process. The hyperparameters are tuned and evaluated on the train and test set using Bayesian optimization implemented with *hyperopt* python package Bergstra et al. (2013). Thereafter, the top N best-performing models with the lowest RMSE are generalized using k-fold cross-validation. This approach finds the most robust model with the highest performance. The method implemented is based on the *hgboost* package by Taskesen (2020), which is explained in Appendix F.

In the final step, represented in the third box in overview 5.2, the resulting best-performing model is fitted on the validation set to explore the performance and feature importance. Furthermore, the GBDTs' performance is compared to the baseline model. The current prediction predicts labor productivity by taking the average of the past four to six weeks and manually adapting it based on the knowledge and experience of the warehouse manager. The Moving Average (SMA) baseline model represents the current prediction method without subjective interference. The expected labor productivity is calculated weekly for each goodsflows, with an SMA of four and six weeks. As the prediction should be on a daily level (except for BBXD), the daily productivity is assumed to be equal to the productivity in that week. This is also what IDL assumes. The results of the different models are presented in the next chapter.

Chapter 6

Solution Implementation

In this section, the results of the feature selection are presented. Whereafter the performance scores of the different model implementations are presented.

6.1. Feature Selection

First, permutation-based feature importance is applied to select the most important features to include in the models. The average permutation score over a hundred permutation iterations across the entire dataset is computed using a sound XGBoost and LightGBM model with Bayesian Optimization. Features with scores exceeding 0.01 are then incorporated into the final model. Table 6.1 and 6.2 presents an overview of the remaining features for each model. For inbound, 42 features were initially included in the model; 20 and 16 remain for the XGBoost and LightGBM models, respectively. Both models identify the total quantity as the most influential feature. The number of orders and orderlines demonstrated a higher correlation with productivity. However, both models include the number of orders with lower importance. Moreover, only the LightGBM model includes the number of orderlines. A potential explanation could be the high correlation of these features with quantity, which might influence the permutation-based feature importance. Additionally, both models highlight the significance of browngoods, CD/DVD, and console quantities. The importance of the two latter features is surprising as these product categories are small, including many zero values. Both models attribute high importance to the high-value cage, mezzanine, and bulk location. The latter is unexpected, as no discernible pattern was found. The racking and mezzanine quantities are only included in the XGBoost model. The LightGBM model surprisingly includes smaller product categories, general and foto. The largest product category, computer, is only included in the XGBoost model, unexpectedly, as it strongly correlates with productivity. The week and month are deemed important features across both models. However, fluctuations are more pronounced for the week, making it a more informative indicator. The days of the week are excluded, indicating relatively stable productivity. The number of full and mixed pallets is included in both models. The exclusion of the mixed pallet ratio aligns with expectations, given the insignificant correlation. The features, average orderline quantity, and master carton adherence, deemed influential by MediaMarkt, were excluded from the LightGBM model.

Table 6.1: Selected features per model based on permutation-based feature importance for inbound and BBXD

Inbound				BBXD			
XGBOOST		LightGBM		XGBOOST		LightGBM	
feature name	importance	feature name	importance	feature name	importance	feature name	importance
1. Quantity	0.61	1. Quantity	0.68	1. SmartBar_x	0.13	1. COMPUTER_x	0.01
2. CD/DVD_x	0.11	2. week	0.13	2. Total Weight	0.08	2. averageOLqty	0.01
3. week	0.08	3. CD/DVD_x	0.07	3. averageOLqty	0.07		
4. full	0.05	4. month	0.04	4. High Value Cage_x	0.06		
5. High Value Cage_x	0.03	5. Bulk_x	0.03	5. Mezzanine_x	0.06		
6. CONSOLE_x	0.02	6. full	0.02	6. COMPUTER_x	0.04		
7. Bulk_x	0.02	7. High Value Cage_x	0.02	7. MC_adherence	0.04		
8. month	0.02	8. CONSOLE_x	0.02	8. #orderlines	0.04		
9. #orders	0.01	9. #orders	0.01	9. month	0.03		
10. BROWNGOODS_x	0.01	10. BROWNGOODS_x	0.01	10. Racking_x	0.03		
11. BROWNGOODS_y	0.01	11. GENERAL_x	0.01	11. #orders	0.02		
12. mixed	0.01	12. #orderlines	0.01	12. COMPUTER_y	0.01		
13. MC_adherence	0.01	13. High Value Cage_y	0.01	13. CONSOLE_x	0.01		
14. Racking_x	0.01	14. Mezzanine_y	0.01	14. week	0.01		
15. averageOLqty	0.01	15. mixed	0.01	15. High Value Cage_y	0.01		
16. High Value Cage_y	0.01	16. FOTO_x	0.01	16. CONSOLE_y	0.01		
17. COMPUTER_x	0.01			17. Mezzanine_y	0.01		
18.Mezzanine_y	0.01			18. BROWNGOODS_y	0.01		
19. Mezzanine_x	0.01						
20. Racking_y	0.01						

For BBXD, 38 features were initially included in the model; 18 and 2 remain for the XGBoost and LightGBM models, respectively. Interestingly, the total quantity is not the primary predictor of labor productivity. Instead, for LightGBM, computer quantities are most indicative, significantly influenced by the fact that the computer category constitutes more than 75% of the total quantities. The average orderline quantity is indicative despite its initial lack of a clear correlation with labor productivity. Further comparative analysis revealed that an increase in average orderline quantity corresponds to an increase in labor productivity up to a certain threshold. Beyond this point, a decline is observed. This observation aligns with MediaMarkt's expectations. The XGBoost model also acknowledges the importance of these two features while putting more emphasis on other predictors. These include the number of orderlines or the number of orders and the quantities and orderlines per warehouse location (except bulk), which could potentially store items from the computer product category. Furthermore, including total weight as a predictor in the model is surprising, given no correlation or pattern was found. A deeper analysis affirms that total weight remains relatively consistent across different levels of labor productivity. Both the week and month features are included in the XGBoost model, with the month potentially offering more insights due to the analysis being conducted at a weekly level. Lastly, the XGBoost model incorporates the master carton adherence feature, in line with expectations due to its observed positive correlation with productivity.

For outbound B2S, initially, 38 features were included in the model; 15 and 16 remain for the XGBoost and LightGBM models, respectively. There are many commonalities between the selected features. Both models identify the total daily quantity as the most influential feature. Moreover, the week number is deemed important in both models, while the month is excluded from the XGBoost model. The week's overall fluctuations are more pronounced, making it a more informative indicator. Outbound B2S productivity is significantly higher on Wednesday; therefore, an important indicator. High-value cage and bulk locations score high in both models. The bulk location is small; therefore, it is unexpected that this location is very influential. However, this could be due to an interaction with the total daily weight. When the

Table 6.2: Selected features per model per goodsflow based on permutation-based feature importance

Outbound B2S				Inbound MDA			
XGBOOST		LightGBM		XGBOOST		LightGBM	
feature name	importance	feature name	importance	feature name	importance	feature name	importance
1. Quantity	0.98	1. Quantity	0.76	1. #orderlines	0.31	1. averageOLqty	0.32
2. week	0.02	2. High Value Cage_x	0.02	2. averageOLqty	0.27	2. Quantity	0.20
3. High Value Cage_x	0.02	3. week	0.02	3. Quantity	0.20	3. Bulk	0.10
4. ordersize	0.01	4. Bulk_y	0.01	4. ordersize	0.20	4. #orderlines	0.07
5. Bulk_y	0.01	5. Wednesday	0.01	5. #orders	0.16	5. #orders	0.02
6. averageOLqty	0.01	6. COMPUTER_x	0.01	6. Bulk	0.13	6. week	0.02
7. Wednesday	0.01	7. Bulk_x	0.01	7. week	0.11	7. Mezzanine	0.01
8. COMPUTER_x	0.01	8. MC_adherence	0.01	8. Racking	0.10		
9. Mezzanine_x	0.01	9. Total Weight	0.01	9. month	0.08		
10. GENERAL_x	0.01	10. BROWNGOODS_x	0.01	10. Mezzanine	0.02		
11. WHITEGOODS_x	0.01	11. ordersize	0.01	11. Friday	0.01		
12. Total Weight	0.01	12. Racking_x	0.01	12. Monday	0.01		
13. #orders	0.01	13. GENERAL_x	0.01	13. Tuesday	0.01		
14. BROWNGOODS_x	0.01	14. averageOLqty	0.01				
15. Bulk_x	0.01	15. FOTO_y	0.01				
		16. month	0.01				

number of bulk orderlines is higher, the total daily weight is relatively higher, leading to more unwieldy handling and faster exhaustion of order pickers. The same reasoning applies to the whitegoods quantities, which the XGBoost model includes. The total weight is also included as an essential feature in both models. The high-value cage quantities displayed diminishing returns on productivity, as well as the computer quantities. The computer items primarily stored in the high-value cage require additional processing steps. This might diminish productivity. Both models also include the browngoods quantity, which strongly correlates with productivity. The XGBoost model includes the mezzanine location, while the LightGBM model includes the racking location; both large product categories strongly correlate with labor productivity. Surprisingly, both models also include the small general product category, and the LightGBM model includes the small product category foto, which both had no discernible pattern. These features could be included due to the inherent. The average ordersize and quantity per orderline are essential features for both models. Additional analysis showed that labor productivity increased when the average orderline quantity increased and diminished after a certain threshold. The ordersize could be important due to an interaction between the feature number of orders and total quantity. The number of orders is included in the XGBoost model but not in the LightGBM model. The quantities are extremely strongly correlated for outbound B2S. Thus, an increase in quantities leads to an increase in productivity. However, productivity growth stagnates when orders increase above a certain threshold. This could indicate an interaction effect via the average ordersize. Finally, the LightGBM includes the master carton value adherence. A negative correlation was found, possibly due to sub-optimal master carton values negatively impacting overall outbound efficiency.

For inbound MDA, initially, 17 features were included in the model; 13 and 7 remain for the XGBoost and LightGBM models, respectively. The seven features in the LightGBM model are also included in the XGBoost model, including the average ordersize, the racking quantity, the month, and the days Friday, Monday, and Tuesday. The days of the week do not seem to be significantly different regarding average labor productivity. Therefore, it is unexpected that these days contribute to the prediction productivity. The week number is

deemed important in both models, while the month feature is not included in the LightGBM model. Overall fluctuations are particularly pronounced for the week feature, potentially making it a more informative indicator. Remarkably, the average orderline quantity is highly important despite lacking a significant correlation with labor productivity. Conversely, average ordersize, which exhibited a slight positive correlation with MDA labor productivity, is omitted from the LightGBM model and has a slightly lower score in the XGBoost model. Quantity is not the primary driver of labor productivity at inbound MDA, which was anticipated as it only moderately correlated with labor productivity. In the XGBoost model, the number of orderlines is more indicative than the total daily and bulk quantities. In the LightGBM model, the number of orderlines slightly scores lower than the total daily and bulk quantities. The bulk location is the most relevant since it represents the primary location for MDA storage. The number of orders and mezzanine quantities also show predictive potential in both models. However, the other features take precedence.

6.2. Hyperparameters Optimization

After selecting the essential features to include per model, the next important step is hyperparameter tuning. The hyperparameters define the conditions and boundaries of the algorithm's learning process. The choice of hyperparameters may affect the performance of the boosting algorithm. Therefore, these parameters must be optimized by testing a range of preselected hyperparameters. An overview of the different parameters for XGBoost, their possible values, and tested values are displayed in Table 6.3. Appendix G provides a detailed explanation of the parameters and presents the final hyperparameter values generated by the optimization (Banerjee, 2020; dlmc XGBoost, 2022).

Table 6.3: Overview (hyper)parameters and tested values XGBoost model

	Parameter	Explanation	Possible values	Tested values
General Parameters	booster	general algorithm type	gblinear, gbtree, dart	gbtree
	verbosity	level of information provided during training	[0,3]	3
booster parameters	eta	learning rate	[0,1]	continuous [0.01, 0.3]
	gamma	minimum loss reduction required for a split	[0,inf]	continuous [0,10]
	max_depth	maximum depth of the tree	[0,inf]	integer [2,22]
	min_child_weight	minimum sum of all observations required in a child	[0,inf]	integer [1,10]
	subsample	proportion of randomly sampled observations	[0,1]	continuous [0.5,1]
	lambda	L2 regularization term	[0,inf]	continuous [0,10]
	tree_method	tree construction algorithm used in XGBoost	[exact, approx, hist, gpu_hist]	hist
learning parameters	objective	loss function which is minimized	any loss function	reg:squarederror
	eval_metric	metric used for data validation	default according to objective	RMSE
	seed	number to initialize pseudorandom number generator	default = 0	0

An overview of the different parameters and ranges for the LightGBM model is displayed in Table 6.4. Appendix H provides a detailed explanation of the parameters and presents the final hyperparameter values generated by the optimization. The general parameters used in the models are listed in Table 6.5.

Table 6.4: Overview (hyper)parameters and tested values XGBoost model

	parameter	Explanation	possible values	tested/set values
Core	objective	the output type of the model	regression, binary/multi-classification, ...	regression
Parameters	boosting	boosting algorithm used	gbdt, rf, dart	default = gbdt
	eval_metric	evaluation metric used to determine optimization	rmse, l2, mse, etc.	rmse
	n_estimator	the number of boosting iterations	continuous[0,inf]	integer[20,205,5]
	learning_rate	controls the learning rate of the model	continuous[0,1]	Continous[0,0.3]
	num_leaves	maximum number of leaves per tree	integer[0,131072]	integer[20,205,5]
	nthreads	the number of parallel threads used to run the model.	[0,max number of cores]	default = max cores
	device_type	device for tree-learning	cpu, gpu, cuda	default = cpu
	seed / random_state	number to initialize pseudorandom number generator	integer[0,inf]	default = 0
Learning	max_depth	maximum depth of the tree	integer[0,inf]	integer [1,30]
Parameters	min_data_in_leaf	minimal number of data in one leaf	integer[0,inf]	integer [1,30]
	min_child_weight	minimum sum of weights (Hessian) in one leaf	integer[0,inf]	integer [1,11]
	bagging fraction	i.e., subsample, the fraction of instances randomly sampled for each tree	continuous[0,1]	continuous[0.8,1]
	early_stopping_round	stops training when the metric of the last stopping rounds do not improve	integer[0,inf]	100
	reg_lambda	L2 regularization term	integer[0,inf]	continuous [0,2]
	verbosity	level of information provided during training	integer[0,3]	verbose = 3

Table 6.5: General parameter overview

parameter	explanation	value
fn	objective of the hyperopt	xgb_reg, lgb_reg
eval_metric	evaluation metric use for optimization	rmse
max_eval	maximum number of evaluations used in the hyperopt	10000
train_size	size of the dataset used for training the model	0.8
test_size	size of the dataset used for testing the model fitted by training	0.2
val_size	size of the independent dataset used for validation	0.2
kfold	the number of folds used in the cross-validation	k = 5
top_cv_evals	the top N models resulting from hyperopt used in cross-validation	100

6.3. Evaluation of Performance

6.3.1. Inbound

The different models' performance measure scores for regular inbound are displayed in Table 6.6. The baseline model results show that the moving average of 4 weeks slightly outperforms the 6-week rolling period. This is because productivity is very volatile over the weeks. A smaller prediction horizon leads to less smoothing of the volatile data over a certain period. A slight improvement is seen when comparing the baseline models to the XGBoost. The LightGBM model scores lower than the baseline models on all performance measures except Mean Absolute Error (MAE). However, it is twice as fast as the XGBoost (LightGBM: 3427 *seconds* vs. XGBoost model: 6953 *seconds*). For both models, the features explain more than half of the variability in the data. Overall, the XGBoost model outperforms all the other models.

Table 6.6: Performance measures score Inbound

Inbound	MA(4)	MA(6)	XGBOOST	LightGBM
MAE	33.09	33.52	27.10	31.21
MSE	2255.00	2313.10	1852.89	2641.00
R^2	N.A.	N.A.	0.62	0.51
RMSE	47.49	48.09	43.05	51.40

Figure 6.1 presents the permutation-based scores from the XGBoost model, which has the highest performance. The most important features are displayed at the top of the graph.

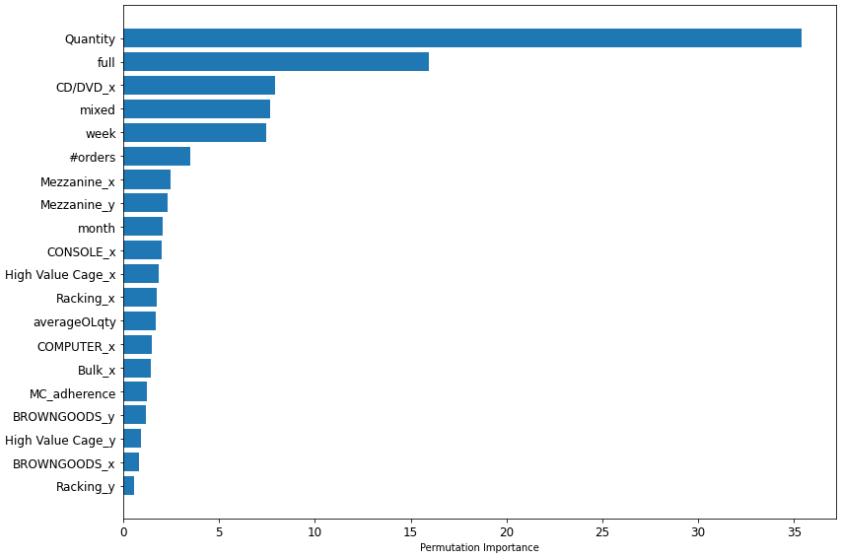


Figure 6.1: XGBoost permutation-based feature importance - Inbound

6.3.2. Inbound MDA

The different models' performance measure scores for inbound MDA are displayed in Table 6.7. For inbound MDA, it seems that the baseline moving average model with a 6-week rolling period performs slightly better than the 4-week rolling period on all performance scores. Although the differences are minimal, it points out that the behavior of productivity at Inbound MDA is more stable over the weeks. When comparing the baseline models' performance scores to those of the XGBoost and LightGBM models, it is noticed that the moving average models outperform the GBDT algorithms. The baseline models score lower on all performance measures. Moreover, the R^2 value indicates shallow scores, indicating that both models do not adequately explain the variation in the dependent variable with the current features used. The GBDT models do not adequately predict labor productivity behavior for Inbound MDA. Therefore, the features of these models are not further investigated, as this might not provide reliable results.

Table 6.7: Performance measures score inbound MDA

MDA	MA(4)	MA(6)	XGBOOST	LightGBM
MAE	5.81	5.73	8.40	6.72
MSE	89.50	88.38	179.41	93.31
R^2	N.A.	N.A.	-0.10	0.13
RMSE	9.46	9.40	13.39	9.66

6.3.3. Outbound B2S

The different models' performance measure scores for outbound B2S are displayed in Table 6.8. The baseline model results show that the moving average of 4 weeks outperforms the 6-week rolling period. This is because productivity is very volatile over the weeks. A smaller prediction horizon leads to less smoothing of the volatile data over a certain period. A considerable improvement is seen when comparing the baseline models to the XGBoost and LightGBM. The error terms all have shrunk considerably. The LightGBM model outperforms

the XGBoost model on all performance measures. Moreover, the LightGBM model is twice as fast (LightGBM: 3367 *seconds* vs. XGBoost model: 7545 *seconds*). The GBDT models have a high R^2 score, indicating that the proportion of the dependent variable variance is very predictable from the independent variables.

Table 6.8: Performance measures score outbound B2S

B2S	MA(4)	MA(6)	XGBOOST	LightGBM
MAE	16.63	16.89	4.33	3.80
MSE	439.44	441.98	52.41	36.33
R^2	N.A.	N.A.	0.88	0.90
RMSE	20.96	21.02	7.24	6.03

Figure 6.2 shows the permutation-based feature importance of the LightGBM model.

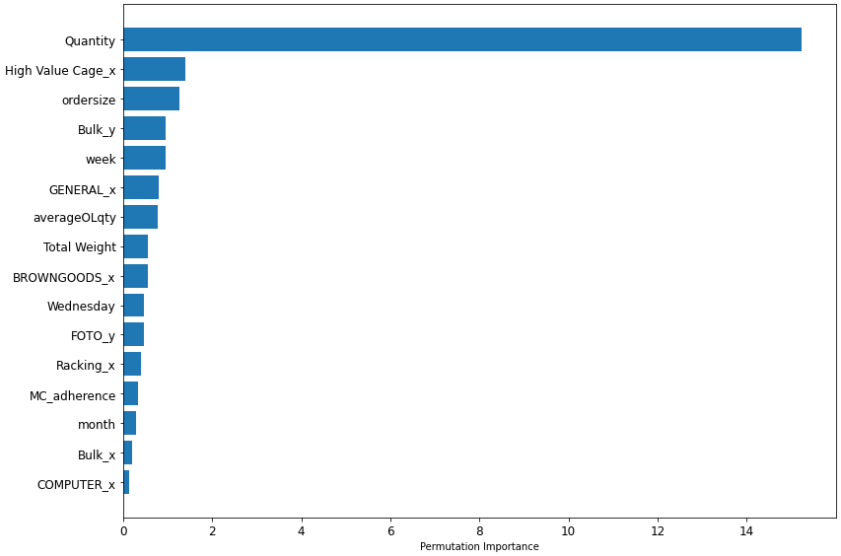


Figure 6.2: LightGBM permutation-based feature importance - Outbound B2S

6.3.4. BBXD

The different models' performance measure scores for BBXD are displayed in Table 6.9. The baseline model results show that the moving average of 4 weeks outperforms the 6-week rolling period on all performance measures. This is because productivity is very volatile over the weeks. A smaller prediction horizon leads to less smoothing of the volatile data over a certain period. When comparing the baseline models to the XGBoost and LightGBM, it is seen that the SMA(4) baseline model's RMSE is outperformed by the LightGBM model but not by the XGBoost model. Overall, the LightGBM model seems to outperform all the other models. Moreover, this model has a lower computational time (LightGBM: 2312 *seconds* vs. XGBOOST: 3450 *seconds*). However, the R^2 is relatively low. Less than 40% of the variance in the dependent variable is explained by the features in the model. From the feature selection, only two important features remained for the LightGBM model: the quantities in the computer category and the average orderline quantity; see Figure 6.3. The other features did not seem to hold much predictive power for BBXD. The current research seems to have missed features that would better predict BBXD labor productivity.

Table 6.9: Performance measures score BBXD

BBXD	MA(4)	MA(6)	XGBOOST	LightGBM
MAE	32.95	36.52	30.96	28.73
MSE	1682.47	2044.72	1978.56	1521.46
R^2	N.A.	N.A.	0.43	0.38
RMSE	41.02	45.22	44.48	39.01

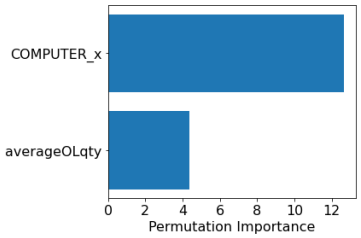


Figure 6.3: LightGBM permutation-based feature importance - BBXD

6.4. Data Limitations

The quality of the data is essential for the final evaluation. The data used in the research was extracted from multiple independent IT systems, leading to inconsistencies as each system works as a silo. For example, not all product categories and current stock placements were available. Orderlines without a product category were removed. Orderlines with no location were given a location based on the SKU characteristics. Thus, not all locations are assigned to the correct location. Furthermore, the master carton data was retrieved and compared from MediaMarkt and IDL information systems. For some SKUs, the values from both systems diverged. Moreover, sometimes, no master carton value was available, so the value was set to one. Therefore, it seems that the master carton value was often adhered to. This could cause the unexpected negative or no correlation found for outbound B2S and BBXD. Finally, suppliers manually enter data on the number of full and mixed pallets in the system, which is prone to human error.

There were also some inconsistencies in the hours data used to calculate the daily labor productivity. First of all, some hours had to be offset. The picking hours for outbound B2S are offset by one day, as it is assumed that picking is performed the day before the goods' packing and loading date, which is used. No offsetting was performed for the regular and MDA inbound hours, as SKUs are 98% of the time booked on the same day as arrival. For BBXD, quantities are assumed to arrive and leave the warehouse in the same week. Although the assumptions are reasonable, it could be that hours booked on a particular day (or week) belonged to quantities timestamped the day (or week) before or after. Moreover, sometimes hours were registered, but no deliveries were planned. These inconsistencies were amended by adding the hours to another day, depending on the day's characteristics. Although these amendments were done carefully, the manipulations can lead to hours not being matched to the correct quantities, thereby incorrectly calculating labor productivity. Although efforts have been made to correct the data inconsistencies with data cleaning and experts' knowledge, data inconsistencies are typical. When evaluating and interpreting the results, it is essential to keep these in mind.

Chapter 7

Evaluation

The models have been trained, tested, and validated. Moreover, the importance of features has been estimated using permutation-based feature importance. This section deeply dives into the features and their interaction effects to better understand how they influence labor productivity. Model-agnostic interpretation methods are used to explain the behavior of the applied machine-learning model. Two model-agnostic global interpretation techniques are used for the evaluation: the Partial Dependence Plots (PDP) and Accumulated Local Effects (ALE).

Partial Dependence Plots (PDP) shows the dependence between the target variable (labor productivity) and one (or two) feature(s) of interest, keeping the other input features constant. This way, the plot provides insight into the interaction between the target variable and the feature of interest. The PDP importance is easily interpretable as the plots show how the average prediction in the data changes when a feature changes. However, the PDP assumes that features are independent. If this assumption is not adhered to, the averages calculated may become more unreliable.

The Accumulated Local Effects (ALE) plot does not assume independence of features. The method also describes how features influence the model's prediction on average. The difference is that ALE calculates differences in predictions instead of the averages of predictions based on the conditional distribution of the features. In general, ALE function calculates the changes in predictions over a specific interval and then accumulates these changes over the whole range. The ALE uses the second-order effect to find the interaction between two features and the target output, which is the additional interaction effect after accounting for the main effects of the features. PDP, on the other hand, shows the total effect of two features. Further theoretical explanation of these methods can be found in Appendix I based on the work by Molnar (2022). The two methods discussed above are used to gain more insights into the behavior of the prediction model and the features' influence and interactions. The insights for the most critical features per goodsflow are discussed.

7.1. Inbound

First, the ALE plots of each feature are inspected, and the results are summarized in Table 7.1.

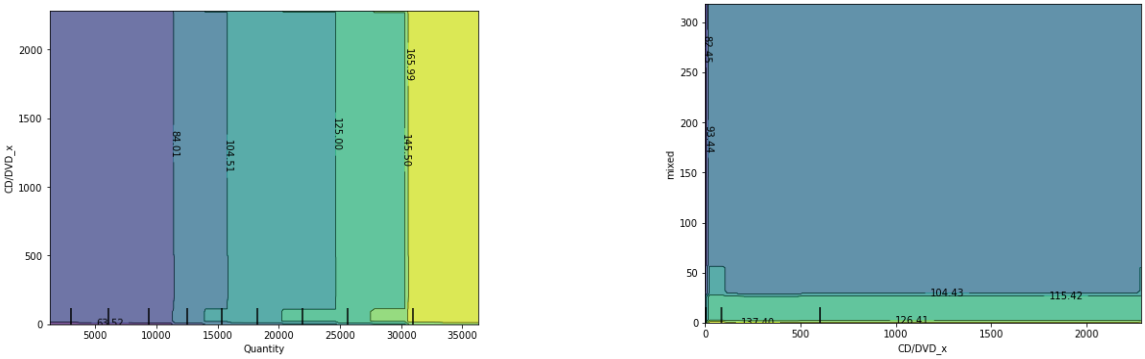
Table 7.1: Results Evaluation ALE plots - Inbound

Feature	**Bounds**	**Impact on prediction**
Quantity	$x < 125,000$	Decreasing productivity
	$125,000 < x < 425,000$	Increasing productivity
	$x > 425,000$	Constant positive productivity
Full pallets	$x < 100$	Decreasing productivity
	$x > 100$	Constant negative productivity
CD/DVD_x	$x > 0$	Increasing productivity
Mixed	$x < 50$	Decreasing productivity
	$x > 50$	Constant negative productivity
#orders	$x < 60$	Decreasing productivity
	$60 < x < 90$	Increasing productivity
	$x > 90$	Constant positive productivity
Mezzanine_x	$x < 2,000$	Decreasing productivity
	$2,000 < x < 3,000$	Increasing productivity
	$x > 3,000$	Constant positive productivity
Mezzanine_y	$x < 175$	Constant negative productivity
	$175 < x < 275$	Increasing productivity
	$x > 375$	Constant positive productivity
Console	$x < 3,250$	No impact
	$3,250 < x < 6,000$	Increasing productivity
	$x > 6,000$	Productivity stagnates
HVC quantities	$x < 3,500$	Decreasing productivity
	$3,500 < x < 20,000$	Increasing productivity
	$x > 20,000$	Decreasing productivity
Racking	$x < 7,500$	Decreasing productivity
	$x > 7,500$	Constant positive productivity
averageOLqty	$x < 50$	Decreasing productivity
	$50 < x < 200$	Increasing productivity
	$x > 200$	Constant positive productivity
Computer	$x < 4,750$	Decreasing productivity
	$4,750 < x < 15,000$	Increasing productivity
	$x > 15,000$	Constant positive productivity
Bulk	$x < 200$	Increasing productivity
	$200 < x < 1,200$	No impact
	$1,200 < x < 1,750$	Decreasing productivity
	$x > 1,750$	Constant negative productivity
MC adherence	$x < 0.93$	Constant positive productivity
	$0.93 < x < 0.94$	Increasing productivity
	$0.94 < x < 0.96$	Decreasing productivity
	$0.96 < x < 0.99$	Constant negative productivity
	$x > 0.99$	Increasing productivity

Overall, the findings reveal that the productivity prediction is negatively impacted when the total daily quantity falls below 125,000 items. This suggests that achieving economies of scale requires quantities beyond this threshold. A similar pattern is observed for the mezzanine, high-value cage, or racking quantities. Conversely, the positive impact plateaus beyond a certain threshold, indicating diminishing returns. Except for the high-value cage, productivity declines after a threshold is reached. Indicating inefficiencies occur when handling numerous small quantities through the high-value cage inbound process. The labor productivity prediction

becomes increasingly positive within the bounds, denoting opportunities for economies of scale. Similar effects are noted for the number of orders.

Remarkably, for the bulk location, a negative impact on productivity prediction is seen as the quantities increase. If the quantities exceed 1200 items, productivity growth declines, as these items are of substantial size and weight, necessitating additional equipment. Above the threshold, an over-utilization of the available capacity occurs, either in terms of trained personnel or additional equipment (i.e., forklifts) availability. The CD/DVD quantities exhibited a high permutation importance score despite being a small product category. The ALE plot demonstrates an overall positive effect on productivity prediction. This is believed to be coincidental rather than directly influencing the product group's SKU characteristics. The rationale is that labor productivity is high whenever the total daily quantities are high. The CD/DVD product category often has no inbound deliveries, but on days when productivity is high, there are CD/DVD inbound orders. This leads to a seemingly positive relationship. The interaction plots with other features show no influence from the CD/DVD quantities on labor productivity. Figure 7.1a demonstrates that an increase in quantity leads to increased labor productivity, independent of the CD/DVD quantities. Similarly, Figure 7.1b, demonstrates that when the number of full pallets is above 30, independent of the CD/DVD quantities, labor productivity is lower. Practical experience does not explain. The importance of this feature is most likely due to the randomness in the XGBoost model.

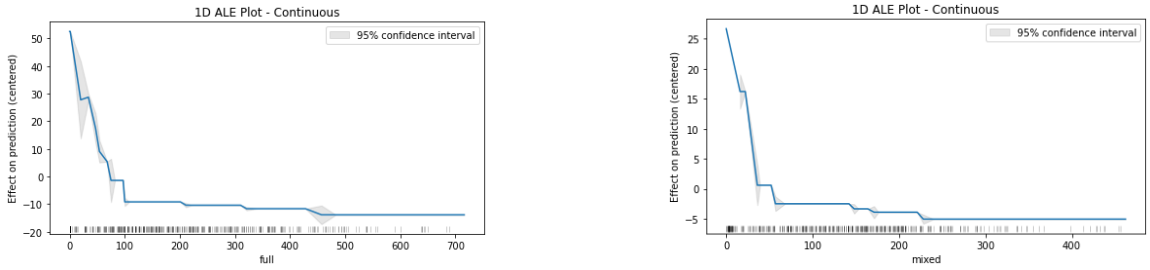


(a) PDP total quantity vs CD/DVD quantity

(b) PDP mixed pallets vs CD/DVD quantity

Figure 7.1: Interaction effect CD/DVD vs. quantities and mixed pallets on labor productivity

Another unexpected insight pertains to the number of full and mixed pallets; both exhibit an increasingly negative effect on productivity as the number increases; see Figure 7.2. However, when a certain threshold is reached, the negative impact on productivity stagnates. For mixed pallets, this was as expected. Mixed pallets complicate operations by increasing the need for sorting and higher put-away time. A positive relationship was expected for full pallets, as an increase in full pallets leads to an increase in economies of scale, and full pallets are more efficient to handle. When the number of pallets increases, the workload increases beyond the available capacity, either in terms of trained employees or available machinery, which negatively impacts productivity growth. Including features such as the availability of equipment and the skills and experience of employees could provide more insights into this behavior.



(a) ALE number of full pallets

(b) ALE number of mixed pallets

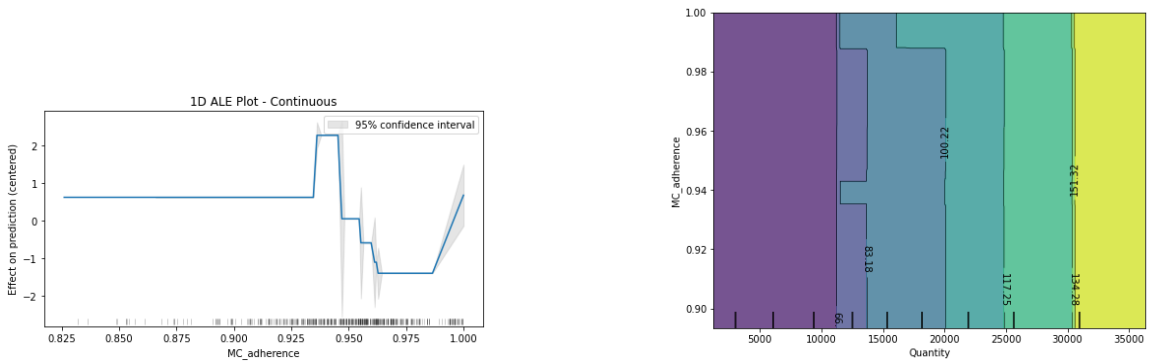
Figure 7.2: ALE plots number of full and mixed pallets - Inbound

The highest productivity is achieved when the average quantity per orderline exceeds 50 items. If the average orderline quantity falls below this threshold, it hurts productivity. Smaller quantities per orderlines lead to lower efficiency as more diverse orderlines are ordered in small quantities. This increases the overall inbound processing time, thereby reducing productivity. The optimal average quantity per orderline is attained when it exceeds 90 items per orderline. The top 10 largest suppliers are evaluated when the average orderline quantity exceeds 90 items per orderline. There is significant room for improvement for some suppliers, especially Samsung and Groupe SEB. When suppliers, especially larger ones, maintain higher average quantities per order line, it increases inbound productivity due to improved handling efficiency.

Table 7.2: Top 10 largest suppliers with adherence to average orderline quantity above 90

Supplier Name	NonAdherence	Total Quantity
Apple Distribution International Lt	24.19 %	1701440
TD SYNNEX Netherlands B.V.	15.10 %	1354645
Samsung Electronics Benelux B.V.	51.20 %	1185146
(Apple) TD SYNNEX Netherlands B.V.	19.05 %	615613
Pirox International BV.	11.90 %	501242
A&C Systems B.V.	22.01 %	455996
Groupe SEB Nederland BV	34.96 %	400002
Philips Domestic Appliances Nederla	18.47 %	369784
Nintendo Benelux B.V.	14.29 %	307056
Sony Europe B.V.	28.92 %	280276

For the master carton value adherence, initially, only a weak negative correlation was found, which was unexpected. However, further analysis did not result in further insights into the behavior of the master carton adherence. Looking at the effect on the prediction, it is seen that a small interval between 0.93 and 0.99 hurts productivity; outside this interval, the master carton adherence seems to be relatively constant. The explanation for the negative impact in this range could be due to the inefficiencies in handling when the case pack sizes are sub-optimal. On the other hand, the master carton data was often set to one, which may distort the true relation with labor productivity. The interaction effect with quantity in Figure 7.3b shows that higher quantities lead to higher labor productivity independent of the master carton adherence, which was previously also found in the analysis and diagnosis phase. This indicates that although some effect is visible, the overall effect of the adherence on labor productivity cannot be defined with the current data. Master carton adherence will not directly lead to improved productivity inbound, potentially due to mitigating effects of the large inbound quantities to be handled.

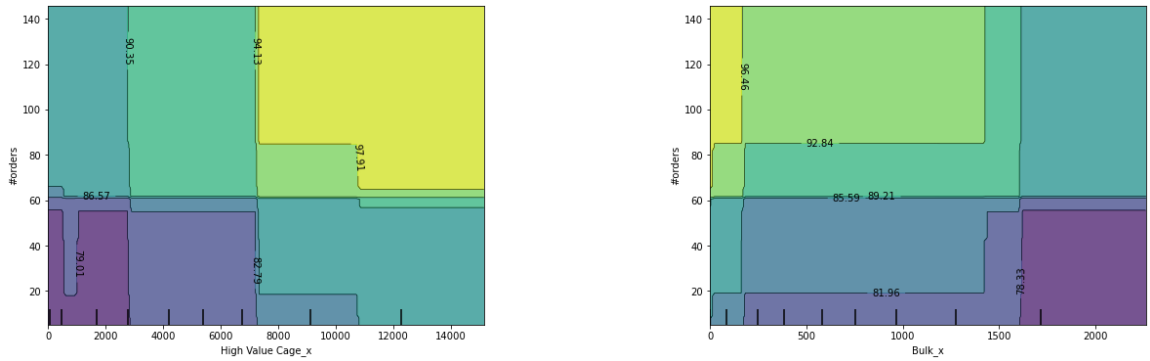


(a) ALE Master Carton Adherence

(b) PDP total quantity vs. MC adherence

Figure 7.3: Evaluation Master Carton Adherence

In summary, the PDP interaction plots between two features and labor productivity reveal that if both features are within their desired bounds (i.e., where productivity has an increasing or constantly positive effect on productivity), the highest labor productivity is attained. For example, in Figure 7.4a, it is observed that labor productivity is lowest when the number of orders is below 60, and the high-value quantities are below 3500. Conversely, when both these factors exceed these thresholds, productivity increases. A similar pattern can be seen in Figure 7.4b, where decreasing productivity is associated with higher bulk quantities. Optimal labor productivity is achieved when orders exceed 60, and the bulk quantities are below 200. Conversely, the lowest productivity is found when the bulk quantities are very high and the number of orders is below 60. These relationships are found for all feature combinations except the ones involving the CD/DVD quantities and MC adherence features, as mentioned above.



(a) PDP high-value quantity vs. orders

(b) PDP bulk quantity vs. number of orders

Figure 7.4: Interaction effect number of orders vs. high value and bulk quantities - Inbound

7.2. Outbound B2S

First, the ALE plots of each outbound B2S feature are inspected, and the results are summarized in Table 7.3. Overall, the findings reveal that the productivity prediction is negatively impacted when the total daily quantity falls below 10,000 items. This suggests that achieving economies of scale requires quantities exceeding this threshold. A similar pattern is observed for the high-value cage, racking, browngoods, and computer quantities. The positive impact plateaus

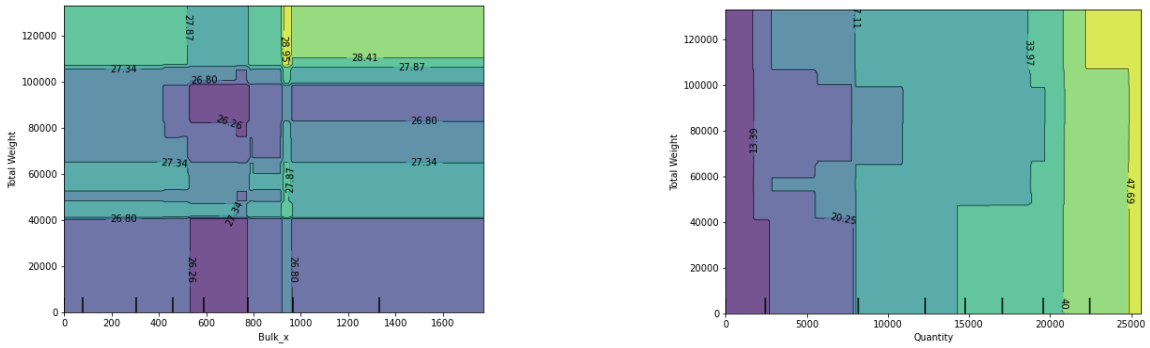
beyond a certain threshold, indicating diminishing returns when these quantities become too high to handle with current capacity. However, an opposite relationship is observed for bulk quantities: labor productivity decreases when the quantities exceed 400 items.

Table 7.3: Results Evaluation ALE plots - outbound B2S

Feature	**Bounds**	**Impact**
Quantity	x <10000	decreasing productivity
	10000 <x <25000	increasing productivity
	x >25000	constant positive productivity
High Value Cage_x	x <2000	decreasing productivity
	2000 <x <11000	increasing productivity
	x >11000	constant positive productivity
ordersize	x <100	decreasing productivity
	x >100	constant positive productivity
Bulk_x	x <400	constant positive productivity
	400 <x <1000	decreasing productivity
	x >1000	constant negative productivity
aveageOLqty	x <4	no impact
	4 <x <6	constant negative productivity
	6 <x <8	increasing productivity
	x >8	constant positive productivity
Total Weight	x <40000	constant negative productivity
	40000 <x <100000	decreasing productivity
	100000 <x <120000	increasing productivity
	x >120000	constant positive productivity
BROWNGOODS_x	x <2500	decreasing productivity
	2500 <x <6000	increasing productivity
	x >6000	constant positive productivity
Racking_x	x <2600	decreasing productivity
	2600 <x <10000	increasing productivity
	x >10000	constant positive productivity
MC adherence	x <0.2	decreasing productivity
	0.2 <x <0.4	increasing productivity
	x >0.4	decreasing positive productivity
Computer	x <2500	decreasing productivity
	2500 <x <7500	no impact
	7500 <x <125000	increasing productivity
	x >125000	constant increasing productivity
day of week	Wednesday	highest productivity

Potentially, the weight of bulk items could result in lower efficiency due to faster exhaustion or more unwieldy handling of items, causing lower productivity. However, the total weight feature only negatively impacts productivity whenever it is low. When the total weight exceeds 100,000 KG, productivity is positively impacted. As weight increases, so do the quantities, and higher quantities drive productivity. However, diminishing returns were expected above a certain threshold. When examining the interaction plot between bulk quantities and total weight in Figure 7.5a, it becomes evident that productivity peaks when both the bulk quantities and the total weight are at their highest levels. Unexpectedly, labor productivity declines when bulk quantities are low relative to the total weight. When both these features are low, labor productivity is even worse. When inspecting the interaction between the total daily quantity and total weight in PDP plot 7.5b, it becomes clear that an increase in quantity leads to increased productivity, regardless of the total weight handled in a day. This indicates that a high total

weight does not diminish productivity returns, as initially believed. Thus, rather than the total weight being the reason for bulk quantities' negative relationship with productivity, it could be due to a lack of trained employees or equipment needed to process the bulk quantities. It would be beneficial to incorporate additional factors into the study to better understand this relationship.



(a) PDP bulk quantities vs. total weight

(b) PDP quantity vs. total weight

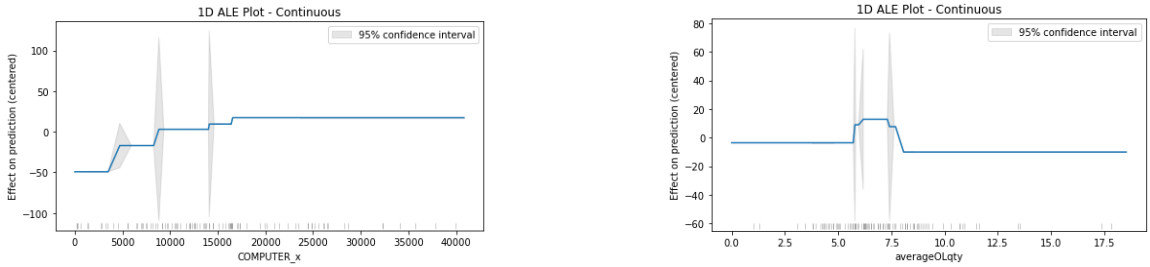
Figure 7.5: Interaction plots total weight vs. bulk and total daily quantities - outbound B2S

For the master carton adherence, again, a negative relationship is observed. Low adherence (below 0.2) appears to decline in productivity, but an adherence above 0.4 also seems to decrease overall productivity. It has previously been shown that the master carton adherence had a negative impact on outbound productivity. It was initially believed that adhering to a specific master carton value could lead to operational efficiencies, as it would eliminate the need to open cartons to count and retrieve the correct number of items. However, the efficiencies gained from this are minimal compared to ensuring that all orderlines adhere to a certain master carton value. Moreover, the master carton values could be non-optimal, introducing complexity in the overall outbound process and consequently negatively impacting labor productivity.

The average order size and quantity per order line are indicative of labor productivity for outbound B2S. Although the initial analysis did not show a particular trend, the prediction models estimate that productivity is higher when the average ordersize is above 100. Thus, higher overall ordersize leads to higher efficiency due to consolidated processing and handling, ultimately increasing productivity. The average quantity per orderline was shown to increase productivity as the value increased. Indeed, the prediction model confirms this finding. When the average orderline quantity is above 6 items, productivity increases. If more items are ordered per orderline, this leads to efficiencies in the handling, as SKU diversity is decreased on a daily level, traveling to a certain location becomes less frequent over the time period, and more items can be received at once. Combining the best bounds of these two features leads to the highest productivity. This is true for all feature combinations. The PDP interaction plots between two features and labor productivity reveal that if both features are within their desired bounds (i.e. where productivity has an increasing or constantly positive effect on productivity), the highest labor productivity is attained, similar to the inbound features interactions.

7.3. BBXD

For BBXD, only two features were included in the best-performing model: the quantities in the computer category and the average orderline quantity. The ALE plots are displayed in Figure 7.6. The ALE plots are quite flat, indicating little influence on the prediction. This explains the low coefficient of variation. It is seen that when the computer quantities are below 7500 items, the productivity prediction is negatively influenced; above this quantity, there seems to be little effect on productivity. The computer product category is the largest product category of BBXD. Thus, having low quantities indicates no economies of scale can be created. The average orderline quantity shows almost no effect on productivity, except for values between 6 and 7.5, for which there seems to be a slight positive impact on productivity.



(a) ALE Computer quantities - BBXD

(b) ALE average orderline quantity - BBXD

Figure 7.6: Comparative analysis: mean labor productivity vs. average orderline quantity

The PDP plot for the interaction effect between the computer quantities and the average orderline quantity is shown in Figure 7.7. When considering low computer quantities, the optimal range for the average orderline quantity lies between 6 and 7.5. Strikingly, excessive and insufficient items per orderline lead to decreased productivity, although the reduction is more pronounced with excessive quantities. In scenarios involving substantial computer quantities, an average orderline quantity of 6 or higher proves most favorable. Consequently, when dealing with significant quantities, enhancing efficiency involves placing orders with higher average orderline quantities, enabling more streamlined ventilation of the various items.

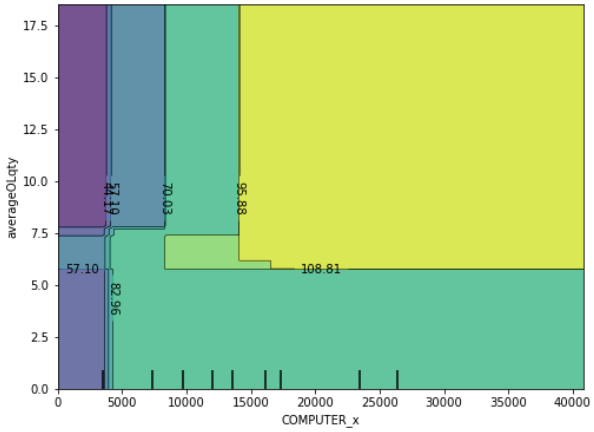


Figure 7.7: ALE Computer quantity vs. average orderline quantity - BBXD

Chapter 8

Discussion

Labor productivity at MediaMarkt seems to exhibit significant fluctuations from week to week, and it has become apparent from theory and practice that these fluctuations are caused due to numerous variables influencing labor productivity. The current prediction method was deemed deficient in accounting for these influencing variables. Therefore, an effort was made to employ advanced models to capture the underlying relationship between labor productivity and the influencing features and more accurately predict it. Gradient Boosting Decision Trees were identified as a suitable model choice due to their accurate, understandable, and interpretable characters. Additionally, no large datasets were retrieved; thus, the computational time remained accessible. Specifically, the XGBoost and LightGBM models were implemented because of their robustness, understandability, accuracy, and flexibility. The most influential features per model per goodsflow were determined and used for the final evaluation using permutation-based feature importance. Although most features' importance scores aligned with expectations, some surprising findings emerged. Particularly regarding the smaller product categories or warehouse locations being important features, as previously no pattern was found in the analysis, and practical insights could not explain the behavior. It's worth noting that, despite being white-box models, GBDT models can exhibit inherent randomness, occasionally producing less intuitive results. However, the permutation-based feature importance might also report lower importance scores for highly correlated variables, thereby influencing the final feature selection.

Based on their permutation score, features were eventually integrated into an optimized XGBoost and LightGBM model. The models were efficiently tuned for speed and accuracy through the implementation of Bayesian optimization, while their robustness was ensured by applying k-fold cross-validation. A comparison was made to a simple baseline model using a moving average to assess the models' performance. Note that the baseline model did not reflect reality, as it did not contain subjective interference based on the warehouse managers' knowledge and experiences. Notably, the LightGBM model outperformed all other models for outbound B2S and BBXD goodflows in terms of performance and speed. However, the XGBoost, although slower, demonstrated superior performance for regular inbound. For inbound MDA, both

GBDT models underperformed compared to the baseline model, and the coefficient of variation indicated that the current features only minimally explained the variability in the dependent variable. Therefore, additional features should be added, such as total weight, workforce size or equipment, and trained personnel availability, to better predict the inbound MDA productivity. This could result in a better understanding of MDA labor productivity. However, currently, some of these features are not recorded. Therefore, trying univariate methods such as time series analysis would be best, which might improve the productivity estimate.

For outbound B2S, the features seem to predict labor productivity well with a coefficient of variation of 0.9. The LightGBM model identified several important features: quantity, high-value, browngoods, racking quantities, average order size and orderline quantity, total weight, master carton adherence, and the week. The bounds for which productivity displayed certain behavior, i.e. increasing, decreasing, or constant, were organized in a Table. These bounds should be considered approximations, as they may change with further model optimization or additional data. However, it gives an adequate overview of when economies of scale are achieved with certain quantities and when diminishing returns are expected. Furthermore, most features displayed similar behavior as in the analysis and diagnosis phase. The total weight did not seem to display diminishing returns on productivity. As the quantity increased, so did productivity, independent of the total weight. Therefore, this feature does not seem to explain the behavior of productivity well. Furthermore, it was seen that if both the average quantity per orderline and ordersize increased, productivity increased. Thus, increasing the overall ordersize and average quantity per orderline, thereby decreasing the number of total orders, could lead to increased productivity at outbound B2S. The number of orders was shown to have diminishing returns. Thus, orders with higher ordersize could potentially lead to higher productivity and not stagnation.

For inbound, while there was a slight improvement in performance with the XGBoost model, it still fell short of capturing the full variability in the dependent variable. The current features do not adequately explain the full behavior of labor productivity, and improvements can be made by adding better features. Overall, the important features found in this research for regular inbound included quantity, full and mixed pallets, number of orders, CD/DVD, mezzanine, console, high-value, racking and bulk quantities, master carton adherence, and average orderline quantity. Unexpectedly, some negative influences on productivity were observed, like the negative impact of bulk quantities, full pallets, and the unpredictable behavior of CD/DVD quantities. The latter may be attributed to the inherent random behavior of GBDT models. Additional features such as the workforce size, employed personnel's skillset, and equipment availability could provide better insights. Additionally, volatility in master carton behavior on productivity was observed. Within a specific range, productivity appeared to decline despite high adherence. This negative impact could be attributed to handling inefficiencies that arise when case pack sizes are sub-optimal. However, noteworthy is that the master carton data contained many values equal to one, potentially distorting its true relationship with labor productivity. It could not be determined whether a master carton value was truly equal to one or because the true value was unknown. It's important to recognize that master carton adherence alone may not directly

translate to improved inbound productivity, possibly due to the mitigating influence of handling large inbound quantities. Similar to outbound B2S, the bounds for which productivity displayed certain behavior, i.e. increasing, decreasing, or constant, were organized in a Table. However, these bounds should be viewed as approximations and should not be generalized.

For BBXD, only two important factors were identified: computer quantities and the average quantity per orderline. The computer product category represents more than 75% of the total quantity and, therefore, understandably, a driver of BBXD productivity. A pattern of diminishing returns was found for extremely high computer quantities, possibly suggesting capacity constraints for the BBXD goodsflow. Therefore, increasing the average orderline quantity is important to reduce SKU diversity and improve handling efficiency at BBXD.

In summary, the study determined the quantities at which economies of scale can be achieved and for which warehouse location or product categories diminishing returns on productivity occurred when the quantities (or the number of orderlines) became too high. The individual effects of each product category are known. However, it was not discovered what the optimal SKU mix would be to achieve the highest productivity. For example, it would be interesting to understand how receiving many large TVs, fridges, and washing machines would impact inbound productivity on a day when available capacity is very low. Overall, the bounds must be interpreted with caution as the values are estimates and may change with model improvements. However, the increasing and decreasing patterns can be used to make relevant decisions. For example, deciding to increase the average quantity per orderline for all goodsflow leads to higher productivity. The GBDT models provided insights for the inbound and outbound B2S and BBXD goodsflow. However, considerable improvement can still be gained in terms of features included and overall performance. Furthermore, the analysis is currently performed for each goodsflow independently. However, bottlenecks in one goodsflow could lead to employees being re-balanced over the goodsflow, thereby influencing productivity. Information on employee shifting functions could provide a holistic view of warehouse productivity across different goods flows. A significant limitation of this study pertains to the utilization of hours used to determine productivity. These hours, by definition, do not account for the truly *"effective"* hours. Although productivity was computed with the direct hours allocated to various activities, it is important to note that these hours contain unnecessary hours. Situations may arise where employees cannot be sent home, resulting in low productivity due to an excessive workforce for the available tasks. Conversely, during periods of high workload, employees may demonstrate higher productivity compared to days with lighter workloads, where they might intentionally slow down work to fill their hours. Currently, the level of effort exerted by employees on a given day is not taken into consideration. Labor productivity is assessed on an aggregated daily level and does not provide insights into the productivity of individual activities.

Chapter 9

Conclusion

This research aimed to understand better what features influence labor productivity and how. A better understanding of labor productivity leads to increased workforce control. This avoids unnecessary re-balancing of labor resources, which leads to inefficiency and decreases either over- or under-capacity of labor. Over-capacity leads to unnecessary logistical expenses for MediaMarkt. Lastly, under-capacity leads to low service levels, low customer satisfaction, and (potential) loss of sales. Therefore, it is essential to control the workforce. A more thorough understanding and prediction of labor productivity can improve workforce control and minimize unpleasant consequences.

The current method of determining the expected labor productivity is done by averaging over the past four to six weeks and manually adapting it based on the knowledge and experience of the warehouse manager. No statistical or prediction methods are used to predict labor productivity. However, labor productivity varies extremely over time and is influenced by many qualitative and quantitative factors, as learned from theory and practice. A prediction based on the influencing factors would result in a workforce more aligned with the capacity needed, minimizing over- and under-capacity and decreasing costs while increasing service levels. The objective was to identify the factors that affect labor productivity and use this information to more accurately predict labor productivity, thereby improving workforce control over all goodsflow in the NDC: outbound B2S, outbound B2C, regular inbound and inbound MDA and BBXD. Therefore, the main research question was formulated as follows:

"What factors influence labor productivity in a warehouse setting, and how can these factors be leveraged to control the workforce, particularly in the context of different goodsflows?"

The main research question is answered through a set of sub-questions. The first sub-question *What are the key characteristics affecting the labor productivity of the different goodsflows (B2C, B2S, 2MH, inbound, inbound MDA, BBXD)?*, has been answered by analyzing literature on operation labor productivity and through stakeholder interviews. Literature indicates that labor productivity may be impacted by multiple factors from one of three sources: the performed activity, the employee performing the activity, and the environment

in which it is operated (Nasirzadeh et al., 2020). Falkenberg and Spinler (2022) were the first authors to categorize the different influential factors into four main categories impacting warehouse employee's productivity. The authors devised a framework with four main impact categories: the warehouse, the operator, the shift, and the product. Numerous impact factors belonging to these categories were found by translating labor productivity influential factors to the warehouse environment. Although the location of the warehouses was included in the research, the warehouse's conditions were not. This includes the design, size, and process maturity of the warehouse. These factors could be relevant indicators of warehouse employees' productivity. However, it is believed that these factors are less relevant in the current research as multiple warehouses were **not** compared. The variables would be constant over the considered timeframe. The factors could, however, be used to explore how automating the warehouse or implementing standard operating procedures could influence productivity in a scenario analysis. Unfortunately, data on employees could not be gathered due to privacy regulations at IDL. Therefore, features such as job role, salary, age, experience, training, days off, sick days, and past performance were not included. These features are highly relevant as they are often mentioned in labor productivity studies and are directly linked to the employee. For the third impact category, shift, the factors included were month, week, day of the week, and the goodsflow type. Factors for which no data was retrieved were work monotony, supervisors, extra payments, and workforce size. The workforce size would have been of great added value. Currently, the actual workforce size is not accounted for. Therefore, it is unknown if specific productivity is caused by under or over-capacity. Productivity could be low on days because too many employees have been planned, and they could not be sent home. Moreover, no data is available on how often employees switch functions per shift, which could lead to re-balancing inefficiencies between goodsflows. Moreover, no distinction is made between the day and night shifts. The different activities performed during these shifts differ. For example, the late shift handles mostly outbound orders, while the first shift typically handles inbound. The difference in quantities and available hours could influence overall daily productivity. Fortunately, data on all the factors listed under the impact category product were available. Moreover, several additional factors were added: the product category, the master carton adherence, the mixed pallet ratio, and the inbound issue score. The latter two were only applied to the inbound goodsflow. The framework provided a sound theoretical foundation to define and research the influential features of warehouse employees' labor productivity per goodsflow.

The second sub-question *How do these characteristics affect labor productivity, and how do they differ across different goodsflows?* was answered through a thorough analysis and diagnosis of the different features that could be gathered from MediaMarkt data. Unfortunately, the data for outbound B2C and 2MH was insufficient to perform a valuable analysis. An overview of the type, strength, and direction of the relationships between the features and labor productivity was presented for the regular inbound, inbound MDA, outbound B2S, and BBXD goodsflows. The unusual behavior of some of these features on labor productivity was further analyzed. However, no conclusion could be drawn for some features with this analysis. Therefore, permutation-based feature selection was applied to better understand which factors influenced labor productivity and how. The most important features were selected to predict labor productivity more

accurately. This also brings us to sub-question 3 *What are the most effective methods for measuring and predicting productivity in a warehouse setting, based on the influencing factors and underlying key patterns?*. The feature selection and productivity prediction was performed using two well-known Gradient Boosting Decision Trees: XGBoost and LightGBM. These models were applied because they are highly customizable, simple to implement, robust against irrelevant features, scale-independent, interpretable, and accurate Natekin and Knoll (2013). Furthermore, there results were made robust by applying Bayesian optimization and k-fold cross-validation. The GBDT models outperformed the baseline model for regular inbound, outbound B2S and BBXD, indicating its effectiveness in predicting. However, only for outbound B2S, the current features explained the behavior of productivity well.

Finally, the last sub-question *How can the established influential factors be leveraged to improve labor productivity and workforce control in a warehouse setting?* was answered through an evaluation of the most essential features from the best-performing model. Model agnostic techniques were applied to understand how the influential features behaved toward labor productivity. The evaluation determined the quantities at which economies of scale could be achieved and for which warehouse location or product categories diminishing returns on productivity occurred. The findings must be interpreted cautiously as the values are estimates and may change with model improvements or changes in the data. However, increasing and decreasing patterns can be leveraged to increase labor productivity. Mainly, it was seen that an increase in the average ordersize and average quantity per orderline for inbound, outbound B2S and BBXD could improve productivity.

9.1. Recommendations

Based on the findings in this study, several recommendations are formulated for the organization. First of all, the advanced GBDT models, XGBoost and LightGBM, have demonstrated the availability to more accurately predict and effectively capture complex relationships in the data. Therefore, it is recommended to continue using advanced models for labor productivity prediction. However, in order to further enhance model performance, a significant emphasis should be placed on feature engineering. Not only in terms of enhancing current available data's availability and quality but also by creating new features that may lead to new, improved predictions. For example, the current data on the inbound issues could be enhanced to include these inefficiencies. Moreover, collecting data on the daily workforce size is recommended to account for possible under- or over-capacity. One could engineer a feature that represents the level of effort exerted by employees on a given day. If data availability is limited, it is advised to consider univariate methods, such as time series analysis, to more accurately predict labor productivity, specifically for more stable flows such as inbound MDA.

Additionally, it is advised to increase the quantity per orderline and the ordersize. The increase of these two features will lead to less diverse orders, reducing travel time and lowering the need for sorting and ventilation, thereby increasing efficiency and labor productivity over the different goodsflows. Furthermore, it is recommended not to surpass the different quantities' thresholds to avoid diminishing returns. Further research could be conducted to understand

how the SKU mix could be used to maximize efficiencies. The current bounds can serve as guidelines. Furthermore, it is advised to analyze inter-dependence between goodsflows within the warehouse. One can explore how bottlenecks in one goodsflow effect impact another. Subsequently, one can implement strategies to balance the workforce effectively. Information on employee shifting functions could provide a holistic view of warehouse productivity across different goods flows. Finally, it should be recognized that although advanced models may provide additional and powerful insights, continuous improvements are needed to provide new insights into the additional data. Moreover, the results must be carefully interpreted due to the random behavior inherent to the GBDT models.

9.2. Contributions and Future research

The existing body of literature on labor productivity predominantly concentrates on sectors such as manufacturing, production, and construction, often emphasizing qualitative considerations. Falkenberg and Spinler (2022) were the first authors to address this research gap. The current research contributed significantly to academic research by introducing novel features to the productivity framework, as defined by Falkenberg and Spinler (2022). However, these dimensions primarily pertain to the retail warehousing context. Future research could explore quantitative and qualitative factors influencing warehouse employees' productivity across distribution centers. Furthermore, future research should systematically investigate and collect data on the factors in the aforementioned framework. This was neither undertaken by the framework's authors nor the current study. However, the current research contributed to the field by implementing the LightGBM model and the highly effective Extreme Gradient Boosting (XGBoost) tree algorithm. Thereby comparing the distinct methods in terms of expediency and performance. Moreover, the study reinforced the efficiency of GBDT models in predictive scenarios and underlined their utility as inferential tools. Although the current research compared the Gradient Boosting Decision Tree methods against a baseline model, it would have been sensible to compare the methods to univariate methods to underscore the success of machine learning methods for prediction even more. To summarize, the current research significantly bridged an existing gap in labor productivity research, particularly within the retail warehousing domain. Yet, substantial room exists for future research on essential features affecting labor productivity in the warehouse context.

References

Al Daoud, E. (2019). Comparison between xgboost, lightgbm and catboost using a home credit dataset. *International Journal of Computer and Information Engineering*, 13(1):6–10. 8, 28, 29, 96, 97

Banerjee, P. (2020). A guide on xgboost hyperparameters tuning. 37, 105

Basahal, A., Jelli, A. A., Alsabban, A. S., Basahel, S., and Bajaba, S. (2022). Factors influencing employee productivity–a saudi manager's perspective. *International Journal of Business and Management*, 17(1):39–51. 5

Bell, J. (2022). What is machine learning? *Machine Learning and the City: Applications in Architecture and Urban Design*, pages 207–216. 7

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24. 31, 99, 100

Bergstra, J., Yamins, D., Cox, D. D., et al. (2013). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. 13:20. 31, 99, 100

Breiman, L. (2001). Random forests. *Machine Learning*, 45. 30, 101

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. 8

De Leeuw, S. and Wiers, V. C. (2015). Warehouse manpower planning strategies in times of financial crisis: evidence from logistics service providers and retailers in the netherlands. *Production Planning & Control*, 26(4):328–337. 1

dlmc XGBoost (2022). Xgboost parameters - xgboost 1.7.6 documentation$_2$022. 37, 105

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27. 8

Falkenberg, S. F. and Spinler, S. (2022). Integrating operational and human factors to predict daily productivity of warehouse employees using extreme gradient boosting. *International Journal of Production Research*, pages 1–20. 6, 7, 8, 12, 13, 14, 25, 54, 56, 63, 64, 83

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Glock, C. H., Grosse, E. H., Elbert, R. M., and Franzke, T. (2017). Maverick picking: the impact of modifications in work schedules on manual order picking processes. *International Journal of Production Research*, 55(21):6344–6360.

Goel, V., Agrawal, R., and Sharma, V. (2017). Factors affecting labour productivity: an integrative synthesis and productivity modelling. *Global Business and Economics Review*, 19(3):299–322.

Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European journal of operational research*, 177(1):1–21.

Hamza, M., Shahid, S., Bin Hainin, M. R., and Nashwan, M. S. (2022). Construction labour productivity: review of factors identified. *International Journal of Construction Management*, 22(3):413–425.

Ishfaq, R., Defee, C. C., Gibson, B. J., and Raja, U. (2016). Realignment of the physical distribution process in omni-channel fulfillment. *International Journal of Physical Distribution & Logistics Management*.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Karim, N. H., Rahman, N. S. F. A., Hanafiah, R. M., Hamid, S. A., Ismail, A., Muda, M. S., et al. (2020). Revising the warehouse productivity measurement indicators: ratio-based benchmark. *Maritime Business Review*.

Kavya, S., Nikhil, T., and Akshayakumar, V. (2022). A review paper on prediction of construction productivity using artificial neural network model.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kembro, J. H. and Norrman, A. (2019). Warehouse configuration in omni-channel retailing: a multiple case study. *International Journal of Physical Distribution & Logistics Management*.

Kembro, J. H., Norrman, A., and Eriksson, E. (2018). Adapting warehouse operations and design to omni-channel logistics: A literature review and research agenda. *International Journal of Physical Distribution & Logistics Management*.

Liu, P. and Li, Z. (2011). Toward understanding the relationship between task complexity and task performance. In *Internationalization, Design and Global Development: 4th International Conference, IDGD 2011, Held as part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011. Proceedings 4*, pages 192–200. Springer.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364.

Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.

Naoum, S. G. (2016). Factors influencing labor productivity on construction sites: A state-of-the-art literature review and a survey. *International journal of productivity and performance management*.

Nasirzadeh, F., Kabir, H. D., Akbari, M., Khosravi, A., Nahavandi, S., and Carmichael, D. G. (2020). Ann-based prediction intervals to forecast labour productivity. *Engineering, Construction and Architectural Management*, 27(9):2335–2351.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

Rahman, N. S. F. A., Karim, N. H., Hanafiah, R. M., Hamid, S. A., and Mohammed, A. (2021). Decision analysis of warehouse productivity performance indicators to enhance logistics operational efficiency. *International Journal of Productivity and Performance Management*.

Sonmez, R. and Rowings, J. E. (1998). Construction labor productivity modeling with neural networks. *Journal of construction engineering and management*, 124(6):498–504.

Sreekumar, M., Chhabra, M., and Yadav, R. (2018). Productivity in manufacturing industries. *International Journal of Innovative Science and Research Technology*, 3(10):634–639.

Taskesen, E. (2020). hgboost is a python package for hyperparameter optimization for xgboost, catboost and lightboost for both classification and regression tasks.

Thomas, H. R. and Sakarcan, A. S. (1994). Forecasting labor productivity using factor model. *Journal of Construction Engineering and Management*, 120(1):228–239.

Van Aken, J., Berends, H., and Van der Bij, H. (2012). Problem solving in organizations (second). *Cambridge University Press*, 12:13.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European journal of operational research*, 226(3):367–385.

Van Gils, T., Ramaekers, K., Caris, A., and Cools, M. (2017). The use of time series forecasting in zone order picking systems to predict order pickers' workload. *International Journal of Production Research*, 55(21):6380–6393.

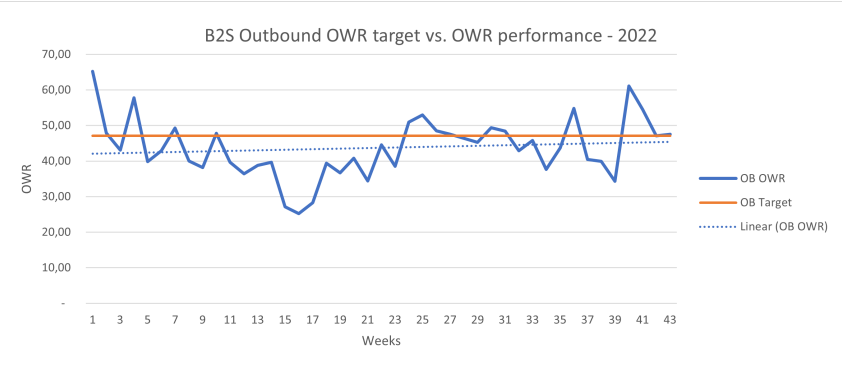
Vanheusden, S., Van Gils, T., Caris, A., Ramaekers, K., and Braekers, K. (2020). Operational workload balancing in manual order picking. *Computers & Industrial Engineering*, 141:106269.

Wensing, T., Sternbeck, M. G., and Kuhn, H. (2018). Optimizing case-pack sizes in the bricks-and-mortar retail trade. *OR Spectrum*, 40:913–944.

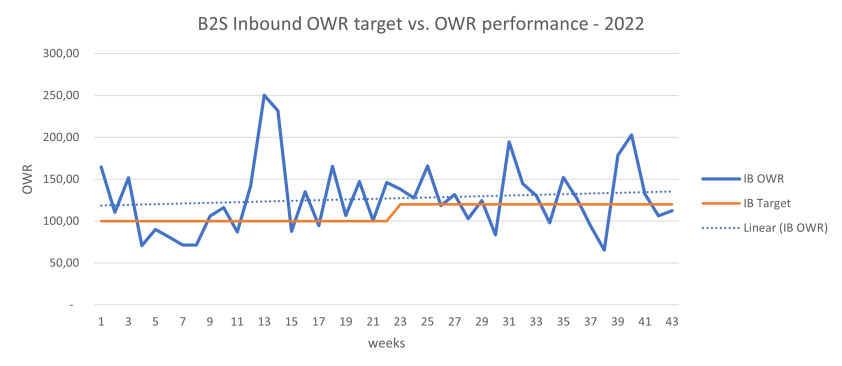
Appendices

Appendix A

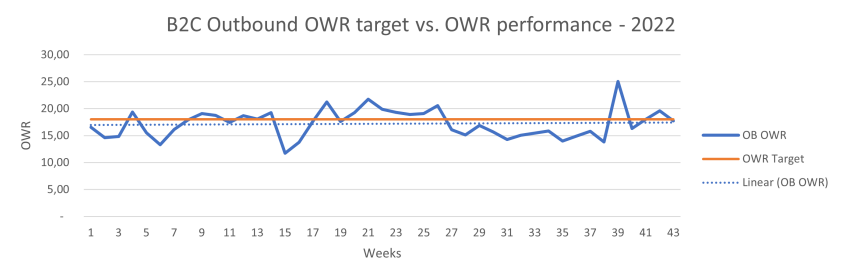
OWR performance vs OWR target per goodsflow



(a) B2S Outbound

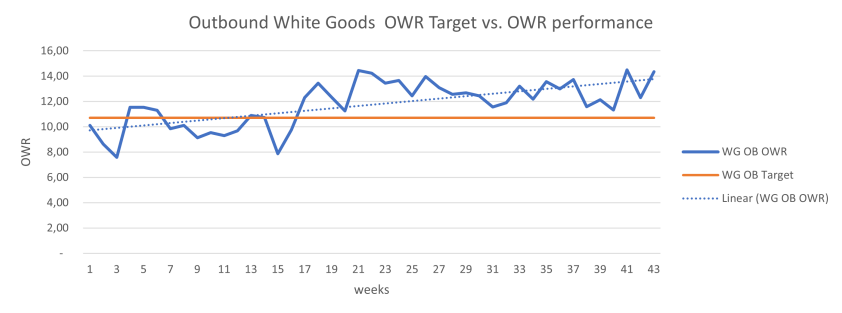


(b) B2S Inbound



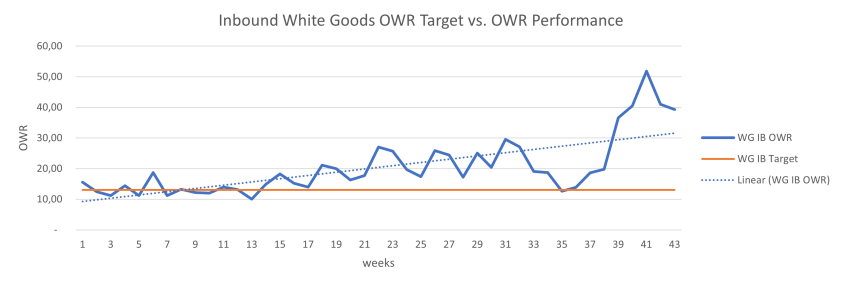
(c) B2C Outbound

Figure A.1: OWR performance versus OWR Target Outbound B2S and B2C, Inbound B2S



(a) Outbound White Goods

Figure A.2: OWR performance versus OWR Target White Goods flow



(a) Inbound White Goods

Appendix B

Theoretical Background Additional Information

"FLOPACE" model by Goel et al. (2017). *"FLOPACE"* is an acronym for focus, leadership style, organization structure, planning, adaptability, control and reward, and entrepreneurial culture. Factors influencing labor productivity found through literature research can be subdivided into seven categories according to the authors.

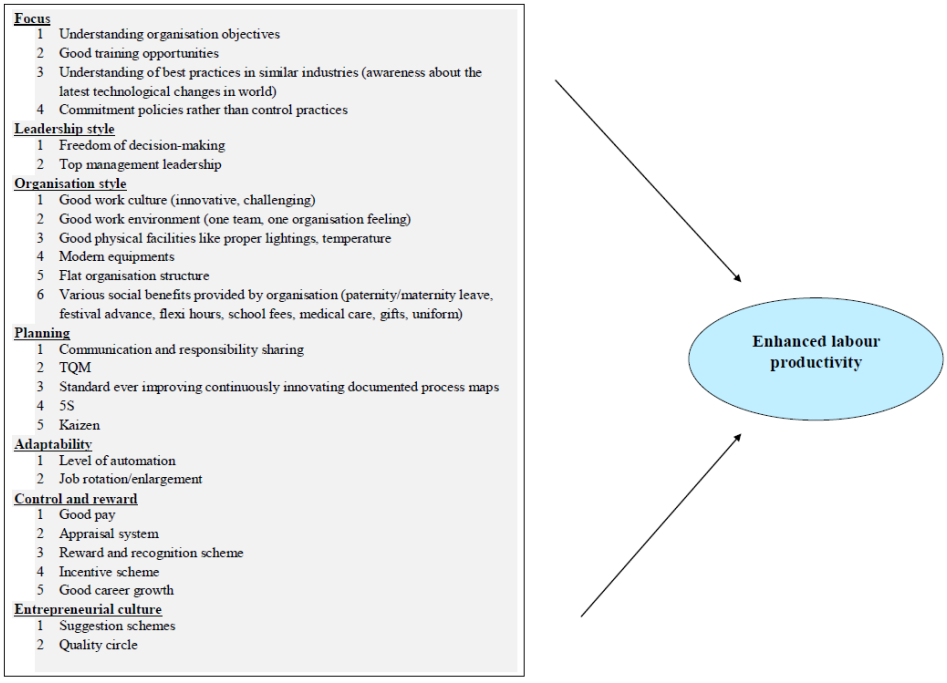


Figure A.1: FLOPACE model by Goel et al. (2017)

Falkenberg and Spinler (2022) translates the factors from the manufacturing, production, or construction environment into relevant features influencing warehouse employees' productivity. The created framework and relevant examples are displayed in Table A.1.

Table A.1: Framework of factors impacting on employees' warehouse productivity by Falkenberg and Spinler (2022)

impact category	impact factor	Examples
Warehouse	Location	level of humidity
	Design	accessibility of the aisles
	Size	short or long travel distances
	Process maturity	standard operating procedures
	Degree of automation	robotization
Operator	Job role	full vs. flex employee
	Salary	hourly wages or fixed contract
	Age	physical fitness
	Experience	accumulated work hours
	Training	onboarding training
	Days off	recreation time
	Sick days	overall health
	Past performance	historical performance indicators
Shift	Date	temperatures on a day, week vs. Weekend day
	Function	differing operations per hour per function
	Shift type	reduced attention during night shifts
	Work monotony	assembly line versus job shop
	Supervisors	ability to motivate
	Extra payments	increased wage for overtime work
	workforce size	congestion in aisles
Product	Quantity	economies of scale for large quantities
	Volume	large products are more unwieldy
	Weight	faster exhaustion when handling heavy products
	Special Handling Requirements	lower operationality of breakable or hazardous products

Appendix C

Data Collection and Manipulations

The context of this research is in the National Distribution Centre (NDC) of MediaMarkt Benelux. This NDC is served by a third-party logistics service provider, ID Logistics (IDL). Both parties have their own independent systems to control their operations. The data used in this research is extracted from four different types of IT systems: SAP, Relex, TESI, and WMS. The inbound and outbound deliveries and characteristics are obtained via the ERP system of MediaMarkt, SAP. The TESI system provides information on the scheduled delivery trucks, the content of their shipments, and possible delivery issues. Master Data information on the SKUs is retrieved from the Relex forecasting system. The Master data contains information on each product code linked to the Material number used in SAP, which represents the unique SKU number. This product code includes additional information on the product, such as product name, brand, supplier code and name, category number and name, Department number and name, MPG number and Name, and PG number and Name. Here, the Category is the highest product group aggregation, followed by Department, PG, and finally, MPG. Each variable describes in more detail the product, up to the type of USB cable. Lastly, The number of hours pertaining to each warehouse activity is extracted from IDL's Warehouse Management System (WMS). The WMS tracks the number of hours employees register on a certain operation. The data from these four sources is collected and combined into the final datasets used to analyze and model labor productivity.

C.1. Collection of deliveries per goodsflow and relevant features

First, the deliveries per goodsflow are collected, and the relevant features are added.

C.1.1. Regular Inbound and Inbound Major Domestic Appliances (MDA)

The inbound deliveries from 2020 until 2022 were retrieved from the SAP system. Each row in the EXCEL sheet represents an inbound orderline. Each row contained the following information: Delivery number, item, Material, Item description, Delivery Quantity, Sales unit (PC), Material availability Date, Actual Delivery Date, Ship-to-Party, Name of the ship-to party, Supplier Name, Purchasing Document, Goods Movement Sts, Storage Location, Storage Type, Reference Document, Delivery Type, Delivery date (From/to), Actual Goods Movement Date, Goods Issue Time. Many data columns are removed as they provide no relevant information or contain no information. The remaining columns are:

1. **Delivery:** a number that represents the order number of the inbound order
2. **Material number:** a number which represents the SKU ordered on that orderline
3. **Delivery Quantity:** which represents the number of items ordered of a particular SKU
4. **Supplier Name:** the name of the supplier who brought the items
5. **Actual Goods movement date:** the date on which the items were handled
6. **Goods Issue Time:** represents the time the goods were processed.

IDL works with different shifts. The first shift starts at 07.00h until 16.00h, and the second shift starts at 16.00h until midnight. Inbound deliveries booked between midnight and 06.00 AM belong to the previous day, and therefore, first, all dates with a Goods Issue Time stamp between 00.00h and 06.00h are set to the previous day, resulting in a new column *"Date"*. The columns *"Actual Goods Movement Date"* and *"Goods Issue Time"* are removed. After correctly amending the date, the Relex Master Data is coupled to the inbound dataframe based on the Material - Product code match.

The Major Domestic Appliances (MDA) Inbound is separated from the regular inbound as handling these white goods items requires certain competencies, which not all employees have. This is done by selecting department numbers 36 and 39, excluding the MPG numbers 141 and 149 in the total inbound sheet. After extracting the Inbound MDA orders and placing them in a separate file, the following actions have been performed on both datasets.

1. The rows where the actual goods movement date or the inbound quantity is zero are removed from the dataset, as these are inbound items that were not delivered, and no work was performed for these rows.
2. The location of the inbound orderline in the warehouse is coupled to each row in the dataset. This is done by linking the material number to the corresponding product code. Each product code has a stock placement in the warehouse: either bulk, racking, mezzanine, smartbar, or high-value cage. For some product codes, no current stock placement location was available, as this SKU might not have been in stock when the data was retrieved. Thus, based on the similarity of characteristics of products/SKUs, a location is derived from a product code with an unknown location. This was done by looking at the most common location for the articles with the same PG number. SKUs without a location were given the most common location for their PG number. However, some locations remain untraceable, and these data rows are excluded from the dataset.
3. The product category name is also coupled to each orderline based on the product code. The product categories are general, foto, CD/DVD, console, whitegoods, browngoods, and computer. Some product groups had a value unknown; these were removed from the dataset.
4. The Master Carton (MC) value was added to each orderline. Data on the MC value was available for each product code from IDL's WMS and MediaMarkt's ERP system. The following rules were applied. Firstly, for the product codes where the values from both systems diverged, the MC value by MediaMarkt was taken. Secondly, the MC value from

IDL was taken for the products with no MC value from MediaMarkt. If no MC value was available for a product code from either system, the value was set to 1. The MC values corresponding to the product code are added to each orderline in the dataset.

5. The inbound issues rate is calculated. First, the number of inbound issues per supplier is calculated and divided by the total inbound issues. Suppliers with inbound issues not mentioned in the entire inbound deliveries file are categorized under other. Vice versa, suppliers in the inbound deliveries, but without supplier issues, are classified under other. Names are researched online to ensure the supplier is not also listed under another name (e.g., Esselte B.V. is part of ACCO). Then, the total number of deliveries per supplier is calculated, multiplied by the issue rate per supplier. Thus, each supplier has a rate weighted based on the number of deliveries the supplier fulfills of the total number of deliveries. This is done because a carrier who supplies more has a higher probability of causing issues. Now, suppliers with more deliveries and the most inbound issues get a higher rate. The higher the rate, the worse the performance. Each orderline in the dataset belongs to a supplier, and the inbound issue rate is added by matching the supplier name.

For the Inbound MDA dataset, only the locations were added. The Master carton value for most MDA SKUs is equal to 1. Thus, this does not provide additional information. Furthermore, the product category is not added as all the items fall in the same category whitegoods, also not providing any additional information.

C.1.2. Outbound B2S and BBXD

For the outbound B2S and BBXD goodsflow, the deliveries from 2020 to 2022 are extracted from the SAP system. Each row in the EXCEL sheet represents an inbound orderline. Each row contained the following information: Delivery number, item, ship-to Party number, Material number, Delivery Quantity, Sales unit (PC), Material availability Date, total weight, weight unit (KG), Volume, Volume unit (CD3), Reference document, Delivery date (From/to), Delivery type, Transptn Plan. Date, Picking Date, Loading Date, Goods Movement Sts, Reference document, Name of the ship-to-party, Goods Issue Date. Some of these data columns are removed as they do not contain any (relevant or correct) data. The columns that remain are:

1. **Delivery:** a number that represents the order number of the order
2. **Material number:** a number which represents the SKU ordered on that orderline
3. **Delivery Quantity:**, which represents the number of items ordered of a particular SKU
4. **Goods Issue date:** when the order is packed and shipped. This date represents the point in time when the outbound deliveries leave the warehouse
5. **Delivery Type:** There are three types of delivery: ZADI, ZMER, and ZALL. ZADI are regular outbound orders to stores, and ZMER are orders for Business-to-Business. The orderlines with delivery type ZALL are separated as these represent the BBXD orderlines.
6. **Total Weight:** which represents the total weight of the SKUs ordered on one orderline
7. **Volume:**, which represents the total volume of the SKUs ordered on one orderline

Similarly, as for the inbound regular goodsflow, the location of the orderline in the warehouse, the product category of the SKU, and the master carton value are added to each orderline.

C.1.3. Outbound B2C and Two-Man-Handling (2MH)

For the outbound B2C and 2MH deliveries, insufficient data could be directly extracted from the SAP system. Therefore, information from the WMS from IDL was used on the actual deliveries of these two goodsflows. However, this means that only the quantity is known for these two streams. For outbound B2C, this quantity can further be split into parcels, One-Man-Handling and NDPU. However, no other features than quantity can be extracted from the system for these two goodsflows.

C.2. Aggregation to daily delivery features

After collecting the inbound, outbound, and BBXD deliveries and adding the needed features, the data had to be aggregated to the daily level. The different features created are explained below. An overview of the features per goodsflow can be found in Table A.1. Note that for BBXD, the features are calculated every week because the cross-dock flow activities are divided over the week, and only the loading date is available for the deliveries.

topsep = 0pt, itemsep=0mm**Quantity:** The quantities per orderline is summed daily. **Number of orders:** The daily number of orders is determined by counting the unique delivery numbers per day. **Number of orderlines:** The daily number of orderlines is determined by counting the number of rows in the dataset per day. **Average quantity per orderline:** The average orderline quantity per day is calculated by dividing the daily quantity by the daily number of orderlines. **Average ordersize:** The average ordersize is calculated by dividing the daily quantity by the daily number of orders. **Locations:** Both the quantities per location and the number of orderlines per location are determined. The locations are bulk, mezzanine, racking, smartbar, and high-value cage. The quantity per location is the sum of the quantity per orderline belonging to a certain location. The number of orderlines is the number of rows in the dataset belonging to the same location. **Product category**: Both the quantities and the number of orderlines per product category are determined. The product categories are general, foto, CD/DVD, console, browngoods, whitegoods, and computer. The quantity per product category is the sum of the quantity per orderline belonging to a specific product category. The number of orderlines is the number of rows in the dataset belonging to the same product category. **Master Carton Value adherence**: The Master carton value adherence is a ratio that indicates the number of orderlines adhering to the MC value on the total number of orderlines. It is calculated on a daily level by noting if the orderline quantity is a multiple of the MC value of that SKU. If this is true, the orderline gets a score of 1, else zero. The average of these scores is calculated on a daily level. Thus, a high ratio indicates that many orderlines adhere to the MC value, while a low ratio suggests that many orderlines did not adhere to the MC value. **Inbound issue score**: Each supplier is given an inbound issue rate, which indicates a supplier's relative number of issues, as

explained above. The daily inbound issue score is calculated by summing the inbound issue rate per unique supplier delivering daily. Thus, the inbound issue rate of a supplier occurring on a day is only counted once. This is done because it does not matter how often a supplier occurs on a day (how many orderlines are delivered per supplier), as the weight of the supplier is already considered in the inbound issue ratio. A supplier is assumed only to have once a day, thus only being able to provide issues once a day. **Mixed pallet ratio:** The mixed pallet ratio is calculated by dividing the number of mixed pallets by the total number of pallets received that day (the number of full and mixed pallets summed). The days where the mix is zero are either because no pallets were delivered or because only full pallets were delivered. **Total Weight and Volume**: Each SKU has a specific volume and weight. Each orderline containing several unique SKUs has the sum of the total weight and volume of these SKUs. The total weight and volume of a day is the aggregation of the weight and volume of each orderline with the same date.

Table A.1: Overview of the features collected per goodsflow

	Quantity	orders	orderlines	Location	Product	MC adherence	Inbound issues	Mixed pallet	Weight & Volume
Inbound	X	X	X	X	X	X	X	X	
Inbound MDA	X	X	X	X			X		
OB B2S	X	X	X	X	X	X			X
OB B2C	X								
OB 2MH	X								
BBXD	X	X	X	X	X	X			X

Certain features are not available for all goodsflows. As mentioned, only the quantity feature is available for the outbound B2C and 2MH. Regular and MDA no weight and volume is registered for the inbound deliveries. Moreover, the MC adherence and pallet mix are not added to the inbound MDA. Because the Master Carton value of MDA products is mostly one (fridges, washing, and drying machines are delivered per piece), and the products seldom arrive on mixed pallets. Furthermore, the product category is always whitegoods, which does not provide any other relevant data. Therefore, the product category is not a feature for Inbound MDA. The inbound issue score and the mixed pallet ratio are inbound features; therefore, these features are not available for outbound B2S and BBXD goodsflow.

C.2.1. Number of hours per activity per goodsflow

Besides collecting and aggregating the quantities and features per goodsflow daily, the daily number of hours per goodsflow is added to the datasets. The daily hours per activity per goodsflow are retrieved from IDL's WMS system. Employees register their time per activity. MediaMarkt and IDL have a contract in which MediaMarkt pays a fee for all hours booked by IDL on the different activities. Several activities are being tracked; see Table A.2. An overview of the hours registered per other activity for each goodsflow is found in Table A.3.

The hours for Business Support and Other are indirect, irrelevant, and, thus, not considered when calculating the OWR. Therefore, these hours are not used in the analysis. An overview of the hours registered per different activity for each goodsflow is found in Table A.3.

Inbound Regular

Table A.2: Daily hours warehouse activities variables

Inbound	Outbound	White Goods	Other
B2C IB - Non-productive	B2C OB - Non-productive	WG - Non-productive	B2S OB - Cross-dock
B2C IB - Unloading	B2C OB - Picking	WG - Receiving	
B2C IB - Receiving	B2C OB - Replenishment	WG - Picking	Business support - Inventory management
B2C IB - Putaway	B2C OB - Packing	WG - Unloading	Business support - Non-productive
B2S IB - Non-productive	B2C OB - Loading	WG - Putaway	Other - Projects
B2S IB - Unloading	B2C OB - Transfers	WG - Packing	Other - Non-productive
B2S IB - Receiving	B2C OB - Smartbar	WG - Loading	Other - Transfers
B2S IB - Putaway	B2S OB - Non-productive	WG - Replenishment	Other - Quality
	B2S OB - Replenishment		
	B2S OB - Picking		
	B2S OB - Packing		
	B2S OB - Loading		
	B2S OB - Quality		

Table A.3: Overview activities per goodsflow

Inbound Regular	Inbound MDA	Outbound B2S	Outbound B2C	Outbound 2MH	BBXD
indirect	receiving	indirect	indirect	indirect	Cross-dock hours
unloading	unloading	replenishment	replenishment	replenishment	
receiving	putaway	loading	loading	packing/loading	
putaway		picking	picking	picking	
			packing		

Before the B2S and B2C inventory consolidation, the inbound hours for each task at B2S and B2C were registered separately. As the total inbound for both B2C and B2S is taken together, the hours for similar tasks are summed together. The non-productive hours are labeled as indirect hours. However, these hours are counted for the OWR score, as these are necessary hours to complete the processes. Furthermore, minimal values were set equal to zero (below $1.0e^{-04}$).

Inbound MDA and Outbound 2MH

The hours for inbound MDA are listed under white goods. Only the hours for the operations at inbound are summed. Thus, receiving, unloading, and putting away activities are considered. The remaining hours belong to the outbound 2MH goodsflow. The packing and loading hours are summed together, as registration for these two operations was consolidated after the 18^{th} of December 2021. Moreover, the non-productive, smartbar, and transfer hours are summed and labeled as indirect. Furthermore, minimal values were set equal to zero (below $1.0e^{-04}$).

Outbound B2S and BBXD

Similarly, as for outbound 2MH, the packing and loading hours are summed together because registration for these two operations was consolidated after the 18^{th} of December 2021. Furthermore, the Quality and non-productive hours are summed together and labeled indirect. The hours for the BBXD goodsflow are all registered under one operation called cross-dock. The deliveries and hours for BBXD are added to a weekly level as the activities occur over the week, but the deliveries are only dated every week. Furthermore, minimal values were set equal to zero (below $1.0e^{-04}$).

C.2.2. Consolidation of deliveries and hours per goodsflow

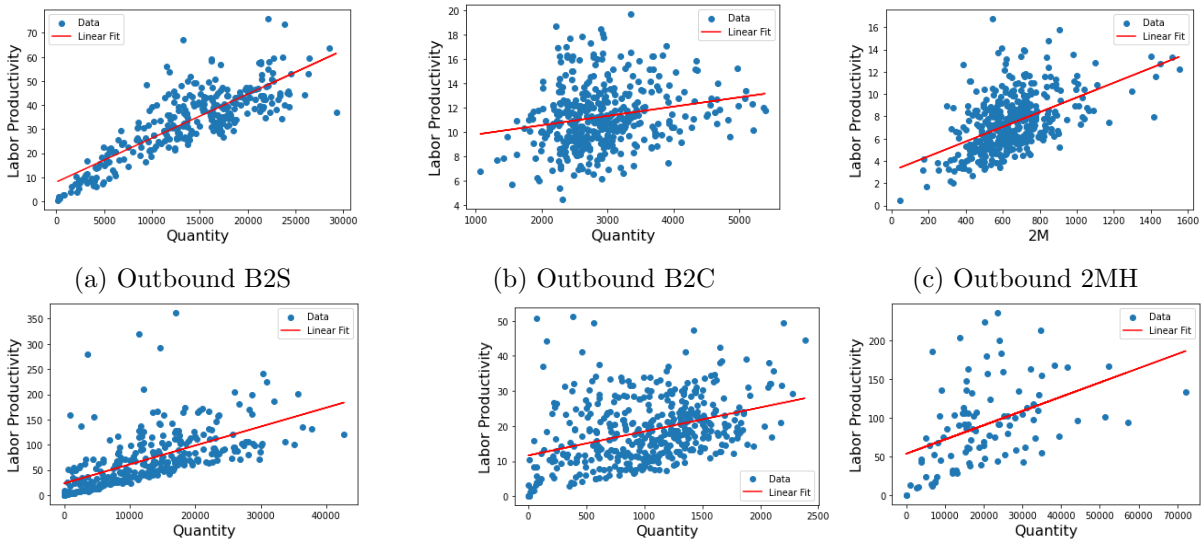
After collecting, preparing, and adapting the goodsflow features and the number of hours, the two datasets are consolidated per goodsflow. Some manipulations were needed to link the correct quantities to the correct hours of the day. For the outbound B2C and 2MH, the picking and packing hours were offset by one day. The delivery date time stamp (i.e., goods issue date) is the date the items leave the warehouse, and with reasonable assumption, the picking and packing for these items is done one day in advance. Similarly, the picking hours for outbound B2S are offset by one day, as this is done the day before the goods' packing, loading, and shipment. For the inbound hours regular and MDA, no offsetting was performed as the process is completed in one day within reasonable bounds as the dock-to-stock measure is above 98%. The cross-dock hours are not offset as well. It is reasonable to assume that all cross-dock deliveries within a particular week are processed and shipped in that same week. Some inconsistencies resulted from merging the daily quantities with the daily number of hours. For example, there were days when hours were registered, but no outbound deliveries were planned. These inconsistencies were amended by adding the hours to the previous day or the following day, depending on whether it was a week or weekend day and if any holidays or other exceptions were found. After merging the two datasets, the OWR could be calculated by dividing the total quantity by the total number of hours daily for all goodsflows, except for BBXD, which is on a weekly level. After this manipulation, the datasets per goodsflow are ready for analysis.

Appendix D

Analysis & Diagnosis Additional Information

D.1. Quantity

The relationships between the total quantity and labor productivity per goodsflows are displayed in Figure A.1. All the relationships have a significant positive linear trend. The linear relationships indicate that labor productivity increases (decreases) proportionally with increases (decreases) in quantities. Quantities create economies of scale, which leads to higher efficiency of operations. Quantity is a strong driver of productivity for all goodsflows, especially for outbound B2S. The relationship between the total daily quantities and labor productivity is extremely strong ($r = 0.79, p < .01$). The relationships between regular inbound and inbound MDA are only moderate ($r = 0.38$ and $r = 0.33$, $p < .01$, respectively). Therefore, other features might be equally critical in predicting labor productivity for these goodsflows.



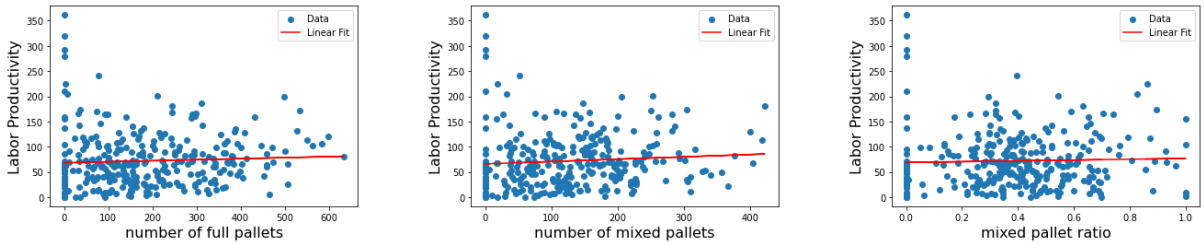
(a) Outbound B2S (b) Outbound B2C (c) Outbound 2MH

(d) Regular Inbound (e) Inbound MDA (f) BBXD

Figure A.1: Relationship between quantity and labor productivity for all goodsflows

D.2. Mixed pallet ratio

The relationships between the number of full pallets, mixed pallets, and mixed pallet ratio and labor productivity are displayed in Figure A.17. The number of full and mixed pallets have an extremely weak positive relationship with labor productivity ($r = 0.14$ and $r = 0.18$, $p < .01$, respectively). Moreover, the relationship between the mixed pallet ratio and labor productivity is insignificant ($p = .14$). The number of full pallets was expected to correlate with labor productivity positively. The rationale is that full pallets simplify operations by reducing the need for sorting and enabling easier receiving and storage than mixed pallets or loose items. More full pallets indicate higher quantities of items processed efficiently. This creates economies of scale, increasing productivity as the number of full pallets increases. The initial expectation was that a pattern of diminishing returns would be found for the number of mixed pallets and the mixed pallet ratio. However, this anticipation was not substantiated. Therefore, these relationships are further analyzed.



(a) Labor Productivity vs. number of full pallets

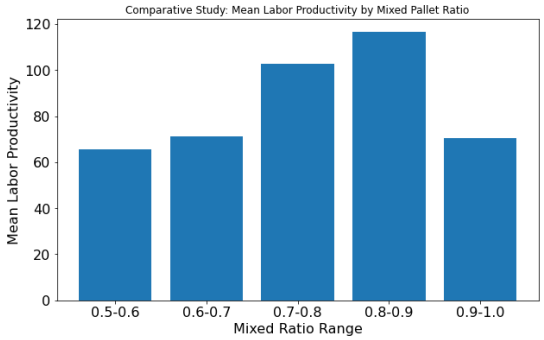
(b) Labor Productivity vs. number of mixed pallets

(c) Labor Productivity vs. mixed pallet ratio

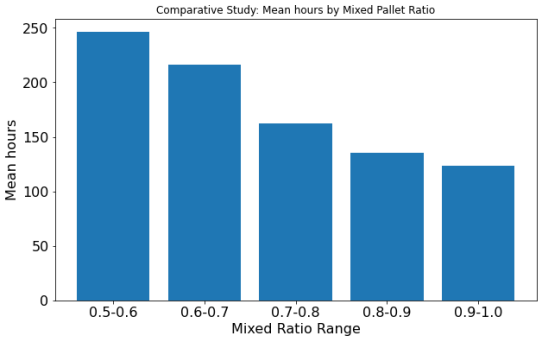
Figure A.2: Relationship of full pallets, mixed pallets, and mixed pallet ratio versus labor productivity

A comparative analysis is performed, which compares the mean labor productivity over different ranges of the mixed pallet ratio; see Figure A.3a. The mean labor productivity seems to increase as the mixed pallet ratio rises. However, a Kruskal-Wallis test indicated no significant difference in mean labor productivity across the different mixed ratio ranges (test statistic = 6.87, $p = .14$). A comparative study with the mean number of hours across the mixed ratio ranges shows a clear decreasing pattern; see Figure A.3b. The Kruskal-Wallis test statistic of 19.35 with $p < .01$ indicates a significant difference between the mean hours across the mixed pallet ratio ranges. Thus, as the mixed pallet ratio increases, i.e., relatively more mixed pallets arrive at inbound, fewer hours are used to process these pallets. No significant differences were found between the mean quantities across the mixed ratio ranges (Kruskal-Wallis test statistic = 5.59, $p = .23$). Thus, the mean quantity remains constant across the mixed pallet ratio ranges. Still, a decrease in the number of hours is confirmed. Therefore, labor productivity increases as the relative amount of mixed pallets increases.

According to the comparative study findings, relatively more mixed pallets lead to higher productivity, contrary to MediaMarkt's beliefs. Processing mixed pallets requires more competency from inbound employees, as the pallets must be more carefully received and sorted compared to full pallets. It could be that when the mixed pallet ratio is higher, inbound



(a) Mean Labor Productivity vs. mixed pallet ratio



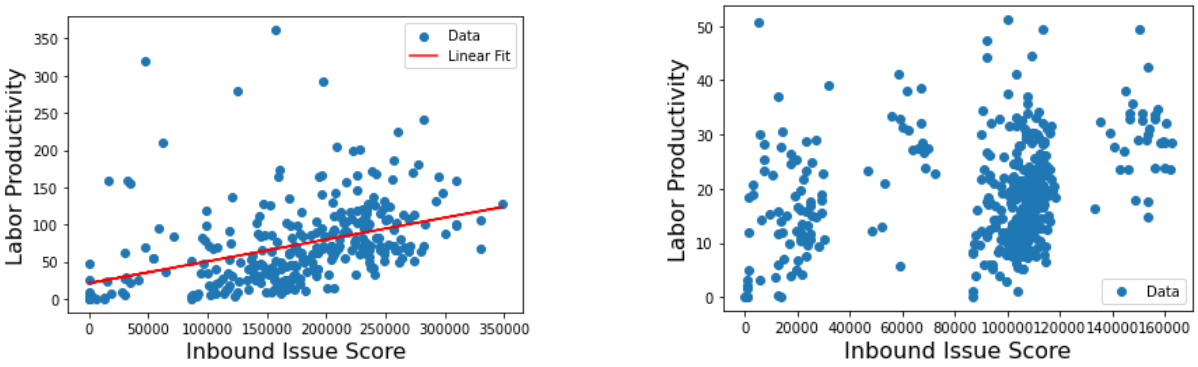
(b) Mean number of hours vs. mixed pallet ratio

Figure A.3: Relationship of full pallets, mixed pallets, and mixed pallet ratio vs. Labor Productivity

employees with more training and skills are scheduled. Another reason could be that higher task complexity and variety associated with an increased mixed pallet ratio may lead to higher productivity. Theory has shown that task complexity is an essential determinant of human behavior and task performance (Liu and Li, 2011). Additional data on employees' skills, training, level of experience, or motivation could provide further insights into the observed patterns in the data. However, this data is currently unavailable.

D.3. Inbound Issue score

Suppliers may have inaccuracies when delivering orders, such as missing information or damaged, missing, or surplus items. These inaccuracies lead to lower productivity as more work must be performed to process these items than when no inaccuracies occur. It was expected that the inbound issue score would negatively impact labor productivity. However, for regular inbound, a strong positive linear trend is found; see Figure A.4a. Furthermore, the inbound issue score for inbound MDA displays an unusual pattern. Only a limited amount of suppliers supply MDA inbound. Therefore, the score falls within certain bounds. There is no discernible pattern between the inbound MDA issue score and labor productivity. The difference between the data and expectation is probably due to the several assumptions underlying the inbound issue score. The inbound issue score is too highly dependent on the total daily quantity. Another method should be applied to address the effect of the number of inbound issues on inbound productivity. The current data does not allow for more straightforward implementation due to the many inconsistencies in the manually entered data. The inbound issue score does not provide insight into the influence of inbound issues on labor productivity and is therefore excluded from the research.

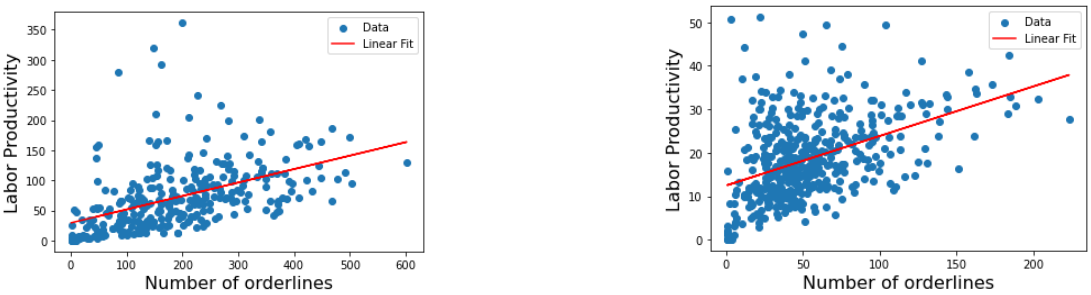


(a) Regular Inbound

(b) Inbound MDA

Figure A.4: Relationship between Labor Productivity and the inbound issue score

D.4. Number of orderlines

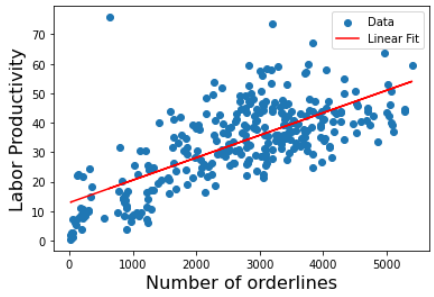


(a) Regular Inbound

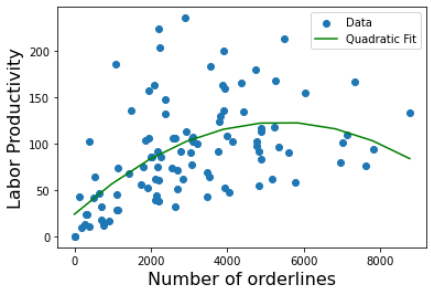
(b) Inbound MDA

Figure A.5: Relationship between Labor Productivity and the number of orderlines

The relationship between the number of orderlines and labor productivity is positively significant for all goodsflows. Generally, the relationship between the number of orderlines and productivity exists due to the strong positive relationship between the number of orderlines and the quantity, and quantity drives productivity as it creates economies of scale. For BBXD, the number of orderlines has a diminishing return on labor productivity. The higher the number of orderlines, the less proportionally the productivity increases. Whenever the number of orderlines exceeds a certain threshold, this leads to an additional processing time for the different SKUs, diminishing productivity return. The relationship between the number of orderlines and labor productivity is mainly driven by the increased quantity that must be handled. Therefore, the number of orderlines does not provide additional information on the relationship with productivity for the goodsflows.



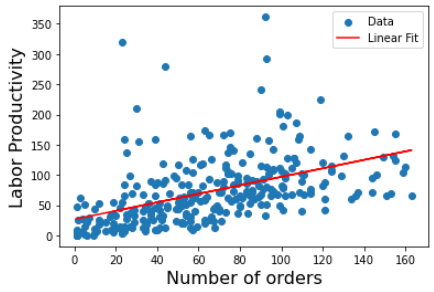
(a) Outbound B2S



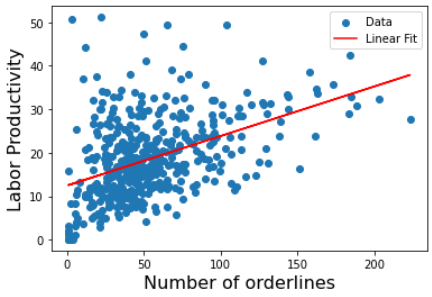
(b) BBXD

Figure A.6: Relationship between Labor Productivity and the number of orderlines

D.5. Number of orders



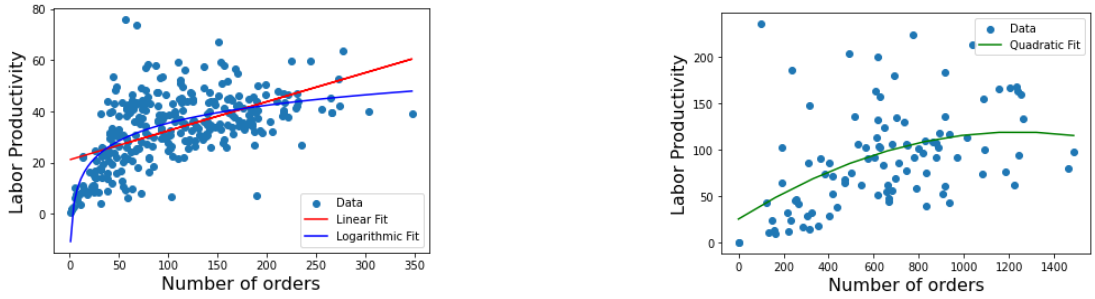
(a) Regular Inbound



(b) Inbound MDA

Figure A.7: Relationship between Labor Productivity and the number of orders

The number of orders is positively related to labor productivity for all goodsflows. Primarily, the increase in productivity is driven by the rise in quantities, which occurs when the number of orders increases, creating economies of scale. For outbound B2S and BBXD, it seems that when the number of orders increases above a certain threshold, the additional activities in order picking, loading, and packing of the varying orders lead to a diminished productivity return. For BBXD, the number of orders follows the same pattern as the total daily quantity. Therefore, the number of orders does not provide additional information. Similarly, for regular inbound and inbound MDA, the number of orders does not provide additional information. For outbound B2S, the quantities are extremely strongly correlated. Thus, an increase in quantities leads to an increase in productivity. However, productivity growth stagnates when the number of orders increases above a certain threshold. This could indicate that the interaction between these two features, for example, via the average ordersize, might be of influence labor productivity.



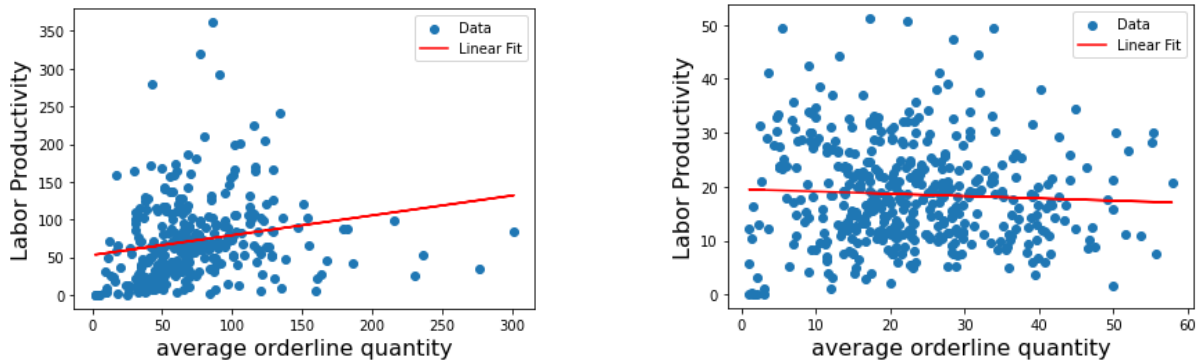
(a) Outbound B2S

(b) BBXD

Figure A.8: Relationship between Labor Productivity and the number of orders

D.6. Average quantity per orderline

An increase in the average quantity per orderline is expected to lead to higher efficiency in warehouse operations due to consolidated processing and handling, thereby increasing productivity. Conversely, decreasing the average quantity per orderline can lead to lower efficiency because it creates more diverse orderlines with smaller quantities, increasing the overall processing time and reducing productivity. The relationship between the average quantity per orderline and labor productivity is displayed in Figures A.9a, A.9b, A.10a and A.10b, for regular inbound, inbound MDA, outbound B2S, and BBXD, respectively.

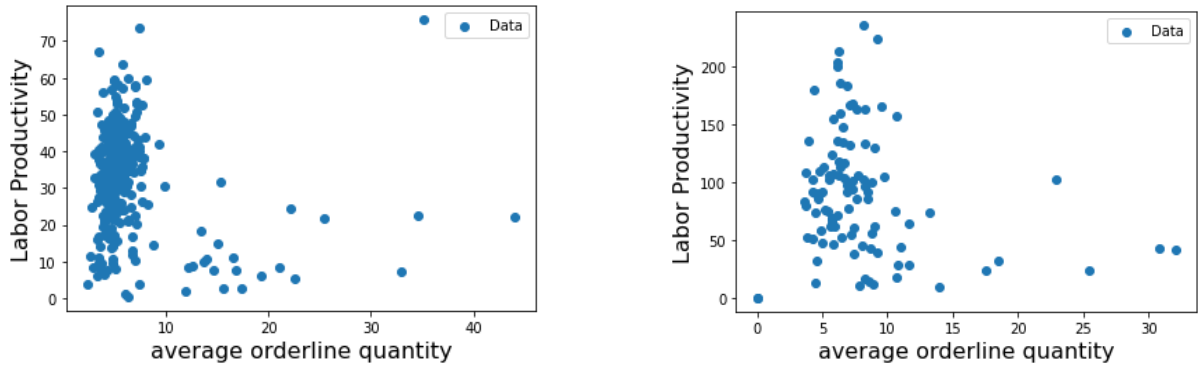


(a) Regular Inbound

(b) Inbound MDA

Figure A.9: Relationship between Labor Productivity and the average quantity per orderline

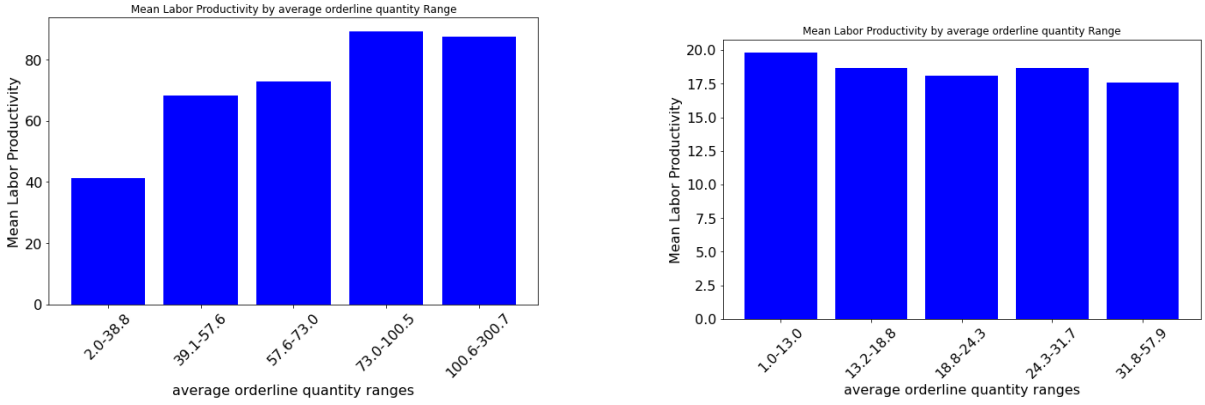
A moderate positive trendline is discovered for regular inbound between the average quantity per orderline and labor productivity. No discernible trendline is established between the average orderline quantity and labor productivity for outbound B2S, inbound MDA, and BBXD. The data is scattered for inbound MDA, indicating no clear pattern between labor productivity and average orderline quantity. For outbound B2S and BBXD, the average orderline quantity is relatively stable and falls between narrow bounds, with some extremely high averages as exceptions. Thus, the average quantity per orderline remains relatively constant, independent of labor productivity. The average quantity per orderline is often high on days (or weeks) before and after promotions. During these periods, large orders are sent to the stores to the warehouse and stores to stock up. Overall, from the visualization, the average orderline quantity does not impact labor productivity.



(a) Outbound B2S (b) BBXD

Figure A.10: Relationship between Labor Productivity and the average quantity per orderline

A comparative analysis is performed because, contrary to expectation, the relationship between the average orderline quantity and labor productivity showed no positive trend for inbound MDA, outbound B2S, and BBXD. For each goodsflow, the average orderline quantity data is divided into five equal bins. For each bin, the average labor productivity is calculated to see how different ranges of the average quantity per orderline affect the mean productivity. The bar charts are displayed in Figure A.11a, A.11b, A.12a and A.12b, for regular inbound, inbound MDA, outbound B2S, and BBXD, respectively.

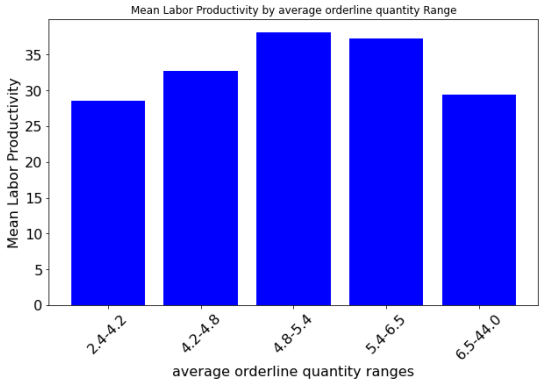


(a) Regular Inbound (b) Inbound MDA

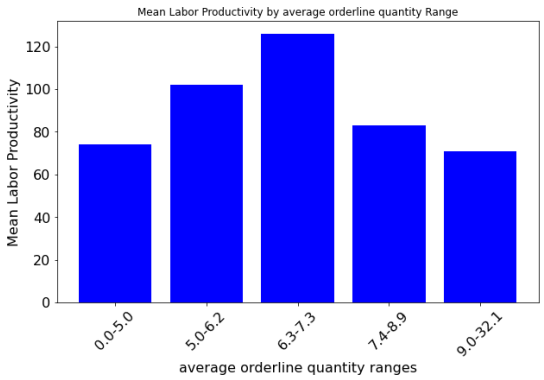
Figure A.11: Comparative analysis: mean labor productivity vs. average orderline quantity

For regular inbound, it is visible that the mean labor productivity increases as the average orderline quantity increases. The difference was deemed significant (Kruskal-Wallis test statistic = 39.50, $p < .01$). Moreover, Figure A.11a shows that the mean labor productivity changes little if the mean labor productivity is very high, indicating possibly a pattern of diminishing returns. For inbound MDA, the mean labor productivity is quite similar for all ranges of the average quantity per orderline. No significant differences between mean labor productivity and the average orderline quantity are found (Kruskal-Wallis test statistic = 3.20, $p = 0.52$). This concludes that the average orderline quantity is not a good indicator of labor productivity at inbound MDA. Other factors have more impact on productivity.

For outbound B2S, the average quantity per orderline remains relatively constant, independent



(a) Outbound B2S



(b) BBXD

Figure A.12: Comparative analysis: mean labor productivity vs. average orderline quantity

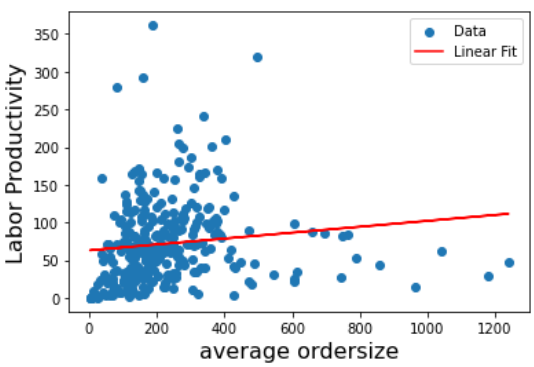
of labor productivity, looking at the scatterplot. However, Figure A.12a shows a slight increase in the mean productivity as the average orderline quantity increases. After a certain average quantity per orderline is achieved, the mean productivity decreases again. The difference between the mean productivity across the average orderline quantity range is deemed significant (Kruskal-Wallis test statistic = 23.37, $p < .01$). A similar significant pattern is found for the BBXD goodsflow (Kruskal-Wallis test statistic = 18.17, $p < 0.01$). Thus, although the average orderline quantity for outbound B2S and BBXD initially seemed entirely independent of labor productivity, some effect is visible in the comparative analysis. However, it is unknown if this threshold is created due to days/weeks where the quantities were extremely high, causing an overload of work for the available capacity, or if the average itself influences the efficiency in handling. The behavior of the average orderline quantity will be further analyzed with the prediction model.

D.7. Average ordersize

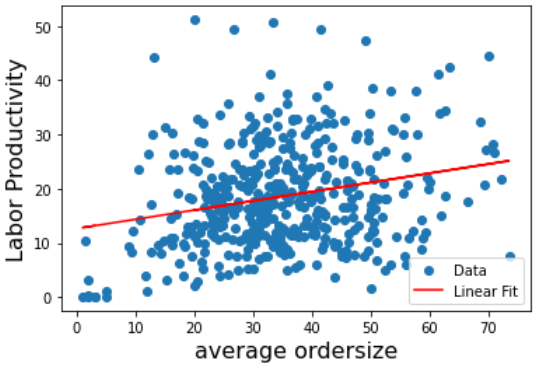
Similar to the average orderline quantity, it is expected that when the average ordersize increases, this leads to higher efficiency due to consolidated processing and handling, thereby increasing productivity. Conversely, if the average ordersize decreases, the handling efficiency will diminish and negatively impact labor productivity. The relationship between the average ordersize and labor productivity is displayed in Figures A.13a, A.13b, A.14a and A.14b, for regular inbound, inbound MDA, outbound B2S, and BBXD, respectively.

Positive linear trendlines are discovered between the average ordersize and labor productivity for regular inbound and inbound MDA. However, for outbound B2S and BBXD, no significant pattern is found. The outbound B2S data is scattered, and the average ordersize and labor productivity move independently. For BBXD, it seems that the average ordersize remains relatively constant independent of the changes in productivity.

A positive relationship was expected between the average ordersize and labor productivity for all goodsflows. Therefore, an additional comparative analysis is performed in a similar fashion as with the average orderline quantity. For regular inbound and inbound MDA, a significant

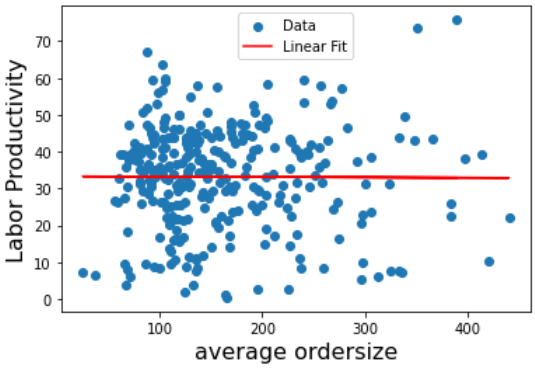


(a) Regular Inbound

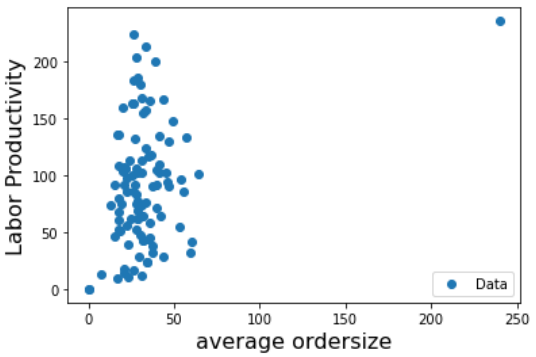


(b) Inbound MDA

Figure A.13: Relationship between Labor Productivity and the average ordersize



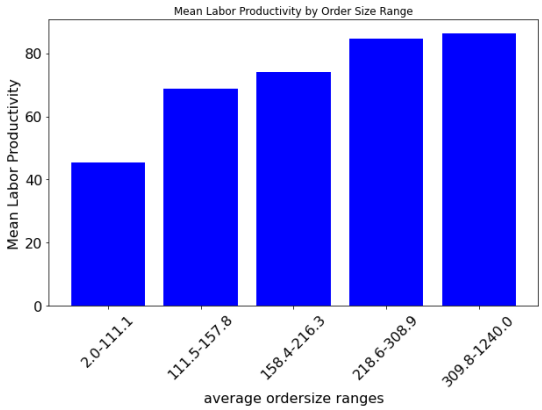
(a) Outbound B2S



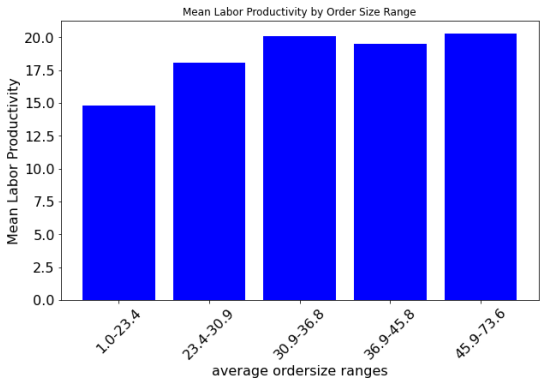
(b) BBXD

Figure A.14: Relationship between Labor Productivity and the average ordersize

difference is found between the mean labor productivity and the average ordersize range with the Kruskal Wallis test statistic of 32.68 and 20.52, $p < .01$, for inbound regular and MDA, respectively. As expected, the mean productivity increases as the ordersize increases for these goodsflow.



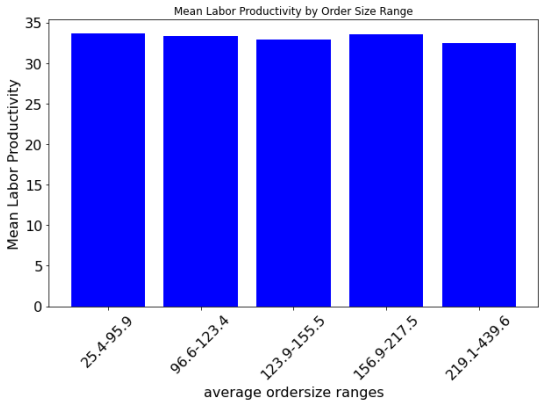
(a) Regular Inbound



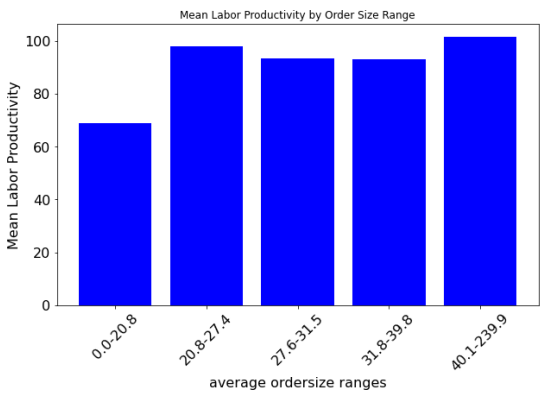
(b) Inbound MDA

Figure A.15: Comparative analysis: mean labor productivity vs. average ordersize

For outbound B2S, the Kruskal-Wallis test indicated no significant difference in the mean labor productivity across the average ordersize ranges (test statistic = 0.61, $p = 0.96$). The average ordersize is not indicative of labor productivity at outbound B2S, and other factors are better predictors of labor productivity. Possibly, average ordersize has little influence as orders are consolidated for picking purposes. The picking operation is the largest outbound warehouse manual operation and, therefore, the most influential on labor productivity. For BBXD, a very small average ordersize does seem to lead to lower average labor productivity, as seen in Figure A.16b. When a certain threshold is reached, the mean productivity seems relatively stable. This would be as expected, indicating that smaller orders at BBXD would lead to more ventilation efforts as more orders with smaller amounts must be distributed over the stores. However, the Kruskal-Wallis test indicates no significant difference (test statistic = 4.61, $p = 0.33$). Thus, this conclusion cannot be drawn based on the current data. The prediction model further tests the behavior of the average ordersize and possible interaction effect with other essential features.

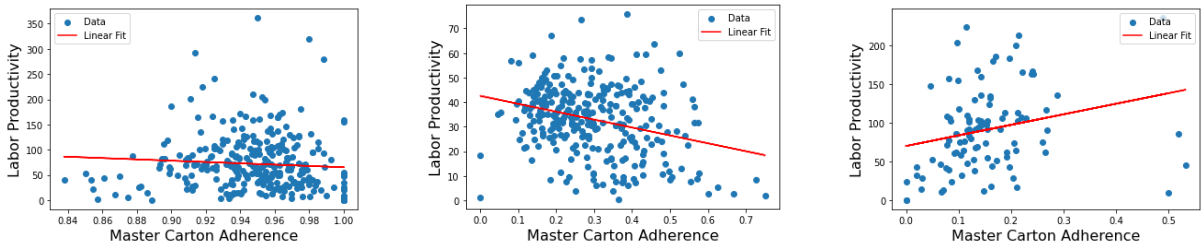


(a) Outbound B2S



(b) BBXD

Figure A.16: Comparative analysis: mean labor productivity vs. average ordersize



(a) Inbound (b) Outbound B2S (c) BBXD

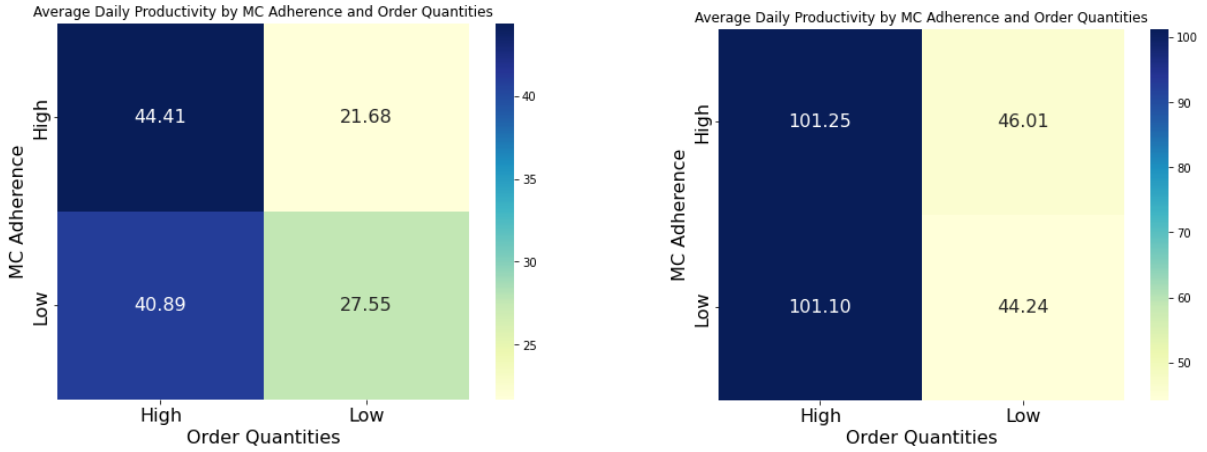
Figure A.17: Relationship between Labor Productivity and Master Carton Adherence

D.8. Master carton adherence

The relationship between the Master Carton (MC) adherence and labor productivity has been displayed in Figure A.17a, Figure A.17b, and A.17c, for inbound, outbound B2S and BBXD, respectively. For BBXD, a positive relationship exists between MC adherence and labor productivity, which aligns with expectations. When the orderlines adhere to the master carton value, the handling time is expected to be lower, as items can more easily be distributed over the different stores when ordered in master carton value. Higher adherence would thus indeed result in higher productivity and vice versa. For outbound B2S and regular inbound, the relationship between the master carton value adherence and productivity is negative. Contrary to the general expectation, labor productivity seems to decrease (increase) when the master carton adherence increases (decreases), and vice versa. An underlying negative relationship between the master carton adherence and the total daily quantity causes this unexpected relationship.

Further analysis is performed to understand the difference in expectations. A 2x2 matrix has been created that categorizes the data into four different combinations: low quantity - low MC adherence, low quantities - high MC adherence, high quantities - high MC adherence, and high quantities - low MC adherence. High and low are determined based on whether the data is higher or lower than the mean. The average labor productivity per category is then computed. Now, one can identify how the different scenarios of MC adherence and daily quantities relate to daily labor productivity. The color-coding and annotations in the matrix visually represent each combination's average daily labor productivity values. Darker colors indicate higher average productivity values, while lighter colors indicate lower values. Furthermore, Tukey's HSD test is used to identify which specific combinations show statistically significant differences in average labor productivity.

For the inbound, the productivity is independent of the master carton adherence. No significant differences were found between mean labor productivity for the scenarios where the quantity is high and the master carton adherence is either high or low ($p < .01$). Moreover, no significant difference was found between mean labor productivity for the scenarios where the quantity is low and the master carton adherence is either high or low. This indicates when the quantity is high, independent of the master carton adherence, labor productivity is high on average. Conversely, when the quantity is low, independent of whether the master carton adherence is high or low, labor productivity is lower on average. The master carton adherence is not a good



(a) Outbound B2S

(b) Inbound

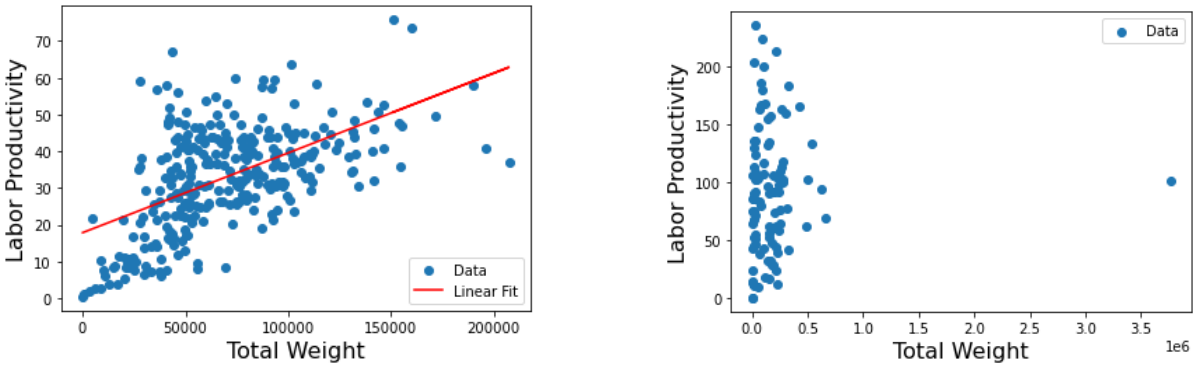
Figure A.18: Average labor productivity by Master Carton adherence and quantity

indicator of productivity, which explains the very weak correlation found. For Outbound B2S, no significant differences were found between mean labor productivity for the scenarios where the quantity is high and the master carton adherence is either high or low ($p < .01$). Indicating that when the quantity is high, the master carton adherence has no influence. Moreover, when the quantities are lower on average, the average labor productivity is also lower, as expected. Still, it is also noticed that the average productivity is slightly lower when the master carton value is more adhered to. The master carton value has been shown to impact distribution logistics efficiency significantly. Defining the optimal master carton value for each SKU is a vital planning problem that affects warehouse operations Wensing et al. (2018). Adhering to master carton requirements (which might be non-optimal) can introduce complexity in the overall outbound process. Thereby negatively impacting labor productivity. The behavior will be analyzed further with the prediction model.

D.9. Total weight and volume

The total volume is assumed to be inaccurately recorded in the system, as many entries equaled zero for both outbound B2S and BBXD. Therefore, the total volume is excluded from further analysis. The relationships between the total daily weight and labor productivity are displayed in Figure A.20a and Figure A.20b, for B2S and BBXD, respectively. No significant relationship was found between the total weight and productivity for BBXD. A strong positive significant relationship is found between the total weight and productivity for outbound B2S. As the weight increases, the quantity increases, which increases the efficiency of handling and, therefore, increases productivity. However, a pattern of diminishing returns was expected as larger products are more unwieldy to handle, needing additional equipment or machinery. Thus, fewer items can be picked simultaneously, and larger items could lead to faster exhaustion of order pickers (Falkenberg and Spinler, 2022). The findings are not aligned with MediaMarkt's beliefs. Therefore, an additional analysis is performed.

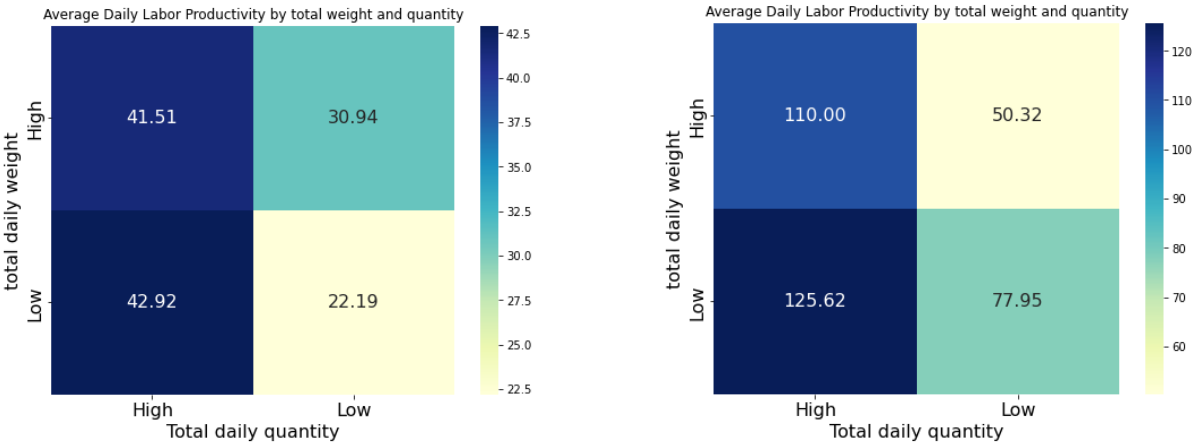
Again, a 2x2 matrix is created to categorize the four different combinations in a similar fashion



(a) Outbound B2S (b) BBXD

Figure A.19: Relationship between Labor Productivity and the total weight

as for the master carton adherence; see Figure A.20. For outbound B2S, no significant difference was found between the average labor productivity in the scenario where both the quantity and total weight are high and the scenario where quantity is high and total weight is low. All other combinations were significantly different from each other. This indicates that when the total daily quantity is high, independent of the total weight being high or low, the average productivity is higher than when the total quantity is low. In the scenarios where the quantity is lower than on average, it seems that productivity is higher when the total weight is high than when the total weight is low. So if the total weight handled is high relative to the total quantity, i.e., predominantly larger items are handled on days where the quantity is low, productivity is higher. When the total weight is low relative to the total quantity, i.e., predominantly smaller items are handled on days where the quantity is lower than average, productivity is low. This could be caused by the inefficiency of picking a few small items. When quantities are low, and many small items have to be picked, orders cannot be adequately consolidated to overcome the inefficiencies of smaller quantities. This is most likely caused by lower average ordersize and quantity per orderline. This relationship will be further investigated with the prediction model.



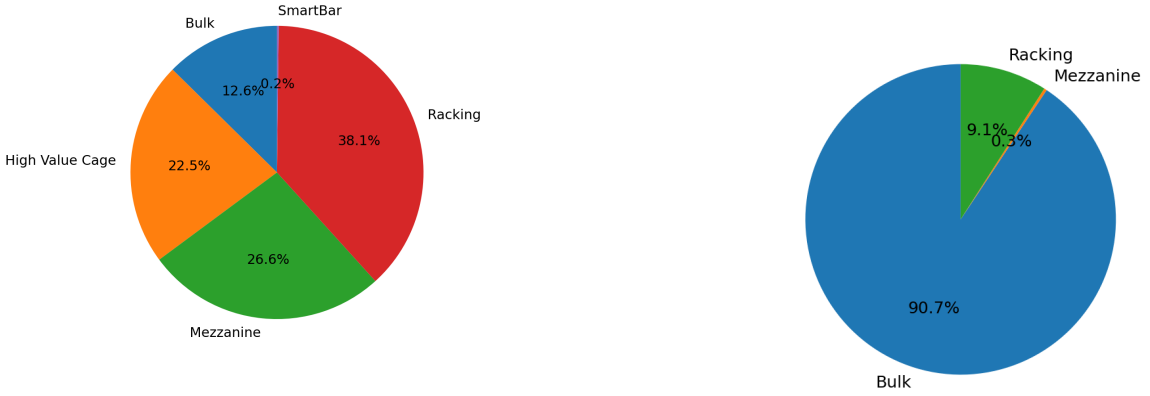
(a) Outbound B2S (b) BBXD

Figure A.20: Average labor productivity by total weight and quantity

For BBXD, no significant differences were found between mean labor productivity for the

scenarios where the quantity is high, and the total weight is either high or low. Moreover, no significant difference was found between mean labor productivity for the scenarios where the quantity is low, and the total weight is either high or low. This indicates that labor productivity is higher on average when the quantity is high, independent of the total daily weight. Conversely, labor productivity is lower on average when the quantity is low, independent of whether the total daily weight is high or low. The total weight is not a good indicator of productivity, which explains why no correlation is found. The interaction effect of the total weight with other features on labor productivity will be further explored with the prediction model.

D.10. Warehouse locations



(a) Regular inbound

(b) Inbound MDA

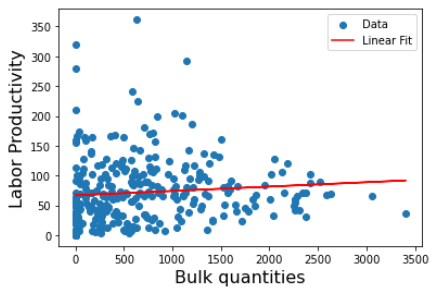
Figure A.21: Overview quantities per location Inbound regular and MDA

It is expected that items put away, stored, or retrieved from the exact warehouse location share similarities in executing operations. On the other hand, items from different locations might require different handling. Differences in the quantities and the number of orderlines per warehouse location might impact labor productivity. These relationships are further analyzed. An overview of the relative inbound quantity per warehouse location in 2021 and 2022 is provided in Figure A.21, for regular inbound and inbound MDA, respectively. The racking location comprises more than one-third of the total quantity for regular inbound. The second and third largest locations are high-value cage and mezzanine. The SmartBar location holds little inventory. For inbound MDA, more than 90% of the inbound items are stored at the bulk location. This is as expected because the inbound MDA goodsflow only contains items from the whitegoods product category.

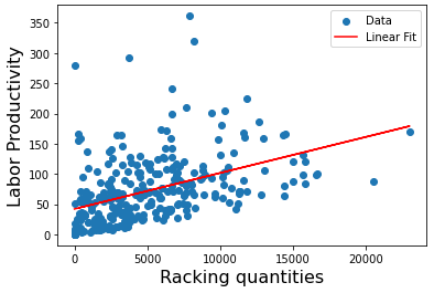
Table A.1: Correlations quantity per location vs. labor productivity

	Inbound	MDA	B2S	BBXD
Bulk quantity	0.25	0.36	0.30	*
Bulk orderlines	0.25	N.A.	0.46	*
Mezzanine quantity	0.61	*	0.56	0.58
Mezzanine orderlines	0.52	N.A.	0.57	0.56
Racking quantity	0.56	0.31	0.66	0.27
Racking orderlines	0.51	N.A	0.61	0.45
High-value cage quantity	0.65	N.A.	0.68	0.44
High-value cage orderlines	0.61	N.A.	0.61	0.35
SmartBar quantities	*	N.A.	N.A.	0.47
SmartBar quantities	*	N.A.	N.A.	0.45
*no significant correlation				

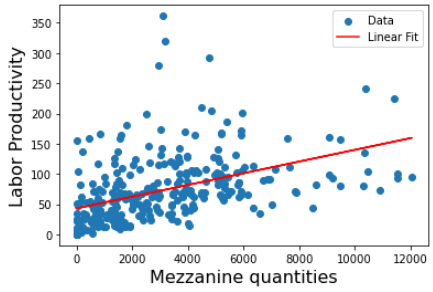
The significant Spearman correlations with $p < .01$ are displayed in Table A.1 for regular inbound. Strong positive correlations are found between the largest inbound regular locations: mezzanine, racking, and high-value cage and labor productivity. The relationships are linear, indicating that an increase (decrease) in these quantities coincides with an increase (decrease) in productivity quite linear. No diverging patterns, such as exponential decay or diminishing returns, are found; see Figure A.22.



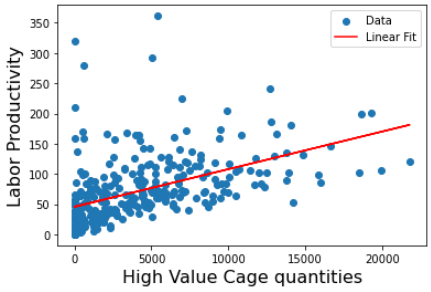
(a) Labor Productivity vs. Bulk quantities



(b) Labor Productivity vs. Racking quantities



(c) Labor Productivity vs. Mezzanine quantities

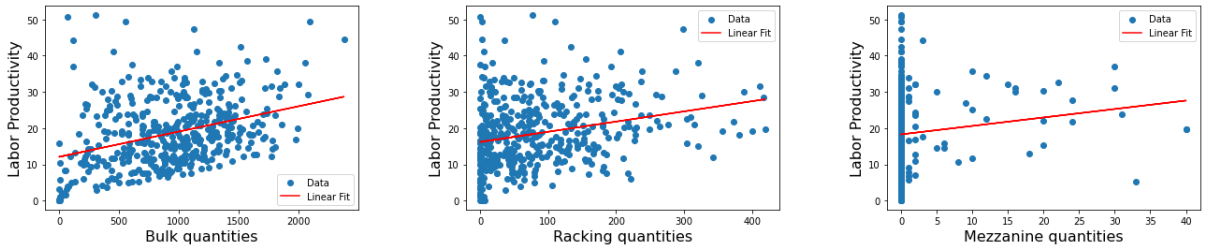


(d) Labor Productivity vs. High-Value Cage quantities

Figure A.22: Relationship between labor productivity and quantities per warehouse location - Inbound Regular

For inbound MDA, there is a strong positive relationship between the quantities at the bulk location and labor productivity, as this category dominates. For racking, there seems to be a somewhat positive relationship. However, there are many days where the quantities are

zero, distorting the relationship. Similarly, for the mezzanine, no significant correlation is found. There are no visible relationships between the labor productivity and the racking and mezzanine locations; see Figure A.23c. Overall, the quantities per warehouse location do not seem to provide additional information about labor productivity at inbound MDA due to the dominance of the bulk location.



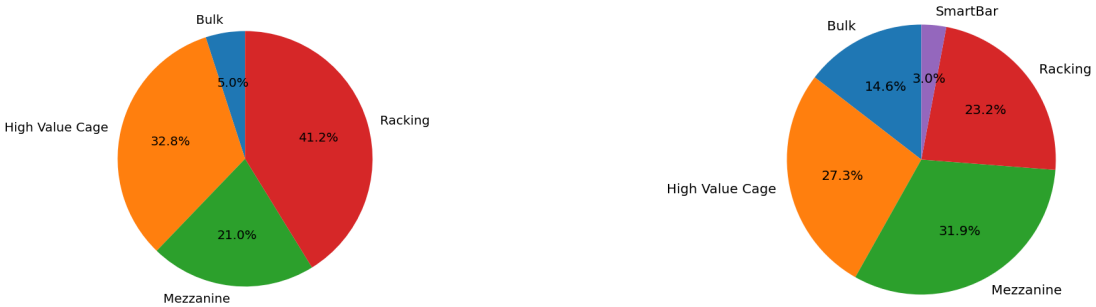
(a) Labor productivity vs. Bulk quantities

(b) Labor productivity vs. Racking quantities

(c) Labor productivity vs. Mezzanine quantities

Figure A.23: Relationship between labor productivity and quantities per warehouse location - Inbound MDA

An overview of the relative quantities per warehouse location in 2021 and 2022 is provided in Figure A.24, for outbound B2S and BBXD, respectively. The characteristics of certain locations are expected to lead to differences in productivity. Therefore, these relationships are also analyzed. The racking location represents more than 40% of the total outbound B2S quantities. Moreover, over one-third of the quantities are picked from the high-value cage location. As expected, the bulk quantities only represent a small percentage of the total quantities picked. Most bulk items are not sent to the store but directly to the customers, which is a separate process (outbound 2MH). For BBXD, the mezzanine is the largest location in terms of quantity. High-value cage is the second largest category. About 60% of the items handled at BBXD belong to the locations mezzanine and high-value cage, indicating that most items handled are quite small and no additional equipment is required, which could positively impact productivity.



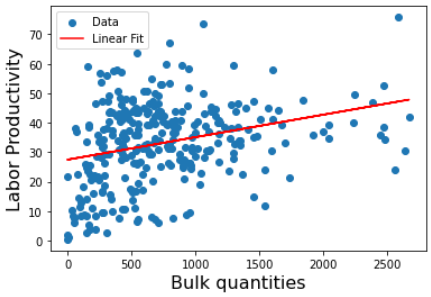
(a) Outbound B2S

(b) BBXD

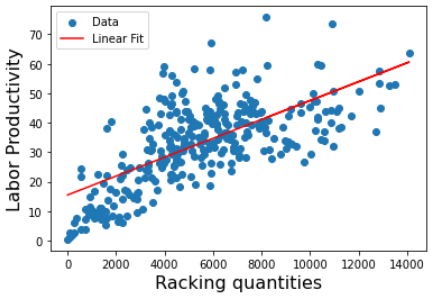
Figure A.24: Overview quantities per location outbound B2S and BBXD

The relationships between the quantity per location and labor productivity for outbound B2S are displayed in Figure A.25. The significant Spearman correlation coefficients are displayed in Table A.1. The relationship between bulk quantities and labor productivity is scattered compared to the other locations. Most likely because these types of items are only sporadically

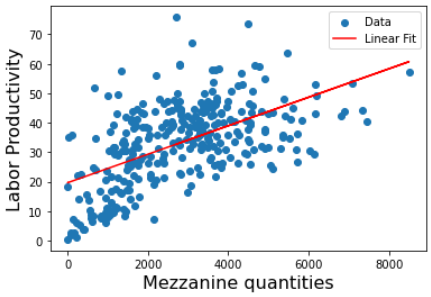
sent to the stores. The quantities picked from the racking and mezzanine location are quite linearly related to productivity, while the relationship with the quantities picked from the high-value cage location is more curved, indicating diminishing returns in productivity. When the quantities at this location reach a certain threshold, the increase in productivity stagnates. The additional process of high-value items leads to inefficiencies in large quantities. Similar patterns are found for the number of orderlines per warehouse location.



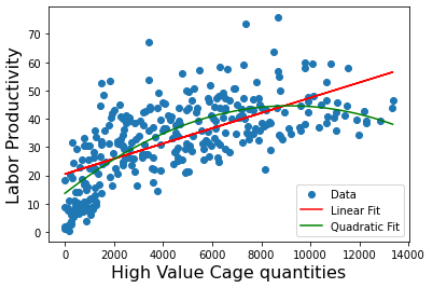
(a) Labor Productivity vs. Bulk quantities



(b) Labor Productivity vs. Racking quantities



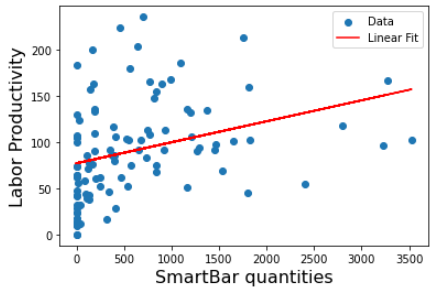
(c) Labor Productivity vs. Mezzanine quantities



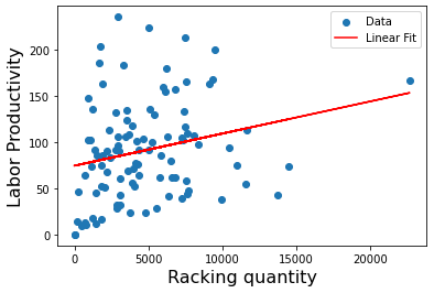
(d) Labor Productivity vs. High-Value Cage quantities

Figure A.25: Relationship between labor productivity and quantities per warehouse location - outbound B2S

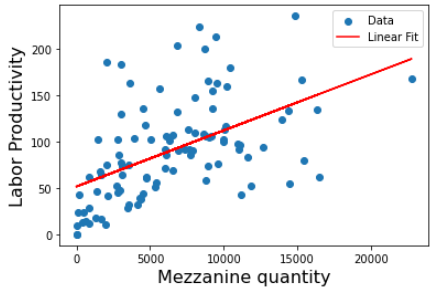
Quantities at BBXD are directly brought to the ventilation area and redistributed per store according to their demand. Although items are **not** stocked at different locations, it still might be that items from similar locations have similar characteristics leading to similar handling, while items stored at different locations might require different handling. The relationships between the quantity per location and labor productivity for outbound BBXD are displayed in Figure A.26. The significant Spearman correlation coefficients are displayed in Table A.1. No significant correlation was found between the bulk quantities and labor productivity. The smartbar, racking, and mezzanine locations are quite linearly related to productivity, while the relationship with the quantities from the high-value cage location is more curved, indicating diminishing returns in productivity. When the quantities at this location reach a certain threshold, the increase in productivity stagnates. The additional process of high-value items leads to inefficiencies in large quantities. Similar patterns are found for the number of orderlines per warehouse location.



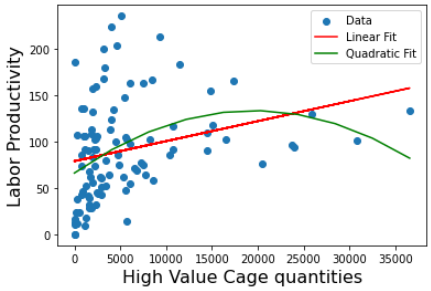
(a) Labor Productivity vs. smartbar quantities



(b) Labor Productivity vs. Racking quantities



(c) Labor Productivity vs. Mezzanine quantities



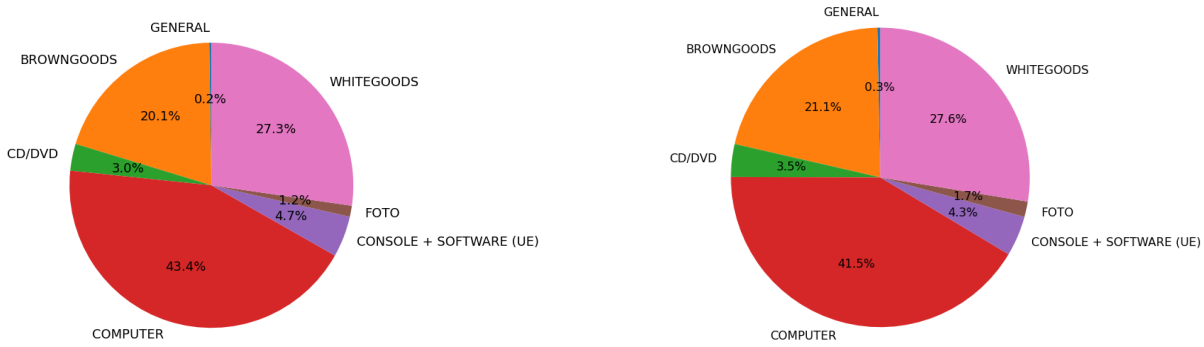
(d) Labor Productivity vs. High-Value Cage quantities

Figure A.26: Relationship between labor productivity and quantities per warehouse location - BBXD

D.11. Product category

Similar to the location mix, it is expected that items from the same product category might influence productivity. Items in the same product category share similar attributes, such as weight, volume, and size, which leads to similar handling. Items in different product groups might differ in several attributes and lead to different handling and processes. Therefore, the relationship between the quantity per product group and labor productivity is analyzed. An overview of the relative quantity per product group in 2021 and 2022 is provided for inbound and outbound B2S in Figure A.27. Inbound MDA is excluded as it has only one product category. The largest product group is computer. It represents more than 40% of inbound and outbound quantities. The second largest group is whitegoods, followed by browngoods. The other product groups account for less than 10% of the total quantities going in and out of the warehouse.

The Spearman correlations between labor productivity and the quantities per product category for inbound, outbound B2S and BBXD, are displayed in Table A.2. Only the significant correlations are displayed ($p < .01$). Figure A.28 displays the strong positive linear relationships, which were not distorted by many zero values in the data. The smaller product categories, general, CD/DVD, foto, and console, had many zero values, which distort the relationships. Positive linear relationships were found for the product categories browngoods, computer, and whitegoods. This indicates that an increase (decrease) in these quantities coincides with an increase (decrease) in productivity proportionally. No diverging patterns, such as exponential decay or diminishing returns, are found. Similar findings were found for the number of orderlines



(a) Inbound regular

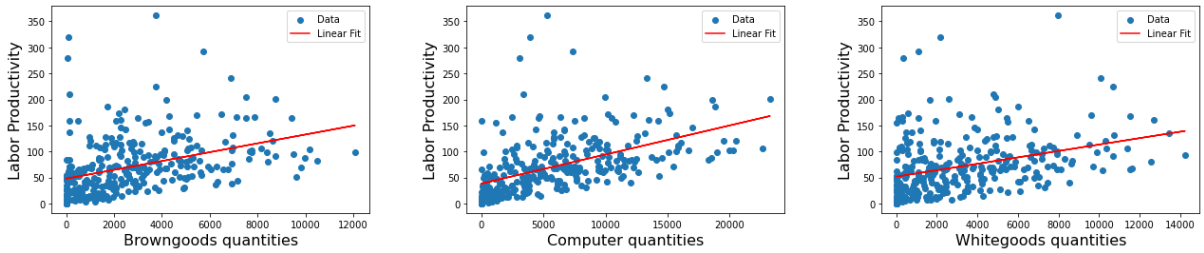
(b) Outbound B2S

Figure A.27: Overview of the quantity per product category

per product group.

Table A.2: correlations quantity per product category

	Inbound	B2S	BBXD
GENERAL	0.25	0.32	*
BROWNGOODS	0.53	0.63	*
CD/DVD	0.39	0.37	*
COMPUTER	0.67	0.72	0.53
FOTO	0.31	0.38	*
WHITEGOODS	0.44	0.55	*
CONSOLE	0.51	0.48	*



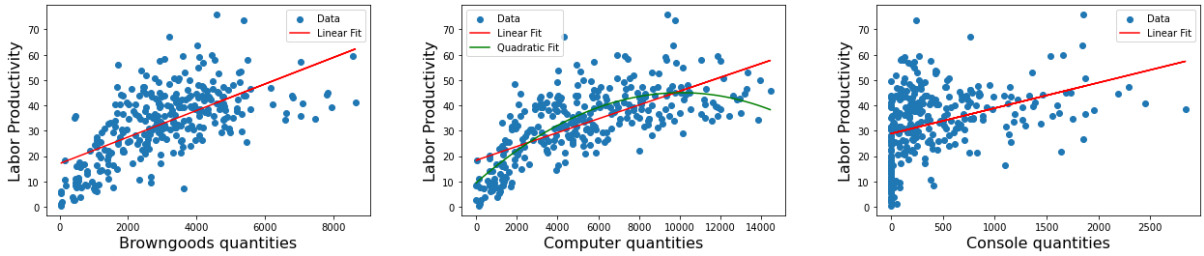
(a) Labor Productivity vs. browngoods quantities

(b) Labor Productivity vs. computer quantities

(c) Labor Productivity vs. whitegoods quantities

Figure A.28: Relationship between labor productivity and product category - Inbound

The relationships between the quantities per product category and labor productivity for outbound B2S are displayed in Figure A.29 and A.30. No discernible pattern was found for the product category CD/DVD due to the many zero values distorting the relationship. For the smaller categories, general, foto, and console positive linear trendlines are found, indicating that an increase (decrease) in these quantities coincides with an increase (decrease) in labor productivity proportionally. Evidence of diminishing returns exists for the browngoods, computer, and whitegoods quantities. Thus when these quantities increase above a certain threshold, labor productivity stagnates. The relationship is most pronounced for the computer category, which is the largest product category for outbound B2S. Similar findings were found for the number of orderlines per product category.

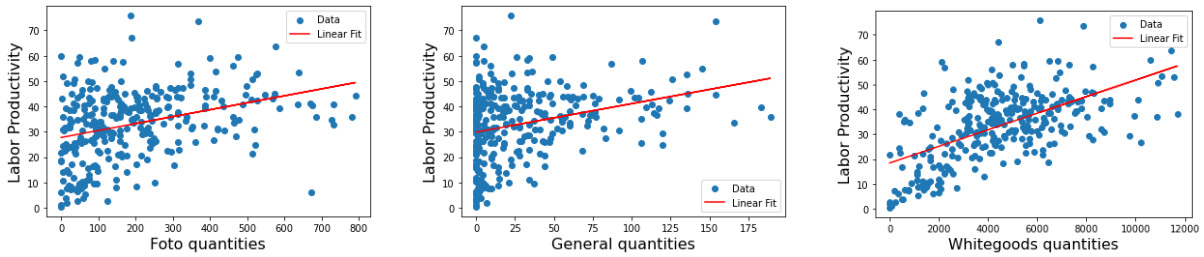


(a) Labor Productivity vs. browngoods quantities

(b) Labor Productivity vs. computer quantities

(c) Labor Productivity vs. console quantities

Figure A.29: Relationship between labor productivity and product category - Outbound B2S



(a) Labor Productivity vs. foto quantities

(b) Labor Productivity vs. general quantities

(c) Labor Productivity vs. whitegoods quantities

Figure A.30: Relationship between labor productivity and product category - Outbound B2S

An overview of the relative quantity per product group in 2021 and 2022 for BBXD is displayed in Figure A.31. The computer category accounts for almost three-quarters of the total products at BBXD. The second largest group is whitegoods, followed by browngoods. The other product groups account for less than 10% of the total quantities at BBXD.

For the BBXD goodsflow, no significant correlations were found between the quantities or the number of orderlines per product group and labor productivity. Except for the computer product group. The relationship even seems to display diminishing returns, indicating that when the computer quantities increase above a certain threshold, the productivity increase diminishes. This relationship also seems to hold for the total quantity at BBXD and the productivity in general and is as expected. Overall, the quantities per product category do not seem to provide additional information about BBXD's labor productivity due to the dominance of the computer

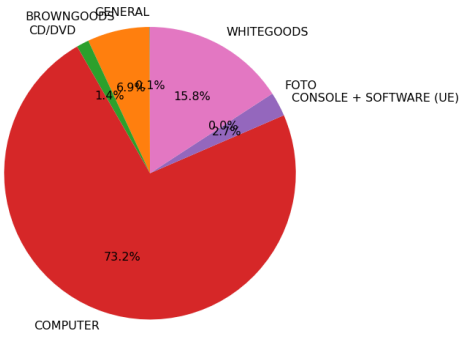
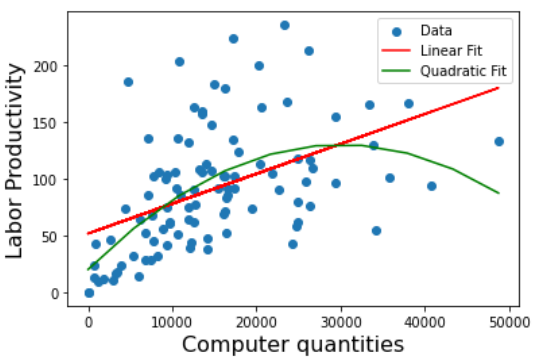
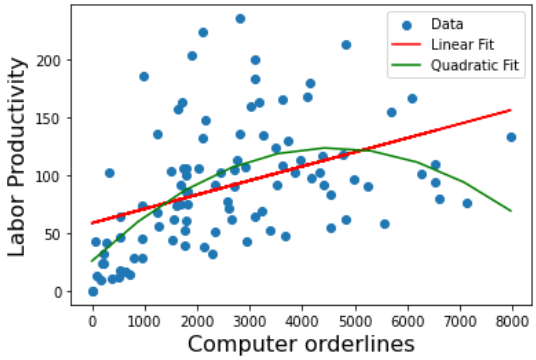


Figure A.31: Overview of the quantity per product category - BBXD

product category.



(a) BBXD



(b) BBXD

Figure A.32: Relationship between labor productivity and Computer product category - BBXD

Appendix E

Theoretical Background Model Development

E.1. Gradient Boosting Decision Trees

The basic methodology of the GBDTs is provided based on the original work by Friedman (2001) and the tutorial by Natekin and Knoll (2013). The classical supervised learning setting aims to find the functional dependence between X and y, where $X = (x_1, ..., x_N)$, represents the set of explanatory input variables, and y is the dependent variable. The estimate of the function $\hat{f}(x)$ must minimize some specified loss function $\psi(y, f)$. The estimation problem can be rewritten in terms of expectations to minimize the expected loss function over the responsible variable $E_y(\psi[y, f(x)])$

$$\begin{aligned}\hat{f}(x) &= y \\ \hat{f}(x) &= \arg\min_{f(x)} \psi(y, f(x)) \\ \hat{f}(x) &= \arg\min_{f(x)} E_x[E_y(\psi[y, f(x)])|x]\end{aligned}$$

The function can take any form; therefore, in order to make the estimation of this function more tractable, the search space of the function can be restricted to a parametric family of the function $f(x, \theta)$. This transforms the function optimization into a parameter estimation problem.

$$\begin{aligned}\hat{f}(x) &= f(x, \hat{\theta}) \\ \hat{\theta} &= argmin_\theta E_x[E_y(\psi[y, f(x, \theta)])|x]\end{aligned}$$

The solution for this closed-form parameter estimation is not available. Therefore, iterative numerical procedures can be considered. The most frequently used procedure is the steepest gradient descent. The aim is to decrease the empirical loss function $J(\theta)$ over the N observed datapoints $(x, y)_{i=1}^N$, resulting in equation A.1

$$J(\theta) = \sum_{i=1}^{N} \psi(y_i, f(x_i, \hat{\theta})) \tag{A.1}$$

The steepest descent optimization is based on consecutive improvements along the direction of the gradient of the loss function $\nabla J(\theta)$. The following steps are taken for the optimization procedure:

1. Initialize the parameter estimate $\hat{\theta_0}$. Repeat the next steps for each iteration.
2. Obtain a collapsed estimate of the whole assemble, i.e. the sum of all the estimate increments from 1 up to t:
$$\hat{\theta^t} = \sum_{i=0}^{t-1} \hat{\theta}_i$$
3. Evaluate the gradient of the loss function $\nabla J(\theta)$ given the ensemble's parameter estimate:
$$\nabla J(\theta) = \nabla J(\theta_i) = [\frac{\delta \nabla J(\theta)}{\delta \nabla J(\theta_i)}]_{\theta=\hat{\theta}^t}$$
4. Calculate the new incremental parameter estimate $\hat{\theta}_t \leftarrow -\nabla J(\theta)$
5. Add the new estimate $\hat{\theta}_t$ to the ensemble

The difference between conventional machine learning techniques and boosting algorithms is that the optimization is held out in the function space instead of the input space. The function space is the space of all possible functions that can be used to model the relationship between the input features and the target variable. Boosting methods parameterize the function estimate $\hat{f}$ in an additive functional form, meaning that instead of directly optimizing the parameters in the input space, an iterative construction of a series of base learners (e.g. single decision trees) is performed and combined to form a stronger model.

$$\hat{f}(x) = \hat{f}^M(x) = \sum_{i=0}^{N} \hat{f}_i(x) \tag{A.2}$$

In A.2, M represents the number of iterations, $\hat{f}_0$ is the initial guess and $(f_i)_{i=1}^M$ are the function increments, or in other words the "boosts". In order to parameterize the family of functions used in the ensemble $\hat{f}$, a base learner function $h(x, \theta)$ is introduced, where x represents the input features and θ represents the parameters of the base learner function. By parameterizing the base-learner functions as $h(x, \theta)$, a set of parameters, θ is introduced that can be learned or optimized to capture specific patterns or relationships in the data. These parameters control the behavior of the base-learner function and determine its ability to fit the training data. In this research, the base-learner function is a single decision tree. However, there are various families of base learners one can choose from.

A "greedy stagewise" approach can be implemented, which is an iterative process of

incrementing the function estimate $\hat{f}_t$ by adding a new base learner at each iteration. The aim is to minimize the loss function.

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t) \tag{A.3}$$

In equation A.3, the function estimate at the t^{th} iteration $\hat{f}_t$, is updated by adding a scaled version of the new base-learner function $h(x, \theta_t)$. The scale factor ρ_t is the learning rate, which determines the contribution of the new base-learner to the overall ensemble.

$$(\rho_t, \theta_t) = argmin_{\rho,\theta} \sum_{i=1}^{N} \psi(y_i, \hat{f}_{t-1}) + \rho h(x_i, \theta) \tag{A.4}$$

Equation A.4 represents the optimization rule used to determine the optimal learning rate ρ_t and the optimal parameters for θ_t for the base-learner at the t^th iteration. This equation aims to find the optimal combination of values for ρ and θ to minimize the loss function over the training data. The boosting algorithm gradually improves the overall ensemble's performance by reducing the training error by updating the function estimate at each iteration using the new base learner with the optimal learning rate and parameters. The iterative nature of the approach allows the model to focus on the training examples that are challenging or have high errors, leading to better generalization and improved predictive accuracy.

The gradient boosting algorithm allows flexibility in choosing the loss function and the base-learner model. However, given these choices, obtaining the solution to the parameter estimates can be difficult. Therefore, the gradient boosting algorithm introduces a strategy to select the new base-learner function $h(x, \theta_t)$ that is most aligned with the negative gradient of the loss function $(g_t(x_i))_{i=1}^N$ along the observed data, see equation A.5.

$$g_t(x) = E_y[\frac{\delta\psi(y, f(x))}{\delta f(x)}]_{f(x)=\hat{f}^{t-1}(x)} \tag{A.5}$$

Instead of searching for the general solution for the boost increment in the function space, the gradient boosting algorithm simplifies the process by choosing the new function increment to be the most correlated with $g_t(x)$, which is the negative partial derivative of the loss function with respect to the predicted function value $f(x)$. This replaces the challenging optimization task with a classic least-squares minimization problem; see equation A.6.

$$(\rho_t, \theta_t) = argmin_{\rho,\theta} \sum_{i=1}^{N} [-g_t(x_i) + (x_i, \theta)]^2 \tag{A.6}$$

The general form of the gradient boosting algorithm as originally proposed by Friedman (2001) can be found in Table A.1.

Table A.1: Retrieved from Natekin and Knoll (2013)

Gradient Boosting Algorithm by Friedman (2001)	
inputs:	- input data $(x, y)_{i=1}^{N}$
	- number of iterations M
	- choice of the loss function $\psi(y, f)$
	- choice of the base-learner model $h(x, \theta)$
Algorithm	1. initialize $\hat{f}_0$ with a constant
	2. for t = 1 to M do:
	3. compute the negative gradient $g_t(x)$
	4. fit a new base-learner function $h(x, \theta_t)$
	5. find the best gradient descent step-size ρ_t
	$\rho_t = argmin_\rho \sum_{i=1}^{N} \psi[y_i, \hat{f}_{t-1}(x_i) + (x_i, \theta_t)]$
	6. update the function estimate: $\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$
	7. end for: stop criteria

This research uses two advanced gradient-boosting decision tree methods: Extreme Gradient Boosting (XGBoost) and LightGBM. The theoretical background for these models is given in the next sections.

E.2. Extreme Gradient Boosting (XGBoost)

XGBoost is a highly scalable boosting system widely used in machine learning implementations. The following theory is based on the paper by Al Daoud (2019). The difference between XGBoost and other gradient-boosting algorithms is the use of a new regularization technique, which is applied by adding a new term to the loss function.

$$\psi(y, f(x)) = \sum_{i=1}^{N} \psi(y_i, f(x_i)) + \sum_{m=1}^{M} \Omega(\delta_m) \tag{A.7}$$

with

$$\Omega(\delta) = \alpha|\delta| + 0.5\beta||w||^2$$

Here α and β are hyperparameters that control the strength of regularization. The term δ represents the number of branches, w represents the value of each leaf, and Ω is the regularization function. XGBoost introduces a balance between fitting the training data and controlling model complexity by adding the regularization term to the loss function. This helps prevent overfitting and improves the model's ability to generalize well to unseen data. The XGBoost uses a different gain function.

$$Gain = \frac{1}{2}[\frac{G_L^2}{H_L + \beta} + \frac{G_R^2}{H_R + \beta} + \frac{(G_R + G_L)^2}{H_R + H_L + \beta}] - \alpha \tag{A.8}$$

Here, $Gj = \sum_{i \in I_j} = g_i$, which represents the sum of the gradient data point i in a particular tree node. I_j represents the set of indices of data points that belong to a specific tree node

j. The gradient g_i is the partial derivative of the loss function with respect to the predicted value of the data point. $H_j = \sum_{i \in I_j} = h_i$ represents the sum of the Hessians (h_i) over all the data points in the same tree node. Hessians (h_i) is the second derivative of the loss function with respect to the predicted value for the data point. This function captures the curvature of the loss function. The *Gain* is a score that quantifies the improvement in the loss function achieved by adding a new split to a tree node. Equation A.8, is the division of the squared sum of gradients by the sum of Hessians minus a regularization term α. The *Gain* score is then used to evaluate potential splits in a tree node and select the split that maximizes the score. The gradients and Hessians are based on the specific loss function chosen. This research uses the Root Mean Squared Error (RMSE), a common loss function in regression tasks.

E.3. LightGBM

LightGBM was created by a team from Microsoft to overcome previous limits of GBDTs in the efficiency and scalability of the implementation when features dimension is high and data sizes are large (Ke et al., 2017). Compared to XGBoost, LightGBM does not apply a level-wise tree growth but a leaf-wise growth. Thus, instead of checking all of the previous leaves for each new leaf, the decision trees are grown leaf-wise Al Daoud (2019). There are two novel techniques used *Gradient-Based One-Side Sampling* (GOSS) and *Exclusive Feature Building* (EFB). Tabular explanations of the GOSS and EFB algorithms by Ke et al. (2017) are displayed in A.2 and A.3, respectively.

Table A.2: Overview GOSS algorithm as defined by Ke et al. (2017)

Algorithm: Gradient-based One-Side Sampling	
Input:	I: Training data d: Number of iterations (number of trees to build) a: Sampling ratio of large gradient data b: Sampling ratio of small gradient data loss: Loss function L: Weak learner (single decision tree)
Initialize:	models: Empty list to store the trained weak learner models (trees) fact: Computed as 1 - (a/b) topN: Number of instances to keep with large gradients, calculated as a * len(I) randN: Number of instances to randomly pick with small gradients, calculated as b * len(I)
Algorithm	For each iteration i from 1 to d: 1. Make predictions on the training data using the ensemble of models already trained. 2. Compute the gradients (g) by calculating the loss between the predictions and the true value in the training data (I). 3. Set initial instance weights w to one (equal weight for all instances). 4. Sort the instances based on the absolute values of gradients (g) and store the sorted indices 5. Select the top N indices from the sorted list to form the top set. 6. Randomly pick N indices from the remaining sorted instances to form the random set. 7. Create the used set by combining the top set and the random set. 8.Scale down the weights of instances in the random set by multiplying them by \frac{1-a}{b} (this gives higher importance to instances with larger gradients) 9. Train a new weak learner using the instances and corresponding weights from the used set with their gradients negated (this aims to fit the model to the errors made by the ensemble of models so far) 10. Add the new weak learner to the list of trained models (models) Final output: an ensemble of trained models stored in models' list.

In general GBDTs, no weights are given to data instances, although different data instances with different gradients affect the computation of the information gain. The gradient represents how much the current model gets the prediction wrong. Instances with larger gradients, i.e. instances for which the model finds it harder to predict, are more important for improving its performance. Therefore, in order to make the training process more efficient, instead of

uniformly randomly dropping the instances, one should drop the instances with small gradients, as larger gradients will contribute more to the information gain. The GOSS algorithm keeps the instances with large gradients (i.e., under-trained instances) and randomly drops the instances with small gradients defined by some threshold or percentile. In doing so, the GOSS algorithm ensures that important instances contributing significantly to the model's improvement are retained, leading to more accurate gain estimation for tree construction than uniform random sampling. Furthermore, EFB is a technique used to reduce the number of features in a dataset while preserving the information they carry. Often, datasets contain many features, of which many are sparse (i.e. many zero-values), especially one-hot-encoded features. EFB bundles together sparse features, thereby reducing the number of effective features without losing much information. Then, a greedy algorithm is applied to efficiently group the exclusive features together, reducing feature dimensionality while maintaining the most critical information.

Table A.3: Overview EFB algorithm as defined by Ke et al. (2017)

Algorithm: Merge Exclusive Features	
Input:	numData = total number of data instances. F: One bundle of exclusive features. Each feature in F has its own set of bins. binRanges = list to store the ranges of bins for different features. totalBin = keeps track of the total number of bins. newBin = new bin that will merge the bins of exclusive features.
Algorithm	Step 1: initalize binRanges with a single element zero, also set totalBin to zero Step 2: For each feature f in the bundle of exclusive features F do: Increment totalBin by the number of bins in feature f (f.numBin). Append the updated totalBin to the binRanges list. Step 3: create new bin newBin with size numData (number of data instances) Initialize all elements of newBin to zero Step 4: Nested loop: Iterate over each data instance index i from 1 to numData. For each feature F[j] in the bundle of exclusive features: - If the bin value of F[j] for the current data instance i is non-zero, update the corresponding newBin
Output	Return newBin, which represents the merged bin containing exclusive features. Return binRanges, which contains the bin ranges for each feature after merging.

E.4. Hyperparamater tuning with Bayesian Optimization

Hyperparameter tuning is a crucial process in modeling, as it defines the model's overall performance. There are several ways to tune your hyperparameters: manual search, random search, grid search, and Bayesian Optimization. Manual search is tuning hyperparameters based on human experience. The best values for different parameters are chosen based on the knowledge and experience of the programmer. A trial-and-error process is followed with various configurations of hyperparameters. Grid search is a simple algorithm of hyperparameter optimization. Here, a range of input values for each parameter are selected, and all possible parameter combinations are tested. The method may be computationally expensive when many different parameter combinations must be tested on a large dataset. Random search resolves this problem by randomly selecting a subset of the parameters. These three methods are often inefficient as parameters are chosen based on evaluating previous results. Therefore, these methods often spend much time tuning parameter values, which can be considered "bad". Conversely, Bayesian Optimization is a probabilistic model-based technique used to find a minimum of a function. It requires fewer iterations than random search or grid search, is computationally less exhaustive, and yields better performance on test sets. The

method takes into account past evaluations when choosing the optimal set of hyperparameters. Bayesian optimization attempts to find a global optimum in a minimum number of steps. The optimization uses a surrogate model to approximate the objective function. Then, an acquisition function is used, which directs sampling to areas where an improvement over the current best observation is most likely (Bergstra et al., 2011, 2013).

Bayesian optimization falls in a class of optimization algorithms called Sequential Model-Based Optimization (SMBO). The algorithm uses previous observations of the loss function to determine the next (optimal) point to sample from. Whenever the true parameter function is too costly, the SMBO algorithm approximates f with a surrogate function $p(y|x)$, where y is the score and x is the hyperparameters. The surrogate model maps the hyperparameters to a probability distribution over the scores achieved by the model. Thus, based on past evaluations, the function serves as a proxy that captures the relationships between hyperparameters and model performance. It provides a probabilistic representation of how different hyperparameters are likely to affect the performance of the model. The point x^* that maximizes the surrogate function becomes the proposal for where the true parameter function should be evaluated. Instead of exhaustively exploring the entire search space, the algorithms use a combination of exploration and exploitation strategies to select hyperparameters that are expected to perform well based on the information provided by the surrogate model. By iteratively evaluating and updating the surrogate model, Bayesian optimization algorithms guide the search process toward promising regions of the hyperparameter space. This approach efficiently explores the space, gradually narrowing down to the optimal hyperparameters and minimizing the number of evaluations required on the actual objective function. Two well-known surrogate models are employed: Gaussian Processes and Tree-structured Parzen Estimator (TPE). The current research uses the TPE model, which has shown effectiveness in high-dimensional and discrete search spaces, where Gaussian-based methods might struggle. Moreover, the TPE algorithm uses a tree-structured approach (Bergstra et al., 2011, 2013).

TPE applies Bayes' rule instead of directly representing the surrogate model $p(y|x) = \frac{p(y|x) \cdot p(y)}{p(y)}$. $p(y|x)$ represents the probability of the hyperparameters given the objective function score. It models the distribution of hyperparameters conditioned on the scores achieved by evaluating those hyperparameters. The distribution is divided into two parts based on a threshold value y^*.

$$p(y|x) = \begin{cases} (l(x)) & \text{if } y < y^* \\ (g(x)) & \text{if } y \geq y^* \end{cases}$$

if y is less than the threshold y^*, the density function is denoted as $l(x)$, representing the lower scores. When the score y is greater than the threshold y^*, the density function is denoted as $g(x)$. The TPE method then samples new hyperparameters from the $l(x)$ distribution, as these hyperparameters are more likely to lead to improved performance based on the observed scores. In such a manner, the TPE focuses the search on hyperparameter regions that are most promising in achieving lower scores on the objective function. TPE effectively explores the search space to find (near) optimal solutions. Intuitively, TPE takes advantage of the

knowledge gained from previous evaluations to draw hyperparameter samples more likely to improve performance, prioritizing exploration in regions associated with better scores (Bergstra et al., 2011, 2013).

Bayesian optimization with the TPE method is very fast and accurate. However, the results of the best hyperparameters might change in different runs of tuning them. Bayesian optimization methods explore the parameter space by sampling different combinations of hyperparameters and evaluating their performance. Since the optimization algorithm combines exploration and exploitation strategies, it may converge to different optimal points in different runs. The algorithm aims to find the best set of hyperparameters that minimizes the loss function. The optimal parameters may vary due to various factors, including the random initialization of the search process, the number of evaluations performed, the random search space, and the complexity of the problem. Therefore, k-fold cross-validation is applied. K-fold Cross-Validation assesses the performance of machine learning models by mitigating issues related to overfitting. It provides are more reliable estimate of the model's general performance. The original dataset is split into k folds, and each fold is treated as a test set once, while the other $k-1$ sets are used as the training set. The model is then trained on the $k-1$ folds, and its performance is evaluated on the k_{th} fold, with the performance metric (RMSE). This process is iterated 1 to k times. The performance metrics over all k iterations are averaged to assess the model's performance overall. The set of hyperparameters that consistently yields the best performance is determined by evaluating the model's performance across the different folds and iterations.

E.5. Permutation-Based Feature Importance Algorithm

Permutation-based feature importance algorithm is an agnostic model inspection technique that can be used for any model. In general, permutation feature importance measures the change in the model error after permuting one feature's value, i.e. randomly shuffling the feature's value. Thereby the relationship between the feature and the target variable is broken, and the drop in the model score indicates the model's dependence on the particular feature. The general algorithm is outlined below.

- inputs: fitted predictive model m, tabular dataset D
- Compute the reference score s of the model m on data D
- For each feature j in dataset D:
    - For each repetition k in $1, ..., K$:
        1. Randomly shuffle column j in D to generate a corrupted version of the data named $\tilde{D}_{k,j}$
        2. Compute the scores $s_{k,j}$ of the model m on corrupted data $\tilde{D}_{k,j}$
    - Compute importance i_j for feature f_j defined as:

$$i_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j} \tag{A.9}$$

The advantages of permutation-based feature importance are that it is model agnostic, does not require the model to be retrained, and captures variable interactions by randomly shuffling variables. However, a disadvantage of the method is that permutation importance might report lower importance scores for highly correlated variables. Moreover, poor performance can result from over-fit models. However, as GBDTs are created to prevent overfitting, this is assumed not to be a problem. Specifically, for decision trees, the feature importance is the features that contribute most to an increase in the *"Gain"* algorithm used in the model. Often, the feature importance prefers features with many possible values (i.e., high cardinality), which potentially skew the feature importance and provide counter-intuitive results. This must be taken into consideration when exploring the results (Breiman, 2001).

Appendix F

Explanation HGBoost Package

For the XGBoost, the function **def xgboost_reg(self, **)** is used. This is the XGBoost regression model that is applied; it contains the *eval_metric* set to RMSE, the *greater_is_better*, which is set to False, as the RMSE is optimal when minimized, the *params*, which represents the search space and is set to default at the start. This function starts the regression, sets the method to *xgb_reg*, and then invokes the regression function. It returns the results. Similarly, the LightGBM model has a function **def lightgbm_reg(self, **)**, which is the regression model applied, it contains the *eval_metric* set to RMSE, the *greater_is_better*, which is set to False and the *params* relevant to the LightGBM model are set to the default values. This function starts the regression, sets the method to *lgb_reg*, and then invokes the regression function.

F.0.1. Explanation general functions

Below, the general functions used in each model class are explained.

1. **def_init_(self,**)**: This function initializes the hgboost class and the user-defined parameters, which are the following:

    - *max_eval* = the maximum number of evaluations that the hyperopt optimization performs. Default = 250, set to 10000.
    - *threshold* = classification threshold, in case of two class models, this is 0.5 (default=0.5)
    - *cv* = cross-validation, the number of folds (default=5)
    - *top_cv_evals* = the number of top best-performing models that are evaluated in cross-validation. If set to none, each iteration *max_eval* is tested; if set to zero, cross-validation is not performed. Default=10, set to 100.
    - *test_size* = percentage split for the test set based on the total dataset (default=0.2)
    - *val_size* = percentage split for the validation set based on the total set. This part is set aside as unseen data and determines the model performance at the end (default=0.2)
    - *is_unbalance* = not applicable as it is control of balance for unbalanced classes for classification (default =True)
    - *random_state* = Fix the random state for the validation and test sets. Note that it is not used for cross-validation. Default=None, set to zero.

- *n_jobs* = The number of jobs to run in parallel for fit. (default = -1, all processors)
- *gpu* = Computing using either GPU or CPU. Note that GPU usage is not very well supported because various optimizations are performed during training, testing, and cross-validation. True: Use GPU. False: Use CPU. (Default = False)
- *verbose* = prints progress to the screen. 0: None, 1: ERROR, 2: WARN, 3: INFO, 4: DEBUG, 5: TRACE (default = 3)
- *algo* = surrogate algorithm used to efficiently explore the random domain space. Set to tpe.suggest, i.e. Tree Parzen Estimator.

2. **def _regression(self,**)**: This function retrieves the parameters' search space by calling the function *def_get_params()*. Moreover, it retrieves the best model fitted on the validation set and its results via the *def_fit()* function.
3. **def _get_params(**):** this function is outside the hgboost class in which the search space for the hyperparameters is defined using Hyperopt functions, and other relevant parameters are defined, such as the *early_stopping_round*, *three_method* and the *predictor*.
4. **def_fit(self,**):** The function first checks the correctness of the input data with the *def_check_input()* function. The dataset is then split into a train/test set (80%) and an independent validation set (20%). Then, the results and model are retrieved from the *def_HPOpt()* function. This model and its results are then fitted on the validation set, and the results are returned to the regression function, which ends the optimization.
5. **def_HPOpt(self,**)**: The function performs a Bayesian optimization over the search space using the Hyperopt python package and uses cross-validation to ensure robust values for the hyperparameters. First, the *def xgb_reg()* or *def lgb_reg()* function is retrieved. This function initializes an XGBRegressor or LGBMRegressor with the corresponding model parameters from the search space. Then, the train and test set is split using the 80/20 rule. The *fmin* function from the hyperopt library is used to find the optimal parameters using the Bayesian optimization approach. The inputs of this function are the objective function to optimize (XGBregressor, LGBMRegressor), the search space for the hyperparameters (space as retrieved from the *def_get_params()* function), the algorithm used (Tree Parzen Estimator TPE), the maximum number of evaluations the optimization algorithm should perform (*self.max_eval*), and finally, the *Trials()* object which keeps track of the optimization process and stores the results of each evaluation. The results of each evaluation are stored in a dataframe using the *def_to_df()* function, which includes storing the best parameter values and the loss score (RMSE). The Hyperopt function then invokes the *def_cv()* function, which performs cross-validation over the top N best models found by the Bayesian optimization. After the cross-validation, the function creates a basic model using the default parameter values. This model is compared to the model which results from the best hyperparameters found by the hyperopt and cross-validation. The default and optimal model are evaluated on the test set, with the *def_eval()* function. The model parameters and results are stored and returned to the *def_fit()* function.
6. **def_eval(self,**)**: The function evaluates the fitted model. It makes a prediction of y based on the given values of x. The function can be used for an evaluation of the train,

test, and validation set. The loss is calculated after fitting the model and predicting y (RMSE). Furthermore, other performance measures have been personally added, including the MAE, MSE, MAPE, and R^2. The evaluated model, the results, and the performance measures are returned to the *def_HPOpt()* function.

7. **def_cv(self,**):** The function performs a k-fold cross-validation over the top N models resulting from the hyperparameter optimization (models with the lowest RMSE). The cross-validation first splits the train and test set with 80/20 rule and then retrieves the output of the *def_train_model()* function. after the k-fold CV is completed, the mean and standard deviation of the losses across the k-folds are computed and stored in *results_summary* dataframe. The hyperparameters of the best model based on the minimum loss-mean resulting from the cross-validation are retrieved and stored as the *best_params*. The function returns the best-performing model object, the *results_summary* dataframe, and the best parameters resulting from the cross-validation to the *def_HPOpt()* function.
8. **def_train_model(self, **):** the function evaluates the models for the cross-validation. It fits each model in the cross-validation on the training sets and then evaluates its performance on the test set with the *def_eval()* function. The function returns the output (dictionary with model details, loss, evaluation time, and status) and the evaluation results.

Appendix G

Extreme Gradient Boosting Model

G.1. Explanation (hyper) parameters XGBoost

In this section, the hyperparameters for the XGBoost are defined, based on the work by Banerjee (2020) and dlmc XGBoost (2022).

G.1.1. General parameters

The general parameters guide the overall functioning of the model, and the three hyperparameters are *booster*, *verbosity*, and *nthread*. There are three types of boosters: *gbtree*, *dart*, and *gblinear*. The two former are tree-based models, while the latter uses linear functions. As the data is not normally distributed and does not exhibit linear relationships with the dependent variable *gblinear* is not used. The XGBoost algorithm combines many regression trees with a small learning rate; early-added trees are more significant than late-added trees. Therefore, a new method in which a dropout technique is used based on neural network methods, which drops trees to solve over-fitting. Generally, *dart* differs from *gbtree* by removing trees during each round of boosting. *gbtree* is the most commonly used booster and default setting.

Verbosity can be set from a value of zero to three. Each level provides more details about the progress of the model throughout the training process. Verbosity is useful for debugging, finding errors in the model, and understanding model progression. However, it slows down the training process. The default is 1.

- verbose = 0: no text is displayed, silent modeling
- verbose = 1: displays the computation time of each fold and the parameter candidate
- verbose = 2: displays the computation time of each fold, the parameter candidate, and the score
- verbose = 3: displays all of the above and the candidate parameter indexes and the computation time

nthread is the number of parallel threads used to run the model. The number of cores one wants to use can be entered into the system. By default, the maximum number of threads available is used. Other general parameters are set automatically and do not require user interference. $Validate\_parameters$ is set to true for the Python interface, which will perform a validation of

input parameters to check whether a parameter is used or not. $Disable\_default\_eval\_metric$ default is set to false and serves as a flag to disable the default metric. $num\_feature$ is set automatically by the XGBoost model.

G.1.2. Booster parameters

eta or *learning rate* is a parameter that slows down learning to prevent over-fitting. It is the step size shrinkage used when updating the tree. After each boosting step, the weights of the new features are gathered, and the learning rate shrinks the feature's weights to make the boosting process more conservative. The range is between 0 and 1. Larger rates introduce faster learning and, thus, a larger probability of over-fitting. On the other hand, lower values allow for smaller steps, thus better optimization. However, smaller steps simultaneously increase computational time. By default, the learning rate is set to 0.3. Typical final values are often between 0.01 and 0.2. As the model is very efficient, larger learning rates than the default setting are not used.

Each node split must result in a positive reduction in the loss function. The *gamma* parameter, also known as the *min_split_loss* specifies the minimum loss reduction required to make a split. A larger gamma makes the model more conservative, i.e. higher values reduce overfitting. However, this comes at the expense of lower granularity. The values can range from zero to infinity. However, a value of 20 is considered extremely high and should only be used when using high *max_depth*. In this research, the maximum depth of the tree cannot be too high as the interpretability of the tree is important. Therefore, the gamma values are continuous in a range of 0 to 10.

max_depth is the maximum depth of the tree. The parameter controls overfitting. Higher depths result in learning too much about the relations in a particular subsample of the data. Larger values thus lead to overfitting and also increasing computation time. The possible range is from zero to infinity. However, zero is only accepted when the *tree_method* is set as *hist*, as a loss-guided growing policy is applied. Typical values are between 3 and 10; the default setting is 6. The algorithm is started at 1 and increased in increments of one until a maximum of 22.

min_child_weight is the minimum sum of weights of all observations required in a child. The default is set to 1, meaning nodes can split data until only one observation is left. Low values allow for the reflection of specific cases. However, it may also lead to over-fitting. Conversely, high values prevent the algorithm from learning relations that might be highly specific to the sub-sample but can lead to underfitting. Generally, the *min_child_weight* ensures that the number of observations in the leaves does not fall below a specified threshold. The possible range can vary from zero to infinity. In the research, we test all integers between 1 and 10.

subsample is the fraction of instances randomly sampled for each tree of the training observations. Subsampling occurs every boosting iteration and prevents overfitting. Lower values make for a more conservative algorithm. The range is from zero to one. The default setting is 1. Typical values are between 0.5 and 1. *colsample_bytree*, *colsample_bylevel*, *colsample_bynode* are part of the family of parameters for subsampling columns. The default is always 1, and the parameters all have a range between 0 and 1. *colsample_bytree* reflects the

proportion of randomly selected features (columns). This occurs each time a tree is constructed. *colsample_bylevel* is the subsample ratio of columns for each level. This fraction is applied once every new depth level is reached in the tree. The *colsample_bynode* is the subsample ratio of columns for each node (i.e. split). By limiting the number of features for building each tree, different insights might be gained from the data. The parameters learn to optimize for the target variable using different features. However, as the data and the number of features are quite limited, the parameters are set to the default value of 1.

alpha and *lambda* are L1 and L2 regularization terms, respectively. Tuning the regularization parameters *lambda* and *alpha* can help reduce model complexity, reduce overfitting, and improve performance. L1 or lasso regression (*alpha)*) adds the absolute magnitude of the coefficients as a penalty term to the loss function. L2 or ridge regression adds the squared magnitude of the coefficient as the penalty term to the loss function. The key difference is that Lasso shrinks the less important features' coefficients to zero, which could remove some features altogether. L1 regularization works well for feature selection in case a huge number of them exist. This is not the case; therefore, it is chosen to set alpha to the default setting of 0. *lambda* values are set to continuous between 0 and 10.

tree_method When training boosted tree models, one can set two parameters in order to choose the algorithm. Namely, *updater* and *tree_method*. The *tree_method* has four built-in algorithms: *exact*, *approx*, *hist* and *gpu_hist*. The parameter *updater* provides a modular way to construct and modify trees. It is an advanced parameter that is often set automatically. Therefore, only the *tree_method* methods are considered here. The *exact* method iterates over all observations of the input data during each split-finding procedure. The *exact* method is more accurate, but the computation time is much higher. Whenever the dataset is too large, the *exact* method is unsuitable as it is slow and not scalable. Therefore, there are approximated training algorithms. The algorithms iterate through a gradient histogram for each node instead of the real dataset for these methods. The *approx* tree method is an approximate greedy algorithm using quantile sketch and gradient histogram. The *hist* tree model is a fast histogram-optimized approximate greedy algorithm, which uses additional performance improvements. The *gpu_hist* tree method is a GPU implementation of *hist* algorithms. When the *tree_method* is set to *hist*, a heuristic is used to choose the fastest tree model. This is chosen for the current model.

Other parameters which are not considered in the research are:

- *max_delta_step*: This parameter is often used in logistic regression when classes are extremely imbalanced and irrelevant in this context.
- *sampling_method* is the method used to sample the training instances. This is uniform by default. Thus, each training instance has an equal probability of being selected. Another method is gradient-based when the selection probability is proportional to the regularized absolute value of the gradients. However, this is only applicable when the *tree_method* is set to *gpu_hist*, which is not the case. So, by default, this parameter is set to a uniform.
- *scale_pos_weight* is a parameter that controls the balance of positive and negative weights.

The parameters are useful for imbalanced classes in classification tasks. Therefore, not relevant to this research's model.

- *max_leaves*, *max_bin*, *grow_policy*, *process_type*, *refresh_leaf*, *monotone_constraints*, *interaction_constraints*, and *multi_strategy* are irrelevant parameters for the current research.

G.1.3. Learning Task Parameters

The learning task parameters define the optimization objective, i.e. the metric to be calculated and optimized at each step. The parameters specify the learning task and corresponding learning objective. The *objective* parameter is the loss function that must be minimized. Several objectives differ for classification and regression tasks. The most common for regression tasks is the *reg:squarederror*, which is also the default objective. The *objective* parameter is set to the default value. Furthermore, the *eval_metric* parameter is defined. This is the metric used for validation of the data and defaults according to the objective. In this case, the objective is regression. Thus, the default is the root mean square error (RMSE). Finally, the *seed* parameters must be set. This is a number used to initialize a pseudorandom number generator. The seed or random state is set to a default value of zero.

G.2. Optimal Hyperparameters

The hyperparameter values belonging to the best model are listed in Table A.1.

Table A.1: Optimal hyperparameters XGBoost

parameters	B2S	INBOUND	MDA	BBXD
gamma	3.85	7.27	8.94	9.13
learning_rate	0.21	0.19	0.20	0.28
max_depth	15	2	15	24
min_child_weight	8	6	1	5
n_estimators	105	95	55	120
reg_lambda	0.02	0.87	5.00	5.26
subsample	0.63	1.00	0.51	0.94

Appendix H

LightGBM model

H.1. Explanation (hyper) parameters LightGBM

This section explains the parameters corresponding to the LightGBM model.

H.1.1. Core Parameters

LightGBM has several core parameters, which must be set before implementing the model.

- **task:** controls the task of the function, which can be either *train*, *predict*, *convert_model*,*refit*. Set to default *train*.
- **objective:** controls the output type of the model, e.g., regression, binary classification, multi-classification, etc. In this research, the goal is to predict based on the regression method. Therefore, the parameter is set to the default *regression*, which has alias *L2*.
- **boosting:** the boosting algorithm that is used, which can be either *gbdt*, *rf*, or *dart*. The traditional Gradient Boosting Decision Tree yielded the best results during exploration. Therefore, this parameter is set to the default boosting algorithm *gbdt*.
- **data_sample_strategy:** either *bagging*, which is randomly bagging sampling, or *goss*, which is Gradient-based One-Side Sampling, a leaf-wise growth method. The default setting is *bagging*. This setting is only effective when *bagging_freq ¿ 0* and $bagging\_fraction < 1$. The *bagging_freq* parameter controls the frequency of bagging. The default setting is zero, meaning disabling bagging. Any other integer value k means that bagging is performed at every k iteration. The *bagging_fraction* parameter controls the fraction of data used in each bagging round. The default setting is equal to 1.
- **n_estimators:** the number of boosting iterations. (default = 100). After initial exploration, the number of estimators is tested on a range of 20 to 200, in increments of 5.
- **learning_rate:** This parameter controls the learning rate to prevent over-fitting. It is the step size shrinkage used when updating the tree. After each boosting step, the weights of the new features are gathered, and the learning rate shrinks the feature's weights to make

the boosting process more conservative. (default = 0.1). Continuous values between 0.01 and 0.30 are tested.

- **num_leaves:** the maximum number of leaves in one tree. This is the main parameter controlling the complexity of the tree model. The *num_leaves* should be smaller than 2^{max} to prevent overfitting. (default = 31). After initial exploration, the number of estimators is tested on a range of 20 to 200, in increments of 5.

- **tree_learner:** parameter for distributed learning, one can use multiple machines to produce a single model. The default is serial, which runs a single machine tree learner.

- **n_threads:** the number of parallel threads used to run the model. The number of cores one wants to use can be entered into the system. By default, the maximum number of threads available is used.

- **device_type:** device for tree-learning *gpu*, *cpu* or *cuda*. The default CPU is used because GPU usage is not well supported. After all, various optimizations are performed during training, testing, and cross-validation.

- **seed or random_state:** number to initialize pseudorandom number generator. The random_state is set to zero.

H.1.2. Learning Control Parameters

The Learning Control Parameters are the main parameters to be tuned in order to get the best performance and prevent overfitting. The most important parameters are discussed and explained. Features important for classification tasks, categorical data, or applied when using *dart* boosting algorithm or *CLI* implementation are not discussed.

- **max_depth:** the maximum depth of the tree. The parameter controls overfitting when the data is small. The default is -1, which means no limit. All integers between 1 and 30 are tested.

- **min_data_in_leaf:** the minimal number of data in one leaf. The parameter controls overfitting based on an approximation based on the Hessian. The default value is 20. All integers between 1 and 30 are tested.

- **min_child_weight:** also known as minimum sum hessian in one leaf, which can be used to prevent overfitting. Low values allow for the reflection of specific cases. However, it may also lead to over-fitting. Conversely, high values prevent the algorithm to learn relations that might be highly specific to the subsample, but it can also lead to underfitting. All integers between 1 and 30 are tested.

- **bagging_fraction:** a.k.a subsample is the fraction of instances randomly sampled for each tree. subsampling occurs every boosting iteration and prevents overfitting. The default setting is 1. The feature can prevent overfitting and speed up training. Continuous

values between 0.8 and 1 are tested. The *feature_fraction* (i.e., colsample_bytree) and *feature_fraction_bynode* (i.e. colsample_bytree) are part of subsample features. As the data and the number of features are quite limited, setting these terms to the default value of 1 has been chosen.

  - **feature_fraction:** fraction of features selected on each iteration (tree)
  - *feature_fraction_bynode:* fraction of features selected at each tree node.

- **early_stopping_round:** stops training if metric has not changed the last *early_stopping_rounds*. The parameter speeds up the model and ensures time is not wasted in areas already improved maximally. It is set to 100 to ensure no further optimization is possible.
- **alpha:** L1 or lasso regression adds the absolute magnitude of the coefficients as a penalty term to the loss function. L1 regularization works well for feature selection in case a huge number of them exist. This is not the case; therefore, it is chosen to set alpha to the default setting of 0.
- **lambda:** L2 or ridge regression adds the squared magnitude of the coefficient as the penalty term to the loss function. The key difference is that Lasso shrinks the less important features' coefficients to zero, which could remove some features altogether. Tuning the regularization parameters lambda may help reduce model complexity, reduce overfitting, and improve performance. After initial exploration, the range is set as a continuous value between zero and two.

H.2. Optimal Hyperparameters

The hyperparameter values belonging to the best model are listed in Table A.1.

Table A.1: Optimal hyperparameters LightGBM

parameters	B2S	INBOUND	MDA	BBXD
learning_rate	0.31	0.29	0.28	0.30
max_depth	11	15	27	2
min_child_weight	7	3	2	1
min_data_in_leaf	14	15	26	1
n_estimators	25	165	110	165
num_leaves	160	35	70	35
reg_lambda	1.10	0.29	2.00	0.00
subsample	0.85	0.97	0.96	0.81

Appendix I

Evaluation Additional Information

I.1. Overview feature interpretation methods

Partial Dependence Plots (PDP) shows the dependence between the target variable (labor productivity) and one (or two) feature(s) of interest, keeping the other input features constant. This way, the plot provides insight into the interaction between the target variable and the feature of interest. Equation A.1 represents the partial dependence function for regression.

$$\hat{f}_s(x_s) = E_{X_C}[\hat{f}(x_s, X_C)] = \int \hat{f}(x_s, X_C)\delta\mathbb{P}(X_C) \tag{A.1}$$

In this equation, x_s represents the features for which the partial dependence function is plotted. X_C are the remaining features in the model $\hat{f}$. The set S represents the features for which the effect on the prediction must be traced. The set consists of a maximum of two features for interpretability reasons. The partial dependence function marginalizes the model's $\hat{f}$ output over the distribution of the features in the set C (remaining features) to show the relationship between the features in set S and the predicted value. The partial function $\hat{f}$ is estimated using the Monte Carlo method, which calculates the averages in the training data. The partial function $\hat{f}$ gives the average marginal effect of the features S on the output value (Molnar, 2022). The PDP importance is easily interpretable as the plots show how the average prediction in the data changes when a feature changes. However, the importance of the PDP captures only the main effect; it ignores possible interactions between feature values. It could be that other methods, such as permutation feature-based importance, find certain features important, but this is not reflected in the PDP score as the feature affects the prediction mainly through interaction with other features. Furthermore, unique feature values with only one instance are as important (i.e. given the same weight) as values with many instances. Another important assumption of the PDP is that the features in set C are not correlated to features in S. If this assumption is not adhered to, the averages calculated will be unreliable (Molnar, 2022).

The Accumulated Local Effects (ALE) plot does not assume independence of features. The method also describes how features influence the prediction of the model on average. The difference is that ALE calculates differences in predictions instead of the averages of predictions based on the conditional distribution of the features. The prediction function $\hat{f}$ is reduced by

averaging the effects of the other features. The function is described in equation A.2.

$$\begin{aligned}\hat{f}_{S,ALE}(x_S) &= \int_{z_0,S}^{x_S} E_{X_C|X_S=x_s}[\hat{f}^S(X_s, X_c)|X_S = z_S]\delta_{z_S} - \text{constant} \\ &= \int_{z_0,S}^{x_S}(\int_{x_C} \hat{f}^S(z_s, X_c)\delta\mathbb{P}(X_C|X_S = z_S)\delta)\delta_{z_S} - \text{constant}\end{aligned} \tag{A.2}$$

Given are the feature value(s) x_S, the features in set C, X_C, and the features in set S, X_S. First, the derivative with respect to the feature in set S is taken. This calculation gives the change in the prediction over a small interval of that feature while keeping the remaining features constant. The partial derivative isolates the effect of the feature in S and provides insights into how the feature impacts the model's prediction locally. Then, the local partial derivatives are integrated over the range of values of the features in set S. This way, the local effects over the entire feature's range are accumulated, which provides an estimate of the overall effect of the feature on the model's prediction. In general, ALE function calculates the changes in predictions and then accumulates these changes over the whole range. The ALE can also be used for the interaction effect between two features. In this case, the integration is performed over rectangular cells instead of intervals to accumulate the effects in two dimensions. The ALE uses the second-order effect of the features, which is the additional interaction effect after accounting for the main effects of the features. PDP, on the other hand, shows the total effect of two features (Molnar, 2022).

Using ALE plots has several advantages. Foremost, the plots do not require independence of features (unlike PDP) and can thus be used when features are correlated. Moreover, they are fast to compute, and their interpretation is clear. The plot provides insight into the relative effect of changing a feature in the prediction, conditional on a given value. Furthermore, the plots are centered around zero, making the interpretation preferable as the value at each point of the ALE plot is the difference from the mean prediction. However, the ALE plots the local effect. Thus, the interpretation of the effect can only be local, not over the whole range. Furthermore, the number of intervals chosen has an effect on the interpretability of the plot. Fewer intervals make the estimates more stable but might hide the complexity of the prediction model. The plot might be unstable with many local fluctuations when using a larger number of intervals. Finally, the second-order ALE estimates might be delicate to interpret as they show the additional effect, not the total effect. So, the technique is adequate for discovering and exploring interaction. However, when the aim is to interpret the effect, PDP are more useful (Molnar, 2022).