Exploring feasibility of AI-driven insights for decision making in an e-commerce environment

Msc. Complex Adaptive Systems

Dan Johansson

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Exploring feasibility of AI-driven insights for decision making in an e-commerce environment

DAN JOHANSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of physics
Chalmers University of Technology
Gothenburg, Sweden 2024

Exploring feasibility of AI-driven insights for decision making in an e-commerce environment
DAN JOHANSSON

Supervisor: Dan Bergman, Viskan Systems - Borås
Examiner: Mats Granath, Director - Complex Adaptive Systems Msc. Program

Master's Thesis 2024
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Data driven E-commerce decision making
Exploring E-commerce platform AI readiness using sales and campaign performance forecasting
DAN JOHANSSON
Department of Physics
Chalmers University of Technology

Abstract

This thesis explores the potential of data-driven decision-making and machine learning inferences within an e-commerce context, focusing on sales and campaign performance modeling at Viskan Systems. The research initiates by dissecting an existing database structure, identifying significant potentials for implementing machine learning methodologies despite encountering systemic data management challenges. These challenges include issues with overwriting campaign instances and handling campaign parameters, which could impede accurate data analysis and modeling.
The study implements and evaluates two distinct machine learning models: XGBoost and NeuralProphet. The XGBoost model reveals limitations in handling the wide variance in sales data, leading to a general trend of overestimation in smaller campaigns and underestimation in larger ones. The NeuralProphet model, employed for time series forecasting, shows a hierarchical structure in model performance, with the meta model yielding the most accurate results. Despite their limitations, these models highlight the feasibility of advanced data analytics in enhancing decision-making processes for Viskan Systems and its customers.
The thesis concludes by recommending strategic modifications to Viskan Systems' data infrastructure to facilitate the integration of data-driven approaches and machine learning. Such enhancements are deemed essential for the system's adaptation to sophisticated analytics, ensuring data integrity while improving compatibility with emerging technologies.

Keywords: Data-Driven Decision Making, Machine Learning, E-Commerce Analytics, Time Series Forecasting, Predictive Modeling, Business intelligence, campaign modeling

Acknowledgements

I extend my sincere appreciation to Mats Granath for his guidance, revision, and examination of this thesis. My gratitude also goes to Dan Bergman and Alexander Söderholm for their unwavering support, insightful feedback, and direction, which have been instrumental throughout the course of this project.
I am thankful to the entire team at Viskan Systems for creating a stimulating and welcoming environment. Their constant encouragement and assistance have significantly contributed to my journey. Special thanks to SCR for generously providing the data essential for this research, and to the SCR e-commerce manager for their engaging and insightful discussions.
Lastly, I am deeply grateful to Mahan Vahid, not only for introducing me to Viskan Systems but also for his friendship and mentorship. His support and guidance, at any hour, have been invaluable throughout the whole project.

Dan Johansson, Gothenburg, January 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AR	Auto regression
SCR	Swedish clothes retailer
NN	Neural network
R/D	Revenue-discount ratio

Contents

List of Figures

List of Tables

1
Introduction

Data-driven decision making plays a pivotal role in the operations of e-commerce businesses, serving as the tool for improving increased sales, understanding customers and managing operations. The leveraging of data can be a make-or-break factor for many enterprises in the e-commerce landscape. The significance of an e-commerce presence for business success has become even more pronounced in the post-COVID-19 era [1]. Across virtually all industries, the market share of e-commerce has experienced continuous growth since the inception of the internet, with the COVID-19 pandemic serving as a catalyst for exponential expansion. As of 2023, e-commerce accounts for approximately 21.8% of all retail sales, a remarkable surge from the 7.4% recorded in 2015. This substantial shift underscores the imperative for businesses to adapt swiftly to the dynamic and ever-evolving e-commerce sector.

Two common targets of data driven analysis and modeling are general sales forecasting and assessing the impacts of a marketing campaigns. Sales forecasting has common pitfalls, such as lack of domain knowledge, data quality and target complexity, often leading to mediocre results. Campaign performance prediction is also a complex task, partially due to the inherent nature of campaigns, which are typically either active or inactive. Direct comparisons between the two states is challenging, as only one condition can prevail at any given time. While it is theoretically possible to run campaigns for certain customers while excluding others from exposure, this luxury is rarely available when working with sales data in a real-world production environment. Consequently, the conventional approach to evaluating campaign performance revolves around the use of key performance metrics. These metrics commonly encompass parameters such as 'Visits,' 'Revenue per visitor,', 'Shopping cart abandonment rate.' and more[2]. There have been attempts to assess the influence of campaigns using other means such as combining time series prediction with a neural network (NN). The method presented by Yunpeng et al. [3] leveraged both customer behaviour through encoding of key statistics and time series prediction of campaign influence fed into a NN to predict real world campaign influence.

With the rapid global growth of data creation and storage [4], companies are naturally inclined to leverage this abundance of information. In 2021, the global big data market was valued at approximately 240 billion US dollars, and forecasts suggest that it is poised to double by 2026 and nearly triple by 2029, reaching an estimated total value of 655 billion US dollars [5]. Indicating significant value of these technologies and their potential effects.

The e-commerce sector is experiencing rapid expansion, accompanied by a parallel increase in data generation. This growing volume of data has the potential to drive

advancements in model complexity and applicability, similar to the developments observed in Large Language Models such as ChatGPT or the recommendation algorithms powering platforms like YouTube. These simultaneous trends in data growth and e-commerce market expansion indicate an evolving landscape where data-driven insights and models may play a transformative role in decision-making related to campaigns and sales strategies in general.
The thesis, data acquisition, data analysis and models implementations were developed in coordination with Viskan Systems. Viskan is an e-commerce platform provider for brands, retail, B2B and subscriptions, servicing companies across many industries.

1.1 Aim

The aim of this thesis is to perform an analysis of an e-commerce production database from a machine learning and data science applicability perspective. By analysis of historical sales data and evaluation of past campaign performance the goal is to implement models to perform general sales forecasting and campaign performance predictions.

1.2 Problem formulation

As the e-commerce market continues to expand its influence, there is a growing imperative to gain deeper insights into its intricate dynamics. In this era of data abundance, organizations are increasingly harnessing the power of available data to gain a competitive edge. However, despite the general availability of large datasets, leveraging this resource effectively remains a formidable challenge. This challenge arises from the inherent complexity of e-commerce platforms, their supporting systems and connections to third party actors such as payment providers. This results in a sophisticated structure where data aggregation, model implementation and integration comes with significant technological, legal and moral intricacies.
The central problem addressed by this study is bridging this knowledge gap. It aims to empower companies with the ability to make more informed decisions by providing insight into data science methodology and systemic choices which can enable an environment for machine learning models to be implemented. By doing so, this thesis seeks to equip businesses with the tools necessary to navigate the intricacies of e-commerce dynamics and enhance their competitiveness.

1.3 Limitations

This study is subject to several limitations that warrant consideration. Firstly, it is confined to a specific e-commerce sector, specifically focusing on clothing articles. The exclusion of other industries and datasets introduces potential limitations regarding the applicability and generalizability of the methods and resulting models proposed in this thesis.

Secondly, the dataset used for this analysis comprises strictly anonymized sales and campaign data, which lacks the granularity required to analyze user behavior and interactions with the campaigns. Consequently, the assessment of campaign performance relies solely on sales data and does not incorporate user behaviors or engagement metrics.
Additionally, the temporal scope of this study is restricted to the period spanning from January 1, 2018, to December 31, 2022. Given the dynamic nature of the e-commerce landscape, it is conceivable that new trends and consumer behaviors have emerged post-2022, which remain unexplored in this thesis. These temporal constraints necessitate cautious interpretation of the findings in light of potential shifts and developments within the e-commerce industry which emerged after the period of study.
During the mentioned period of study the considered retailer has also expanded its business to other countries, to remain consistent throughout the whole study period all data (purchases, campaigns) is restricted to only Swedish data.

1.4 Data introduction

The foundation of this thesis is the production SQL database obtained from a prominent Swedish clothing retailer, hereafter referred to as SCR, who has opted for anonymity. SCR is a prominent customer of Viskan System and have provided their data for this thesis. This database serves as a production database for SCR's e-commerce platform, encompassing all transactions and campaigns from mid-2017 to the present time of writing. To uphold privacy and data protection standards, rigorous anonymization processes were applied to remove customer-specific information and other sensitive data prior to its usage in this study.
Given its origin in a production environment, this database is rich in information, although not all aspects are directly pertinent to the objectives of this thesis. Notable elements within the dataset include detailed purchase records, including individual product prices, applied discounts, and the influence of various campaigns on purchases and products themselves. While a full enumeration of the database's contents may prove exhaustive, pertinent information will be introduced contextually and as needed throughout the course of this study.
After reducing the dataset according to the limitations mentioned in the previous section a total of 2,248,905 unique purchases were found, accounting for 1,201,269,235 SEK in generated income through the sale of 11,854,160 products.

1.5 Thesis structure

This thesis will start with a chapter on data analysis to provide a deeper understanding of SCR's database, its features, dynamics and shortcomings. This will allow us to make an informed choice of machine learning algorithms for modeling the data's intrinsic behaviours, such as sales forecasting and campaign performances. Subsequent chapters provide a theoretical foundation for the chosen models, NeuralProphet and XGBoost, followed by a detailed discussion of their implementation.

The model's performance will then be explored in the results and discussion chapters as well as discussing the potential systemic improvements that could enable this type of analysis to be better and easier in the future. Some additional material and topics that did not fit in the main body of the text will be provided as appendix instead for reader completeness.

2
Data analysis

This chapter will act as a prologue to the theory and method chapters in the sense that it will introduce some important data analysis concepts and theories but also provide some important insights of sales and campaign dynamics in the data set. These insights will steer the direction by having a direct effect on what theory and methods will be discussed in the following chapters.
To have a solid understanding of the purchase and campaign behaviours found within this dataset a thorough analysis needs to be performed. This analysis will serve as a foundation for what type of modeling approach should be taken for modeling the sales and campaign dynamics found within this system. This chapter will start off with a section covering general sales analysis of all recorded sales within the database, this includes sales both related and unrelated to a campaign. Thereafter a look into the behaviours of the campaigns and their related metrics, this will hopefully provide a deeper understanding of what data features are relevant to predicting campaign performances. Both of the aforementioned sections will also include what filtering and cleaning steps were performed on the dataset.
In summary this chapter is aimed to enrich us with key information to allow us to make well-informed decisions regarding the continuing analysis and model construction.

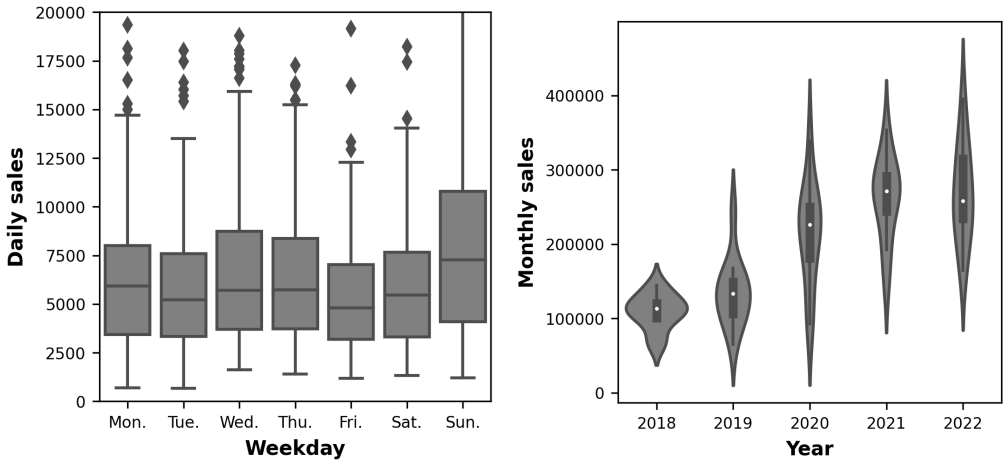
2.1 General sales analysis

In retail, particularly in clothing, it is reasonable to expect some degree of seasonality in sales patterns, influenced by the nature of products sold. For instance, categories like jackets and swimwear are likely to show increased sales during fall and summer, respectively. A basic introduction to seasonality including some examples can be found in appendix A. To explore potential seasonal trends in SCR's sales data, various plots in figure 2.1 display the quantity of products sold over different time frames.
Firstly, weekday sales behavior (Figure 2.1a) reveals a consistent pattern from Monday to Saturday, with a median of approximately 5-6 thousand units sold daily. Sundays, however, demonstrate a higher median of around 7,500 units and a broader quartile spread, suggesting a weekly seasonality in the data.
Secondly, the monthly sales per year Figure 2.1b indicate significant year-over-year variability, both in median sales and month-to-month fluctuations. This trend might reflect evolving customer purchasing behaviors and SCR's overall sales growth.
Further, the analysis of day-of-the-month sales (Figure 2.1c) and monthly sales

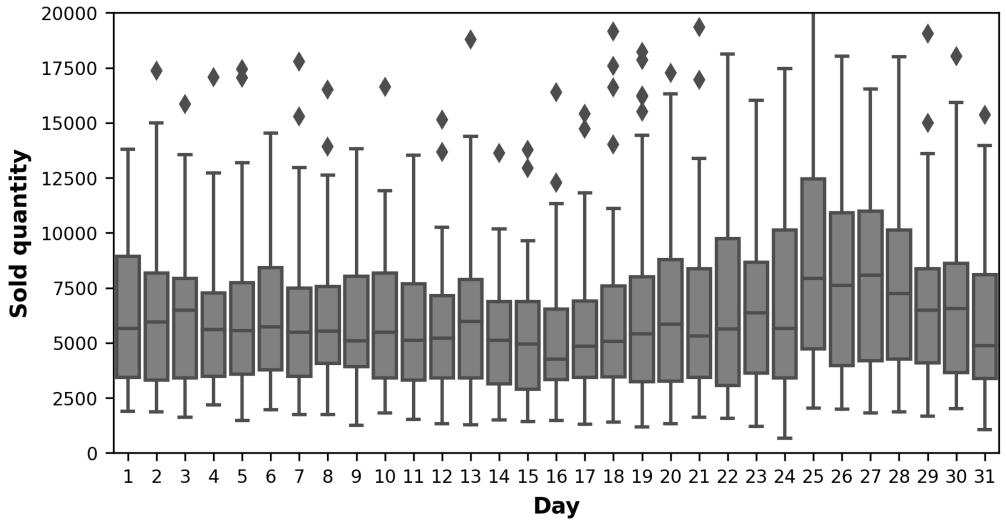
(Figure 2.1d) reveals additional insights. While no pronounced monthly seasonality is evident, an uptick in sales around the 25th of each month is observed, potentially linked to the common payday in Sweden. The yearly trend (Figure 2.1d) more clearly exhibits seasonality, with sales dipping during December-February and mid-summer, and rising in spring and fall. One month that seem to exhibit some additional complexities is November which shows the overall highest median and month to month variance, the source of this behaviour is mostly likely multifaceted but one key contributor is "Black week", which is a large week long yearly sale in conjunction with Black Friday which historically has a large impact on the SCR's sales. From this analysis it can be said with some confidence that there is a presence of seasonal patterns in the data, this must be considered when developing models for sales and campaign performance prediction.

Another behaviour found within the data which was hinted at from figure 2.1, is that there seem to exist quite extreme volatility in sales from day to day. Figure 2.2 provides a clearer picture of this volatility, as seen by the high day to day variance in sales. Some of the seasonality previously discussed can be seen even among this extremely noisy data. The downtrend in July and the influx of sales towards the end of November due to Black week are still present but less obvious. One reoccurring theme throughout all plots shown in figure 2.1 is that there seem to exist large variance on all time scales shown, this is illustrated by the large interquartile range (fig. 2.1a, 2.1c) and the density distribution of the data points in the violin plots (fig. 2.1b, 2.1d.

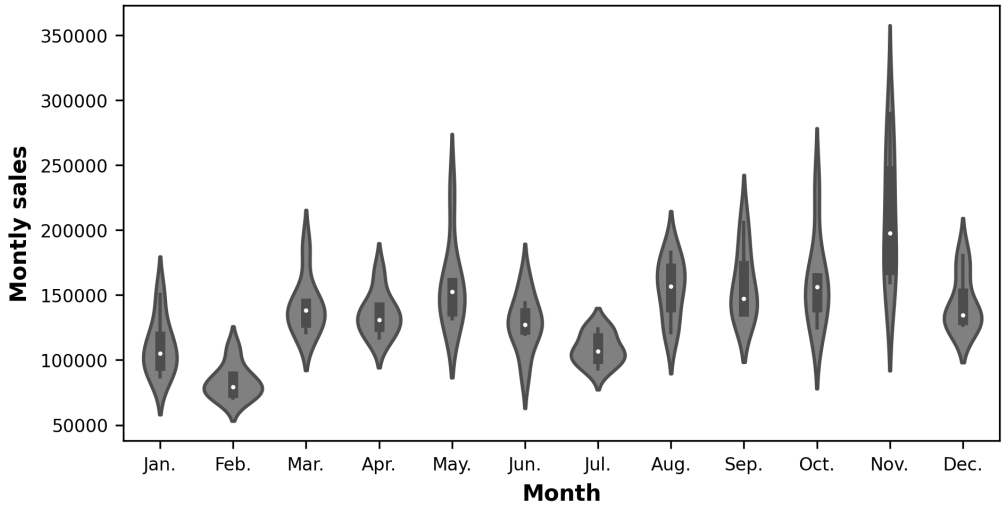
So far we have treated the data as uniform i.e we have not differentiated campaign and non campaign sales data, to get a better understanding of the dynamics inherent to these two types of data lets start by separating them and looking at the daily sales volume over time. Figure 2.2 illustrates some important properties of the data, firstly looking at the gray line showing the daily sales for all data, non campaign and campaign we see strong day to day volatility which was indicated previously. The black and red line shows the 7 day moving average for campaign and non campaign sales, a couple of things stand out immediately. Firstly that 2018-2021 campaign sales and non campaign sales are similar in magnitude and variance, with campaign sales showing some larger peeks around black week and holidays. Then suddenly for 2022 the behaviour of data changes drastically, non campaign sales drop very close to zero while the campaign sales seem to greatly increase. This behaviour could indicate a major shift in SCR business behaviour, seemingly in 2022 almost all sales are connected to a campaign in some shape or from. The source of this behaviour could also be multifaceted, instead of a shift in sales practice it is more more reasonable that the cause could be a shift in how SCR handles campaigns within their e-commerce system, which could affect how the data is generated. Without further inquiry to SCR it is impossible to tell.



(a) Daily product sales per weekday. **(b)** Monthly product sales per year.



(c) Daily product sales per day of month.



(d) Monthly product sales per month.

Figure 2.1: Sales Trends Analysis: Daily and Monthly product Sales Patterns.

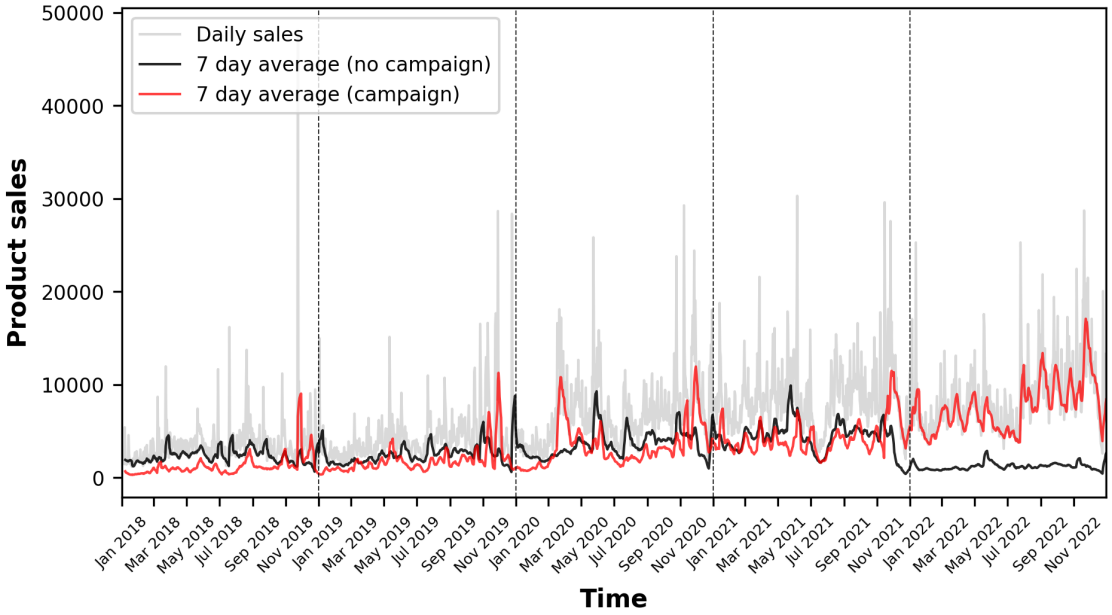


Figure 2.2: Product sales over time, red and black line showing the 7 day moving average for non-campaign and campaign sales, grey line showing the daily sales for the combined data.

Table 2.1 provides a concise description of how the distribution between campaign and non campaign sales have evolved over time. Before 2022 campaign sales was the minority, albeit close to 50% of the total sales. But in 2022 campaigns sales made up 86% of total sales, thus resulting in 56.6% of all sales coming through campaign sales aggregated over all years. A side effect of the majority of sales coming for campaigns sales is that there is very little non campaign sales data in which from a sales baseline could be drawn. You can frame this as, if there always is a sale, there never is a sale because that is just the new normal behaviour. This phenomenon, known as 'discount fatigue', has been a subject of debate. Some researchers, like Tanase (2023), acknowledge its impact and propose potential strategies to mitigate it [6]. Conversely, other studies, such as Dholakia (2011), find no substantial evidence supporting the existence of 'daily deal fatigue' [7]. While this thesis will not delve further into this phenomenon, it remains a potentially significant area for SCR to investigate, given its implications for consumer behavior and sales strategy.

Table 2.1: Campaign and non campaign sales over time, for each respective year expressed as percent of total sales.

	Product sales (%)					
	2018	2019	2020	2021	2022	Total
Campaign	35.06	41.87	48.07	49.85	86.03	56.59
Non campaign	64.94	58.13	51.93	50.15	13.97	43.41

Another fundamental aspect of the business model change observed in 2022 is, since a campaign performance model naturally is a forecasting model that learns from

historic data it might render the pre 2022 data futile to predict 2022 campaigns since it exhibits very different behaviour. Unfortunately the answer to this can only be learned by implementing a model and analysing the results.

2.2 Campaign analysis and cleanup

The preceding analysis shed light on general sales trends within SCR, this section however, shifts focus towards dissecting the dynamics of campaign sales data, encompassing aspects such as campaign duration's, impacted products, and corresponding sales figures. Rather than examining each campaign in isolation, they have been categorized by their respective offer types. A comprehensive list and detailed descriptions of these offer types are available in Appendix C.
In the process of analysis, certain offer types were excluded due to specific challenges related to analysis, accountability, or their unique nature. The reasons for excluding these offer types are as follows:

- Offer Types 7, 16, 17, 19: These were excluded as it was not feasible to determine the specific products or categories responsible for the discounts. The discounts in these offers are applied directly to the customer's shopping cart, rather than to individual items.
- Offer Type 8: This offer, providing free delivery, was removed from the analysis. It is almost perpetually active and is not linked to any specific product or category, but rather applies to the cart as a whole.

This approach ensures a focused and efficient analysis by concentrating on the most relevant and analyzable campaign data.
Unfortunately analysis of the campaign data immediately presented certain challenges that needed to be addressed. To provide a clearer understanding, let's introduce how a campaign is defined within the database:

campaign ID	campaign name	start date	end date	creation date
1	Buy X pay Y	2018-01-20	2018-02-05	2018-01-05

Table 2.2: Campaign data point example.

In table 2.2, an example campaign is presented. The campaign ID serves as a primary key for campaign identification and is referenced across multiple database tables. Campaign name, start date, and end date are self-explanatory, while creation date marks the initial entry of the campaign into the database. However, somewhat counterintuitively, not every campaign within the database represents a unique campaign instance. To clarify, consider another example, this time involving a purchase:

purchase id	purchase date	quantity	price (SEK)	discount	campaign ID
1	2018-01-10	2	100	50	1

Table 2.3: Purchase data point example.

In Table 2.3, a purchase example is shown, featuring a unique purchase ID, purchase date, quantity, price, discount, and the associated campaign ID. It's crucial to note that the "discount" reflects the total discount awarded for the entire purchase, not on a per-item basis, and the price represents the price per article. What becomes evident when examining Tables 1 and 2 is a discrepancy in the dates. In this instance, a purchase was made before the campaign became active according to the start date in Table 2.2.

This inconsistency arises from the fact that in the current Viskan Systems e-commerce platform campaigns can be reused by using previous campaigns as templates and reactivating them by simply declaring new start and end dates. Consequently, purchases may occur during campaigns, but if those campaigns have been reactivated multiple times, one lacks a comprehensive history of campaign dates and possesses information only on the latest instance of the campaign. This poses a challenge, especially when attempting to compare campaigns equitably. A common method to achieve this involves normalizing campaign performance metrics by the campaign's runtime. However, calculating the runtime using the difference between the start and end dates provides an inaccurate representation because it considers only the latest instance of the campaign, potentially incorporating purchases from many different periods in time. This can distort the performance analysis of campaigns that have been run multiple times. This also implies that one can not look at the dates of the first and last purchases connected to a campaign since it also does not account for the multiple instances of campaign.

To address this challenge, a solution for estimating campaign runtimes that is more accurate than the information provided by the campaign data is imperative. This is achieved by identifying prior instances of campaign runs and consolidating their durations. Essentially, this task resembles a time series segmentation problem, for which numerous sophisticated algorithms are available. However, these algorithms tend to be intricate and often require extensive hyperparameter tuning to yield satisfactory results for a given time series. To avoid introducing unnecessary complexity, a simpler approach is sought. Notably, the absence of known answers for true runtimes in the data makes training supervised algorithms essentially unfeasible. Unsupervised algorithms, on the other hand, introduce their own set of challenges. Hence, a pragmatic and straightforward solution is preferred.

To prepare for the forthcoming explanation of the approaches attempted, let's illustrate the campaign instance problem with an example. In Figure 2.3, a time series depicting the products sold through an example campaign. In other worlds products which have been assigned a campaign ID within the dataset. However, it's critical to acknowledge that there may still be sales indirectly influenced by a campaign but not explicitly linked to it. For instance, if a campaign offers a "Buy 3, pay for 2" deal and a customer purchases only one product, this transaction, despite being influenced by the campaign's presence, would not be assigned a campaign ID. The green-shaded area represents the duration when the latest instance of this campaign was active, as per its specified start and end dates. Preceding the green section, a flat region with zero product sales exists, indicating a period when the campaign was inactive. Before this inactive phase, there is another segment, akin to the green one, indicating active sales. The challenge at hand revolves around the

identification and removal of these flat, inactive sections from the timeline, retaining only the segments when the campaign was actively running. By summarizing the active days, one can approximately determine the campaign's true runtime. It is worth noting that a simplistic approach of merely eliminating days with zero sales is inadequate, as even less successful campaigns may have days with zero sales while still being active.

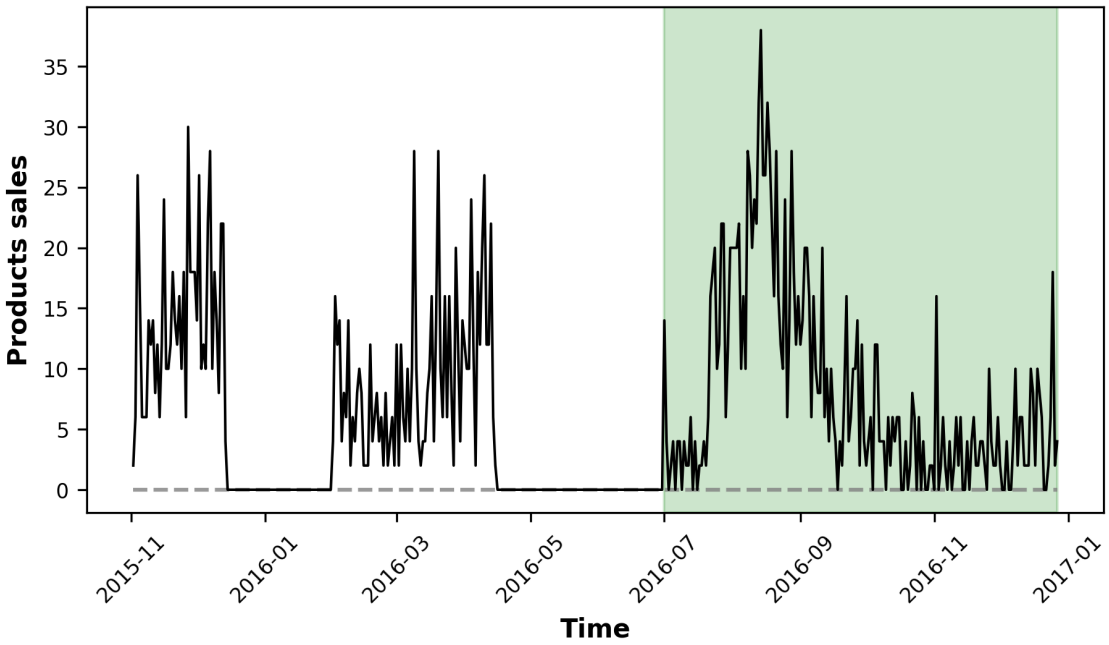


Figure 2.3: Time series of products sold through an example campaign, green area shows campaign runtime according to the dataset.

The first step to find these inactive segments was to calculate the time between days in which the sold products count was larger than zero. First define the time series of sold products as $\mathrm{Y} = (y(t) : t \in \mathrm{T})$ where T is the index set. Then define the subset $\mathrm{T}_d \subseteq \mathrm{T}$, where $\mathrm{T}_d = \{t \in T \mid y(t) > 0\}$ containing the days in which there were active sales. The set containing the number of days between days with sales is then defined as

$$\Delta \mathrm{T}_d = \{t_{i+1} - t_i,\ \forall i \in \mathrm{T}_d\}$$

which means that $\Delta \mathrm{T}_d$ is of length $|\mathrm{T}_d| - 1$ due to the dependency of the previous time step. Now that the time between days of sales for given campaign is know, the goal is to find the outliers i.e large values of $\Delta \mathrm{T}_d$ that its more likely to be an inactive regions and not poor sales performance. The inactive segments are defined by their change points, which state the start and end point for each segment. The problem now becomes finding these change points for each individual campaign.

Upon inspection of the distributions of $\Delta \mathrm{T}_d$ which can be seen in appendix A.2 it was clear that it did not follow a normal distribution but more something more

similar to a long tail distribution. This means the simple statistical approach of using the mean value and standard deviation of T_d to find out outliers i.e change points would not work. As a result of this insight both algorithmic and statistical approach seemed reasonably unfeasible, especially due to the fact that there is no way to confirm the accuracy resulting segments. Instead a much more simple rule based approach was taken, in which if an element of $\Delta T_d > 7$ i.e if there is no sales for 1 week, assume that the campaign is inactive. this gives us the binary classification for each time step in ΔT_d where

$$\begin{aligned} &\text{t is a change point if } \Delta T_d > 7, \quad \text{for each } t \in \Delta T_d \\ &\text{t is not a change point, otherwise} \end{aligned}$$

1 week was chosen based on the fact that 97.6% of all $\Delta T_d \leq 7$, and by manual inspection of the resulting segmentations when testing different values for the rule based approach. For the remainder of this thesis the calculated campaign instances from the rule based approach will be used for campaign performance analysis, even though this segmentation is highly approximate. The resulting segmentation of the campaign shown in figure 2.3 can be seen in figure 2.4. This example highlights the successful segmentation of the time series using the 7 day rule. The approximated runtime of the campaign was calculated to 298 days while the data would have given 180, this discrepancy would as previously mentioned result in a large error when normalizing the campaign performance by their runtime. A few additional examples of campaign segmentation's can be seen in appendix A.2

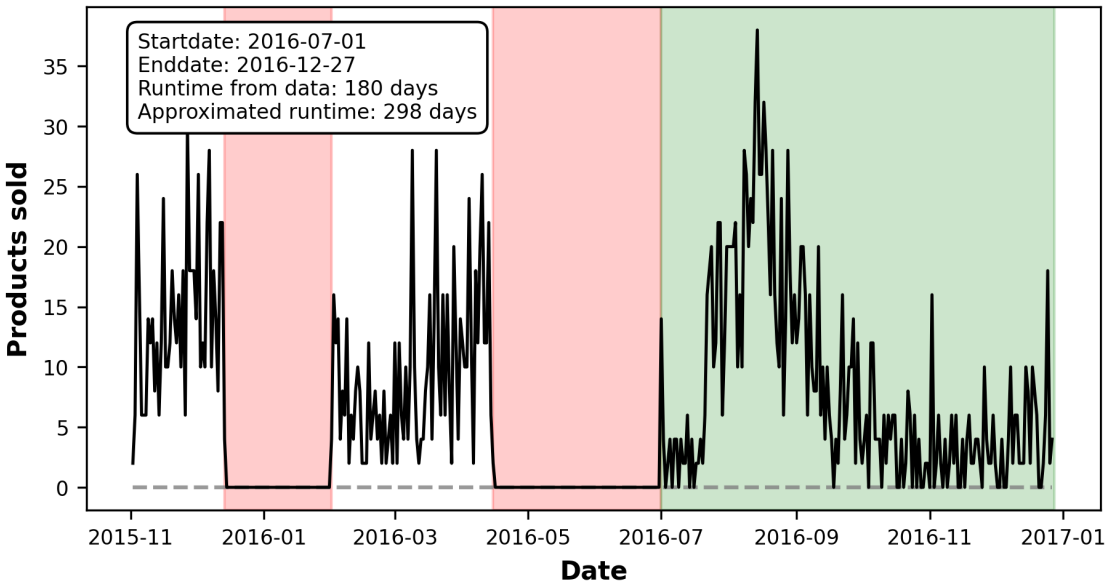


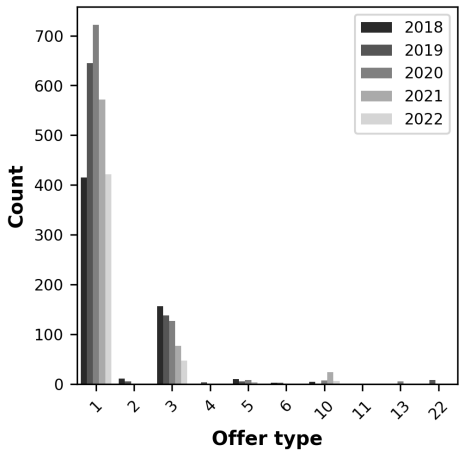
Figure 2.4: Time series of products sold through an example campaign, green area shows campaign runtime according to the dataset. Red segments highlight the periods when the campaign was inactive, as determined by the applied runtime segmentation rule. Calculated runtime according to both data records and the segmentation rule is shown.

With the acquired campaign instances from the time series segmentation an analysis

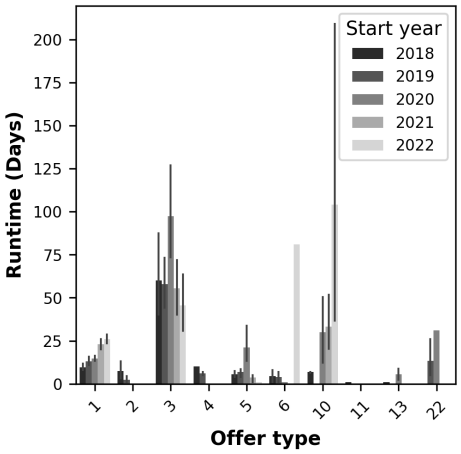
of campaign metrics was feasible. To determine if a machine learning modeling approach is possible for sales and campaign performance forecasting, a deeper analysis into the behaviours and distributions of the offer types needed to be done.

Figure 2.5 shows a broad overview of a few interesting metrics of the system. Firstly looking at figure 2.5a the occurrence count for each offer type per year is displayed. There is a large discrepancy in the number of occurrences among the offer types, offer type 1 and in some measure offer type 3, show a disproportionate count compared to other offer types. Another interesting behaviour is the year over year decrease in the number of offer type 3 campaigns. Moving on to figure 2.5b, plotted in the same fashion we see the runtimes for each respective offer type, showing the mean and 95% confidence interval. Most notably is that there seem to exist a large variance between the offer types, additionally there seem to exist a noticeable variance year to year as well, offer type 3 even showing a large yearly variance. Additionally offer type 1 shows a consistent year over year increase in runtimes, this coupled with the large occurrence count indicate that a large portion of SCR sales come from offer type 1 campaigns. This statement is further strengthened by figure 2.5c showing a year over year increase in the median number of product sales per day for offer type 1. Similarly to other metrics the sales per day shows a large variance between the offer types, presenting no easily interpretable results. Finally, figure 2.5d shows the revenue-discount (R/D) ratio of the offer types, R/D ratio can be interpreted as the revenue generated per SEK offered as discount i.e a profitability metric. Two takeaways from this figure is the year over year increase in R/D ratio for offer type 1, indicating that the offer type has become more profitable over time as SCR potentially became better at applying this certain offer type. As for the rest of the offer types, the R/D ratio show no obvious results, except for offer type 3 which seemingly is quite consistent over the years.

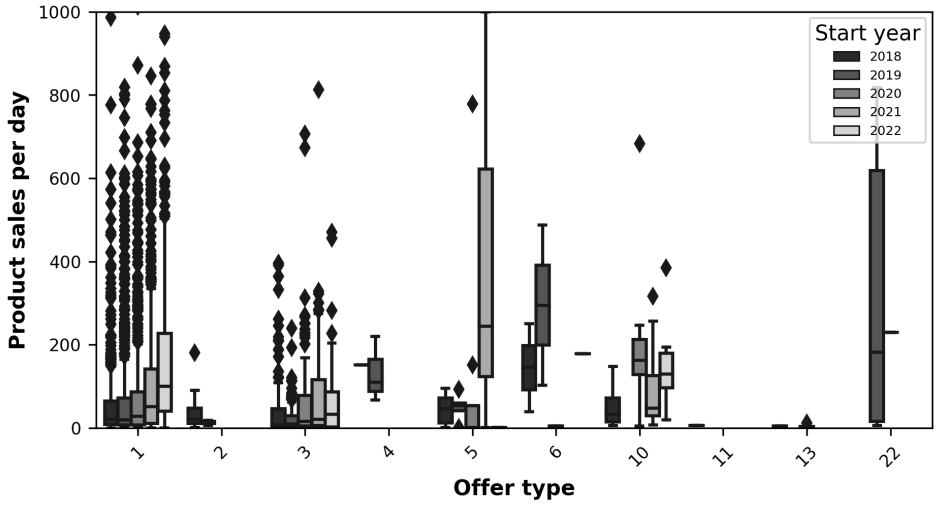
To conclude this segment on the offer type analysis, it is hard draw any data driven conclusion on the performance of campaigns, except maybe for what was mentioned on offer type 1. However, we can say without much doubt that data hungry algorithms such a neural networks, most likely wont perform optimally due to the lack of data in for some offer types. In cases with strong presence of data sparsity, statistical approaches or highly data efficient machine learning algorithms would be more appropriate.



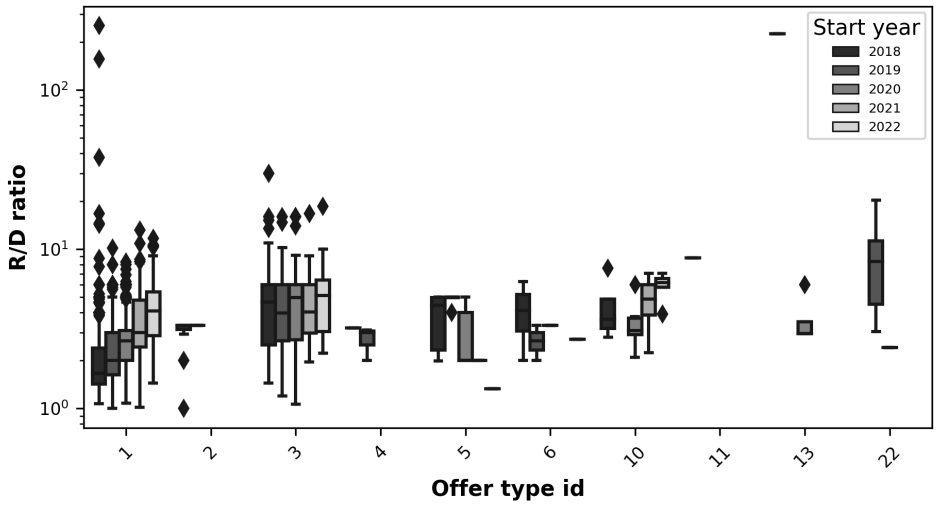
(a) Occurrence count for each offer type.



(b) Runtime distribution for each offer type, showing mean and 95% confidence interval.



(c) Product sales for each offer type.



(d) Revenue-discount (R/D) ratio for each offer type.

Figure 2.5: Campaign analysis showing key metrics for the offer types.

2.3 Model selection

From the general and campaign data analysis we have learned that SCR's business model, like most large businesses, is highly complex with many factors affecting campaign performances. A strong presence of seasonality was exhibited, both natural seasonality and induced seasonality as result of reoccurring yearly events, keeping the time dependency in mind and also considering the sparsity of data for some offer types, two models were selected. Firstly, a time series based approach using NeuralProphet [8], a powerful and user friendly framework with built in support for exogenous variables such as campaigns. This model will hopefully be able to model the general sales trends of SCR's e-commerce system. Secondly, a regression based approach was chosen due to the high daily variance which might make a time series model hard to use. For the regression XGBoost [9] was chosen due it being a gradient-boosted trees algorithm, offering better general explainability then a pure black box model, and shown to have good performance on small data sets [10]. This model will hopefully be able to predict the performance of campaign instances.

2.4 Missing data

The analysis thus far has revealed the complexity of SCR's campaign system, highlighting multiple influencing factors on campaign performance. Regrettably, certain potentially impactful data features were unavailable during the analysis. These missing features, listed without prioritizing their importance, include:

- Social media campaigns
- Campaign placement on platform
- Specialty events (Singles day, black week,...)
- Email campaigns
- Re-marketing

Starting with social media campaigns, SCR's extensive presence on social media platforms like Facebook, Instagram, and TikTok is noteworthy. Their content include product launches, product highlighting and campaign promotions through various mediums. However, this analysis did not account for the impact of social media on campaign performance using any sort of engagement metrics such as click through rate, likes or views. This omission leads to a gap in knowledge of its influence, as numerous studies have underscored the significance of social media in e-commerce [11, 12, 13].

Moreover, the placement of campaigns on SCR's website could be a crucial factor. The ease of navigation and the visibility of campaigns, whether prominently on the homepage or concealed within menus, can significantly affect sales performance [14].

Additionally, special events like Singles Day and Black Week, which are annual occurrences, might require distinct consideration. The dataset does not specify if a campaign is linked to such events, and any attempt to infer this from campaign dates and names is speculative and inconsistent.

Furthermore, the influence of email marketing and re-marketing strategies, which involve notifying customers about campaign launches or reminding them of incom-

plete purchases, was not assessed due to data unavailability.
In summary, this subsection highlights the incompleteness of the available information for a complete analysis. The mentioned examples represent only a subset of potentially crucial factors, engagement metrics that could prove invaluable in this form of analysis. A deeper collaboration with SCR would be beneficial to acquire specific domain knowledge and identify additional relevant data features that could provide useful to Viskan Systems in it's customers.

3
Theory

The theory chapter of this thesis will be divided into two sections, firstly the introduction and definition of *NeuralProphet* which is a highly popular time series forecasting framework, secondly the introduction of XGBoost a state of the art tree boosting machine framework. These two frameworks were chosen based on the what was learned from the data analysis chapter.

3.1 Neural prophet model

NeuralProphet, as introduced by Triebe et al. in [8], is a powerful time series forecasting framework built upon Facebook's forecasting framework known as Prophet. Both frameworks provide scalable and user-friendly environments for time series forecasting. However, NeuralProphet distinguishes itself from Prophet in several key ways.
One significant difference lies in NeuralProphet's introduction of local context through auto-regression, which enhances its capability for near-future forecasting. Additionally, NeuralProphet is developed using PyTorch, a well-established framework for machine learning models. This choice of backend technology grants NeuralProphet greater flexibility and complexity compared to Prophet, which relies on the STAN backend.
Before delving into the specifics of NeuralProphet, it is essential to provide a more general introduction to the field of forecasting. Accurate forecasting plays a vital role in decision-making processes, influencing areas such as business strategy, resource allocation, and risk management. In a business environment, forecasting is often challenging due to the costly consequences of errors. Decision-makers typically prefer interpretable models, which have become scarcer with the rise of highly performant black-box models based on neural networks, such as recurrent neural networks.
Fortunately, NeuralProphet offers a level of explainability that can be harnessed by domain experts to interpret results and make adjustments to both the model and the data. This approach, known as "human in the loop" [15], empowers domain experts to contribute to the model training process effectively.
Historically, statistical methods have dominated time series forecasting, with techniques like ARIMA (autoregressive integrated moving average), SMA (simple moving average), and ES (exponential smoothing) being extensively studied and used [16]. However, these models come with assumptions and parametric limitations that can restrict their applicability in real-world scenarios, especially when dealing with complex dynamics.

In recent years, driven by the explosion of available data and the growing popularity of neural networks, models powered by neural network methods have gained traction. Frameworks like NeuralForecast have provided user-friendly environments for implementing and training neural network-based models. However, these models often retain their black-box nature, making them less interpretable.
This is where NeuralProphet steps in as a hybrid model that combines the interpretability of statistical methods with the strength of neural networks to handle intricate data patterns. In this section, we provide a foundational understanding of NeuralProphet's key concepts. For an in-depth exploration of the model and its components, we refer readers to the original paper by Triebe et al. [8].

Model components

NeuralProphet is constructed in a highly modular way, in which most components can be set to be additive or multiplicative to the final forecasting result. The predicted forecast $\hat{y}_t, ..., \hat{y}_{t+h}$ where h is the forecasting horizon is comprised of said components in the following way

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(t) \tag{3.1}$$

where,

$T(t) =$ Trend at time t
$S(t) =$ Seasonal effects at time t
$E(t) =$ Event and holiday effects at time t
$F(t) =$ Regression effects at time t for future-known exogenous variables
$A(t) =$ Auto-regression effects at time t based on past observations
$L(t) =$ Regression effects at time t for lagged observations of exogenous variables

All of the above components can be configured individually or turned off to best handle the task at hand. The final forecast is then a result of the component effects as seen in equation 3.1.

Trend

NeuralProphet models the trend component using a continuous piece-wise linear series in which the trend effect can change at each segment. The segments are defined by a set of change points C containing n_c points at different times defining $C = \{c_1, c_2, \ldots, c_{c_n}\}$. Each segment has a growth rate δ, the growth rate at time t is calculated by summarising the initial growth rate δ_0 and all growth rates δ_j at all the change points up to time step t. Similarly an offset ρ_j for change point c_j is defined by $\rho_j = -c_j\delta_j$ and is calculated by summarising the initial offset δ_0 and all offsets ρ_j at all the change points up to time step t. This effectively results in n_C number of δ_j to be fitted to the data. To represent if the time t is past a change point $\Gamma(t) \in \mathbb{R}^{n_c}$ is introduced which is a binary vector defined as

$$\Gamma(t) = (\Gamma_1(t), \Gamma_2(t), \ldots, \Gamma_{n_c}(t))$$
$$\Gamma_j(t) = \begin{cases} 1, & \text{if } t \geq c_j \\ 0, & \text{otherwise} \end{cases}$$

Thus resulting in the final equation for the trend component

$$T(t) = (\delta_0 + \Gamma(t) \cdot \delta) \cdot t + (\rho_0 + \Gamma(t) \cdot \rho)$$

where

$$\delta = (\delta_1, \delta_2, \ldots, \delta_{n_C})$$
$$\rho = (\rho, \rho, \ldots, \rho_{n_C})$$

NeuralProphet provides a semiautomatic way of selecting change points, given the number of change point n_c they are equidistantantly spread out along the series. The user can of course also manually define change points anywhere along the series.

Seasonality

Seasonality in NeuralProphet is modeled using Fourier series. Where for each seasonality $p \in \mathbb{P}$, where $\mathbb{P}$ is the set of all seasonalities, each seasonality is defined as

$$S_p(t) = \sum_{j=1}^{k} \left(a_j \cdot \cos\left(\frac{2\pi jt}{p}\right) + b_j \cdot \sin\left(\frac{2\pi jt}{p}\right) \right) \tag{3.2}$$

Seasonalities are of course data dependent but some common examples with daily data is yearly seasonality ($p = 365.25$) or monthly seasonality ($p = 12$). The total effect of all seasonalities is defined by equation 3.3

$$S(t) = \sum_{p \in \mathbb{P}} S_p^*(t) \tag{3.3}$$

where

$$S_p^*(t) = \begin{cases} S_p^\dagger(t) = T(t) \cdot S_p(t), & \text{if } S_p \text{ is multiplicative} \\ S_p(t), & \text{otherwise} \end{cases} \tag{3.4}$$

As mentioned previously some components can have additive or multiplicative effects on the forecast result as seen in equation 3.4. NeuralProphet offers some default settings for common seasonalities, $k = 6$ for $p = 365.25$ yearly, $k = 3$ for $p = 7$ weekly, and $k = 6$ for $p = 1$ daily seasonality. These seasonalities can be customized and more can be added to the model.

Auto-regression

To better handle near future effects Auto-Regression (AR) is introduced into the model, which is based on the assumptions that future values are dependent on the previous ones with some noise. AR in it's simplest form can be defined as

$$y_t = c + \sum_{i=1}^{i=p} \theta_i y_{t-i} + e_t$$

where c is an intercept, θ_i is a coefficient fitted for each past value and e_t is noise. p is referred to as the order AR and defines how many time steps in the past the AR takes into account. This form of AR is known as linear AR, however if one wants to perform forecasting more than one time step in the future and introduce non linear effects it becomes a bit more complicated. Without going too in depth, NeuralProphets AR component is based on a modified version of AR-Net by Triebe et al. [17] which introduces the capability to do forecasts h steps into the future which can be linear or non-linear using hidden layers. Linear AR is then defined

$$y = Wx \tag{3.5}$$

where

$$x = (y_{t-1}, y_{t-2}, \ldots, y_{t-p})$$
$$y = (A^t(t), A^t(t+1), \ldots, A^t(t+h-1))$$

Equation 3.5 and results in $y \in \mathbb{R}^h$, a vector containing the AR effects for h time steps into the future and p is the order of the AR component i.e the number of immediately preceding values in the series that are used to predict the value at the present time. Where $A^t(t+d)$ represents the predicted AR-effect for forecast $\hat{y}_{t+d}$ at $t+d$, predicted at forecast origin t using time steps up until and including $t-1$, which is generalized as

$$A^t(t), A^t(t+1), \ldots, A^t(t+h-1) = \text{AR-Net}(y_{t-1}, y_{t-2}, \ldots, y_{t-p}) \tag{3.6}$$

The weight term $W_{i,j} \in \mathbb{R}^{h \times p}$ in equation 3.5 embeds the coefficients defining the linear impact of previous time step y_j on AR-effect $A^t(t=i)$. The $h \times p$ parameters are to be fitted using the data. As previously mentioned NeuralProphet also features non-linear AR in the form of deep AR based or AR-Net, this is implemented by having the module train a fully connected neural network with user specified layers. Introducing deep AR may improve forecasting results for complex data but does reduce interpretability. The NN is constructed using p previous observations as inputs and h outputs. Note that the final layers outputs, are not transformed by an activation function and have no bias. A NN containing p inputs, h outputs, l layers of dimension d is then defined as

$$a_1 = f_a(W_1 x + b_1) \tag{3.7}$$
$$a_i = f_a(W_i a_{i-1} + b_i),\ i \in [2, \ldots, l] \tag{3.8}$$
$$y = W_{l+1} a_l \tag{3.9}$$

where

$$f_a(x) = ReLU(x) \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

and the weights and biases are defined as, $b \in \mathbb{R}^d$, $W_1 \in \mathbb{R}^{d \times p}$, $W \in \mathbb{R}^{d \times d}$ and $W_{l+1} \in \mathbb{R}^{h \times d}$. Similarly to the linear AR the parameters of the NN are fitted on the data.

Lagged regressors

Lagged regressors are independent variables that are correlated to the dependant variable of the model i.e the forecast variable. The future of lagged regressors are unknown but the past values can be used to have an additive effect on the forecast. NeuralProphet handles lagged regressors identically to the AR component discussed in the previous section, resulting in

$$L(t) = \sum_{x \in \mathbb{X}} L_x^t(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) \tag{3.10}$$

where L_x^t is defined similarly to the AR components in equation 3.6

$$L^t(t), L^t(t+1), \ldots, L^t(t+h-1) = \text{AR-Net}(x_{t-1}, x_{t-2}, \ldots, x_{t-p})$$

Given the set of covariates $\mathbb{X} \in \mathbb{R}^{T \times n_l}$ to the dependant variable y, a lagged regressor component is created for each of the n_l covariates containg T past values. The combined effect of all the lagged regressors are then calculated using equation 3.10.

Future regressors

To improve forecasting performance information that is both know in the past and in the future can be used as a forecasting component, information such as calendar variables or weather forecasts. Given a set of future regressors $\mathbb{F} \in \mathbb{R}^{T \times n_f}$, where n_f is the number of future regressors which values are known for T past time steps. The combined effect of all future regressors is defined in equation 3.11, where d_f is coefficient for each future regressor $f \in \mathbb{F}$.

$$F(t) = \sum_{f \in \mathbb{F}} F_f^*(t) \tag{3.11}$$

where

$$F_f(t) = d_f f(t)$$

$$F_f^*(t) = \begin{cases} F_f^\dagger(t) = T(t) \cdot F_f(t), & \text{if } f \text{ is multiplicative} \\ F_f(t), & \text{otherwise} \end{cases}$$

Event and Holidays

Events and holidays effects are modeled similarly to future regressors but instead now each event e is treated as a binary variable $e \in [0, 1]$, indicating if the event or

holiday is active or not for a given time step t. Thus for a set of events $\mathbb{E} \in \mathbb{R}^{T \times n_e}$, containing n_e events and a time series of length T, the effect of events is denoted by equation 3.12. Where z_e is coefficient of effect for each event $e \in \mathbb{E}$.

$$E(t) = \sum_{e \in \mathbb{E}} E_e^*(t) \tag{3.12}$$

where

$$E_e(t) = z_e e(t)$$
$$E_e^*(t) = \begin{cases} E_e^\dagger(t) = T(t) \cdot E_e(t), & \text{if } e \text{ is multiplicative} \\ E_e(t), & \text{otherwise} \end{cases}$$

NeuralProphet contains functionality to automatically add country holidays for a specified country and custom event can be configured as well. For events that stretch out over multiple time steps, an event window can be configured. Then for a given event at time t_e the event window is defined as $[t_e - i, t_e + j]$, in which each day of the event is treated as unique event with its own additive or multiplicative effect.

3.2 XGBoost

XGBoost (eXtreme Gradient Boosting) is an implementation of gradient boosting algorithms introduced by Tianqi et al. [9], known for its efficiency, flexibility, and portability. It has gained widespread popularity in machine learning competitions and practical applications due to its robust performance in a variety of tasks. XGBoost is built upon the principles of gradient boosting, a powerful ensemble learning technique that combines multiple weak predictive models to create a stronger overall model. This section presents the condensed version of the core components and algorithms that constitute the XGBoost framework, for a deeper explanation of XGBoost we refer the reader to the original paper.

Gradient boosting

At its core, XGBoost utilizes the gradient boosting framework. This involves sequentially adding predictors to an ensemble, each one correcting its predecessor, more specifically XGBoost primarily uses gradient boosting regression trees as its base learners. Unlike traditional boosting methods that minimize a loss function by adding models that correct the errors of all predecessors, XGBoost uses a gradient descent algorithm to minimize the loss function. This approach allows for more effective optimization, particularly in the context of large and complex datasets. For a more general introduction into the history and mechanics of decision trees we refer the reader to paper "Decision trees" [18] written by B. De Ville.

For a given data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$ with $i = (1, \ldots, n)$ data points, m features and K predictors, the final output is defined as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$$

f_k corresponds to independent tree structure with the map $q : \mathbb{R}^m \to T$, mapping m features to T output leaf nodes. T is the number of leaves in the tree, with leaf weights $w \in \mathbb{R}^T$, thus q is the regression tree space. Each regression tree contains a continuous score on each of the leaf, w_i to represents score on the i-th leaf. Instead of finding the set K of functions which minimizes loss, the model is trained in an additive manner $\Rightarrow$ let $\hat{y}_i^{(t)}$ be the prediction of the i-th instance at the t-th iteration $\Rightarrow$ each iteration f_t is found which minimized the objective function

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \tag{3.13}$$

Where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. Since the model is trained in an additive manner, the f_t that most improves the model according to equation 3.13 is added greedily to the next iteration. Finally $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||w||^2$ is a regularization term which penalizes complexity within the model, resulting in less over-fitting. γT punishes having large tree nodes with many leaf nodes, while $\frac{1}{2}\lambda||w||^2$ punishes large leaf weights. This regularization component is not present in standard gradient boosting methods and is a significant factor in XGBoost's improved performance. Figure 3.1 presents a high level overview of the XGBoost architecture, where the leaf nodes are marked in green. As described, the final model output $\hat{y}$ is derived through a cumulative process. Initially, for each decision tree in the ensemble, the outputs from its leaf nodes are aggregated to form the tree's individual contribution, f_k. Subsequently, the overall model output is obtained by summing up these contributions from all the decision trees within the ensemble.

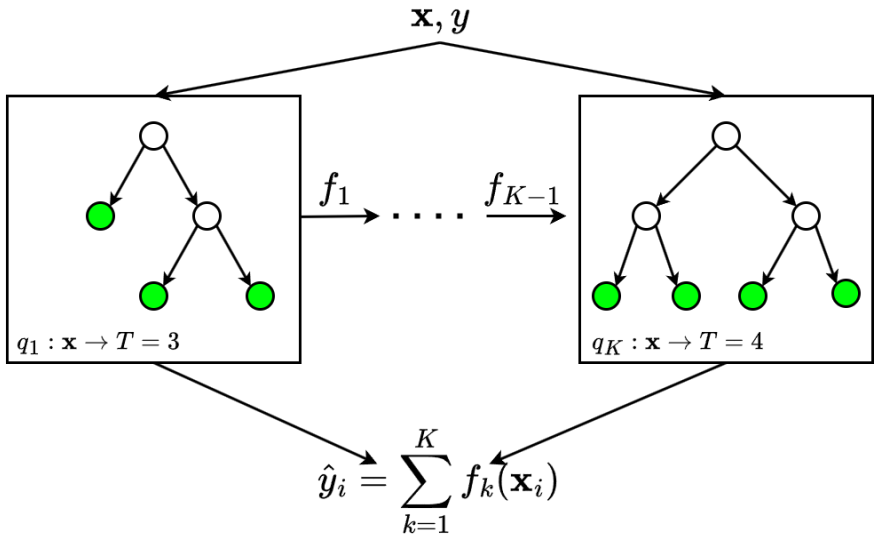


Figure 3.1: High level overview of how the regression tree space q maps from data to model output. Leaf nodes are marked in green.

By then applying gradient descent methodology the loss is minimized, details on the derivation of gradient decent used in XGBoost have been left out to reduce the complexity of this section. However a problem not discussed yet is that of finding the best tree structure q, practically it is impossible to enumerate over all the possible structures for q. To solve this XGBoost implements an approximate split finding algorithm i.e how the decision tree split the features between the nodes and leaves. The algorithm first proposes candidate splitting points according to percentiles of feature distribution, then the features are split into buckets based on the candidate points and find the best split based on aggregated statistics. Within the framework there are two versions of the approximate algorithm namely the global and the local version. The global variant proposes all the candidate splits during the initial phase of tree construction, and uses the same proposals for split finding at all levels trough out the tree. The local variant re-proposes after each subsequent split. The global variant is less computationally heavy, while the local variant can potentially be more appropriate for deeper trees.

There are some other factors of the XGBoost framework which makes it highly attractive for practical use.
XGBoost accelerates the tree building process by leveraging parallel and distributed computing. It distributes the computation of the individual trees across multiple cores in a CPU or even across multiple machines, significantly reducing the training time. This feature makes it particularly suitable for datasets that would otherwise take a long time to process on a single machine. This might be less of a problem in the case of this thesis but having quick training times allows us to do rigorous hyperparameter optimization with limited resources in an efficient manner.
Another feature that XGBoost offers is that sparsity-aware split finding algorithm, which is highly important when working with categorical data such as category ID's which require some sort of encoding, usually one-hot encoding, resulting in sparse data.

4

Methods

The method chapter is divided into two chapters, starting with the implementation NeuralProphet. This section will contain the steps taken to prepare the data set, model settings and hyper parameter optimization. The second section will similarly for XGBoost go trough the process of data preparation, model implementation and hyperparameter optimization.

4.1 NeuralProphet

To predict SCR's sales as a time series, a dedicated dataset with daily resolution was constructed. This decision was based on the weekly and yearly seasonal patterns identified in the general sales analysis (section 2.1). The target feature for forecasting was set as the quantity of sold products, rather than revenue or the number of orders. This choice is grounded in the rationale that revenue and order numbers are derivatives of sold quantities, influenced by pricing strategies and consumer purchasing behavior. Opting for the more fundamental metric, sold product quantity, minimizes the introduction of extraneous dependencies.

For testing the model a 3 month forecasting horizon was chosen, this resulted in training data based on dates in the range 2018-01-01 - 2022-09-01 and testing 2022-09-02 - 2022-12-31. The choice of a 3 month forecasting was not based on any preconceived forecasting horizon from SCR, but simply based on a quarterly forecast.

To model sales dynamics, we developed three hierarchical sales models:

1. Meta Model: Aggregates all sales data across product categories into a single time series.
2. Category level Model: Focuses on sales within a specific category, in this case, the 'Basic' category, which includes popular staple items like T-shirts, sweatshirts, joggers, and underwear.
3. Product level Model: Concentrates on the sales of a single, anonymized but highly popular product from the 'Basic' category.

Each model in this hierarchy was selected to explore the accuracy and practicality of each tier. Due to time constraints, only one representative time series from each level was analyzed, a full hierarchical structure is illustrated in figure 4.1. Ideally, the entire system would be modeled comprehensively.

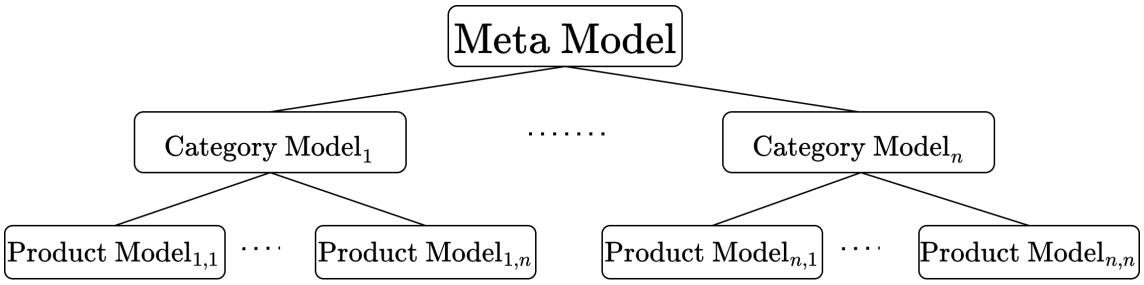


Figure 4.1: Diagram of hierarchical model structure for sales prediction.

The hierarchical structure serves multiple purposes. Firstly, the aggregate models (meta and category levels) inherently lose specificity regarding the source of sales, thus providing a general forecast without product-level granularity. Ideally, a highly precise model could be developed for each product, as depicted in figure 4.1, which would enable the parent models to fully account for the origin of sales. However, developing such a detailed model for each product in SCR's portfolio is beyond the scope of this thesis. It presents challenges including significant computational demands due to the need for individual training and hyperparameter optimization of potentially thousands of models. Additionally, there are risks associated with compounded errors and data sparsity when integrating numerous low-level models.

4.1.1 NeuralProphet implementation

The NeuralProphet model was configured with the following key settings and features based on the hyperparameter optimization described in the next subsection.

- **Seasonality**: Incorporated yearly and weekly seasonalities in a multiplicative mode.
- **Auto-regression**: Utilized a Deep AR model with a two-layer architecture, consisting of 32 and 8 neurons, respectively. Where the AR order i.e how many previous time steps including in modeling is 78.
- **Future Regressors**: Campaign data was integrated using the future regressor component in multiplicative mode (Equation 3.11). To simplify, campaigns were encoded into time series, one for each offer type, containing the count active campaigns for each time step.
- **Events and Holidays**: The model's event component was employed to capture complex campaigns and holidays. Notable events like Blackweek and the after-Christmas sale (Mellan dags rean) were defined by their specific dates. Swedish holidays were included automatically using NeuralProphet's country holiday feature. Multiplicative mode was used for specific events, whereas additive mode was applied for general holidays.

4.1.2 Training and hyperparameter optimization

The models were trained using the AdamW optimizer [19], with the learning rate determined via a range test to identify the optimal rate that maximizes the negative

loss gradient [20]. Training was conducted over 250 epochs, supplemented by early stopping criteria based on the non-improvement of training loss.
Hyperparameter optimization for the hidden AR layers, seasonality, and event modes was performed using a random grid search with expanding window cross-validation. This method progressively increases the training data set while advancing the evaluation data forward. A total of a 1000 models (200 samples 5-fold cross validation each) were trained to find the best hyper parameters that would fit across all three model hierarchies. The finalized model configuration is detailed in Table 4.1. By default, NeuralProphet applies minmax scaling to all components. Additionally, the target feature, sold products, underwent a logarithmic transformation to mitigate the high daily variance observed (Figure 2.2).

Hyperparameter	**Value range**	**Best value**
Seasonality	Multiplicative - Additive	Multiplicative
AR layers	$8-48, 2-8$	$32, 8$
AR order	$1-1073$	78
Future regressors	Multiplicative - Additive	Multiplicative
Event	Multiplicative - Additive	Multiplicative
Holiday	Multiplicative - Additive	Additive
Target feature	**Value range**	**Transform**
Sold quantity	$0-\infty$	Log

Table 4.1: Hyperparameter set for NeuralProphet model, feature ranges and best found parameter value. Additionally, Target feature and applied transform.

4.2 XGBoost

To predict campaign performance with XGBoost, a dataset representing each campaign instance as a separate data point was constructed, utilizing approximated campaign instances from section 2.2. The features employed in training the model are enumerated in Table 4.2.
The transformed features include general attributes like range, distribution, and processing methodology. An initial univariate linear regression analysis was conducted to gauge the significance of these features in relation to the quantity of products sold during a campaign. Most features demonstrated significance at a 95% confidence level. However, considering XGBoost's capability to model non-linear relationships, features deemed non-significant in this linear context were still retained.
Transformed features include:

- **Runtime**: Duration of each campaign instance.
- **3 Month Sales**: Total sales in the three months preceding the campaign, filtered for sales from categories involved in the campaign.
- **Previous Year Sales**: Sales during the same period in the previous year as the campaign instance.
- **Discount Percent**: Calculated as $1-\frac{\text{Discount}}{\text{Revenue}}$, where 'Discount' and 'Revenue' are totals for a campaign instance. This metric aims to standardize the diverse

discount mechanisms across different offer types, under the hypothesis that higher discount percentages generally yield better campaign performance.
- **Start/End Day and Month**: Specific days and months marking the start and end of each campaign instance.

Categorical features such as "offer type" and "category IDs," though numerical, required special handling. One-hot encoding was chosen for its effectiveness in dealing with categorical data, enabling a straightforward inclusion of these features in the model.

Transformed features	**Value range**	**Distribution**	**Transform**	**p-value**	**Significant**
Runtime	$0-\infty$	Lognormal	Boxcox	2.33e-5	Yes
3 month sales	$0-\infty$	Lognormal	Boxcox	9.08e-35	Yes
Previous year	$0-\infty$	Lognormal	Boxcox	4.03e-1	No
Discount percent	$0-1$	Normal	Minmax	1.55e-19	Yes
Start day	$1-31$	Roughly uniform	Minmax	8.23e-4	Yes
End day	$1-31$	Roughly uniform	Minmax	6.33e-1	No
Start month	$1-12$	Roughly uniform	Minmax	3.09e-5	Yes
End month	$1-12$	Roughly uniform	Minmax	1.90e-6	Yes

Encoded Features	**Values**	**Encoding**
Offer type	1,2,3,5,6,10,22	One-hot
Category IDs	84 unqiue IDs	One-hot

Target feature	**Value range**	**Distribution**	**Transform**
Sold quantity	$0-\infty$	Log normal	Log

Table 4.2: Feature set for XGBoost model, feature ranges, distribution, transform and significance level.

To address the high variance of the target feature, we filtered the dataset, removing campaign instances with fewer than 250 or more than 10,000 sold products. This pruning resulted in a significant reduction of the data set, from the original 3,431 data points to a final count of 1,854 unique campaign instances. Our motivation for this pruning stems from the observation that the smaller campaigns likely represent instances of poor campaign segmentation, as discussed in Chapter 2. Additionally, these small campaigns typically have a minimal impact on the overall business performance and should not be the primary focus of our predictive modeling efforts. Similarly, we excluded campaigns with more than 10,000 sold products due to their exceptional nature, often associated with special events like Black Friday or post-Christmas sales, rather than reflecting normal business operations.

4.2.1 XGBoost implementation

The implementation of the XGBoost model was facilitated using the Scikit-Learn library [21], a versatile tool for machine learning in Python. We constructed a model pipeline where all features were transformed or encoded as shown in table 4.2. The log transform was chosen for the long tailed features since it is good at reducing right skewness in data. It compresses the long tail and expands the values closer to

zero, which often leads to a distribution that resembles a normal distribution more closely, which is desirable. Minmax transform was chosen simply for being a industry standard,it maintains the original distribution and makes comparison between features easy. For categorical features like Offer type and Category IDs, one-hot encoding was employed, increasing the feature space considerably but justified due to XGBoost's efficient handling of sparse data through its sparsity-aware splitting algorithm.

4.2.1.1 Hyperparameter optimization

Hyperparameter optimization was conducted using a random grid search approach with cross-validation, as implemented in Scikit-Learn. This process involved sampling various parameter combinations from the specified range (Table 4.3), resulting in the training of a total of 1000 models (200 samples with 5-fold cross-validation each). The parameters explored included learning rate (Eta), minimum loss reduction for further partitioning (Gamma), maximum tree depth (Max depth), minimum sum of instance weight in a child (Min child weight), maximum delta step (Max delta step), subsample ratio (Subsample), and L1 and L2 regularization terms on weights (Alpha and Lambda, respectively). The optimal values for each parameter, determined through this process, are indicated in the table.

Table 4.3: XGBoost parameter space, parameter range and best found value for each respective parameter.

Parameter	Range	Best value
Eta	$0.01 - 0.3$	0.2
Gamma	$0 - 4$	0.5
Max depth	$2 - 12$	4
Min child weight	$1 - 10$	2
Max delta step	$0 - 10$	1
Subsample	$0.5 - 1$	0.7
Lambda	$1 - 5$	3
Alpha	$0 - 2$	0.1

5

Results

5.1 NeuralProphet

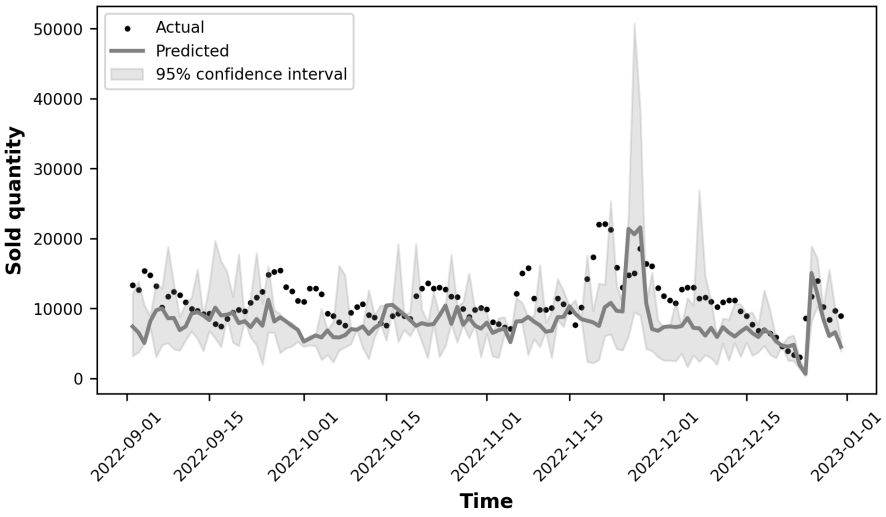
The performance metrics of the NeuralProphet models, detailed in Table 5.1, reveal a trend in predictive accuracy across different hierarchical levels. The meta model exhibits the highest accuracy with a Mean Absolute Percentage Error (MAPE) of 29%, followed by the category and product models with MAPEs of 38% and 51%, respectively. This increasing error trend down the model hierarchy is a notable observation.

In Figure 5.1, the sold quantity predictions for each model are depicted along with a 95% confidence interval. Further, Figure 5.2 illustrates the normalized error $\frac{y-\hat{y}}{y}$ distributions for the models. Across all models, a consistent pattern of underestimation in actual sold quantities is evident.

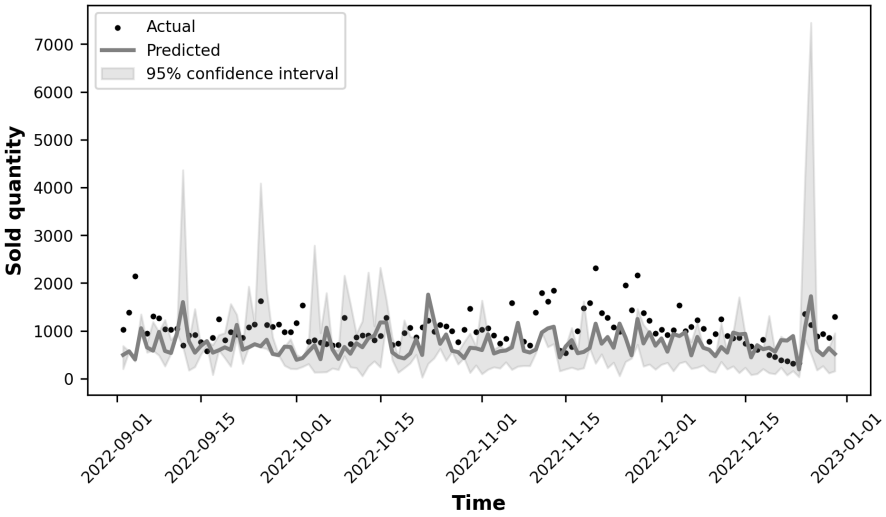
Analyzing the error characteristics, it is observed that both the magnitude and variance of errors escalate further down the hierarchy. The product model (Figure 5.1c) demonstrates significant day-to-day variance, indicative of a challenging prediction landscape. However, it is noteworthy that both the meta model and to lesser extent category model effectively capture major sales events such as Black week and the post-Christmas sale, whereas the impact of these events is less pronounced in the product model.

Metrics	**Meta model**	**Category model**	**Product model**
Mean absolute error	3551	502	205
Mean averege percent error	29%	38%	51%

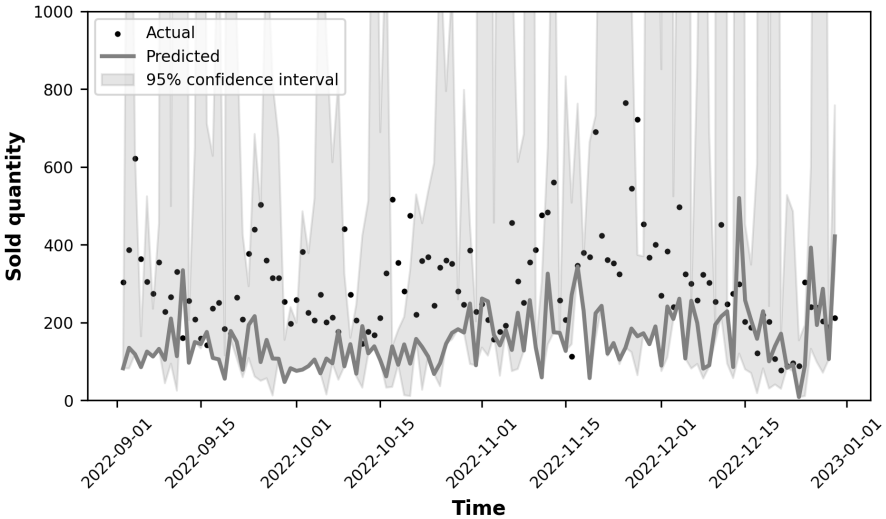
Table 5.1: Performance metrics for NeuralProphet models.



(a) Meta model prediction and actual sold product quantity for test set.



(b) Category model prediction and actual sold product quantity for test set.



(c) Product model prediction and actual sold product quantity for test set.

Figure 5.1: NeuralProphet hierarchy model result.

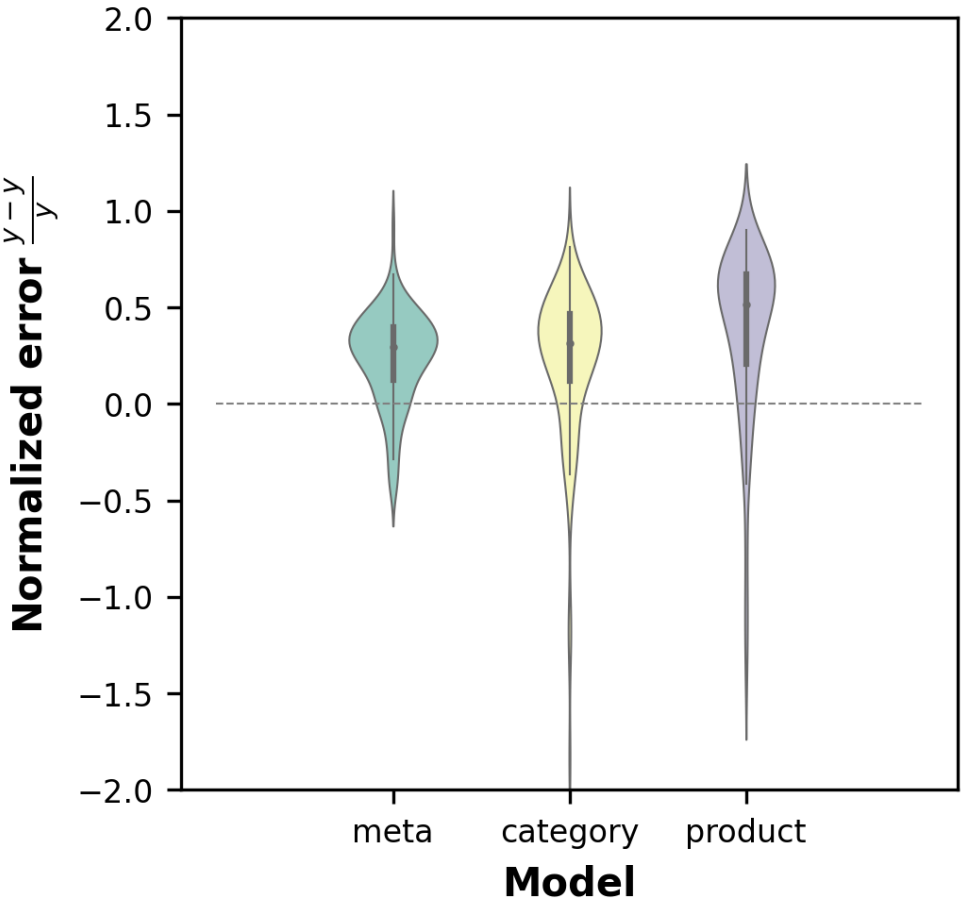


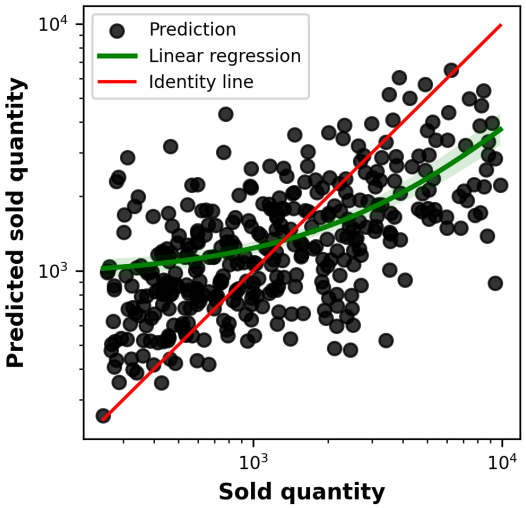
Figure 5.2: Normalized error for meta, category and product model.

To keep the results section focused on the key results, the seasonality analysis of the NeuralProphet model structure have been placed in appendix B.

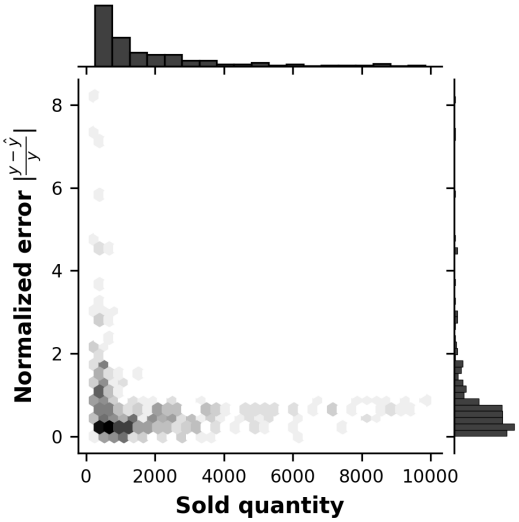
5.2 XGBoost

The primary outcomes of the XGBoost model are depicted in Figure 5.3. Analysis begins with a scatter plot comparing actual versus predicted campaign sales (Figure 5.3a), where a green regression line (with a 95% confidence interval) illustrates the relationship between these variables. This line deviates from the identity line, indicating a tendency of the model to overestimate sales for smaller campaigns and underestimate for larger ones. However, the model does capture campaign behaviors to a noticeable extent.
In Figure 5.3b, the normalized error $\frac{y-\hat{y}}{y}$ is plotted, revealing substantial errors in smaller campaigns. Despite this, the error distribution predominantly resides within the 0-1 range, suggesting a concentration of relatively smaller errors.
Key performance metrics are summarized in Table 5.4. The model shows a MAPE of 69%, and a Mean Absolute Error (MAE) of 1053. Given the typical range of campaign sales (1000-2000 units), these errors are notably significant.

Feature importance analysis, as shown in Figure 5.5, highlights 'Category IDs' and 'End Month' as the most influential features. Surprisingly, features like 'End Day', 'Start Day', 'Start Month', 'Same Period Previous Year', and 'Runtime' exhibit zero importance, which warrants further investigation.



(a) Sold quantity against predicted sold quantity



(b) Normalized error against sold quantity.

Figure 5.3: XGBoost model results.

Metric	Value
Mean absolute error	1053
R^2	0.24
Mean average percent Error	69%

Figure 5.4: Xgboost performance metrics

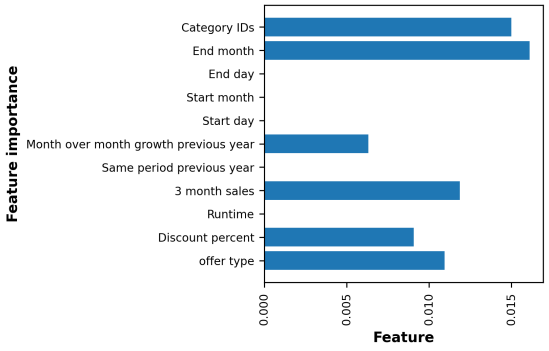


Figure 5.5: XGBoost feature importance.

6
Discussion

6.1 Result interpretation

The NeuralProphet model presents a nuanced set of results. The Meta model, aggregating all sales data, emerges as the most accurate, likely benefiting from reduced daily variance due to the aggregation of multiple sales sources. However, all models exhibit a systematic underprediction error (as shown in Figure 5.2), the causes of which are multifaceted. Potential sources of this error and future mitigation strategies are proposed:

- **Campaign Data Limitations**: The model currently incorporates the number of active campaigns using NeuralProphet's future regressor feature. This approach, while simplifying the model, overlooks the complexities of parameters for each campaign type, such as discount percentages and activation requirements. Future iterations could benefit from an encoding strategy that integrates these aspects without significantly expanding the feature space, thus avoiding issues with sparse data.
- **Variance Challenges**: Particularly in the product model, high daily variance could be contributing to prediction errors. More extensive forecasting on the product level needs to be conducted to determine if product level models are generally feasible. Implementing a moving average approach might help in smoothing out this noise, retaining broader trends while reducing day-to-day variability. The main concern with this method is the potential leakage of campaign and event effects into inactive periods, although this risk could be mitigated by using a smaller averaging window.
- **Cross-Pollination Effects**: The models currently do not account for cross-pollination effects between products or categories, which could be particularly relevant for category and product-level models. This phenomenon, stemming from cross-selling dynamics, can significantly influence sales. A feasible approach might involve calculating correlation values between products or categories and incorporating these as lagged regressors, weighted by their respective correlation strengths, to capture these interdependencies.

Shifting focus to the XGBoost model, results are bit more clear, the model performs quite poorly. This should maybe not come as a surprise considering the large approximation made when defining the campaign instances (Section 2.2) which might have induced large errors in the data itself. Comparatively this effect is less noticeable in the NeuralProphet model, considering that the NeuralProphet model takes effects such seasonality and holidays into account, and not only campaign instance

data, this seems reasonable. Additionally a fraction of the error can most likely be attributed to the large variance observed of sold products, from a couple of hundred products to tens of thousands the model with the current feature set can not support this variance. This is why we observe the over predication for small campaigns and under predictions for large campaigns as the model tries to be as general as possible. To solve this issue future modeling, a multiple model approach could be taken focusing on campaigns of different magnitude. Future analysis to determine if campaigns of different magnitude exhibit different purchase behaviours could be conducted to determine the validity of this alternative approach. Looking at the feature importance of the XGBoost model we interestingly find that the runtime of a campaign is not significant to its performance, instead features such as what categories are connected to the campaigns are much more important. This indicates that category specific purchase behaviours are very important to model campaign performance correctly, intuitively this makes sense since product such as underwear most likely exhibit different purchase behaviour from a winter jacket which might be perceived as less of an expendable piece of clothing.

6.2 Systemic issues and potential solutions

From the data analysis chapter there was the realisation that there are some inherent systemic issues with how the current platform system handles campaign data. As shown these issues leads to low confidence and accuracy in the results, when performing data driven analysis and modeling.

A critical problem identified is the practice of modifying the start and end dates of existing campaign instances rather than creating new ones. This approach leads to substantial data loss, since it is very difficult to accurately fix this in post processing, and thus hampers accurate campaign analysis. It is hypothesized that this behavior might stem from operational practices, where personnel extend successful campaigns for convenience. To address this, implementing a 'campaign template' system is recommended. This system would allow for easy replication of well performing campaign settings, while ensuring accurate assignment of sales to the appropriate campaign instances. This would result in a small amount of additional campaign data, but considering the substantial data volumes Viskan System handles this increase is dwarfed in comparison.

Additionally the handling of campaign parameters could be improved, currently the database only records the latest instance of all campaign related parameters, resulting in the loss of historical parameter data. This limitation renders any analysis or modeling that incorporates campaign parameters highly speculative, as it assumes consistency throughout a campaign run. A straightforward solution is to log all changes to campaign parameters for each instance. While this approach might increase the volume of data, especially if frequent changes are made, it would significantly enhance the accuracy and depth of campaign performance analysis.

Addressing these issues would benefit the platform as a whole, ensuring that Viskan Systems can leverage its data effectively, especially in scenarios where data-driven insights are pivotal for decision-making and strategy development.

6.3 SCR perspective

After the results was established as presented above, they were presented to SRC e-commerce manager during a meeting to get their perspective on the assumptions, analysis and results. This section will provide some much needed insight into the current workflow and sales strategy of SCR. The meeting was not recorded, nor transcribed, instead the following section will paraphrase SCR's response from notes taken during the meeting.

What became immediately obvious during the meeting is that SCR works very differently than expected, SCR stated that they don't operate like other clothing companies. SCR stated that they operate very personally with their products in a dynamic and reactive way, utilizing day to day micro management of the product offering to maximize sales. Additionally they don't heavily rely in any data analysis or business intelligence methods, stating that "The numbers become a comfort blanket". This is heavily contrasted by the purely data driven analysis performed in this thesis.

In the data analysis chapter it was observed that there had been a major change in SCR business model starting in 2022, where sales connected to campaigns increased from 40-50% to 86%. A more realistic estimate stated by SCR during the meeting is 40-50% during 2022, which is consistent with previous years. The consequences of this discrepancy is multifaceted, firstly it means that there is a flaw in, either in how SCR assigning products to campaigns or somewhere in Viskan System platform pipeline. Assuming there isn't some fundamental flaw in how the data analysis were performed. Further investigation is required to determine the source of the discrepancy. Secondly, this means that there are many sold products getting wrongly assigned to a campaign instance, potentially having a large impact on both the data analysis and modeling performed in this thesis. Addressing this discrepancy would improve the accuracy and integrity of further analysis and modeling efforts.

Variance and its effects have been heavily discussed in this thesis and SCR was able to provide some valuable insight. As mentioned SCR work very dynamically and this is potentially a major factor of the observed daily variance, SCR stated that "If there is a lot of rain forecast, we push rain coats" i.e. operating very reactivity to outside factors. SCR make use of loss leaders i.e. take a loss on some product to drive sales on other profitable products. SCR also stated that due to their low overhead, they have very good margins on products, allowing them to be aggressive and have dynamic in pricing if a product is not performing well enough. Additionally SCR stated that they are not afraid too take a loss on a product if warehouse space is needed for other products. Considering these highly dynamic behaviours the volatile variance observed is not to unexpected.

When asked about campaign operations SCR stated that metrics are not used heavily to evaluate their campaigns post finish. Instead SCR stated that they push a campaign, let it run its course, and by the time it ends they are already working the next campaigns. They instead follow all running campaign day to day, and if one is under performing they make some changes. SCR stated that they only look at sold products, always trying to keep things simple, and don't get lost in the numbers. When asked about the model features used for the XGBoost model SCR stated

that discount percent is not a good metric for their campaigns since the discount given is very product dependent. Additionally SCR stated that good products, with good prices, don't need strong discounts to sell. This statement really provides and excellent image of how SCR run their business. When asked what factors are most important to the campaigns they run, these factors were stated in no particular order

- **Relevance**: Products are trendy or otherwise relevant in the current market.
- **Product availability**: Enough product inventory to sustain a campaign with momentum.
- **Timing**: Somewhat connected to relevance, but timing the market with the launch of campaign or products is important.
- **Price competition**: SCR pricing compared to competitors is one of the biggest factors affecting their sales, always being cheaper and aware of competition pricing.
- **Marketing**: Strategic use of social media and email marketing to generate hype and spread awareness for product launches and campaigns.

Social media and email marketing was two of the factors brought up in the missing data section (Section 2.4). SRC was able to provide some much needed insight on the effects of these factors. SCR stated that email is their strongest marketing tool, SCR loyalty customers even receiving up to 3 mails on a daily basis. Email recipients receiving everything from product launches, restock information and campaign information. Social media was also stated as a highly important marketing tool for SCR, allowing them to operate on essentially on zero marketing costs. SCR uses social medias Facebook, Instagram and Tiktok, were the focus is not driving direct sales through product links (Often not even providing them, especially on Tiktok), but building brand recognition. Social media is also used to interact with customers and staying on top of customer needs and developing trends. Thus considering how large of an impact these marketing strategies have on SCR's sales performance, incorporating these features in future model development would be highly recommended.

Campaign placement online, as discussed in the missing data section, can have tangible effects on sales performance. SCR stated that overall they have a pretty good scroll rate i.e. how far down the average users scrolls on the platform, but always has one main campaign running at the top of the page to quickly capture customer interest. Interestingly SCR are the most active of all of Viskan System customers in regard to updating the platform content, providing additional context to the dynamic variance seen during analysis.

7
Conclusion

This thesis set out to evaluate whether the current database structure of Viskan Systems is conducive to data-driven decision-making and machine learning applications, particularly in the context of sales and campaign performance modeling. While the analysis covered only a fraction of the available data streams, it revealed significant potential for integrating data-driven approaches and machine learning technologies, given the breadth of data accessible to Viskan Systems. However, the investigation also uncovered systemic issues that introduce uncertainties into the data, highlighting the need for a comprehensive review and rectification of these challenges across all data streams to improve future data driven decision making and modeling efforts.

It is apparent that Viskan Systems' infrastructure, primarily optimized for high-traffic e-commerce operations, was not initially designed with machine learning applications in mind. As such, adapting their systems for advanced data analytics and machine learning will require some strategic modifications to ensure data integrity and compatibility with these technologies.

Regarding the sales and campaign models developed in this thesis, their applicability in a production environment remains to be assessed, potentially by a role such as a sales manager. The models, although not yielding optimal results, suggest several avenues for improvement. Addressing the identified data aggregation issues and integrating additional data features, like marketing strategies, could enhance future models' effectiveness. The proposed hierarchical modeling approach, as detailed in the methods section, offers a promising framework for capturing both macro-level sales trends and more granular product or category-level insights, potentially achieving a balance between model performance and explainability.

In conclusion, while the current campaign performance models faced challenges, primarily due to compromised data quality and lack of knowledge about how SCR's business structure operates. The models nonetheless demonstrate the feasibility of developing sophisticated modeling solutions within the Viskan Systems platform. With further development and refinement of the data aggregation highlighted, these models could significantly contribute to their system's enhancement and support data-driven decision-making processes.

Bibliography

[1] 201. *Impact of covid pandemic on eCommerce*. URL: https://www.trade.gov/impact-covid-pandemic-ecommerce.

[2] Snezhana Sulova. "A system for E-commerce website evaluation". In: *International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM* 19.2.1 (2019), pp. 25–32.

[3] Yunpeng Xiao et al. "Influence prediction model for marketing campaigns on e-commerce platforms". In: *Expert Systems with Applications* 211 (2023), p. 118575. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118575. URL: https://www.sciencedirect.com/science/article/pii/S0957417422016360.

[4] Petroc Taylor. *Data Growth Worldwide 2010-2025*. Aug. 2023. URL: https://www.statista.com/statistics/871513/worldwide-data-created/.

[5] Petroc Taylor. *Size of the big data analytics market worldwide from 2021 to 2029*. okt 2022. URL: https://www.statista.com/statistics/1336002/big-data-analytics-market-size/.

[6] Emilia Tanase. "3 Tips to Overcome Discount Fatigue in Your Email Marketing". In: *Mention Blog* (July 2023). URL: https://mention.com/en/blog/discount-fatigue-email-marketing/.

[7] Utpal M Dholakia and Sheryl E Kimes. "Daily deal fatigue or unabated enthusiasm? A study of consumer perceptions of daily deal promotions". In: *A Study of Consumer Perceptions of Daily Deal Promotions (September 11, 2011)* (2011).

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

[9] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *CoRR* abs/1603.02754 (2016). arXiv: 1603.02754. URL: http://arxiv.org/abs/1603.02754.

[10] Miao Zou et al. "Optimized XGBoost model with small dataset for predicting relative density of Ti-6Al-4V parts manufactured by selective laser melting". In: *Materials* 15.15 (2022), p. 5298.

[11] Kee-Young Kwahk and Xi Ge. "The Effects of Social Media on E-Commerce: A Perspective of Social Impact Theory". In: *2012 45th Hawaii International Conference on System Sciences*. 2012, pp. 1814–1823. DOI: 10.1109/HICSS.2012.564.

[12] Mayank Yadav and Zillur Rahman. "The influence of social media marketing activities on customer loyalty: A study of e-commerce industry". In: *Benchmarking: An International Journal* 25.9 (2018), pp. 3882–3905.

[13] Curzi Valerio, Lecoq William, and Quéré Noémier. “The impact of social media on E-Commerce decision making process”. In: *International Journal of Technology for Business (IJTB)* 1.1 (2019), pp. 1–9.

[14] Elizabeth Grandon and C Ranganathan. “The impact of content and design of web sites on online sales”. In: *AMCIS 2001 Proceedings* (2001), p. 179.

[15] Marc Anderson and Karën Fort. “Human Where? A New Scale Defining Human Involvement in Technology Communities from an Ethical Standpoint”. In: *International Review of Information Ethics* (Aug. 2022). URL: https://inria.hal.science/hal-03762035.

[16] Spyros Makridakis and Michèle Hibon. “The M3-Competition: results, conclusions and implications”. In: *International Journal of Forecasting* 16.4 (2000). The M3- Competition, pp. 451–476. ISSN: 0169-2070. DOI: https://doi.org/10.1016/S0169-2070(00)00057-1. URL: https://www.sciencedirect.com/science/article/pii/S0169207000000571.

[17] Oskar Triebe, Nikolay Laptev, and Ram Rajagopal. *AR-Net: A simple Auto-Regressive Neural Network for time-series*. 2019. arXiv: 1911.12436 [cs.LG].

[18] Barry De Ville. “Decision trees”. In: *WIREs Computational Statistics* 5.6 (2013), pp. 448–455. ISSN: 1939-5108. DOI: 10.1002/wics.1278.

[19] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[20] Leslie N. Smith. “No More Pesky Learning Rate Guessing Games”. In: *CoRR* abs/1506.01186 (2015). arXiv: 1506.01186. URL: http://arxiv.org/abs/1506.01186.

[21] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[22] Nicole Koenig-Lewis and Eberhard E Bischoff. “Seasonality research: The state of the art”. In: *International journal of tourism research* 7.4-5 (2005), pp. 201–219.

[23] Mustafa N. Gultekin and N.Bulent Gultekin. “Stock market seasonality: International Evidence”. In: *Journal of Financial Economics* 12.4 (1983), pp. 469–481. ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(83)90044-2. URL: https://www.sciencedirect.com/science/article/pii/0304405X83900442.

[24] J.C.F. Ehrenthal, D. Honhon, and T. Van Woensel. “Demand seasonality in retail inventory management”. In: *European Journal of Operational Research* 238.2 (2014), pp. 527–539. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2014.03.030. URL: https://www.sciencedirect.com/science/article/pii/S0377221714002690.

[25] Simeon Lisovski, Marilyn Ramenofsky, and John C Wingfield. “Defining the Degree of Seasonality and its Significance for Future Research”. In: *Integrative And Comparative Biology* 57.5 (2017), pp. 934–942. ISSN: 1540-7063. DOI: 10.1093/icb/icx040. URL: https://dx.doi.org/10.1093/icb/icx040.

[26] *Google Trends*. Nov. 2023. URL: https://trends.google.com/trends/explore?date=today%205-y&geo=SE&q=jackor,badkl%C3%A4der&hl=sv.

A

Supporting data analysis material

This appendix contains some more information and surplus material related to chapter 2 that didn't quite fit in the main body of the thesis.

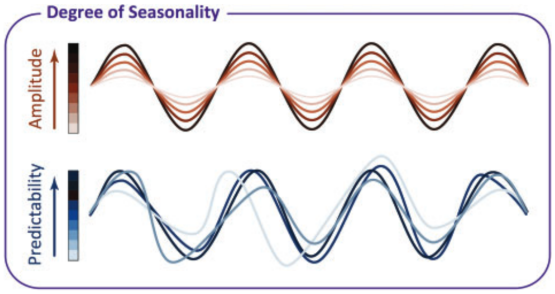


Figure A.1: Illustration of seasonal amplitude and predictability, the two main components of a systems seasonality. Taken from [25]

A.1 Time series seasonality

The concept of seasonality is nothing new, countless papers have been published tackling seasonality from different perspectives, everything from meta studies [22], proving its existence in a market [23], or modeling seasonal behaviour [24]. The degree of seasonality for any given system can be defined as the combination of its seasonal amplitude i.e difference in magnitude between low and high season, and the seasonal predictability i.e consistency of its period, these concepts are illustrated in figure A.1. For excellent introduction introduction and further reading into seasonality i refer the reader to Lisovski et.al paper on seasonality [25] written from a global environmental perspective.

By now the reader has hopefully understood that seasonality being a quite well studied phenomenon, unfortunately this does not mean that seasonality is easy to model or understand, this is due to it being highly specific. For example seasonality in the electric grid, weather, food consumption and retail all exhibit unique seasonal complexities which needs to taken into account when studying them. Another behaviour present in some seasonal system is that of induced seasonality, this can be in the form of seasonal goods such as ginger bread cookies, which are in many locations in Sweden only available during the winter season, this seasonality is then not from the lack of demand but the decision of the stores to remove the product from the market. A concrete example of seasonality in clothing retail can be seen in figure A.2 where relative search intensity from Google Trends for "Jacket" and

"Swimwear" in Sweden from the past 5 years can be seen. A strong seasonality exists for both product categories, jackets showing a peek around September trough January, and swimwear peeking heavily at the start of summer in July. For a retailer offering a very broad selection of categories such as SCR a complex system of seasonalities quickly form, some categories that in a vacuum does not show seasonality, might exhibit it due to purchase correlation with a heavily seasonal category.

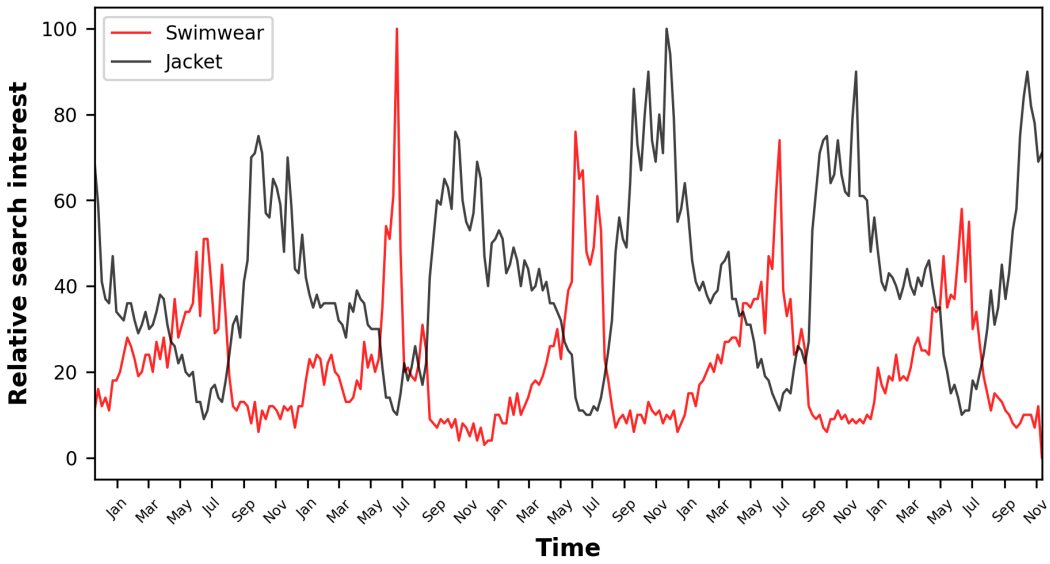


Figure A.2: Relative search interest over the past 5 years. A value of 100 indicates the highest interest for the period shown. Source: Adapted from [26].

A.2 Campaign runtime segmentation

This appendix chapter contain some more details regarding the campaign segmentation analysis discussed in section 2.2. Figure A.3 displays the probability distribution off $\Delta\mathbf{T}_d$, we observe that it represents some form of a long tail distribution. We also observe that $\Delta\mathbf{T}_d = 1$ is very close to 1 and as $\Delta\mathbf{T}_d$ there is a sharp drop off in $P(\Delta\mathbf{T}_d)$.

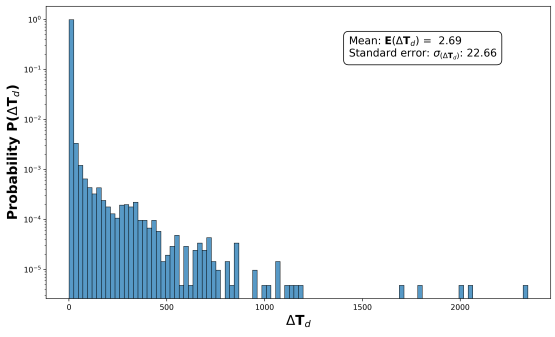


Figure A.3: Histogram showing the probability distribution off $\Delta\mathbf{T}_d$

To gain a better understanding of $\Delta\mathbf{T}_d$ the cumulative distribution function (CDF) off $\Delta\mathbf{T}_d$ was plotted aswell

which can be seen in figure A.4. Using the presented CDF we calculate that the probability of $\Delta\mathbf{T}_d < 7 = 0.976$, in other words that means that 97.6% of all days without sales are 7 days or shorter. Based on this discovery it was decided that 7 days will be the break point for the campaign segmentation rule.

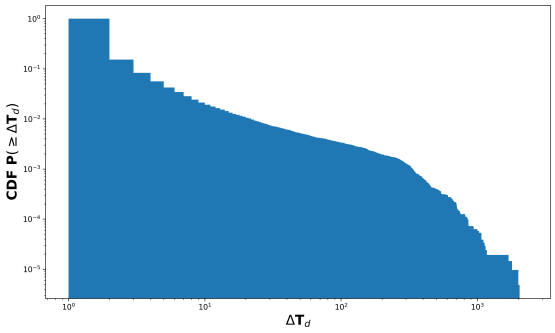


Figure A.4: Histogram showing the cumulative distribution function off $\Delta\mathbf{T}_d$

Figure A.2 shows a set of example campaign and their respective runtime segmentation as a result of the method described in section 2.2. To highlight a few behaviours of these examples in figure A.5a we observe that the latest instance of the campaign shown in green is only a small portion of the complete time series for this campaign, according to the data it had a runtime off 43 days, but after removing the red segment according to the segmentation rule the actual runtime of the campaign was 373 days. This shows a discrepancy of almost a order of magnitude which would have a great effect on the results when normalizing campaign results by its runtime. Similar results can be seen in figures A.5b, A.5e, A.5f.
Figure A.5c shows another type of behaviour, here the campaign is active for the whole time series shown by the green overlay but has a segment of 0 sales, shown by the overlapping red segment. This is a result of the campaign being active in the data base but temporarily un-published by the managers of SCR e-commerce platform. This causes the campaign to not be seen or usable to customers which naturally results in there being 0 sales related to the campaign. Unfortunately these segments of the campaign being non-public is not recorded in the data. Similar behaviour can be seen in figure A.5d, A.5g.
Lastly as seen in figure A.5h there are situations in which the campaign has only been run one time which results in the start and end date in the data base providing the correct runtime for the campaign.

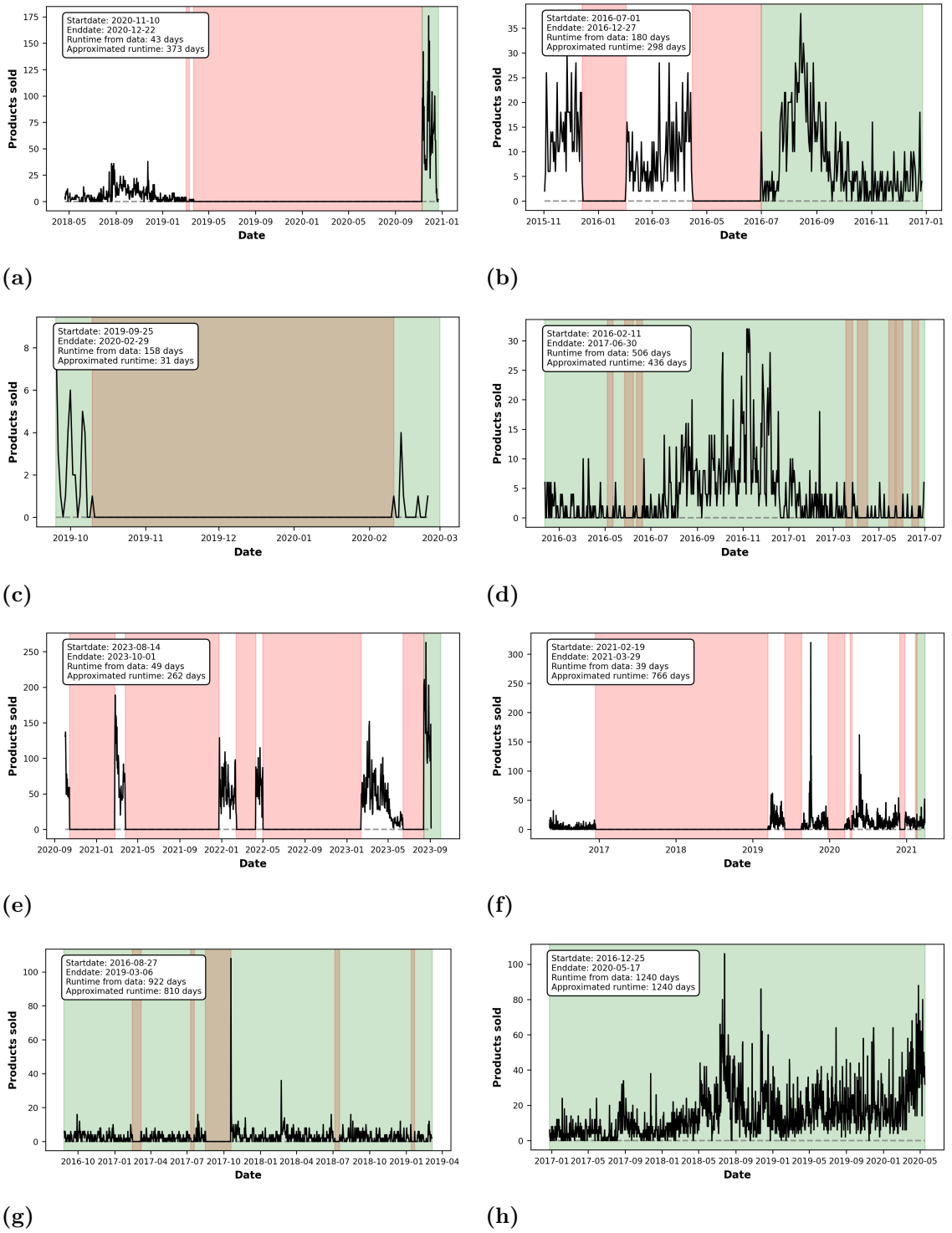


Figure A.5: Estimated runtimes for selected campaign examples illustrating varied behavioral patterns. Each figure presents the duration of a campaign with its start and end dates, as indicated by the green segments according to the data. The red segments highlight the periods when the campaign was inactive, as determined by the applied runtime segmentation rule. Additionally, the figures depict the runtime for each campaign, according to both data records and the segmentation rule.

B

NeuralProphet seasonality

Figures B.1, B.2, and B.3 depict the seasonal components extracted by the NeuralProphet model system. The meta model's seasonality patterns align with the general sales data, both on yearly and weekly scales (Fig. 2.1d and 2.1a). This correlation is expected given that the meta model is derived from the aggregate dataset.

More intriguing are the distinct seasonal behaviors observed in the category and product models (Fig. B.2, B.3). The category model, which focuses on the "Basic" category, demonstrates peak sales during the winter and a decline in summer. In contrast, the product model shows an almost inverse pattern, suggesting higher popularity in the summer months. While the specific product represented in this model remains is unknown due to the anonymization, the data suggests it likely to be a summer-favored item, such as shorts or a similar product. As for the weekly seasonality, the category model shows very little weakly variation (4%) while the product model exhibits up a large difference between between best and worst days (20%). This analysis shows that products and its parents category can exhibit clearly different seasonal behaviours, which should not be too surprising considering categories can contain a varied assortment of products.

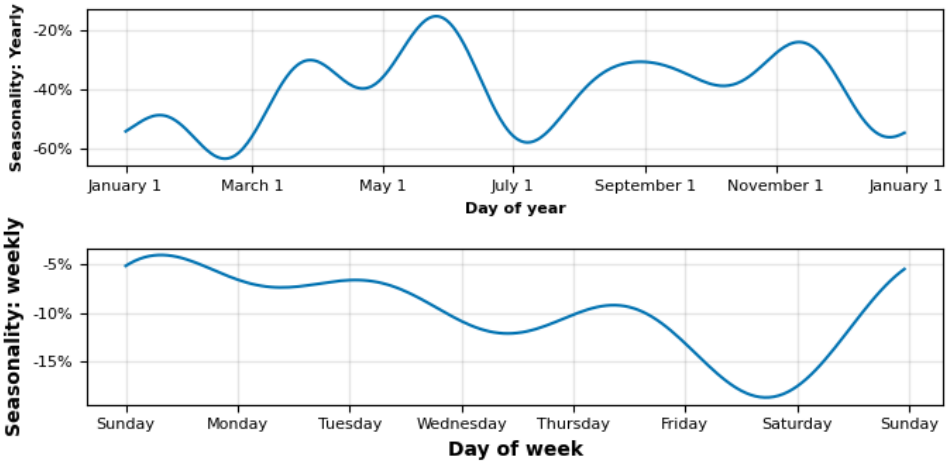


Figure B.1: Meta model seasonality component.

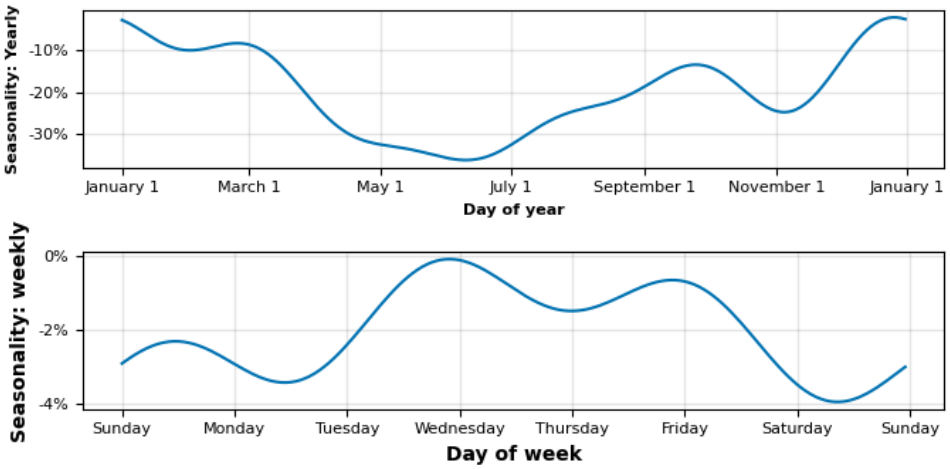


Figure B.2: Category model seasonality component.

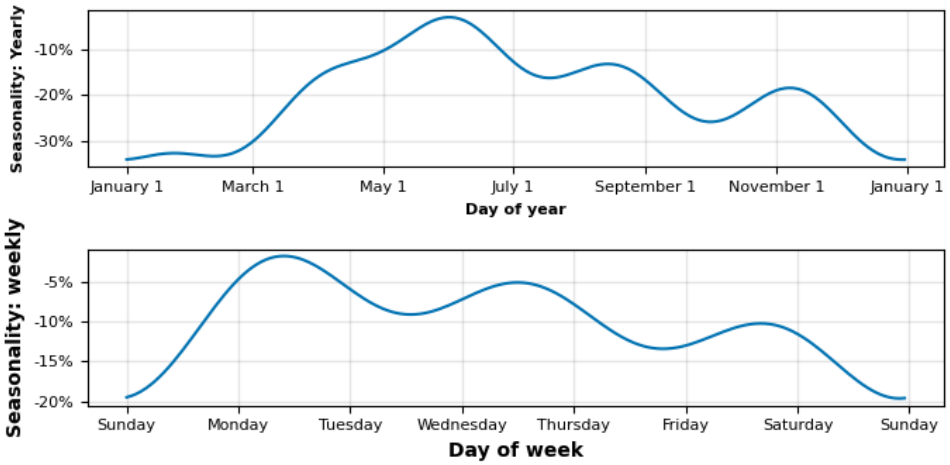


Figure B.3: Product model seasonality component.

C

Offer types

- **Offertype 1**: New price X - Product A receives a new temporary price X.
- **Offertype 2**: Discount X percent - Product A receives X percent in discount.
- **Offertype 3**: Buy X pay Y - Buy X units of product A, pay Y.
- **Offertype 4**: Buy X pay for Y - Buy X units of product A, pay for Y units.
- **Offertype 5**: X percent discount on one article - A article gets X percent discount before checkout, A is chosen by campaign settings.
- **Offertype 6**: X percent discount on everything - Whole shopping cart recives X discount before checkout.
- **Offertype 7**: Buy for X get Y discount for every X - Buy items for X price and receive Y discount for every multiple of X spent.
- **Offertype 8**: Carriage discount - Free shipping depending on some campaign setting.
- **Offertype 9**: Buy X units get Y discount - Buy X units of product A and receive Y discount.
- **Offertype 10**: Buy A and B pay Z for both - Buy product A and B and pay X.
- **Offertype 11**: X percent extra discount - X percent extra discount on some product or category according to campaign settings
- **Offertype 12**: Change prentype?
- **Offertype 13**: Buy X different articles get Y discount - Buy X amount of unqiue articles and recive Y discount at checkout.
- **Offertype 14**: Buy X pay for Y leave expensive - ??
- **Offertype 15**: Buy products for X get Y discount one or many times - ??
- **Offertype 16**: Buy for X get Y discount on total value - Buy products for X get Y discount on total pruchase value.
- **Offertype 17**: Buy for X get free product from category A
- **Offertype 18**: Buy X units get Y percent discount on one item -
- **Offertype 19**: Buy for X get Y discount on total value -
- **Offertype 20**: Bonus ladder buy X units and get Y percent discount.
- **Offertype 21**: Subscription discount per delivery -
- **Offertype 22**: Buy for X get free article from Y category add if only one.
- **Offertype 23**: Buy X get free article from category A

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY