Building Inventory Simulations for High Velocity Garment Retail Stores

by

Davy Qi

Submitted to the MIT Sloan School of Management and
Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Civil and Environmental Engineering

in conjunction with the Leaders for Global Operations program
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2024

Author by: Davy Qi
MIT Sloan School of Management and
Department of Civil and Environmental Engineering
May 10, 2024

Certified by: Georgia Perakis
William F. Pounds Professor of Management Science; Codirector, Operations Research Center
Thesis Supervisor

Certified by: Patrick Jaillet
Professor of Civil and Environmental Engineering; Codirector, Operations Research Center
Thesis Supervisor

Accepted by: Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering; MacVicar Faculty Fellow; Graduate Officer

Accepted by: Maura Herson
Assitant Dean, MBA Program, MIT Sloan School of Management

Building Inventory Simulations for High Velocity Garment Retail Stores

by

Davy Qi

Submitted to the MIT Sloan School of Management and
Department of Civil and Environmental Engineering
on May 10, 2024, in partial fulfillment of the
requirements for the degrees of
Master of Business Administration

and

Master of Science in Civil and Environmental Engineering

Abstract

To facilitate agility in store inventory planning for a brick-and-mortar retail business with high sales velocity and product portfolio complexity, this project created a Monte Carlo tool that simulates how upstream shipment decisions impact capacity utilization and product complexity. The simulation model was built in two steps, first a Monte Carlo model for aggregated store inventory, followed by machine learning models that predict the display inventory and the number of store and display unique articles based on Monte Carlo outputs. In the process of building the Monte Carlo model, the project examined methods to model inventory trends, developed a quantification technique for daily demand stochasticity, and explored possibilities to control the simulation stochasticity. These methods and techniques, novel to retail inventory modeling, were able to model store inventory with little systematic biases and store daily mean absolute inventory deviations within 2-4%. Meanwhile for the machine learning models, the project systematically examined the efficacy of linear regression, tree and fully connected neural network models at making time series predictions using two time series as inputs. It also rigorously dives into the limitations and advantages of various model architectures, including the selection of variables, treatment of multiple time series, order of predictions, and the scope of loss functions. The final machine learning model results showed some systematic biases with daily mean absolute deviation ranging from 3-10% for display inventory and up to 10-20% for unique articles.

Thesis Supervisor: Georgia Perakis
Title: William F. Pounds Professor of Management Science; Codirector, Operations Research Center

Thesis Supervisor: Patrick Jaillet
Title: Professor of Civil and Environmental Engineering; Codirector, Operations Research Center

Acknowledgments

I would like to thank my colleagues on the business analytics team at Zara, especially Candela and Belen for helping me navigate the business. Additionally, huge thanks for my advisors, Georgia and Patrick for your guidance throughout the project.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Glossary

Section: there are a total of three sections, women (section 1), men (section 2) and kids (section 3)

Upstream: company internal movements from the perspective of stores, including shipments, backstocks and transfers

Downstream: customer facing movements from the perspective of stores, including sales and returns

Shipment: broadly refers to shipment of products to stores, including new article shipment and existing article replenishment

Replenishment: shipment of products that already exist in stores (in contrast to new article shipment)

Backstock: return of products from stores to warehouses

Sales: sales of products to customers, including a variety of online and in-person channels

Return: return of products from customers, including products purchased from a variety of online and in-person channels

SINT: orders placed online and shipped from store inventory directly to customers, as opposed to from warehouse inventory

IPOD: orders placed in-person in store (on tablet devices, hence the name) but shipped from warehouses, mostly due to lack of store inventory

Click and Collect: orders placed online and picked up in-person with inventory coming from warehouses (shipped to store); note that there is the option to place orders online for existing store inventory to be picked up in-person as well, but is categorized as in-store sales

CD1: a type of distribution center that ships new and existing articles to stores around the world

CD2: a new type of secondary distribution center for certain stores

Cycle 1 and cycle 2: all stores receive two new (not already in stores at time of shipment) product shipments twice a week from CD1; CD1 shipments happen on Mondays and Thursdays, barring few exceptions, and the period of Monday Tuesday Wednesday is referred to as cycle 2 and Thursday Friday Saturday Sunday as cycle 1.

Chapter 1

Introduction

For a brick-and-mortar retail business with omni-channel fulfillment, high sales velocity, product portfolio complexity, limited store capacity, and inherent demand stochasticity, agile store inventory planning is crucial to its success. To facilitate such agility in shipments, backstocks, transfers and in-store movements, this project aims to create a tool that simulates how these decisions directly impact store operations (i.e. capacity utilization and product complexity), in order to inform better store inventory planning.

1.1 Business context

Originally started in northwestern Spain in 1975 by founder Amancio Ortega, who had been running a dressmaking workshop since 1963, Zara is now a multi-national fashion retailer with over 1800 stores around the globe [7]. Organizationally, Zara has been a part of the Inditex Group since 1985, which was started by the same founder and currently owns fashion brands beyond just Zara [7]. With an annual revenue of 32.6 billion euros in fiscal year 2022, over 70% of which coming from Zara (including Zara Home), Inditex is the biggest company by market capitalization in Spain [6]. Geographically, Spain accounts for 14% of Inditex revenue, the rest of Europe 48%, the Americas 20%, and Asia and the rest of the world 18% [6].

A key differentiator for Zara is its large and constantly evolving assortment. Dozens or hundreds of new products arrive in every store weekly, with differing assortments for each store to cater to demand. However, managing such a dynamic business comes

with its challenges. The need to frequently design, source and replenish new products means constant commercial and operational planning. In addition, a larger assortment also implies lower volume for each product, which leads to higher variability.

Although traditionally a brick-and-mortar retails, Zara has ventured into online business, which currently accounts for over 20% of revenue [6]. Online business spans a number of formats. The point of sales can be either online or in-person, orders can be fulfilled in warehouses or stores and products can reach customers via parcels or be picked up in-person. Along with similar considerations for returns, the online business adds significant complexities to store and warehouse operations.

1.2 Project motivation

Inventory planning is at the heart of any retail business. Too much inventory results in high holding and overage cost, as well as difficulty in capacity management. Too little inventory leads to lost sales. In addition to quantity, positioning the right inventory at the right place at the right time is also highly crucial, since different assortments are not substitutes for one another. As a result, it is crucial to understand how retail store inventory will behave in order to more accurately inform store operations planning and shipment decisions.

Zara currently does not have the capability to precisely model future store inventory. Store inventory is driven by two sets of inputs and outputs, downstream (customer facing) and upstream (company internal) movements. Future downstream movements are currently modeled using machine learning algorithms in the form of sales and returns forecasts, although they can be on various levels of granularity and not always available for all channels of sales and returns. Upstream movements are driven by real-time downstream movements and dynamically calculated for a pre-specified shipment schedule, with some real-time manual inputs. Using the aforementioned forecasts and shipment schedule, one can get a sense of how inventory will evolve in the coming weeks for each store, but not precisely. Details of various sales and returns channels are discussed in Chapter 2.

Motivated by this lack of modeling capability, this project sets out to build a tool that integrates modeled downstream movements, upstream movement patterns, as well as other store and product attributes, in order to understand how store inventory will vary precisely on a daily basis for a forward-looking period. With such a tool, store

operations managers would hopefully have more knowledge at their disposal to inform decisions that impact how stores should be served.

A specific dynamic at Zara is the incorporation of CD2 warehouses in their distribution network. Most stores receive clothing products twice a week from CD1 warehouses, including new and existing articles (those that don't exist in stores at the time of shipment and those that do). For a small set of stores with tight capacity constraints, like those in busy commercial districts with small stock rooms and high sales, such a shipment model can cause significant operational stress, as each bi-weekly shipment contains large volume of inventory. The connection to CD2 can often be a capacity solution for these stores, as the store can receive smaller and more frequent shipments from both CD1 and CD2. Therefore, another motivation for the store inventory model is to be able to quantify the operational benefits of CD2 replenishment through simulations, which causes less operational disruptions, compared to the alternative of putting actual stores through pilots.

1.3 Previous LGO work at Zara

For the last 10+ years, approximately one LGO thesis project per year was carried out at Zara. Previous work tends to focus on two areas: 1) improving forecasting capabilities 2) improving inventory shipment and fulfillment policies. A quick synopsis of thesis projects in the past few years is provided below:

- The 2023 thesis examines possible ways to featurize new articles, using verbal descriptions of the garments and a combination of natural language processing (NLP) techniques [15]. New articles always pose challenges for demand forecasting due to the absence of historical data. By exploring potential correlations between the verbal descriptions of articles and demand profile, the project aims to provide better forecast accuracy for new articles.

- The 2022 thesis explores an optimization-based heuristic that dynamically calculates optimal days of inventory coverage [17]. By using static estimates of inventory coverage and customer behavior features, the heuristic is able to reduce inventory with minimal impact on stockouts in a simulation.

- The 2019 thesis focuses on a optimization heuristic approach to calculate integrated target inventory across both online and physical channels [14]. By looking

at the channels holistically, the heuristic is able to outperform single channel approaches in a simulated cost analysis.

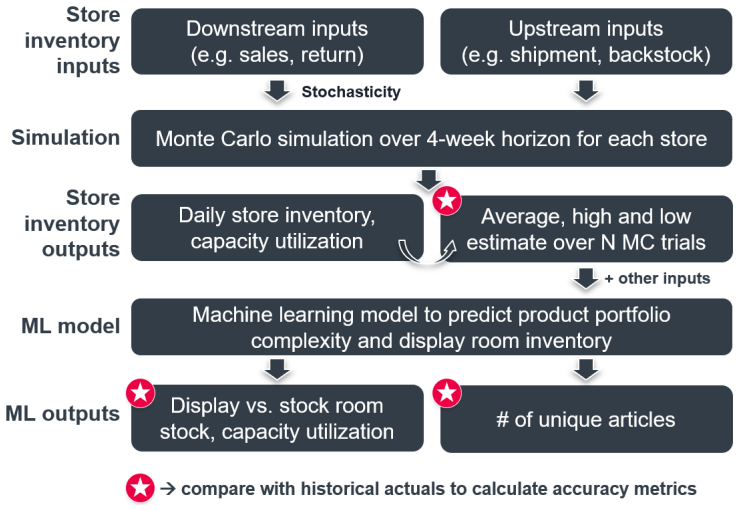
- 2018 saw two LGO theses at Zara. One focuses on improving demand forecast using e-commerce data, such as opt-in user tracking data and article age data [5]. The other explores the potential to increase system-wide profits through a fulfillment model that leverages multi-period demand information [4].

Fundamentally different from previous LGO theses, this work does not seek to make direct improvements on business operations. Instead, the goal is to build a simulation model that accurately approximates store operations, which can be used to inform real-time decision-making. As a result, success of the project is not measured by how much incremental improvement can the model achieve on top of current operations. Instead, it is measured by how accurately can the model simulate current operations. The higher the accuracy, the more reliable the model will be to inform future decisions. The literature review for topics relevant for the project will be covered in Section 3.1.1 and 4.2.

1.4 Project methodology and contributions

The simulation model was built incrementally in two steps. An illustration of the simulation model architecture is provided in Figure 1-1.

Figure 1-1: Illustration of simulation model architecture



First, a Monte Carlo simulation model for aggregated store inventory was built. The model uses upstream inputs (i.e. shipments, backstocks, transfers) and downstream

inputs (i.e. sales, returns), calculates daily inventory changes, and arrives at daily inventory level for each product category in the store. The stochasticity of the model comes from the inherent uncertainty in customer behavior and downstream inputs, which is quantified through historical calculations and simulated through a Monte Carlo sampling process that generates downstream inputs for each trial of simulation. Due to the complexity of the business and simulation task, a large number of assumptions were adopted in the model. In order to guide the assumptions, a set of customized accuracy metrics was introduced to quantify model performance against a historical time period, when store inventory data is available. Across 6 stores over a recent time period, no significant systematic biases are observed, while the daily mean absolute inventory deviation remains under 2-4% for the store total and under 2-8% for the section total (i.e. men, women, kids). Detailed discussion of the Monte Carlo modeling process and results are covered in Chapter 3.

On top of the Monte Carlo model, machine learning models that take store inventory as inputs were built to predict the display inventory and the number of unique articles in both the store and display room. Due to data and modeling complexity, similar Monte Carlo approaches face logistic challenges and are not expected to perform well. With machine learning approaches, correlations with store inventory could be learned using historical data and applied to the store inventory output of the Monte Carlo model to predict the display inventory and store and display number of unique articles for the simulation period. Thus, the machine learning models in conjunction with the store inventory model can provide a complete picture of inventory and product complexity relevant to store operations. Across the same 6 stores and time period, machine learning outputs perform worse than store inventory outputs. A slight upward bias is observed for display inventory and a downward bias for store and display unique articles. Daily mean absolute deviation ranges from 3-10% for display inventory and can be up to 10-20% for unique articles, which is quite significant considering how little number of unique articles tends to fluctuate within a few weeks. The outcome is not unexpected, given the limit numbers of predictive features available, the complexity of the prediction tasks, and the machine learning models using store aggregate inventory as inputs, which is a Monte Carlo output that already contains inaccuracies. Detailed discussions of the rationale behind machine learning methods, choice of machine learning models and model performance are covered in Chapter 4.

Last but not least, the Monte Carlo model and the machine learning models are integrated in a dashboard format to facilitate usability. The dashboard design is

discussed and presented in detail in Chapter 5.

Overall for the Monte Carlo inventory simulation, the project examined various methods to model inventory trends, developed a quantification technique for daily demand stochasticity, and explored possibilities to control the stochasticity in the simulation. These methods and techniques, novel to inventory modeling in a retail context, are able to achieve reasonable modeling accuracy for the specific business of Zara.

For the machine learning models, the project systematically examined the efficacy of linear regression, tree and fully connected neural network models at making time series predictions using two time series as inputs. It also dives into various model architectures, including the selection of variables, treatment of multiple time series, order of predictions, and the scope of loss functions. This rigorous endeavor shed light on the various limitations and advantages of each model and architecture, which allowed the project to arrive at a recommendation for a model that is most suitable for an inventory simulation tool.

Chapter 2

Navigating the data landscape

This chapter lays out all of the data used in the project. All data is pulled directly from Zara's enterprise systems, using a combination of Scala, SQL and PySpark. The data discussion is structured with the Monte Carlo model in mind, but is inclusive of data leverage by the machine learning models, as it is a subset of data used in the Monte Carlo model. The specifics of machine learning data processing and featurizations are discussed in Section 4.4.

2.1 Historical store stock and volume flow data

To simulate inventory level and product complexity, it is necessary to know historically how much inventory there is and how much flows in and out of every store on a daily article level.

Two types of store inventory data are utilized at Zara:

- RFID inventory: actual inventory counted by scanning the RFID tags in physical stores. Despite RFID inventory being the "actual" inventory, it is by no means the single source of truth, because RFID tags and scanners can malfunction, products can be misplaced, and store associates may improperly conduct the scanning process. RFID inventory data here is able to distinguish between articles as well as whether they are placed in stock or display room.

- Theoretical inventory: calculated by summing all the store volume in and out flows. Due to the discreteness of fashion retail inventory, data for volume in and

out flows IS generally considered to be accurate. Nonetheless, it is impossible to fully capture the dynamics of physical stores through theoretical data, as volume flow can still be improperly recorded.

Store volume in and out flows (inventory movement data), which are used in theoretical inventory calculations, can be broken down into the following two categories:

- Upstream movements (company internal movements): shipments from warehouses, backstocks to warehouses and transfers between stores.
- Downstream movements (customer facing): (gross) sales to customers, returns from customers.

The identification of upstream movements is straight-forward, as all company internal movements are recorded under different movement codes at Zara with clear identification of origin and destination. In contrast, special cautions need to taken when analyzing downstream movements. For the purpose of the project, any downstream movement that leads to changes in physical store inventory levels is relevant. Due to the omni-channel fulfillment model of the business, articles purchased online many come from store inventory with the point of sales recorded as the e-commerce warehouse, and vice versa. Similarly, articles returned physically to stores can be purchased from any sales channel with volume attributed to the point of sales, not point of returns.

For sales to customers, the following precautions need to be taken:

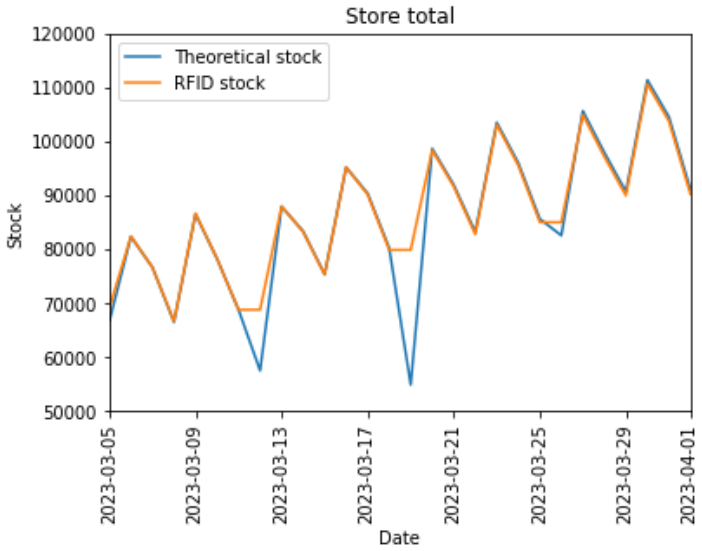
- In-store sales are relevant. Point of sales is always the stores where physical inventory comes from.
- SINT sales are relevant. Point of sales needs to be the stores where physical inventory comes from, not the e-commerce warehouse.
- Click and Collect sales are not relevant. Inventory comes from e-commerce warehouses, even though point of sales may be recorded as stores.
- IPOD sales are not relevant. Inventory comes from e-commerce warehouses, even though point of sales may be recorded as stores.

For returns from customers, all channels of sales are relevant, including in-store, SINT, Click and Collect, IPOD, and general online sales, and should be recorded against

the point of returns. Specifically for in-store sales, articles that are returned to the same store or other stores both need to be taken into account in general but are only relevant if the store of interest receives the returned articles.

To ensure that the inventory movement data is accurate, daily movements can be aggregated to calculate theoretical inventory for comparison with RFID inventory. Results for a particular store in Spain over a 4-week period are shown in Figure 2-1. Despite small differences on select days, the two sources agree with each other very well, thus validating the accuracy and completeness of historical inventory data sources. Note that the days with the most differences are Sundays and RFID inventory remains the same on Sundays as the leading Saturdays, which is not surprising given that Sunday is usually a day with the least retail activity in Spain and stores are generally open for shorter hours, if not completely closed to foot traffic in many cases. Therefore, one can reasonably conclude that the cause for discrepancy between the two sources on select days is simply that RFID inventory is not recorded and updated on Sundays, especially given that the two data sources converge immediately the following Mondays. In other words, the data discrepancies on select days between the two sources are not of concern and should not lead to overall biases.

Figure 2-1: Comparison of theoretical inventory constructed from daily inventory movements and RFID inventory for a particular store in Spain over a 4-week period



2.2 Forecast data

Because the goal of the project is to simulate store operations for a forward-looking 4-week period, forecast data is necessary in addition to historical data, for both sales and returns. Upon discussions with relevant teams, the only available sales and returns forecasts that meet the forward-looking time horizon are provided on section level (i.e. women, men and kids). They include products beyond just clothes (i.e. shoes) and do not strictly take into account all the downstream volume in and out flows that impact store inventory (i.e. SINT). The inconsistencies with historical sales and returns data laid out in Section 2.1 are not ideal, but are not of major concerns, because forecast data is not used verbatim in the simulation. Instead, forecast accuracy parameters are calculated to adjust for systematic biases between forecasts and actuals, which feed directly into the Monte Carlo sampling of the stochastic simulation. The sampling process is discussed in detail in Section 3.1.3.

The forward-looking section-level forecast described above is used in the business for medium term planning, but do not directly impact shipment decisions on a day-to-day basis. Section 3.1.7 lays out the simulation model requires a mechanism to correct shipment volume based on downstream stochasticity in the simulation. One of the ways to model such a mechanism is through the calculation of historical shipment volumes and how they actually react to higher/lower demand and higher/lower inventory level. Actual shipment volume in the business is currently mostly driven by an article level one-week forward-looking forecast grounded in weighted average historical sales, commonly referred to as VMP (ventas medias propuestas, or average suggested sales). This source of data is also utilized in the project.

2.3 Other miscellaneous data

2.3.1 Article master

Article master that contains necessary mapping of various article codes to article attributes (e.g. buyer categorization, family categorization, hanging vs. folding) is leveraged throughout the project. Additionally, article codes for the same product may change within the same campaign or across different campaigns, often due to shift in suppliers. The article master also serves the purpose to reconcile these different codes as each article should have a unique identifier.

2.3.2 Location master

Location master contains necessary mapping of location numbers to attributes (e.g. name, location, types of location).

2.3.3 Store capacity

In order to translate inventory level to capacity utilization, quantification of store capacity is necessary. Store capacity is currently modeled separately by section and by hanging stock room, folding stock room and display room. Each capacity quantity is dynamic over time, as room staging and product mix can vary. For example, capacity tends to be lower during the winter campaign due to winter clothes being bulkier than summer clothes. For simplicity, store capacity is not treated as time-dependent in this project as they don't tend to vary much over the simulation time horizon of a few weeks.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Monte Carlo simulation model for store inventory

The project first simulates daily store aggregate inventory for each buyer-family combination in the store over a 4-week horizon using a Monte Carlo approach. A conscious decision was made to model stores on aggregated buyer-family levels.

First of all, the product portfolio of Zara is highly complicated, with up to hundreds of thousands of unique articles sold every year and most having life cycles of a few weeks in physical stores. An article level model would not only require a deep understanding of historically how new products are released and how old products are phased out, but also predictions of when and how many new products in each product category will impact store inventory over forward-looking simulation horizons, dramatically increasing the model complexity. Therefore, building a reasonable store simulation on an article level within the time frame of the project was deemed impossible, leaving buyer-family level as the most granular option that is still reasonable.

Secondly, similar inventory behavior can be reasonably expected from articles within the same buyer and family categorization. Articles are categorized into approximately 20 buyers, 90 families and 320 buyer-family combinations across all three sections. Buyer categorization indicates natural divisions between styles of clothing (e.g. basic vs. athletic) and do not span across sections, with each buyer having their own designs and suppliers. On the other hand, family categorization is used to distinguish between types of clothing (e.g. t-shirts vs. pants). Although the same family of products may come from different buyers, they tend to share similarities in demand profiles, such as

seasonality.

Thirdly, the simulation requires quantification of sales and returns forecast accuracy and inherent stochasticity in sales and returns, which are calculated from historical data. With 300+ buyer-family combinations, the majority of which having little daily volume for a single store, collecting enough data for such quantification is already challenging. In fact, a volume cutoff threshold was already applied to each buyer-family for each store, to ensure data availability. Therefore, disaggregating the modeling beyond buyer-family levels would only be a disservice to the simulation fidelity.

Additionally, a 4-week horizon is also chosen because it strikes the balance between providing medium term insights that are currently unavailable, and keeping the simulation tractable. Depending on the store, it may receive 2-6 shipments a week, and shipment decisions are made one at a time, the night before the shipment, based on the latest information up to that point. Therefore, store managers don't typically have insights into store capacity more than one week in advance, or sometimes even a few days. A 4-week window would provide a view into the medium term future that is not currently available.

On the other hand, building simulations of much longer time horizon can also be problematic. There are two campaigns each year, interspersed with end of campaign sales, promotions and holiday sales throughout the year. "Steady state" operations within a year typically only span two 3-month periods, one starting in March and the other in September. Any simulation on the order of 3 months or longer would have to take into account the complicated dynamics of changing consumer behaviors and corresponding operational responses due to sales and promotions, which are not the main focuses of the project.

Lastly and most importantly, a forward-looking simulation requires some understanding of future sales and returns, which are modeled by demand forecasts in this case. Existing sales and returns forecasts of helpful granularity are only 7-weeks forward-looking. Therefore, any simulation longer than 7 weeks would require extrapolating the forecasts beyond what is currently available.

3.1 Building the Monte Carlo model

The store aggregate inventory simulation takes the expected volume in and out flows and returns the daily store inventory level over 4-weeks. It is a Monte Carlo based model that derives its stochasticity from sales and returns uncertainty.

3.1.1 Literature review of Monte Carlo simulations

In essence, a Monte Carlo simulation is a "mathematical model that simulates a real system", where "a large number of random sampling of the model is applied, yielding a large number of random samples of output results from the model" [16]. First adopted by scientists working on the Manhattan project, the method was used to study systems where "input variables, and a series of algorithms that were too complicated to analytically solve" [16]. With "authentic" algorithms and the proper "choice of input probability distributions", a Monte Carlo simulation can inform the distribution of outcomes, including the average and the range [16].

In the context of inventory management, Monte Carlo methods are typically applied to model how system-wide behaviors can be optimized in the presence of probabilistic inputs: Brits and Bekker 2016 uses the Monte Carlo method to introduce stochastic behavior in power generation and calculates the corresponding optimal coal stockpile inventory [2]; Montororing and Widyantoro 2022 uses stochastic demand generated from Monte Carlo methods to identify optimal supermarket inventory with considerations of competitors and in-store stimulus strategies [13]; Mansur, Mar'ah, and Amalia 2020 examines the optimal inventory replenishment policies for blood used for transfusion, given demand for blood simulated using Monte Carlo methods [12] (For more examples of similar work, see Widyadana, Tanudireja and Teng 2017 [18], Baharom and Hamzah 2018 [1]).

For the aforementioned research in inventory management, the value of Monte Carlo methods lies in the fact that they can convert variables with analytical distributions into sets of physical numbers, which can then be optimized over numerically. In contrast, this project does not have an optimization component. Instead, the value of Monte Carlo methods is in the range of outcomes. If an inventory simulation is built deterministically, the outcome will be singular, which allow any insight into neither the potential inventory variations from uncertainty in customer behaviors, nor the potential correlations between the stochasticities across categories of products. This information can be valuable, as capacity and inventory planning does not typically focus on the

mean outcome, but also the upper and lower estimates. Therefore, despite the vast literature on Monte Carlo simulations, particularly concerning inventory management, this project is unique in the sense that Monte Carlo methods are deployed for very different reasons. Although the fundamentals of quantifying, modeling and sampling stochasticities are still applicable, many of the difficulties encountered here are specific to the project, and do have tend of have parallels in most academic literature concerning Monte Carlo simulations in the context of inventory management.

3.1.2 Reshaping sales and returns forecast

The customer demand input of the Monte Carlo model hinges on forecasts, because the tool is designed to be forward-looking. As mentioned in Section 2.2, sales and returns forecasts are provided on section levels and include products beyond clothing. However, the project is solely focused on clothes and the simulation model is built on buyer-family levels. To bridge the differences, historical sales and returns actuals are leveraged to allocate sales and returns forecasts to the right granularity. First, a specific time period is chosen, from which historical sales and returns data is pulled for all the product categories that the forecasts include. The historical time period examined should have similar sales and returns volume breakdowns as the simulation period, as they can be quite different depending on the season. Then sales and returns for each buyer-family combination, as a percent of total sales and returns from the historical period, are calculated, and applied to the sales and returns forecasts to get to buyer-family level forecasts.

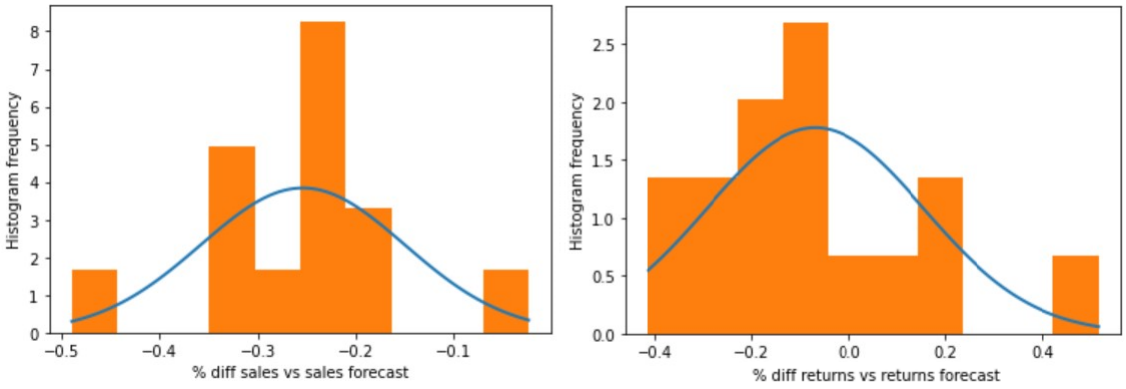
3.1.3 Quantifying sales and returns accuracy and weekly demand stochasticity

Weekly buyer-family level sales and returns forecasts are compared to historical actuals to calculate weekly forecast accuracy and demand stochasticity.

Percent differences between forecasts and actuals are first calculated on a weekly aggregate level and normal functions are fitted to the distribution of weekly percent differences over a time period. The means of the normal fits give us the systematic bias of the weekly forecast and the standard deviations give us the weekly stochasticity of demand. An example is included in Figure 3-1 for a specific buyer-family in a specific store over a 3-5 month period. Normal functions are able to roughly capture the mean and standard deviations of the distribution.

The Monte Carlo incorporates two types of sales and returns stochasticity, one for the weekly aggregate total, and the other for each day within the week, the latter of which will be discussed in Section 3.1.4.

Figure 3-1: Histogram of weekly sales (left) and returns (right) % difference between forecast and actuals for a specific buyer-family in a specific store over a 3-5 month period with normal distribution fit. Both histograms and normal fits are density functions with area normalized to 1.

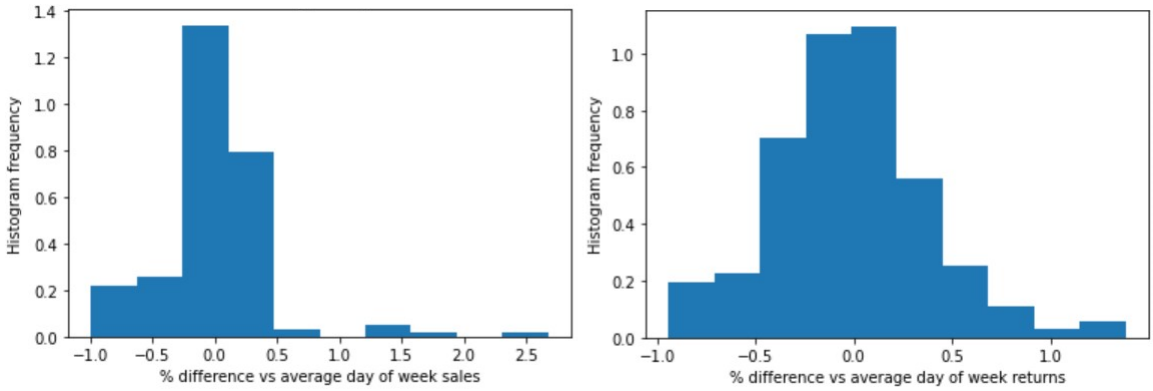


3.1.4 Calculating day of week sales and returns proportions and stochasticity

Sales and returns forecasts are provided daily, but are aggregated to weekly level before stochastically disaggregating to each day of the simulation. First, a historical time period is chosen when volume for each day of week as a percent of volume on an average day of that specific week is calculated for each week. Averages of the calculated percentages are taken across weeks to calculate average day of week sales and returns proportions, which are used to disaggregate the weekly total. The approach to calculate stochasticity is similar to that in Section 3.1.3. Volume on each day of week as a percent of volume on an average day of that specific week is calculated and collected across days and weeks. Subsequently, percent differences with regards to the average of the collected distribution are calculated, to which normal functions are fitted. The standard deviations of the normal fits provide a quantification of stochasticity for each day of week. As an example, if Saturday sales are on average 150% of an average day of sales for the week, there may be some Saturdays at 165% and some at 135%. These percentages are subtracted and divided by the average of 150% to get a percent difference, such as +10% and -10%. The standard deviation of this percent difference distribution, collected across not just all Saturdays, but all days of weeks, quantifies the magnitude of stochasticity for each day of week, across

various weeks. An example of such a distribution is shown in Figure 3-2. Note that unlike in Figure 3-1, this distribution is centered around 0 by design, and hence the mean is not of particular interest.

Figure 3-2: Histogram of day of week sales (left) and returns (right) % difference between each week and the average across weeks for a specific buyer-family in a specific store over a 6 month period. Histograms are density functions with area normalized to 1.



Finally in conjunction with weekly demand stochasticity, we now have a way to introduce stochasticity to each day of the Monte Carlo simulation. Each simulation trial will sample randomly in both the weekly and day of week normal distributions and generate different sales and returns profiles that mimic historical uncertainty in demand.

3.1.5 Quantify sales and returns covariance between buyer-family combinations

One way to build the simulation bottoms-up from buyer-family levels is to treat each buyer-family as its own independent mini-model with no correlation, which is implicitly assumed in the normal function fit approaches described in Section 3.1.3 and 3.1.4. However, buyer-family independence is not entirely realistic, as there are often demand correlations between them. For instance, if someone buys a pair of jeans, chances are they might also buy a denim jacket to go with the jeans. Article level quantification of this correlation is an incredibly difficult task, since new articles are introduced frequently and understanding the behavior of these new articles without historical data is non-trivial. Fortunately, this simulation model is built on buyer-family levels and it can be reasonably assumed that historical inter-buyer-family correlations will

hold in the future.

To quantify the correlations, a total of four covariance matrices are calculated: sales weekly aggregate, sales day of week, returns weekly aggregate, and returns day of week, corresponding to each type of stochasticity described in Section 3.1.3 and 3.1.4. The diagonals of the matrices are expected to give us the variance, consistent with the standard deviations of the normal function fits (due to the equivalence of ordinary least square approach maximum likelihood estimation approach for normal functions). The off-diagonals of the matrices are expected give us the covariance, which quantify the correlations of stochasticity between various buyer-family combinations. By allowing the Monte Carlo model to sample from the multi-normal distributions with calculated covariance matrices, inter-buyer-family correlation can be simulated and is expected to lend more fidelity to the model.

The simulation model is test both with and without multi-normal sampling using covariance matrices and the comparison of outcomes is discussed in Section 3.3.

3.1.6 Setting up shipment and backstock inputs

Balancing upstream and downstream movements

The simplest approach to model upstream movement is to assume that backstock is zero and shipment volume is equal to the expected weekly net downstream movements. By balancing upstream and downstream movements this way, inventory is expected to stay flat on a weekly basis.

Assuming that a store receives two shipments per week on CD1 shipment model on Mondays and Thursdays, each shipment needs to cover the expected net downstream movements before the next shipment comes. Therefore, the Monday shipments should account for the net downstream movements on Mondays, Tuesdays and Wednesdays (cycle 2), and the Thursday shipments for Thursdays, Fridays, Saturdays and Sundays (cycle 1). By looking at the proportions of the expected net downstream movements of cycle 1 and 2, weekly shipment volume can be broken down into volumes for the two weekly shipments. This approach to set up shipment and backstock inputs, perhaps simplistic, is the foundation for the simulation model and more complicated mechanisms are incrementally built on top of it.

Inventory build-up and depletion trends

Inventory on buyer-family levels may demonstrate strong seasonal trends. For example, shorts are generally sold in the summer, not in the winter, and inventory can be observed increasing around March and April each year. Therefore, to inform shipment volume for the forward-looking inventory simulation, we hope to extract weekly historical inventory build-up and depletion trends, preferably from the same time period of the previous year (e.g. simulation intended for March of 2023, historical inventory trends extracted from March of 2022). Inventory trends are also particularly important for the quantitative evaluation of the model, as the outputs are compared to actuals from historical periods to calculate accuracy metrics. If the model is unable to capture general inventory trends, the accuracy metrics would not be helpful. A number of different methods were tested to extrapolate historical inventory trends.

First, a regression based approach using data from the prior year is tested. Despite averaging over days of the week, weekly historical inventory data still tends to be noisy due to delayed shipments, promotions, and inherent unpredictability in demand, especially on buyer-family levels. Therefore, instead of simply subtracting the starting inventory from the ending inventory of a similar time period the year prior to the simulation period, a single variable linear regression is applied to each buyer-family to calculate on average how much the inventory increases or decreases on a weekly basis. From there, weekly inventory differences are divided by the average weekly downstream movements (sales minus returns) of the same historical time period to arrive at the parameter for inventory build-up and depletion trends.

There is no guarantee that historical inventory trends from a similar time period in the prior year will produce accurate the results from the simulation period. Therefore, an additional approach is tested using the same linear regression analysis but performed on the actual simulation period. Although not helpful for forward-looking simulation time horizons, this approach may allow us to better quantitatively evaluate the model accuracy. Lastly, it is possible that weekly inventory build-up and depletion trends are inherently too noisy on buyer-family levels for linear regressions to yield accurate results. An option is also included in the model to manually input inventory build-up and depletion trends, which are set to be the exact linear extrapolation of the beginning and ending inventory of the actual 4-week simulation period. Again, this approach is also not possible for forward-looking simulations, but may be helpful when calculating accuracy metrics and quantitatively evaluating the model.

Shipment and backstock balance

In actual store operations, twice as week shipments are not meant to only cover the expected amount of sales until the arrival of the next shipment, especially given that each article is unique and not exactly substitutable. Instead, backstocks are sent accordingly to adjust for capacity and deplete articles that are not longer needed in stores. Functionally, this makes total business sense if done carefully without impacting capacity, as the cost of not having products in stock is almost always higher than the cost of having to move extra inventory to and from the store, given demand uncertainty. Therefore, the simple assumption that backstock is zero and shipment volume matches net downstream movement is almost certainly simplistic.

It is very difficult to tell exactly how each store should balance shipments and backstocks, as it is an ongoing area of improvement for the business. Therefore, rather than fixing the model to any pre-calculated value, a functionality is built in, where the amount of backstock can be adjusted manually. It is set up such that shipment volume can be manually increased by a given percentage, and the total increased amount for the week is even split between each day of the week as backstocks. This way, net upstream movement is conserved, and the shipment and backstock balance mechanism effectively makes the day of week inventory "spikier" by inflating the volume of each shipment.

CD2 replenishment

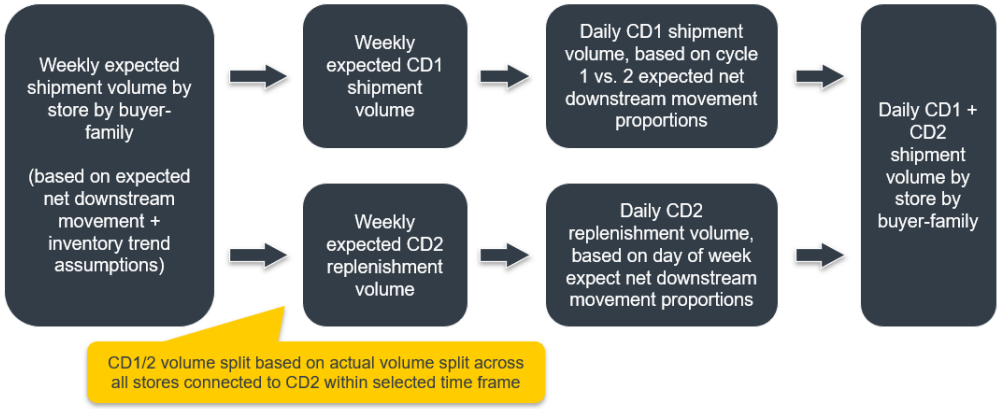
As discussed in Section 1.2, understanding the impact of CD2 replenishment is one of the key objectives of the simulation model.

To simulate store shipments under a CD2 replenishment model, one must first understand how much volume comes from CD2 versus CD1, if a store is connected to a CD2, which largely depends on how much volume to the store is replenishment for existing products and shipment for new products. This quantity can be calculated from historical data, by looking at the type of origin warehouses of all product shipments for stores connected to a CD2 over a specified time period. The calculated proportions can be applied to the simulation to determine how much weekly shipment volume should come from CD1 versus CD2.

From this point, the allocation logic of weekly to daily volume remains exactly the same. CD1 weekly volume is allocated to Monday and Thursday, based on the

proportions of the expected net downstream movements of cycle 1 and 2. CD2 weekly volume is allocated to each day, based on based on the proportions of the expected net downstream movements of every day of week. The logic of setting up CD2 replenishment is illustrated in Figured 3-3.

Figure 3-3: Illustration of CD2 replenishment logic



3.1.7 Shipment and backstock adjustment mechanism

One of the key challenges of building a stochastic store simulation is that there is an inherent and unavoidable misalignment in the decision timings between the simulation and reality. Suppose a store gets shipments twice a week. In reality, the volume of the shipments will be determined based on real-time inventory and sales (forecast) data twice a week, right before each shipment. If the sales have been higher and inventory has depleted more than expected, the warehouses would send a little more in the next shipment and vice versa. Recreating such a decision process exactly is obviously not possible for a simulation that requires 4 weeks of shipment and backstock volume up front as inputs. However, if the shipment and backstock volume inputs are used verbatim, the stochasticity in the sales and returns compounds over time and results in the Monte Carlo range of the inventory level continuously increasing further into the simulation. Such an outcome can be observed in the 6th column of Figure 3-7, where stochasticity is allowed to impact inventory freely.

To properly simulate upstream decision-making, a shipment and backstock adjustment mechanism is introduced. The mechanism looks at the inventory on Wednesday and Saturday (when shipment decisions for Thursday and Monday are made), calculates the difference versus a baseline value, and applies the difference multiplied by an adjustment coefficient to the following Thursday or Monday, either as extra shipment

or backstock volume. For example, if inventory is at 1000 on Wednesday, but the baseline is 1100, and the adjustment coefficient is set to be -0.5, an extra 50 units will be added to the Thursday shipment. The adjustments are applied twice a week, resulting in a total of 7 adjustments over the 4-week simulation horizon (not needed for the first Monday, since the simulation starts on a Sunday).

For the inventory adjustment to function well, the values of baseline inventory and adjustment coefficient need to be carefully calibrated, yet both of them are parameters introduced for the simulation and do not have operational equivalences. For the baseline inventory, a straight-line is extrapolated from the starting inventory of the simulation (at the end of Sunday) by buyer-family combination and the expected ending inventory from the inventory trends (assuming no stochasticity). This way, as the simulation progresses, the baseline inventory can increase and decrease accordingly. Similarly, the baseline inventory for the Wednesday/Thursday adjustment is set to be the average of the preceding and following Saturday/Monday adjustment. For the inventory adjustment coefficient, historical inventory and sales forecast data is examined. If actual shipment quantity tends to be higher if inventory is low and sales forecast is high, one might be able to quantify this trend by looking at their historical correlations. Linear regressions are run using historical shipments and the corresponding RFID inventory and VMP (see Section 2.2 for details) on the days when the shipment decisions are made. Of course, it is unclear whether historical calculations will yield sensible results, as data tends to be noisy, especially on buyer-family levels. Options to manually input values for the inventory adjustment coefficient are built into the simulation as well.

3.1.8 Defining minimum and maximum inventory level

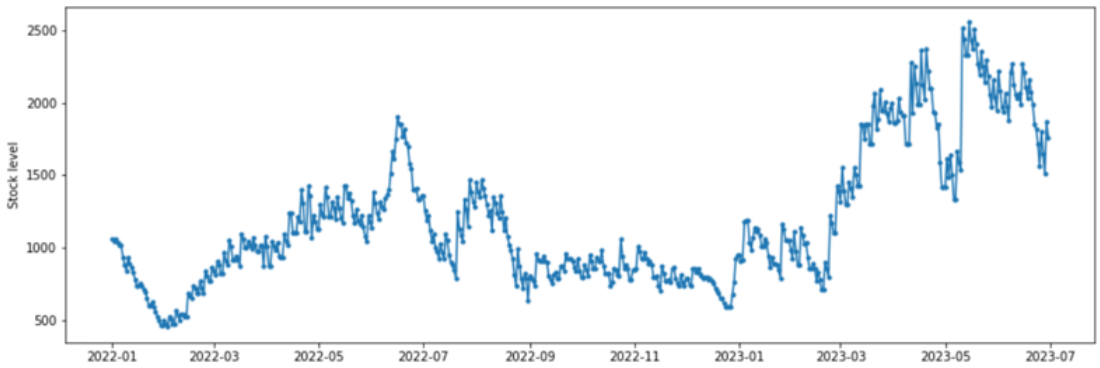
Due to the stochasticity introduced in sales and returns, it is possible for inventory for each buyer-family combination to be unrealistically high or low in each Monte Carlo trial (e.g. below 0). Therefore, inventory guardrails are necessary in the simulation.

Dramatic inventory level fluctuation can be observed for each buyer-family over time, due to seasonalities, business cycles, operational changes, and shifts in consumer preferences, which make it difficult to predict what the min and max inventory can be. As a result, an empirical approach is taken, where inventory min and max are calculated as the inventory of the day with the (2nd percentile) lowest inventory and

the (100th percentile) highest inventory, for the days when inventory data for that buyer-family is available (barring data noises).

Nonetheless, the empirical approach can still be problematic for inventory max. In Figure 3-4, inventory level for a specific buyer-family for a specific store is shown from January 2022 to June 2023. If the simulation is run for March/April of 2023, the starting inventory level would be higher than the highest level of inventory on all of 2022, implying that the empirically calculated inventory max will artificially suppress the inventory and interfere with the simulation. Unfortunately, this is a fundamental limitation of the approach. It implicitly assumes that future inventory levels are similar to historical levels, which is true for the most part, but exceptions are certainly not uncommon. Therefore, the model is set up in such a way that the max works as a soft cap, meaning inventory can still go above the max, but backstocks are added each day, proportional (one-third) to the amount the simulated inventory exceeds the empirical inventory max. Additionally, inventory max is not enforced when starting or expected ending simulation inventory is above a certain percent of the empirically calculated max. In Section 3.3.2, results from different enforcement thresholds of the inventory max are contrasted to explore the quantitative impact on simulation accuracy.

Figure 3-4: RFID inventory for a specific buyer-family in a specific store from January 2022 to June 2023



3.1.9 Calculating store and section capacity and utilization

Store capacity, in units of number of articles, is already modeled by Zara. It is disaggregated into display inventory (hanging only), stock room hanging inventory and stock room folding inventory. Furthermore, capacity is a dynamic quantity that is adjusted based on the estimated product bulkiness (e.g. coats are bulkier than

t-shirts) and the racking and staging of the rooms. Fortunately, capacity is a relatively stable quantity within the same season, and a simple average can be taken across a historical time period representative of the simulation period.

To convert store inventory level to utilization, it is necessary to know how much of each buyer-family is folding or hanging. Each article is designated either folding or hanging in the article master. Combined with the historical mix of article level inventory in store, the percent of volume in each buyer-family that is folding versus hanging can be inferred.

3.2 Qualitative evaluation

To understand whether the model mechanisms are performing as expected, a series of simulations are run with varying parameters.

First and foremost, Figure 3-5 and 3-6 provide a comparison of daily simulated and actual sales and returns volume for a specific store over a select simulation time horizon. The blue lines indicate historical actual sales and returns, the various lines in the background indicate outcomes from 50 Monte Carlo trials, while the orange line indicates the average of the Monte Carlo trials. The spikiness results from demand fluctuation over the days of week, as well as store closure on Sundays. The simulated trials are based on sales and returns forecasts, with stochasticity introduced using parameters calculated historically. One would expect actual sales and returns to generally agree with the simulation averages with little systematic biases, and largely fall within the range of outcomes of the various Monte Carlo trials, which is exactly what is observed for sales. The observation is less true for returns, which is not surprising given that returns forecasting is more difficult due to higher stochasticity. But given that sales volume is an order of magnitude higher, slight inaccuracies in returns are not expected to impact model fidelity materially.

Following the verification that the stochastically simulated sales and returns are generally accurate, Figure 3-7 includes a series of simulation inventory outputs for two selected buyer-family combinations (both within section 1) and section 1 (women) total for a specific store with varying forms of stochasticity included. Buyer-family combination A and B are intentionally chosen as examples with relatively low and high stochasticity. The x-axes represent days of simulation, y-axes inventory level and lines with different colors correspond to different trials of Monte Carlo simulation. A total of

Figure 3-5: Comparison of daily simulated and actual sales volume

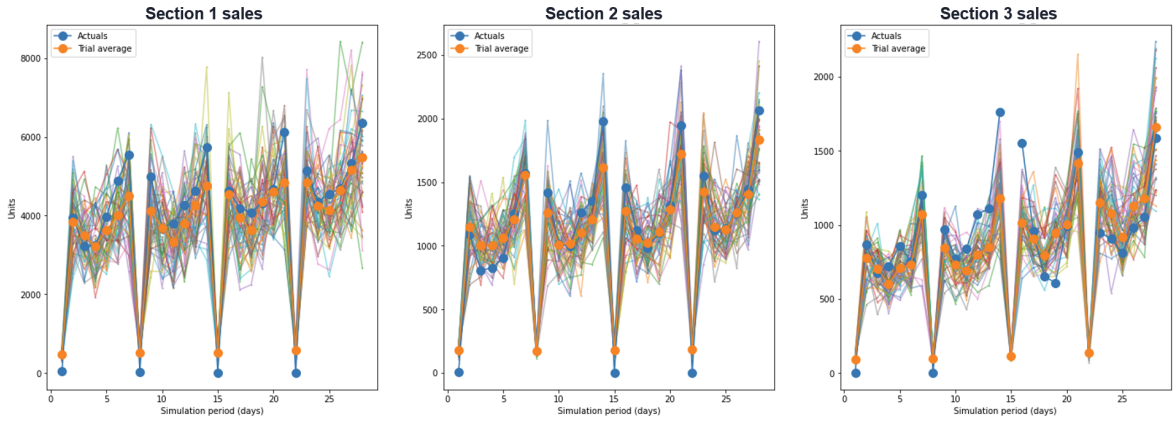
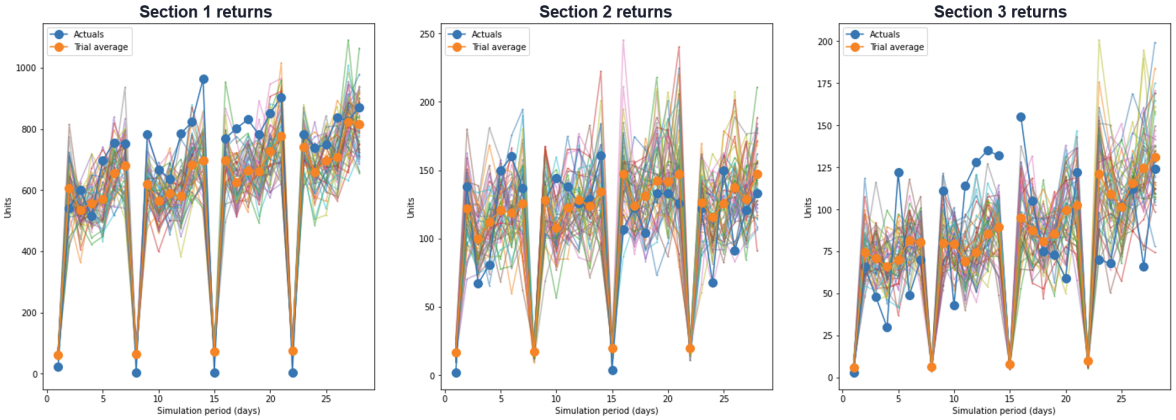


Figure 3-6: Comparison of daily simulated and actual returns volume



50 Monte Carlo trials are run for each scenario. Also note that all upstream movement and inventory adjustment mechanisms are disabled, meaning the inputted shipment and backstock volume is used verbatim and the store inventory is allowed to evolve freely without any bounds. Lastly, inventory trend is set to be zero, meaning net upstream movements are balanced with the expected net downstream movements.

Starting from the no stochasticity scenario in the 1st column, all Monte Carlo trials converge, as the model becomes fully deterministic with all stochasticity set to zero. Additionally, inventory remains flat over the course of 4 weeks (day 0 vs. day 7, 14, 21, 28) with spikes over the course of weeks from the two weekly shipments. It is also interesting to note that day of week variation in week 3 and 4 is higher than in week 1 and 2, despite inventory trend being set to flat. This is due to higher demand forecast

for week 3 and 4, resulting in bigger shipment volumes during the weeks.

In the 2nd and 3rd column, sales stochasticity is introduced, one with just weekly and the other with just day of week. Weekly stochasticity is expected to impact weekly aggregate sales, while day of week is expected to impact individual days within each week. The difference is evident for buyer-family A. With just weekly stochasticity, the day of week spikiness is maintained and the differences between Monte Carlo trials stem from the spiky "shapes" shifting up and down for each week. In contrast, the day of week spikiness demonstrates very different shapes across Monte Carlo trials with day of week stochasticity introduced. Similar results can be observed in buyer-family combination B and section 1 total, but not as clearly. Buyer-family B has high sales stochasticity compared to its mean, implying that the stochasticity tends to dominate the day of week spikiness. Section 1 total includes a variety of buyer-family combinations, which are currently assumed to be independent from each other (covariance is not included) and the same observations are more difficult when patterns aggregated.

In the 4th and 5th column, similar stochasticity is introduced but for returns only, and the same observations still hold. It is worth noting that the magnitude of variations between Monte Carlo trials dramatically decreases, because the magnitude of returns is much smaller than sales and therefore impacts a smaller fraction of downstream volume. Finally in the 6th column, all stochasticities are combined to generate the daily inventory level over the 4-week simulation horizon. It is fair to say that the model has behaved as expected so far, thus instilling confidence that it is working as intended.

Problems arise from the simulation, if the stochasticity is allowed to impact store inventory freely, as discussed in Section 3.1.7 and 3.1.8. Therefore, a number of upstream movement and inventory control mechanisms are introduced and their impacts are visualized in Figure 3-8.

Starting from the 1st column, which is the same as the 6th column of Figure 3-7, inventory adjustment mechanism as described in Section 3.1.7 is introduced in the 2nd and 3rd column with two levels of strength. With an adjustment coefficient of -0.5, half of the deviation from the expected baseline is added to the next shipment, while with -1, all of the deviation is added. Upon first glance at the inventory time series, it is clear that the range of Monte Carlo outcomes has been significantly tightened with the introduction of inventory adjustment mechanism. For instance, the range of the

Figure 3-7: Impact of various stochasticities on two buyer-family combinations (within section 1) and section 1 (women) total for a specific store; upstream movement and inventory adjustment mechanisms disabled and inventory trend set to be flat

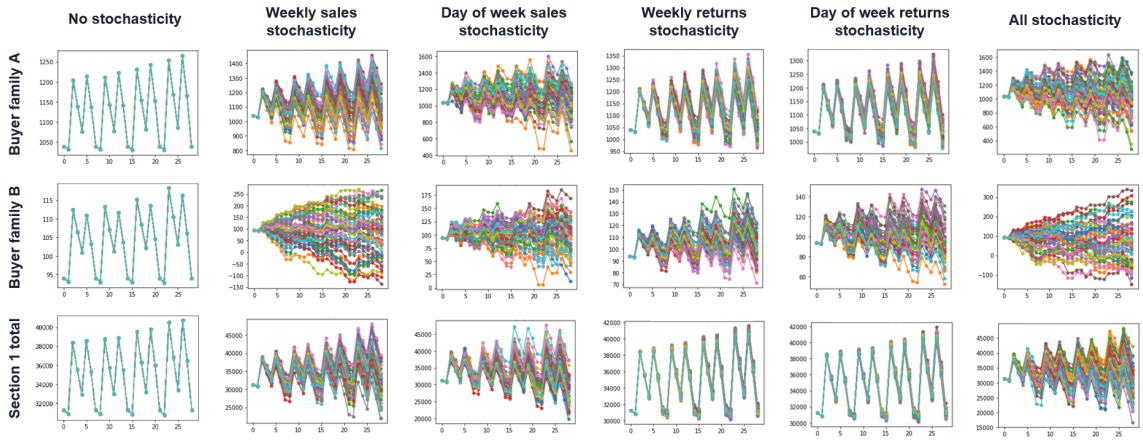
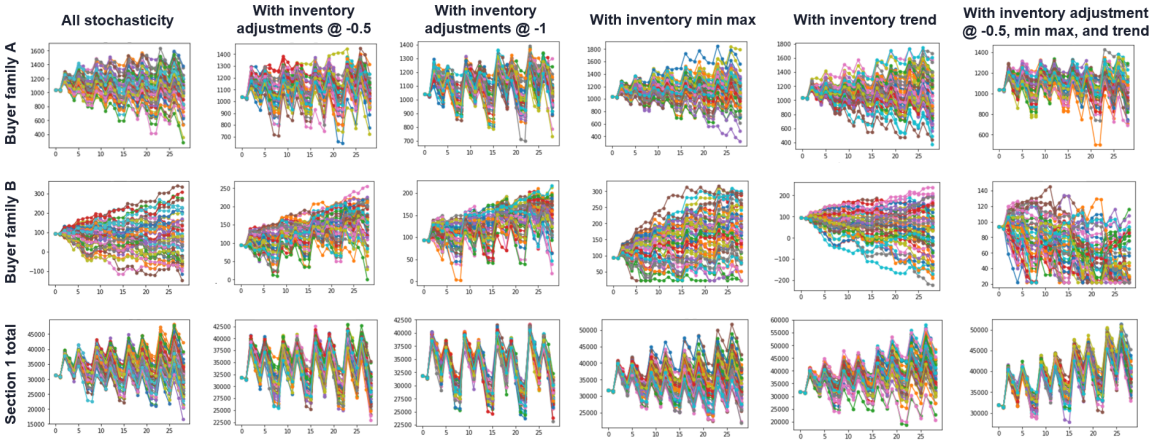


Figure 3-8: Impact of various upstream movement and inventory control mechanisms over the simulation period on two buyer-family combinations (within section 1) and section 1 (women) total for a specific store



charts for section 1 total decreased from 15,000-45,000 to 22,500-42,500. Comparing the 2nd and 3rd column, increasing the strength of the inventory adjustment to -1 can still tighten the bound, but the impact is much smaller. The diminishing return is not surprising, given the stochastic nature of the simulation and day of week inventory variations.

The 4th column applies the inventory minimum and maximum to the simulation, as described in Section 3.1.8. Buyer-family A is not impacted, because the thresholds

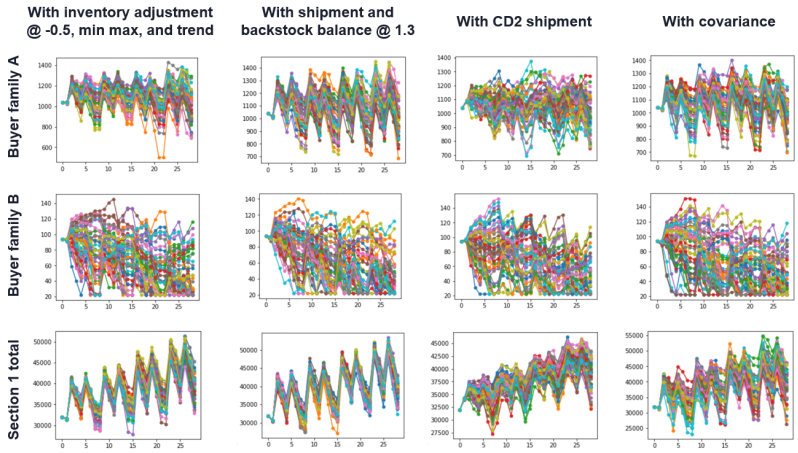
are not reached (chart still looks different because each Monte Carlo trial is unique), but buyer-family combination B is clearly bounded. Instead of containing negative inventory, the lower limit is capped at around 10 units. The same impact on the lower bound can also be observed in section 1 total. On the other hand, the impact on the upper bound is somewhat limited. This is because the upper bound is only selectively enforced and as a soft cap, as discussed in Section 3.1.8.

The 5th column applies a calculated inventory trend to the simulation. The trend is relatively flat for buyer-family combination A, negative for B and overall positive for section 1 total. Finally, the 6th column combines the inventory adjustment coefficient at -0.5, inventory min and max, and an inventory trend, along with all the stochasticities. With all these mechanisms, the model is able to not only introduce stochasicity based on historical buyer-family level behavior, but also control the stochasticity such that the inventory does not freely evolve to reflect operational realities.

Lastly, Figure 3-9 visualizes the impact of shipment and backstock balance, CD2 shipments and covariance, on top of the mechanisms already discussed. With shipment and backstock balance at 1.3 in the 2nd column (meaning shipment volume is 30% more than before and net weekly upstream volume stays the same), variations within the weeks are increased. With daily shipments from CD2 in the 3rd column, variations within the weeks are decreased for each Monte Carlo trial as the volume of the two weekly CD1 shipments is spread throughout the week. With the incorporation of covariance between buyer-family combinations, individual buyer-family inventory is not impacted but the section aggregate spans a wider range. Without covariance, combining independent stochasticities of each individual buyer-family allows them to cancel each other out. When each buyer-family is not independent, the effect of stochasticities cancelling out from aggregating is reduced, resulting in bigger ranges in the Monte Carlo outcomes.

In conclusion, the qualitative evaluations provide visual verification of the various mechanisms built into the Monte Carlo simulation model. The outcomes have been fully consistent with expectation, thus suggesting that the model has been properly formulated.

Figure 3-9: Impact of shipment and backstock balance, CD2 shipments and covariance on two buyer-family combinations (within section 1) and section 1 (women) total for a specific store



3.3 Quantitative evaluation

3.3.1 Defining simulation accuracy metrics

By running the Monte Carlo simulation on historical time periods and comparing the results to historical actuals, accuracy metrics can be calculated. That said, precautions need to be taken to carefully define the accuracy metrics. The Monte Carlo outputs are given by Monte Carlo trials, by days, and by buyer-family combinations. The method and order of operations in which they are aggregated as summary accuracy metrics can lead to dramatically different conclusions. For instance, the store total of the 75th percentile Monte Carlo outcome is very different from the 75th percentile Monte Carlo outcome of the store total, because the former aggregates the Monte Carlo variations across each buyer-family combination, while the latter allows the variations to cancel each other first within each trial by aggregating them to the store level.

Therefore, for the sake of consistency and clarity throughout the project, five custom accuracy metrics are defined:

- Daily % mean absolute difference (MAD), for each buyer-family combination, section total and store total: calculate the Monte Carlo median by buyer-family combination by day; aggregate by section and store; find the absolute differences between the Monte Carlo median and historical actuals; aggregate the absolute differences across days (e.g. week 1, all 4 weeks); divide the aggregated absolute

differences by the aggregated historical actuals over the same time period

- Weekly % mean absolute difference (MAD), for each buyer-family combination, section total and store total: calculate the Monte Carlo median by buyer-family combination by day; aggregate across days to week level; aggregate by section and store; find the absolute differences between the Monte Carlo median and historical weekly actuals; [optional] aggregate the absolute differences across weeks (e.g. all 4 weeks); divide the aggregated absolute differences by the (aggregated) historical actuals over the same time period

- Daily % mean difference (MD), for each buyer-family combination, section total and store total: calculate the Monte Carlo median buyer-family combination by day; aggregate by section and store; find the differences between Monte Carlo median and historical actuals; aggregate the differences across days (e.g. week 1, all 4 weeks); divide the aggregated differences by the aggregated historical actuals over the same time period

- Days out of Monte Carlo estimate range, for each buyer-family combination, section total and store total: aggregate by section and store; calculate the Monte Carlo upper estimate (75th percentile) and lower estimate (25th percentile) by buyer-family combination by day; count the number of days historical actuals are outside the range given by the upper and lower estimate on each day, over selected time periods (e.g. week 1, all 4 weeks)

- Monte Carlo estimate range size as % of median, for each buyer-family combination, section total and store total: calculate the Monte Carlo median by buyer-family combination by day; aggregate by section and store; calculate the Monte Carlo upper estimate (75th percentile) and lower estimate (25th percentile) by buyer-family combination by day; calculate the daily upper minus lower estimate range size; aggregate the range size across selected time periods (e.g. week 1, all 4 weeks); divide the aggregated range size by the aggregated median over the same time period

The first two MAD metrics are designed to examine the magnitude of deviation of the Monte Carlo median from historical actuals. Daily % MAD, as the name suggests, tells us on how many percent off is the simulation from actuals on an average day. This metric can be arbitrarily high and can be as low as 0, with lower values indicating higher accuracy. Note that daily % MAD is expected to be smaller on section and store

levels, compared to buyer-family levels. While individual buyer-family combinations can experience significant inaccuracies, aggregating them to store or section levels before calculating the absolute difference with historical actuals is expected to cancel out some of the fluctuations, hence stabilizing aggregated metrics. Weekly % MAD adopts the same concept, except that the comparison between Monte Carlo median and historical actuals takes place on weekly aggregate levels. Similar to the logic of aggregating buyer-family combinations to sections and stores, aggregating from day to week is expected to offset fluctuations between days and result in slightly smaller weekly % MAD values compared to daily % MAD.

The third metric, daily % MD, is designed to examine the systematic bias of the Monte Carlo median versus historical actuals. The calculation is exactly the same as daily % MD, except that the absolute value of the differences is not taken. This metric can be arbitrarily low or high, with proximity to 0 indicating lower systematic biases.

The last two metrics are designed to provide insights into the possible ranges of outcome from the Monte Carlo simulation, as it is difficult to predict inventory level precisely, given downstream stochasticity. It is crucial to stress that the upper and lower estimates here do not have statistical significances like confidence intervals. They can be higher or lower, depending on precise metrics definitions, or model mechanisms. However, by defining those ranges, we have a way to systematically compare models with different configurations and parameters, as well as gaining insight into how historical data may differ from the simulation. For days out of estimate range, the maximum over the course of 4 weeks is 28 and the minimum is 0. Nonetheless, days out of estimate range in a vacuum is not a helpful metric, as larger ranges naturally lead to fewer days out of range. Therefore, Monte Carlo estimate range size as % of median is introduced as the final metric, and the insights lie in the balance of days out of range and range size. A very important distinction between these two metrics compared to the first three is the order in which buyer-family level data is aggregated for the upper and lower estimate. Specifically, buyer-family level data here is aggregated first to section and store totals for each Monte Carlo trial, before taking the 75th or 25th percentile. This order is deliberately chosen for two reasons. First, the estimate range is much smaller than if the order is reversed, because the Monte Carlo variations of each buyer-family combination are allowed to cancel each other out first instead of completely stacked on top of each other. Second, this method of aggregation allows section and store totals to be susceptible to the incorporation

of covariance. If each buyer-family is treated independently with upper and lower estimates taken before aggregation, the correlation introduced with covariance would be pointless.

Following the same logic, one might wonder why should the Monte Carlo median be taken before aggregating to section and store totals. With an arbitrarily large number of buyer-family combinations and Monte Carlo trials, the aggregation order for the median should not matter. However, with fewer than 100 buyer-family combinations simulated per store and a few dozen trials per simulation, the section and store totals of each Monte Carlo trial can carry varying levels of randomness over a large range, and taking the median across limited Monte Carlo trials can yield higher MAD. Instead, each buyer-family combination has much smaller range variation and taking the median first before aggregating to section and store levels ensures more consistent results and lower MAD.

3.3.2 Accuracy metric results

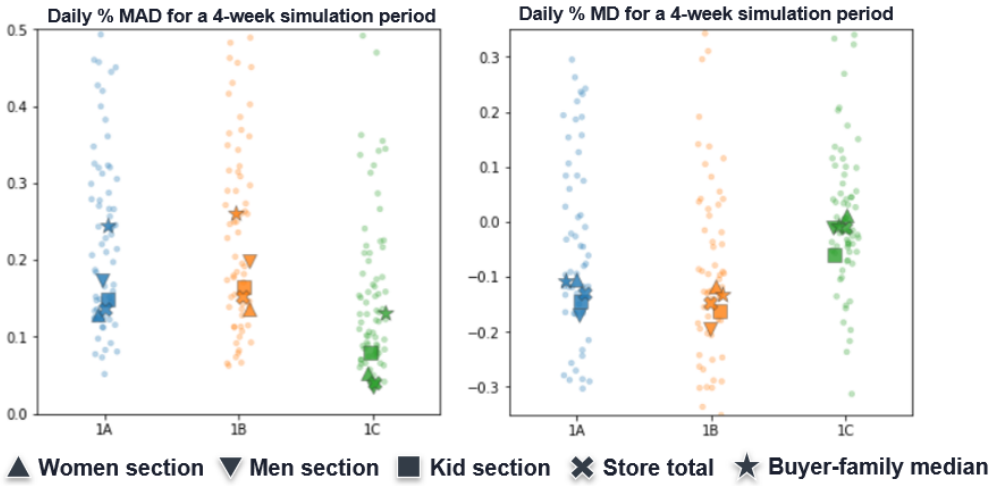
Equipped with precise definitions of accuracy metrics, the simulation model is run with various parameters to understand what combinations of model parameters result in the most desirable accuracy metric results.

Inventory trend options

Section 3.1.6 discussed three options to set up inventory build-up and depletion trends. Option 1 is a regression based approach using data from one year prior to the simulation period. Option 2 is the same regression based approach using data from the simulation period. Option 3 is based on linear extrapolations using the actual beginning and ending inventory of the simulation period. The daily % MAD and MD for the three inventory trend options are presented in Figure 3-10 as 1A, 1B and 1C. The accuracy metrics very clearly point to the inadequacy of the regression based approach. Significant systematic biases (MD $>$ 10% for section and store totals) can be observed for 1A and 1B, while 1C shows very small MD and much lower MAD. This result suggests that buyer-family level inventory build-up and depletion trends are inherently very noisy from week to week and therefore any pattern extracted using regressions has little predictive power. Therefore, trying to reproduce the inventory trends of a time period using observations from prior (or even slightly different) time periods has proven to be difficult. That said, this inability is not a fundamental issue for the simulation tool. At the end of the day, the goal of the tool is to produce sensible

inventory results with known upstream and downstream inputs, rather than predicting inventory trends itself. If the tool can accurately reproduce inventory levels given beginning and ending inventory over the simulation period, it can still be effective. To simulate future time periods when the beginning and ending inventory is not known, the user would simply have to rely on their domain knowledge.

Figure 3-10: Daily % MAD and MD for various inventory trend options for a 4-week simulation period for a specific store; 1A: regression inventory trends from previous year, 1B: regression inventory trends from simulation period, 1C: linear extrapolation using beginning and ending inventory from the simulation period; all inventory adjustment mechanisms turned off except for inventory minimum and maximum; background dots indicate individual buyer-family combinations



Shipment adjustment options

Section 3.1.7 discussed the various options to set up inventory adjustment coefficients, either through historical calculations or manual inputs. Figure 3-11, 3-12 and 3-13 explore how these options impact our defined accuracy metrics.

Figure 3-11 shows that daily median MAD and MD do not vary significantly based on how the inventory adjustment is set up, nor should it. Because the inventory adjustment does not have any systematic upward or downward impact, the median remains unaffected when adjustments of varying levels are introduced. As a result, no significant difference in MAD or MD is observed across different simulations in Figure 3-11.

The same observation cannot be generalized to the upper and lower inventory estimate. As illustrated in Figure 3-11, inventory adjustments is effective at tightening the

Figure 3-11: Daily % MAD and MD for various inventory adjustment options for a 4-week simulation period for a specific store; 2A/2B/2C: inventory adjustment at 0.5x/1x/1.5x of historically calculated value, 2D/2E/2F/2G/2H/2I/2J/2K: inventory adjustment at -0.1/-0.25/-0.4/-0.5/-0.6/-0.7/-0.8/-1; inventory trend using the actual beginning and ending inventory of the simulation period; inventory minimum and maximum included; background dots indicate individual buyer-family combinations

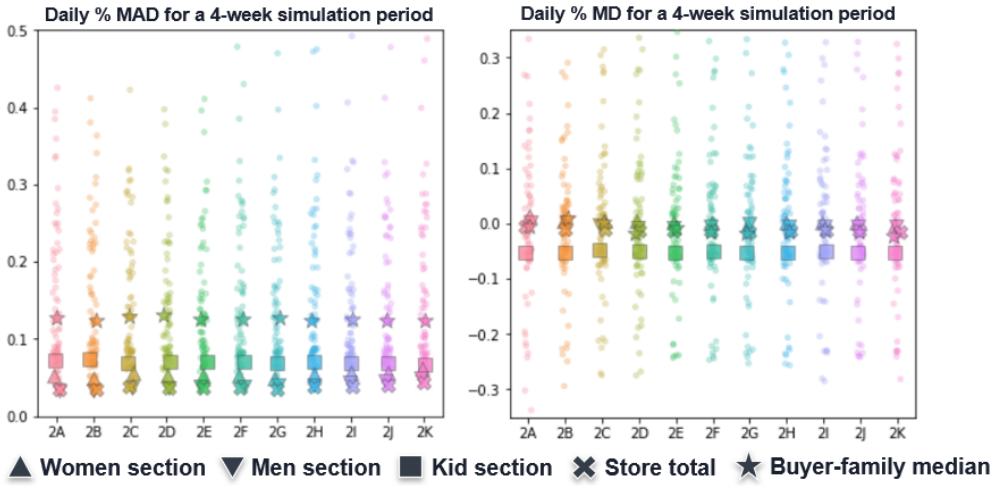
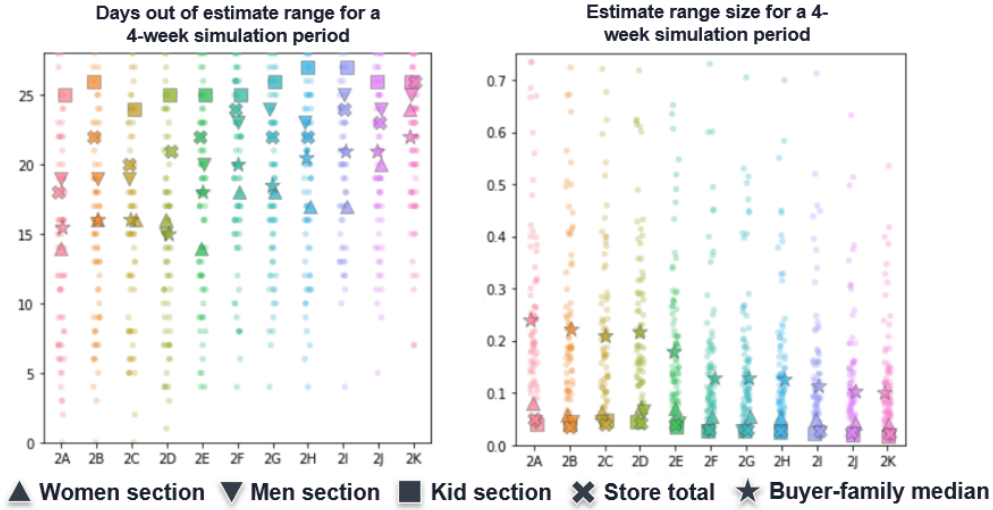
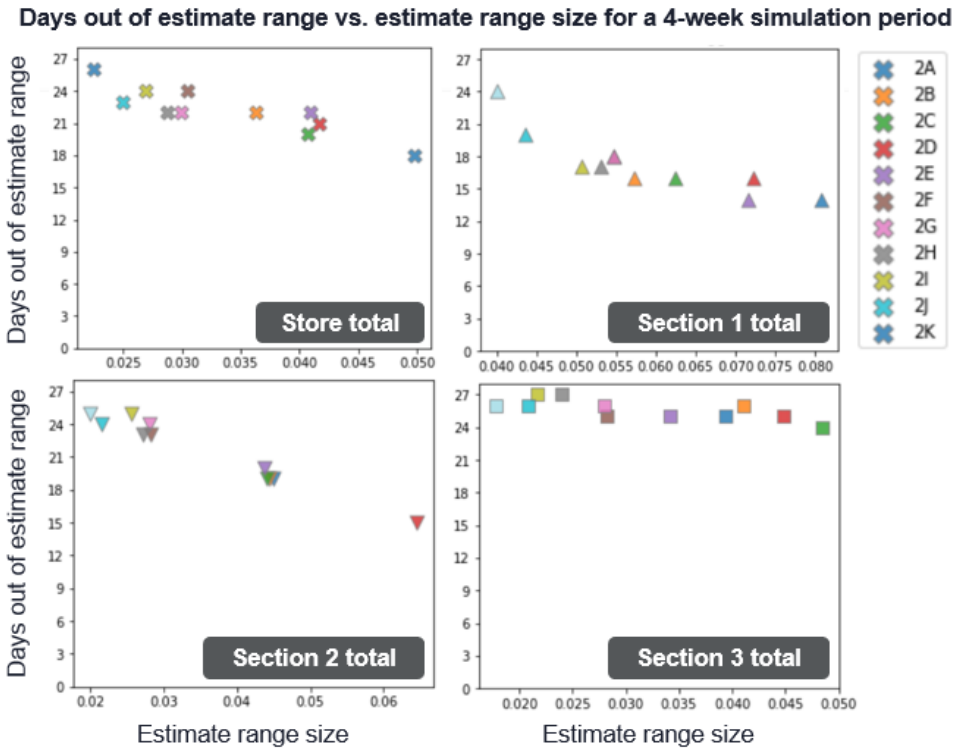


Figure 3-12: Days out of estimate range and estimate range size for various inventory adjustment options for a 4-week simulation period for a specific store; 2A/2B/2C: inventory adjustment at 0.5x/1x/1.5x of historically calculated value, 2D/2E/2F/2G/2H/2I/2J/2K: inventory adjustment at -0.1/-0.25/-0.4/-0.5/-0.6/-0.7/-0.8/-1; inventory trend using the actual beginning and ending inventory of the simulation period; inventory minimum and maximum included; background dots indicate individual buyer-family combinations



range of Monte Carlo trials further into the simulation horizon. Therefore, as the the strength of the inventory adjustment is increased (i.e. from -0.1 to -1), it is expected that the Monte Carlo estimate ranges would decrease as well, providing more precision to the estimates. The right hand side of Figure 3-12 shows exactly that. Nonetheless, the size of the estimate range in isolation is not a meaningful metric, as the size can be arbitrarily reduced by reducing the model stochasticity. It is only relevant in conjunction with days out of estimate range. The ideal model parameters should hopefully reduce the size of the estimate range without increasing days out of estimate range. Upon examination of the left hand side of Figure 3-12, days out of estimate range seems to increase as the estimate range gets smaller. To more systematically evaluate the trade-off, days out of estimate range and estimate range size are plotted against each other in Figure 3-13.

Figure 3-13: Days out of estimate range and estimate range size trade-off for various inventory adjustment options for a 4-week simulation period for a specific store; 2A/2B/2C: inventory adjustment at 0.5x/1x/1.5x of historically calculated value, 2D/2E/2F/2G/2H/2I/2J/2K: inventory adjustment at -0.1/-0.25/-0.4/-0.5/-0.6/-0.7/-0.8/-1; same parameters as Figure 3-12



Overall, we see that days out of estimate range and estimate range size largely go hand in hand with each other. As one increases, the other decreases accordingly, without a

particular level of inventory adjustment outperforming the rest on both fronts. This is interesting as it suggests that manually inputted inventory adjustment coefficients perform just as well as those set to historical calculated values (2B). Thus, for the sake of simplicity, it is recommended that the model adopts manually inputted values. As to which manually input value to use, because there isn't a particular one that stands out, the choice becomes a subjective matter of balance, rather than optimality. For the rest of the project, inventory adjustment coefficient of -0.5 (2G in Figure 3-13) is selected as the default going forward.

Inventory max options

Section 3.1.8 discussed the difficulty of empirically defining a maximum inventory level, as past inventory levels aren't always indicative of the future. Nevertheless, the need to define a inventory max can still be theoretically important, because the enforcement of inventory minimum alone may cause some systematic biases in the simulation due to the asymmetries of inventory bounds. Figure 3-14 examines inventory maximum of varying restrictiveness, and compares their respective accuracy results. We see that both MAD and MD are practically unaffected regardless of the restrictiveness of inventory maximum. Intuitively, only a subset of buyer-family combinations will be noticeably impacted by the inventory maximum. Even if they are, only the ones deviating much higher than the median will be impacted directly. In this case, the level of impact is not enough to noticeably affect the median, and thus accuracy metrics related to the median. Going forward, the least restrictive inventory maximum (3A) is chosen and applied to all simulations.

Covariance option

Section 3.1.5 discussed how covariance can be added to introduce correlation between otherwise independent buyer-family combinations in the simulation and the impact on section and store totals was visualized in Figure 3-9. To quantify this impact systematically, days out of Monte Carlo estimate range and range size metrics are calculated with the covariance option in Figure 3-15.

First and foremost, compared to Figure 3-13, the estimate range size is much bigger in Figure 3-15, which shouldn't come as a surprise as the effect of covariance widening range of store and section aggregate estimates was already illustrated in Figure 3-9. What is worth noting is that covariance with the lowest inventory adjustment coefficient value (-0.3) tested still produces a larger estimate range than the highest

Figure 3-14: Daily % MAD and MD for various inventory maximum adjustment options for a 4-week simulation period for a specific store; 3A/3B/3C: inventory maximum only enforce if starting or expected ending inventory is below 65/80/95% of empirically calculated inventory max; inventory trend using the actual beginning and ending inventory of the simulation period; inventory adjustment coefficient at -0.5

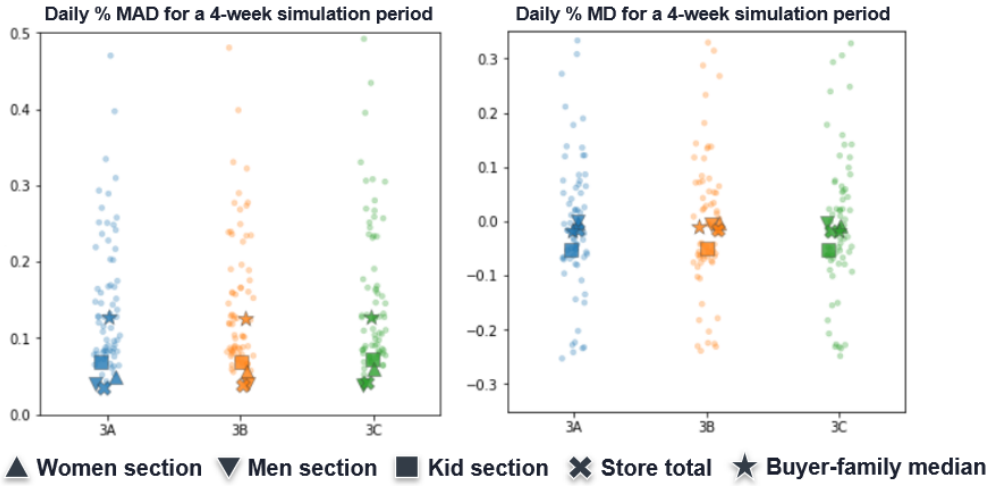
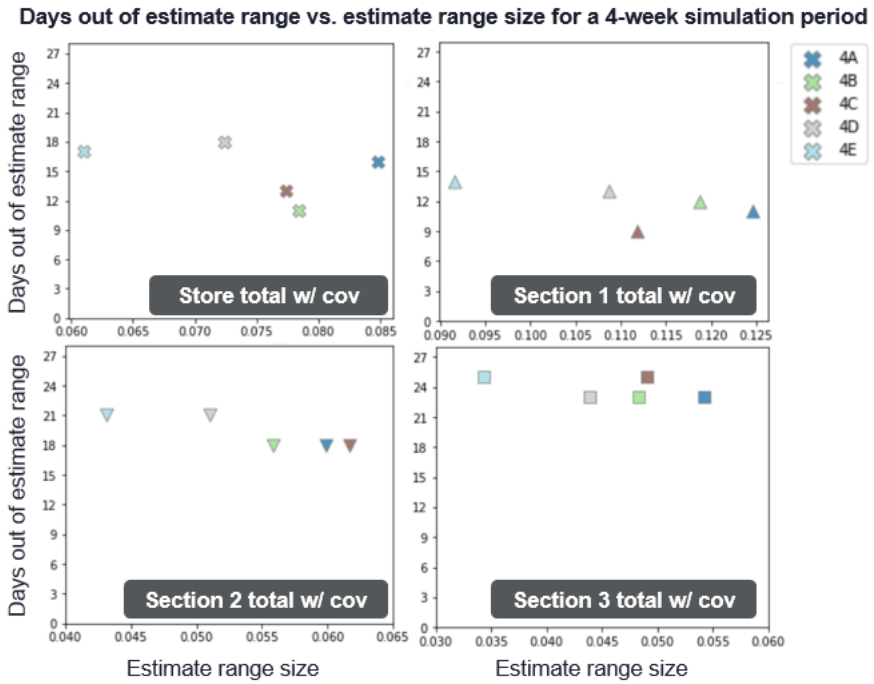


Figure 3-15: Days out of estimate range and estimate range size trade-off for covariance option for a 4-week simulation period for a specific store; 4A/4B/4C/4D/4E: with covariance and inventory adjustment coefficient at -0.3/-0.4/-0.5/-0.6/-0.7; inventory trend using the actual beginning and ending inventory of the simulation period; inventory minimum and maximum included; inventory adjustment coefficient at -0.5



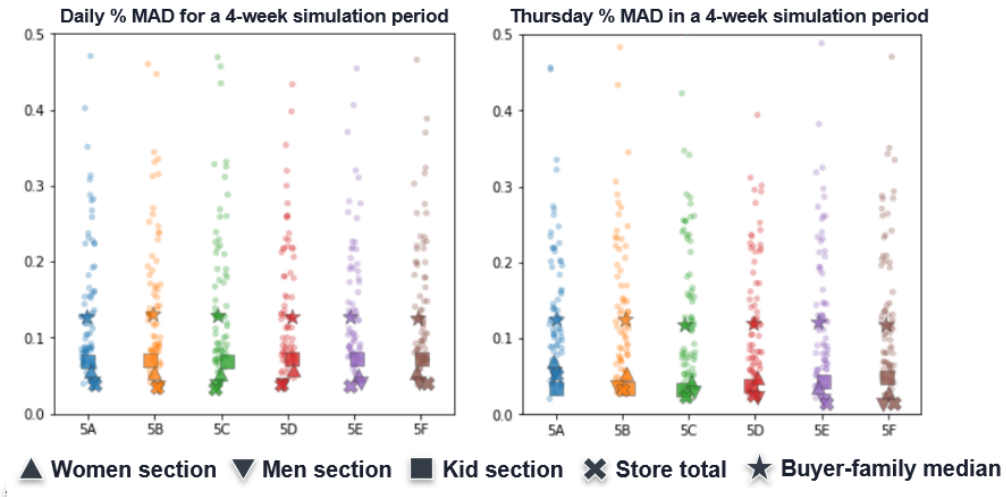
inventory adjustment coefficient value (-1) test without covariance. In the meantime, days out of the estimate range is steadily reduced from the larger estimate range, with inventory adjustment coefficient of -0.5 (4C) performing among the best. On top of it, buyer-family level estimate ranges remain tight, because using covariance has no impact on individual buyer-family combinations and impacts only the aggregates.

In summary, including covariance in the quantification and sampling of downstream stochasticity with an inventory adjustment coefficient of -0.5 strikes a good balance between the estimate range size and days out of estimate range on various levels. The range is reasonably relaxed for store and section aggregate, avoiding unrealistic precision from the simulation. Meanwhile, the range is relatively tightened for individual buyer-family combinations, which can have tendencies to be much bigger, due to high stochasticity of select buyer-family combinations relative to their inventory level. As a result, this particular configuration is selected for the store inventory simulation.

Shipment and backstock balance options

Section 3.1.6 discussed the potential need to increase shipment volume beyond covering the expected downstream movements exactly until the next shipment. Accuracy metrics with varying levels of of shipment and backstock balance are presented in Figure 3-16, for all days and all Thursdays in the simulation.

Figure 3-16: Daily and Thursday % MAD for various inventory adjustment options for a 4-week simulation period for a specific store; 5A/5B/5C/5D/5E/5F: shipment and backstock balance set to +0/10/20/30/40/50%; inventory trend using the actual beginning and ending inventory of the simulation period; inventory minimum and maximum included; inventory adjustment coefficient at -0.5; covariance included



Daily % MAD is not significantly impacted. As shipment volume is increased, backstock is increased accordingly such that the net weekly upstream movements remain the same. Therefore, even if accuracy may be increased on some days of the week, it may also be reduced on other days, thus netting very little improvement on average. However, if we isolate a few single days, improvements can be observed. Looking at Thursdays only when new shipments are expected, increasing shipment and backstock balance is effective at reducing MAD, implying that there is some truth operationally to sending more shipment than the exact expected demand before the next shipment. It is difficult to determine the optimal setting for the shipment and backstock balance, as every store tends to display slightly different behaviors. For the purpose of this project, it was set to +30% for CD1 stores and +0% for CD2 stores, as CD2 stores receive shipments daily and purpose of replenishment shipments from CD2 is to restock exactly what was sold the previous day.

CD2 option

Section 3.1.6 discussed CD2 replenishment as an option for upstream movements and the impact on day of week inventory level was visualized in Figure 3-9. For a store during a historical time period on CD2 replenishment, accuracy metrics are calculated for simulations with and without the CD2 replenishment for comparison in Figure 3-17.

With CD2 shipments, the simulation performs better both in MAD and MD across the board, meaning it is more accurate with smaller systematic biases. Therefore, the CD2 shipment option has a quantifiable and positive impact on modeled inventory of stores connected to CD2s. Thus for model evaluation, CD2 option will only be used for stores during time periods connected to CD2s. In addition, it is a key feature in the integrated dashboard, as understanding the inventory impact of serving stores currently only connected to CD1 with CD2 replenishment is of particular interest to business planning. The details will be discussed in Section 5.1.

Final store inventory accuracy results

With all the model mechanisms examined individually through quantitative assessments, final store inventory simulation outputs for six select CD1 and CD2 stores (connected to CD1 and CD2 during the historical time period when the accuracy metrics are calculated) are presented in Figure 3-18. Approximate ranges of all the accuracy metrics for the six specific stores are presented in Table 3.1.

Figure 3-17: Daily % MAD and MD for various inventory adjustment options for a 4-week simulation period for a specific store during a historical time period connected to a CD2; 6A/6B: upstream inputs without/with CD2 replenishment; inventory trend using the actual beginning and ending inventory of the simulation period; inventory minimum and maximum included; inventory adjustment coefficient at -0.5; shipment and backstock balance and covariance not included

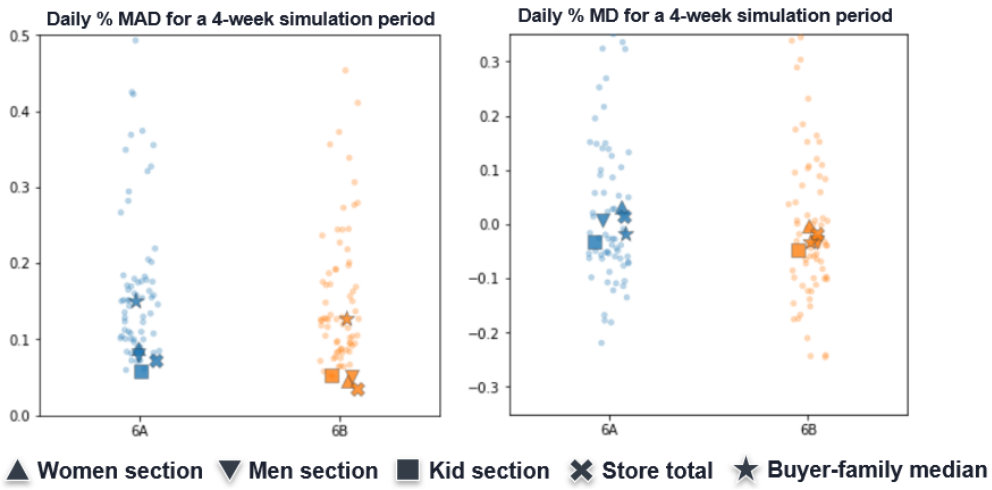
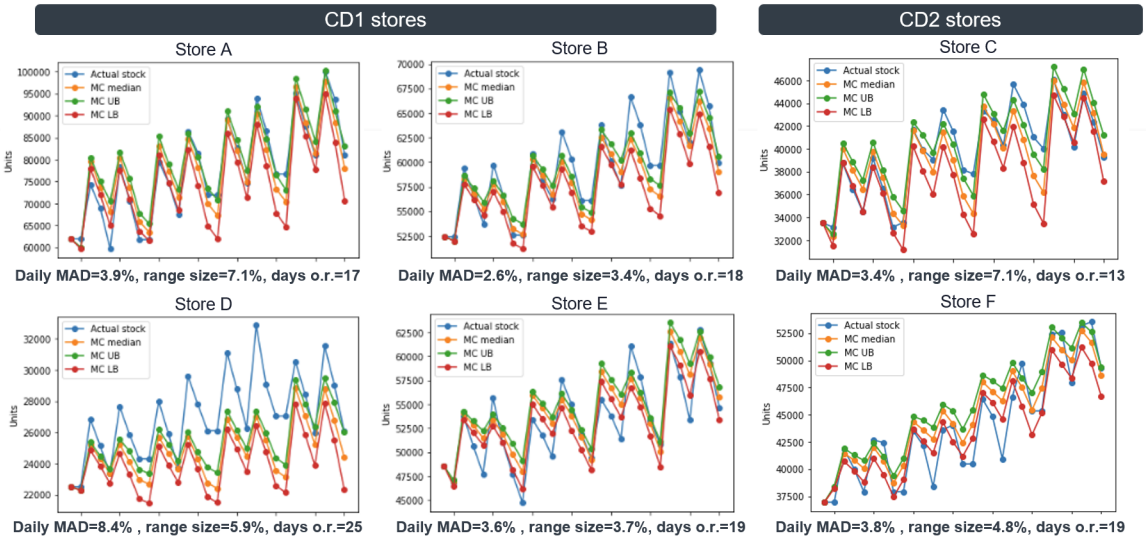


Figure 3-18: Final Monte Carlo store inventory results for a 4-week simulation period for 6 select stores in Spain; daily % MAD, estimate range size, and days out of estimate range indicated for simulation; all stores using linearly extrapolated inventory trends, inventory adjustment coefficient of -0.5, inventory max and min and covariance; CD1 stores include shipment and backstock balance of +30%; CD2 stores includes upstream inputs with CD2 replenishment



	Store	Section	Buyer-family median
Daily % MAD	2 to 4%	2 to 8%	10 to 15%
Daily % MD	$<\pm 3\%$	$\pm 8\%$	$\pm 8\%$
Weekly % MAD	2 to 3%	2 to 7%	9 to 12%
Estimate range size	3-8%	2 to 10%	10 to 20%
Days out of estimate range	12 to 20	15 to 25	$\sim$20

Table 3.1: Ranges of store aggregate inventory accuracy metrics for the six specific stores examined

Overall, the simulation is able to replicate historical store inventory in many cases, but not always. Therefore, it is particularly important to discuss how this model should be used and where its limitations lie.

First of all, the simulation model is not a predictive model like a forecast. A forecast looks at historical patterns, applies a set of assumptions in the form of an underlying model, and projects certain values forward. In addition to historical patterns, the store inventory simulation also relies on a set of user inputs, especially to structure upstream inventory movements, requiring domain knowledge from the users. Once the model receives the user inputs, it is able to replicate inventory accurately, but it cannot do so in a vacuum without user inputs. In other words, the model does not predict inventory purely from historical data. It simply simulates the inventory, with knowledge of how much inventory is expected to flow in and out of the store over the simulation period, which is derived from both historical data and user inputs.

Second, the simulation model is currently not able to simulate any non-linear inventory trends. Take store D as an example in Figure 3-18. Over the 4-week simulation period, an increase is expected, but it is far from linear as inventory peaks in week 3, which the model is not able to simulate, resulting in high MAD and MD. Inventory trends are intentionally designed to be linear for user-friendliness purposes, as they rely on user inputs for forward-looking time horizons. Therefore, the trade-off that any non-linear inventory trend behavior like that for store D cannot be simulated is one we have to accept.

Third, the quantitative and qualitative evaluation of the model is a conceptually a rigorous approach, but the model parameters and features are far from truly optimized based on those exercises. As we have observed, many different model options do not vary significantly in performances, and there isn't always a clear answer as to which trade-offs are better. Therefore, the selection of model parameters and features may

be somewhat subjective and dependent on intuitions. As the model continues to evolve and is perhaps deployed one day, the subjectivity of some of the model design choices is worth noting.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Machine learning model for display inventory and product portfolio complexity

Using store aggregate inventory from the Monte Carlo model, the project proceeds to disaggregate display inventory and further simulate the number of unique articles in each buyer-family combination, for both the entire store and the display room, using machine learning approaches.

4.1 The case for a machine learning approach

A Monte Carlo approach to simulate store aggregate inventory as described in Section 3.1 fundamentally depends on two important aspects about the data and operations. First of all, historical store volume in and out flows are known, from which inventory at any given point in time can be reconstructed with accuracy. Without being able to construct historical inventory, any attempt to simulate future inventory based on store volume in and out flows would have no ground of credibility. Secondly, future store volume in and out flows can be reasonably modeled, whether relying on forecasts, historicals, assumptions, or a combination of all of them. These are inputs that directly impact inventory level, and therefore are crucial to model fidelity.

Unfortunately, similar data and operations properties cannot be said about display inventory. Thus, extending the Monte Carlo approach to disaggregate display from

store inventory has proven to be difficult.

- When in-store sales are made, there is no direct indication of whether the inventory is coming from the display or stock room from a data perspective. Although the vast majority is presumably coming from the display room, there is no way to know for sure and small daily deviations in the data can accumulate and result in significant inventory deviation over a period of time.

- Although in-store movements between the stock room and display room are recorded, the data is messy. These movements happen very frequently and store associates may not follow operational guidelines fully or perform the scanning process accurately. In fact, to accurately estimate in-store movements in the business, RFID inventory data at various points in time is often leveraged to back into store movements. This approach is not helpful for the purpose of inventory simulation, as using in-store movement data derived directly from RFID inventory to reconstruct inventory leads to circular logic.

- Some stores have both an internal and external stock room. Internal stock rooms are physically attached to display rooms while external stock rooms are not, but rather in close vicinity. Due to physical space limitations, some stores are not able to fit both the display and stock room in the same space, hence requiring external stock rooms. Inventory may move first from the external stock room to the internal stock room and then to the display room, or directly to the display room. The multitude of in-store movement channels further contributes to the difficulty of reconciling inventory data.

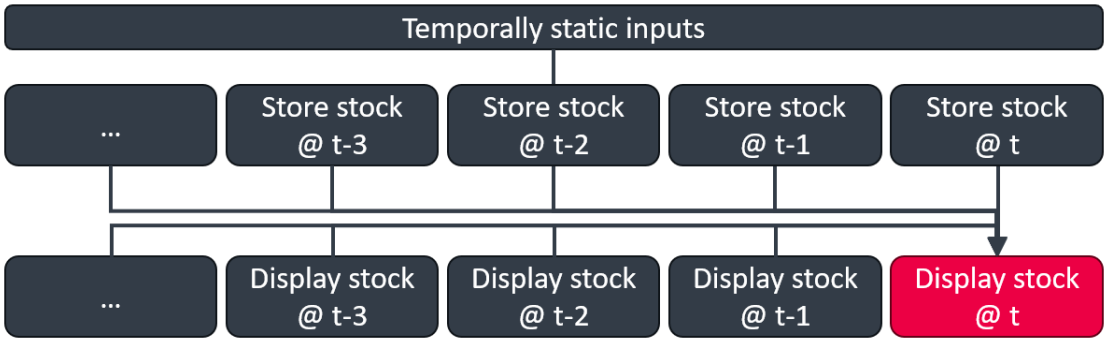
For all the reasons above, a machine learning approach is adopted to predict display room inventory, based on store inventory output from the Monte Carlo simulation, along with other store, product and temporal attributes. Compared to the Monte Carlo approach, a machine learning approach treats display room volume in and out flows as a "black box" and predicts display room inventory directly from other features. Store inventory and display inventory are typically highly correlated, as the display is meant to showcase articles available in store and the store is meant to carry articles that will go on display for sale. Therefore, a machine learning model is expected to be able to learn some of the correlations.

The case for modeling number of unique articles using machine learning approaches stems from the similar difficulty of not being able to reconstruct the quantity based

on volume in and out flows. In fact, the obstacles are much more fundamental. The model is built on buyer-family levels, and it is mathematically impossible to calculate the number of unique articles with just inventory data of such granularity, hence requiring machine learning approaches.

For all three prediction tasks at hand (display inventory, unique articles in store, unique articles in display room), the desired outputs are 4-week time series per buyer-family combination per store. Time series store inventory data, historical time series of the prediction quantity, along with temporally static inputs such as display capacity and day of week will be utilized to make the predictions. An illustration of the display inventory prediction task is show in Figure 4-1 (i.e. predicting display inventory for the next day).

Figure 4-1: Illustration of display inventory prediction



4.2 Literature review for time series predictions

4.2.1 Traditional time series techniques

Traditional time series forecast problems typically look at historical patterns of the same time series (e.g. using historical sales to forecast future sales). In the words of Lütkepohl 1991, "if time series observations are available for a variable of interest and the data from the past contain information about the future development of a variable, it is plausible to forecast it as some function of the data collected in the past" [9] [10]. Mathematically, this statement can be expressed as Equation 4.1, where $\hat{y}$ denotes predictions and y denotes historical data [9]:

$$\hat{y}_{t+h} = f(y_t, y_{t-1}, ...) \tag{4.1}$$

Traditional time series prediction tasks are a well studied class of problems and treatments have also evolved dramatically over the decades, especially with the development of modern machine learning [19]. But regardless of the specific forecasting techniques, the general approach remains the same:

- First, a model is proposed to relate any historical time series with future time series. Broadly speaking, the model can be anything from linear functions to neural networks.

- Subsequently, the model parameters are optimized using training data. The optimization can be performed analytical or numerical. More traditional forecasting models can be optimized analytically since solutions can be found symbolically from model properties. With more advanced and complicated machine learning approaches, the optimization process is typically performed numerically, usually through non-linear convex optimization techniques.

- The trained model with optimized parameters is then used to make forecasts. Because the model has some understanding of the temporal patterns within the time series, it is able to predict future time series based on historical inputs.

The literature for time series prediction models is vast. For the purpose of brevity, overview of a select few approaches is presented here, from which the project draws inspirations.

Auto regressive models with linear functions

Mathematically, auto regressive models with linear functions can be expressed generally as Equation 4.2, where prediction at time t is expressed as a linear combination of past time series from time $t-p$ until $t-1$, with an offset of ν and an error term of u_t [9]. ν and α do not have t subscripts, meaning they do not vary across time steps. By training those coefficients using historical data and applying them to forward-looking time periods, the model allows us to make $\hat{y}$ predictions incrementally into the future. For the purpose of this project, auto regressive models are too restrictive and not suitable for the tasks, as they only use past values of the forecast time series to make predictions.

$$\hat{y}_t = \nu + \alpha_1 y_{t-1} + ... + \alpha_p y_{t-p} + u_t \tag{4.2}$$

Linear regressions

The general form of linear regressions looks quite like auto regressive models, as demonstrated in Equation 4.3, where y stands for the target variable to forecast, xs the explanatory variables, βs the linear coefficients and ϵ the error term [8]. Unlike auto regressive models that strictly use past values of the forecast time series to make predictions, linear regressions build upon the techniques with added flexibility. No restrictions are placed on explanatory variables x, as long as they serve the prediction purpose. Once the explanatory variables are chosen, the process of using linear regressions to make predictions is exactly the same as that of auto regressive models, which involves training the linear coefficients and applying them to forward-looking time periods.

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon \tag{4.3}$$

Note that linear regression models have a number of regularization options. Without regularization, the training process typically minimizes the sum of squared differences between the training y and predicted y as the loss function. With regularization, additional terms, typically proportionally to β or β^2 are added to the loss function, which encourages the model to not overfit the training data and may lead to better performances. The inclusion of regularization only impacts the optimized value of β from the training process.

Tree models

The idea of regressive tree models for time series forecasting is very similar to linear regressions because the two models leverage the same inputs and outputs. Linear regressions require a set of explanatory variables as scalar inputs, and output a single scalar prediction at a time. Regressive trees, although different in their fundamental methodologies, have the same inputs and output dimensions. Therefore, linear regressions and tree models are direct substitutes for time series predictions, with the only functional difference being their performances in varying situations.

Regressive trees come in various different forms, the most fundamental one being decision trees. Using "a long list of if-else statements", the model is able to "predict some result x if a certain condition is true, and it will predict y otherwise" [8]. The biggest differentiation from linear regression is the introduction of non-linearity, because the

"if-else" statements are not mathematically linear. The training process focuses on finding the parameters of "if-else" statements that minimize loss functions, specifically the choice of explanatory variables to examine, the criteria of examination (i.e. bigger or smaller than a certain value for numerical variables, equal to a certain value or not for categorical), and the order of the statements.

Decision trees alone do not tend to perform well, because the "if-else" mechanism is too flexible, but not powerful in small quantities. Random forests and gradient boost trees are two models based on decision trees with additional mechanisms to improve model performances.

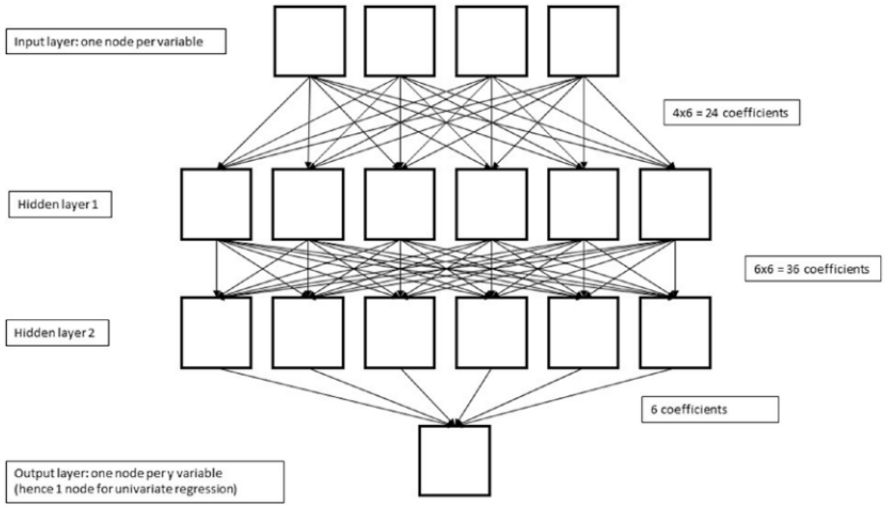
Random forests consist of a large number of decision trees in parallel and use the average outcomes as predictions. While one decision tree model "can sometimes be wrong, the average prediction of a large number of machine learning models is less likely to be wrong" [8]. For the average to be meaningful, each decision tree needS to be slightly different, which is achieved by training them on different subsets of training data acquired through a resampling process called bootstrapping [8].

In contrast, gradient boost trees consist of a large number of decision trees in series and use the final outcome as predictions. While the first tree is trained exactly like a decision tree, each subsequent tree is trained on the error of the previous trees, allowing them "focus on learning the things that are not yet understood" [8]. This incremental approach ensures that the training error can be further minimized compared to a single decision tree, but also makes the model much more susceptible to overfitting. To prevent overfitting, regularization can be adopted to penalize the models with higher complexity (e.g. more layers of boosted trees). A common example of such a regularized gradient boost tree model is XGBoost, where complexity scores are given to each leaf node in the trees and added to the training loss function as regularization terms [3].

Fully connected neural network models

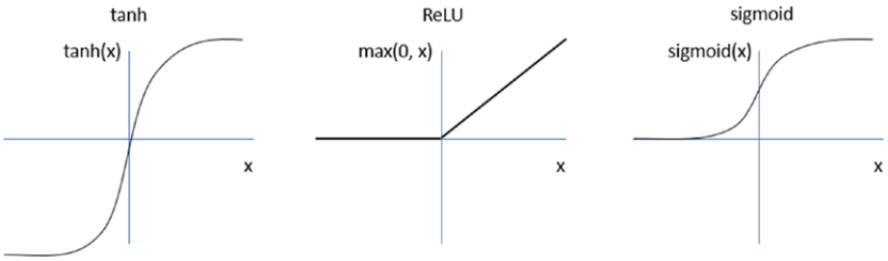
Fully connected neural networks (FCNNs) take scalar inputs of any dimension, apply a series of mathematical transformations (typically non-linear), and produce scalar outputs of any dimension. The inputs, mathematical transformations and outputs are mapped out layer by layer in the form of nodes, with linear operations connecting all of them. A visual illustration of a FCNN with 4 input nodes, 1 output node, and 2 hidden layers with 6 nodes each is showed in Figure 4-2 [8].

Figure 4-2: A fully connected neural network with 4 input nodes, 1 output node, and 2 hidden layers with 6 nodes each; source: Korstanje 2021 [8]



The mathematical transformations are often referred to as "activation functions", with tanh, ReLu and sigmoid as common selections. The behaviors of these activation functions are shown in Figure 4-3 [8].

Figure 4-3: Behavior of tanh, ReLu and sigmoid activation functions; source: Korstanje 2021 [8]

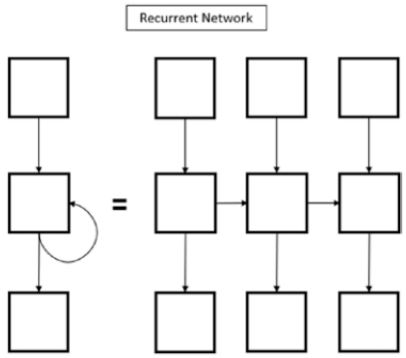


Compared to linear regressions or tree models, FCNNs can be even more effective at identifying patterns, as the activation functions can approximate any non-linear function, and any number of them can be included in a model. Furthermore, a major advantage of FCNNs is the output flexibility. Even though Figure 4-2 shows an output layer of a single node, there is no limitation on the output dimension as long as the outputs remain as scalars. In other words, a single model is able to make multiple predictions at once, unlike linear regressions and tree models.

Recurrent neural network models

Recurrent neural network models (RNNs) build on the concept of FCNN, but incorporate a loop, where "the inputs of have a feedback relation with each other" as shown in Figure 4-4 [8]. The feedback process forces the model to examine the inputs and make predictions incrementally, which makes it particularly suitable for "data that has sequences", such as time series and written texts [8]. How RNNs are able to learn sequences is a fundamental differentiation of the approach.

Figure 4-4: Illustration of a simple recurrent neural network; source: Korstanje 2021 [8]



For time series predictions, a simple recurrent neural network typically is not used directly and is further adapted with more complexity. One of the state-or-the-art example is long short-term memory (LSTM), which is able to understand sequences on multiple time scales, and hence is suitable for time series predictions with cycles and seasonalities [8].

4.2.2 Application of traditional techniques to our prediction tasks

Despite the vast literature on time series predictions, none of them are directly applicable to our predictions tasks, as the problem here is rather uncommon. Specifically, the difference lies in the fact that our prediction makes use of two different time series both of importance, namely store inventory and historicals of the prediction time series, as opposed to just the historicals. Historicals are included such that the time series can maintain temporal consistency as new predictions are made. Meanwhile, store inventory is included not only because it is the output of the Monte Carlo store simulation from Chapter 3, but also because predictions need to vary as store inventory

varies, as part of the simulation tool. As a result of this need to use two time series as inputs, two complexities arise.

First of all, not all models are able to easily take two time series as inputs. Take RNNs for example. Although suitable for time series predictions, ultimately the models look at historical time series and carry them forward for predictions. Tools like DeepAR developed by Amazon are able to combine RNNs with extra scalar regressors (e.g. holidays), but fundamentally are unable to use both time series as inputs with inherent temporal properties [8]. That said, for models that only take scalar inputs (i.e. linear regressions and tree models), values at each time step are can be featurized separately. Although temporal adjancencies are lost, they are able to take multiple time series as inputs and treat them equally in the training process.

Second, it is difficult to control how much the models will draw from each time series to inform predictions. All the training process does is finding the combination of model parameters that minimize the loss function, even if minimizing the loss function implies putting no weight on one time series and all on the other in the extreme case. This behavior can be quite problematic for the purpose of our tasks, because intuitively both time series are important and serve different purposes. Without a way to control how the models draw upon the two inputs, it is possible that the models do not learn the patterns as intended.

4.3 Exploring design choices for ML model

Broadly speaking, there are six sets of design choices to be made: 1) how to choose X and Y features, 2) how to treat multiple time series across buyer-family combinations and store, 3) how to featurize the temporal aspects of time series, 4) in what order to make time series predictions, 5) over what scope of predictions to minimize the loss functions, and 6) which machine learning models to use. The choices are not truly independent and this section will go through the options and inter-dependencies considered in this project.

4.3.1 Choice of X and Y variables

The most straight-forward choice of Y features is to simply predict display inventory, number of unique articles in store and on display directly. Although completely valid, this approach could potentially lead to lower model performances.

First of all, historical time series of these prediction quantities are highly correlated. If the display inventory was high yesterday, it is also very likely that the display inventory was high the day before, because the level of display inventory tends to literally carry over to the next day. As shown in Figure 4-1, historical time series of the prediction quantities are used as X features in the actual predictions, and using highly correlated X features can lead to degeneracies between variables, which can lower model performance.

Secondly, this straight-forward selection of Y variables may result in intrinsic inconsistencies across various machine learning models. For instance, if the number of unique articles in store and on display are predicted directly and separately, there is no guarantee that the number of unique articles on display will be smaller than that in store for a specific buyer-family and store on a particular day across different models. Although one can use the number of unique articles in store as additional inputs to predict the number of unique articles on display, such an approach requires using the output of one machine learning model as inputs to another for predictions, which may result in compounding prediction errors.

To alleviate these potential concerns, percent of store inventory on display and percent of store unique articles on display are selected as Y variables instead of display inventory and number of unique articles on display. In comparison, these quantities do not display as much correlation in the historical time series. For example, even if the quantity of display inventory remains similar in level over a few adjacent days, the percent of store inventory on display may still vary, due to changes in total store inventory. Additionally, given that these quantities in the training data will all fall in the range of 0 to 1, it is less likely that prediction will exceed 1. Similar intuitive percentage based variables are not available for unique articles in store and therefore will be predicted directly as Y variables of the ML models.

In terms of X features, the following are explored based on domain knowledge:

- Historical store inventory time series, up until the day of prediction: store inventory is necessary because the prediction tasks examine in how Y features will change as a result of store inventory changes.

- Historical time series of Y feature, up until one day before prediction: the model needs to know the Y feature values in the days leading up to the prediction to maintain temporal continuity.

- Day of week of the day of prediction: new product shipments arrive in stores twice a week (usually Mondays and Thursdays, but can vary by store) and have significant impact on store inventory level and product assortment.

- Store capacity and display capacity: the capacity of the display and store is expected to limit what products go on display. Note that this feature is only relevant for models that involve multiple stores/sections, because each store/section only has one single capacity.

Note that due to the inherent correlations between time series data, not all of them will be important in the prediction tasks and the inclusion of too many X features may result in overfitting the data and lower overall model performance. Therefore, the time series features will be selectively chosen, based on their relatively importance to the models.

4.3.2 Multiple time series treatment across buyer-family combinations and stores

The prediction tasks involve multiple time series across buyer-family combinations and stores, and each of them shares some level of common characteristics with another. Just like with any machine learning model where there is always a trade-off between allowing enough freedom in the models to pick up desirable patterns and avoiding overfitting the training data using models with too much freedom, the treatment of multiple time series here is no exception. If each store and buyer-family combination is treated as a separate model with its own prediction task, the model will be very flexible, but may be computationally intensive and result in overfitting, especially if data is limited for each buyer-family in each store. On the other hand, if a single model is trained for all buyer-family combinations and stores (with buyer-family and stores as categorical variables), the model may not be able to pick up patterns that impact certain buyer-family combinations differently from the others. Unfortunately, there is no silver bullet to which methods tend to perform the best, since each problem is unique in its own ways. In this project, five approaches are explored, with varying level of grouping buyer-family combinations and/or stores before using them to train the models, the inspirations of which are drawn from Maharaj, D'Urso and Caiado 2019 [11]. Prediction accuracy will be computed for each approach to identify if there are systematic trends in their performances for the prediction tasks of interest:

1. Separate ML model for each buyer-family combination for each store. This approach allows the most freedom, as each model is built independently, using data only for the specific buyer-family and store.

2. Separate ML model for each section for each store. Sections are natural divisions of products within stores, often with different operational procedures. Buyer-family combinations within each section is expected to share more similarity than across sections, potentially leading to higher prediction performances.

3. Separate ML model for each buyer-family clusters for each store, where the buyer-family clusters are defined using unsupervised algorithms (K-means, with varying numbers of clusters). Along the lines of building separate ML model for each section, models are built on clusters of buyer-family combinations within each store that share similarities. However, instead of relying on domain knowledge about sections within stores, the similarities are defined through an automated process based on distance metrics between each buyer-family combination in the clustering space. It is worth noting that selecting the right clustering features is not a trivial task, because the "similarity" of interest between buyer-family combinations cannot be any similarity, but rather similarity in how each X feature contributes to the ML model predictions. An example of a bad clustering feature may be the average price of products within each buyer-family, as there is no obvious reason to believe that buyer-family combinations with similar average prices can lead to store inventory at time t having similar impact on the percent of store inventory on display at time t. In the case here, the clustering features are selected to be the coefficients of linear regression models built separately for each buyer-family combination for each store, as laid out in approach 1. Intuitively, the coefficients can be interpreted as how each X feature linearly impact the Y predictions and similarity in the coefficients can imply similarity in how each X feature contributes to the ML model predictions. This clustering approach is applied to non-linear machine learning models as well, despite that the coefficients only capture linear contributions. Quantifying non-linear contributions of each X feature to the Y predictions is difficult, and simplification assumptions are made here.

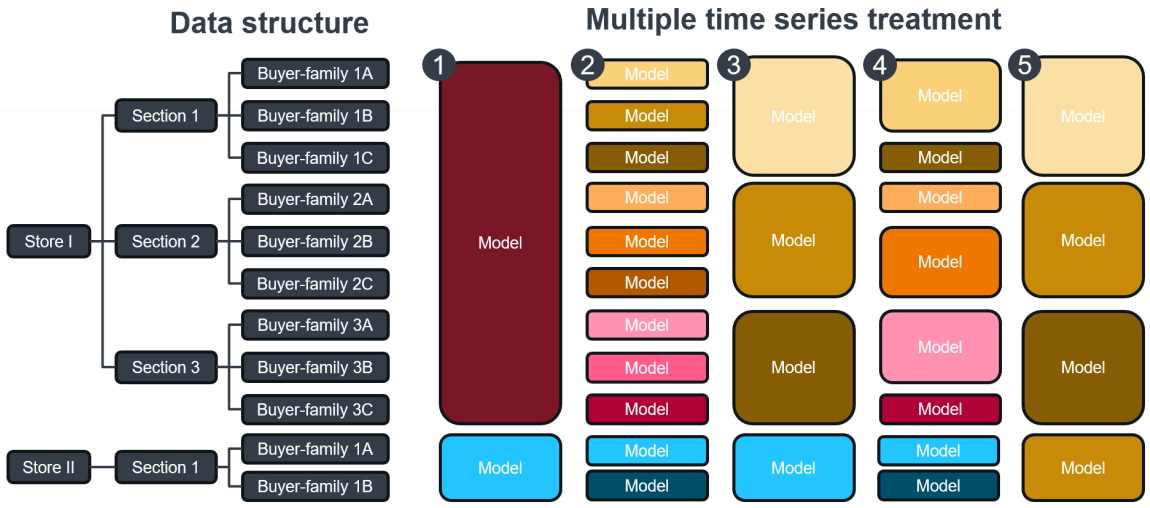
4. Separate ML model for each store. If all buyer-family combinations have considerable similarities but not across stores, this approach is expected to perform relatively well, given that it allows the freedom for each store to receive

its own independent model.

5. Separate ML model for each store/section cluster, where store/sections are defined using unsupervised algorithms. Like buyer-family clusters, store clusters are defined through a K-means algorithm, where stores with high operational similarities are grouped together. In order for the algorithm to explore these similarities, four clustering features are carefully selected: daily unit of sales, display capacity, total store/section capacity, and store/section classification (used internally to indicate tiers of stores, which are typically an important indicator derived from sales velocity, store size, locational importance, volume of new articles, etc.). Additionally, clusters are built separately for stores connected to CD1 and CD2 for replenishment because significant operationally differences are expected.

A visual illustration of the five multiple time series treatments is presented in Figure 4-5.

Figure 4-5: Visual illustration of the five multiple time treatments explored in this project



4.3.3 Temporal featurization of time series

For the prediction tasks, both the inputs and outputs involve time series. The inherent temporal correlations in the inputs and outputs lead to choices of embedding and model output dimensions that may lead to fundamentally different outcomes.

Time series inputs and outputs can be embedded as either vectors or scalars. Temporal

adjacencies are fundamental properties of time series, as the time series value at time t versus $t+1$ and $t+1$ versus $t+2$ share temporal similarities. To accurately represent the temporal adjacencies in machine learning models, the time series inputs need be embedded as vectors, which place limitations on the possible choice of machine learning models as not all of them are compatible with vector embeddings. In contrast, scalar embeddings allow much more freedom in the choice of machine learning models, but cannot fully capture the temporal adjacencies like vector embeddings can, because every single value in the time series becomes its own input and output with no inherent temporal relationship between each other.

4.3.4 Order of time series predictions

When making time series predictions, the order in which they are generated can have a large impact on the final outcome.

Suppose that a historical time series is given until time t. Making a prediction for $t+1$ using data up until t is straight-forward as it only requires historical data. However, to adopt the same approach for $t+2$, $t+1$ predictions are necessary as inputs. In other words, each prediction needs to be made incrementally as the next prediction depends on the previous. This step-by-step method can be problematic at times, as any systematic bias to each prediction will accumulate and lead to significant deviations further into the prediction time horizon.

This problem can be circumvented, if each prediction is made only using historical data, meaning all predictions for $t+1$ and onward are made simultaneously and independently from each other using only data at time t or before. However, new challenges are presented with this workaround. Because each prediction is independent, there needs to be a separate (or at least partially separate) model for each time step in the prediction horizon and temporally adjacent predictions may not be fully consistent with each other. Furthermore, model performances may be low, because making time series predictions using data up until a number of time steps ago can be a difficult task.

4.3.5 Prediction scope of loss function minimization

Independent from the order of time series prediction, the loss function of the proposed model can be minimized for each time series prediction individually, or all the predictions at once in the training process. This choice becomes highly crucial if the

time series predictions are made incrementally when the next prediction depends on the value of the previous time step, whether it is data or past predictions. If the model is trained to minimize each time series prediction and the predictions are made incrementally, the model only learns to predict the next time series value well, which can lead to longer term deviations if each prediction is subject to systematic biases. This problem may be avoided if the loss function can be minimized holistically over the entire time series prediction at once, since the process of minimizing the loss function will prevent longer term deviations that may stem from systematically biases. However, not all models allow loss function minimization over multiple predictions.

Note that if each prediction is made only using store inventory and not historical time series data, the prediction scope of loss function minimization is not expected to cause significant differences. Because each prediction is made somewhat independently from each other, minimizing the loss function for each prediction tends to lead to minimization globally.

4.3.6 Selection of machine learning models

The discussion of model choices cannot be carried out in a vacuum without the context of actual machine learning models. Each model has its own specific input and output limitations and not all permutations of design choice options are possible, nor are they all useful. Table 4.1 lays out how linear regression, tree, fully connect neural network and recurrent neural network models are able to accommodate various design choices.

Featurization	Order	Scope	LRs and trees	FCNNs	RNNs
Scalars	Incremental	Individual	Standard	Standard	N/A
Scalars	Incremental	Holistic	N/A	N/A	N/A
Scalars	Simultaneous	Individual	Independent model for each prediction time step	Independent model for each prediction time step	N/A
Scalars	Simultaneous	Holistic	N/A	Single model with one output layer node per prediction time step	N/A
Vectors	Incremental	Individual	N/A	N/A	N/A
Vectors	Incremental	Holistic	N/A	N/A	Standard
Vectors	Simultaneous	Individual	N/A	N/A	N/A
Vectors	Simultaneous	Holistic	N/A	N/A	N/A

Table 4.1: Machine learning model design choices

4.3.7 Summary of ML model design choices

Section 4.3 covers various choices that impact the design of machine learning models.

X and Y variables and multiple time series treatment are choices independent from the machine learning model selections, as they impact the underlying dataset used to train models. In contrast, the temporal featurization, prediction order and loss function scope are closely tied to the machine model selection, as illustrated in Table 4.1. For this project, RNNs will not be explored, because it is unable to accommodate two time series inputs effectively unless custom architecture is adopted, as explained in Section 4.2.2.

For linear regressions and trees, only the standard architecture will be explored (Section 4.5.1 and 4.5.2), because training a separate model for each prediction time step is not practical for simulations. For fully connected neural network models, only the architecture with simultaneous prediction order and holistic loss function scope will be explored (Section 4.5.3). FCNNs tend to be computationally intensive and are not expected to significantly outperform linear regressions and trees for the same tasks in this project. The advantage of FCNNs lies within their abilities to produce multiple predictions at the same time and tune the model parameters to minimize loss function for all predictions holistically. Therefore, only one FCNN architecture will be explored to complement the standard linear regression and tree model architecture.

4.4 Preparing the dataset

The source of all time series data is article level RFID inventory by day by store, from which store inventory, percent of store inventory on display, number of unique store articles and percent of store articles on display can be calculated daily.

As laid out in Section 4.3, historical time series data of the predicted time series is used to make predictions. For models that make one prediction at a time (linear regressions and trees), historical data from $t-1$ to $t-7$ is leveraged to make predictions at time t, meaning the data set used to train the models is constructed by looking at the RFID time series inventory in the trailing 7 days of each day. In the case that any data is missing from t to $t-7$, it will not be used to train the machine learning model, because linear regressions and trees cannot handle missing data easily. Note that the same day can appear multiple times in the data as different time steps (e.g. as $t-1$

for one data entry, and $t-2$ for another). For models that make one prediction for each prediction time step all at once (FCNNs), the same historical data is leverage to make predictions from time t until $t+27$. Because each data entry looks at a total of 35 days and missing data is sometimes inevitable for a time window of this duration, a linear extrapolator is used to fill in the missing data. Otherwise, the process of constructing the data set remains the same.

While preparing the data set, two additional assumptions are made:

- If data is not available on a Sunday, it is assumed that Sunday inventory is the same as the leading Saturday inventory for the buyer-family. A significant proportion of stores are closed on Sundays and it is not uncommon that RFID inventory data is not recorded when the stores are closed. It can generally be assumed that inventory will remain the same as the previous day in these cases. Sunday extrapolation is performed before the linear extrapolation for the FCNN data set.

- The data set is only constructed from time periods considered as "steady state", without disruption from end of campaign sales and campaign transitions.

For each buyer-family combination and store, approximately 400 data points with complete X and Y features are compiled from 2021, 2022 and 2023 February to May and August to mid-November for the linear regression and tree data set. There are just over 200 stores in Spain and approximately 80 buyer-family combinations are modeled at a time for the simulation, implying that the entire data set is roughly 6 million rows across relevant buyer-family combinations and stores in Spain from 2021 to 2023 in the selected 6 months for each year. Not all of them are always used in the models and the specifics depend on the treatment of multiple time series across stores and buyer-family combinations, as discussed in Section 4.3. The data set is further split into train, validate and test, at proportions of roughly 60/20/20. The data is intentionally not extended to 2020 or earlier, given the potential impact of COVID on store operations during the months following March 2020. For FCNN, the size of the data set is similar, because the impact of missing data is avoided through linear extrapolation, while each data entry itself consists of a longer time series (35 time steps vs. 8) and therefore fewer can be prepared from the same time periods.

4.5 Machine learning models training and testing

4.5.1 Regression models training and testing

Regression models are first examined as potential candidates for the simulation. All results presented use L2 regularization, meaning the sum of squares of the regression coefficients is included in the loss function as an additional term to minimize. The strength of regularization is optimized using the validation data set.

Multi-colinearity and P values

For linear models, explanatory variables that are highly correlated linearly introduce redundancy to the model, lower model interpretability, and may lead to lower accuracies in some cases (although not guaranteed). Therefore the best practice is to look at the multi-colinearity of the explanatory variables and select the necessary ones first before evaluating model performances.

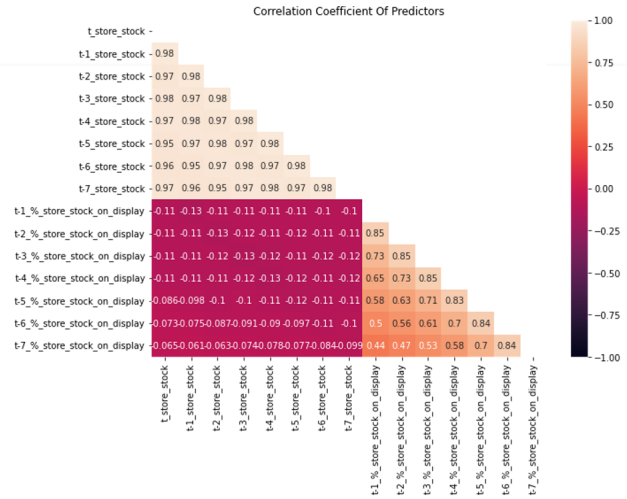
In addition, because linear regressions are originally based on statistical theories, P values can be calculated to examine how statistically significant each explanatory variable is at explaining the data. Intuitively, P value can be understood as how likely the natural statistical fluctuations in the underlying data can reproduce the patterns explained by each explanatory variable, with lower values meaning less likely for the natural fluctuations and more statistically significant for the explanatory variables. For a specific store, P values are calculated with all the explanatory variables included. Along with multi-colinearity, they should inform which variables to keep when evaluating accuracy results.

For a specific store, the P value and multi-colinearity results for all numerical explanatory variables for % of store inventory on display, number of unique articles in store and % unique articles in store on display are shown in Figure 4-6, 4-7 and 4-8.

For % of store inventory on display, it is not surprising that historical store inventory is highly correlated. If inventory is high the previous day, it is likely that inventory is also high the next day. In contrast, historical % of store inventory on display is significantly lower, because as store inventory changes, display inventory will not exactly mirror these movements, and therefore as a percentage, historical display inventory is less correlated. Lastly, the correlation between store inventory and % of store inventory on display is weak as expected since the two quantities are driven by different factors.

In addition, the P values (and regression coefficients, for reference) for each numerical explanatory variable are provided. Historical % of store inventory on display generally has low P values, while results for store inventory are mixed. In particular, store inventory at time t (same day as prediction), $t-1$ (the day before prediction), and $t-7$ (a week before prediction) have the lowest P values. This result makes strong intuitive sense, and can be generally observed across the same analysis for other stores as well. Display inventory is strongly influenced by store inventory that same day and the day before, as well as a week before, due to the weekly cyclical nature of store operations. Given that store inventory is so highly correlated from the multi-colinearity test, a decision for the linear regression models is made to keep only store inventory at these three time steps, while keeping all historical % of store inventory on display.

Figure 4-6: Multi-colinearity, P value and regression coefficients for % of store inventory on display for a specific store



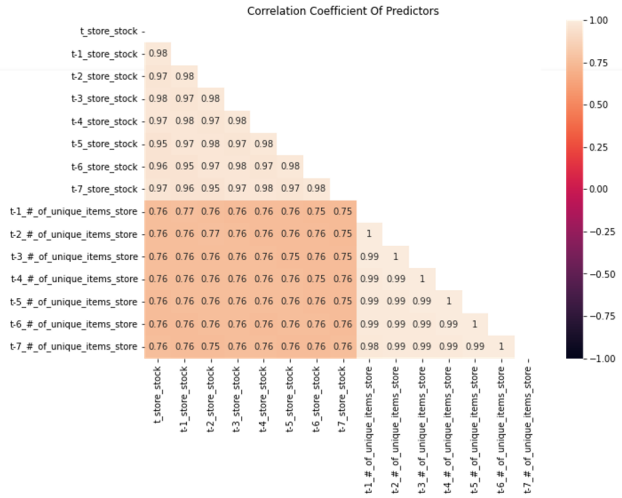
X_features	p_value	coef
t_store_stock	0.000	-0.085
t-1_store_stock	0.000	0.053
t-2_store_stock	0.916	-0.001
t-3_store_stock	0.146	0.010
t-4_store_stock	0.181	0.009
t-5_store_stock	0.000	-0.024
t-6_store_stock	0.611	0.003
t-7_store_stock	0.000	0.024
t-1_%_store_stock_on_display	0.000	0.135
t-2_%_store_stock_on_display	0.046	-0.004
t-3_%_store_stock_on_display	0.002	-0.007
t-4_%_store_stock_on_display	0.000	0.018
t-5_%_store_stock_on_display	0.097	-0.003
t-6_%_store_stock_on_display	0.001	-0.007
t-7_%_store_stock_on_display	0.000	0.008

For number of unique articles in store, the biggest difference in multi-colinearity compared to % of store inventory on display is that historical number of unique articles in store itself is much more correlated, for the same reason store inventory is. Furthermore, number of unique articles don't tend to fluctuation very much from day to day, as the size of assortment remains relatively constant over a week.

For explanatory variables with such high multi-colinearity, P values (and regression coefficients) are not particularly insightful, as the choices will unlikely result in any meaningful accuracy differences. Store inventory from time t to $t-2$ and % of store inventory on display at time t and $t-1$ are selected, as they consistently display low P values (and non-zero regression coefficients) across stores, including the one examined

here.

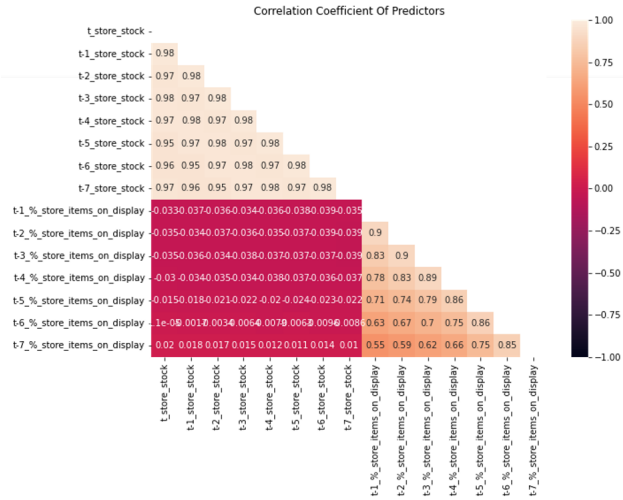
Figure 4-7: Multi-colinearity, P value and regression coefficients for number of unique articles in store for a specific store



X_features	p_value	coef
t_store_stock	0.000	13.500
t-1_store_stock	0.000	-10.123
t-2_store_stock	0.000	-1.846
t-3_store_stock	0.001	-1.126
t-4_store_stock	0.749	-0.113
t-5_store_stock	0.000	1.574
t-6_store_stock	0.810	-0.085
t-7_store_stock	0.006	-0.812
t-1_#_of_unique_items_store	0.000	44.026
t-2_#_of_unique_items_store	0.000	2.756
t-3_#_of_unique_items_store	0.000	3.341
t-4_#_of_unique_items_store	0.000	2.438
t-5_#_of_unique_items_store	0.000	-2.923
t-6_#_of_unique_items_store	0.058	-1.311
t-7_#_of_unique_items_store	0.003	1.557

Last but not least, the same tests are performed for % of unique articles in store on display. Prediction of this quantity is much harder than the previous two as it combines predictions of unique articles and display room into one, making it challenging using only historical time series and store inventory.

Figure 4-8: Multi-colinearity, P value and regression coefficients for % of unique articles in store on display for a specific store



X_features	p_value	coef
t_store_stock	0.000	-0.036
t-1_store_stock	0.000	0.028
t-2_store_stock	0.900	-0.001
t-3_store_stock	0.873	-0.001
t-4_store_stock	0.878	0.001
t-5_store_stock	0.005	-0.019
t-6_store_stock	0.009	0.018
t-7_store_stock	0.369	0.005
t-1_%_store_items_on_display	0.000	0.144
t-2_%_store_items_on_display	0.000	0.016
t-3_%_store_items_on_display	0.033	0.005
t-4_%_store_items_on_display	0.000	0.012
t-5_%_store_items_on_display	0.060	-0.004
t-6_%_store_items_on_display	0.523	-0.001
t-7_%_store_items_on_display	0.002	0.005

P value results indicate that store inventory at time t and $t-1$ tend to be more statistically significant. Beyond $t-1$, results are mixed for different stores. P values

for historical % of unique articles in store on display also tend to be mixed, while multi-colinearity is lower than that of number of store unique articles, due to larger relative fluctuations in percentage quantities. Given that there isn't a strong case for which explanatory variables to keep or drop, the same set as % of store inventory on display is used.

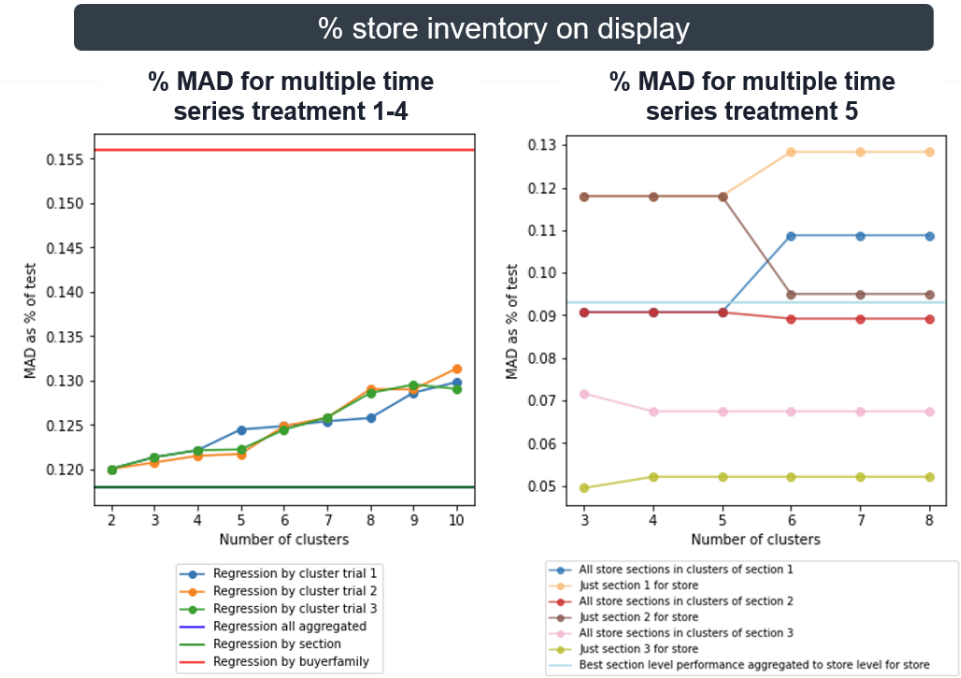
Comparison of various multiple time series treatments

Based on the selected numerical explanatory variables, accuracy results of the 5 multiple time series treatments illustrated in Figure 4-5, are shown in Figure 4-9, 4-10, and 4-11. Accuracy results are based on the test data set mean absolute deviation (MAD), weighted as a percentage of test data values. For multiple time series treatments based on clusters (buyer-family or store/section), MAD will vary based on the number of clusters. For store/section clusters (treatment 5), MAD is calculated for all store sections in the same cluster as the store section of interest, and for just the store section of interest. Results from the number of clusters that perform the best for each store section of interest are aggregated for store total results, which can be compared to outcomes from other multiple time series treatments.

For % of store inventory on display, regressions by section and store aggregate perform better than regression by buyer-family and buyer-family clusters, an observation consistently across stores. Store cluster regressions further outperform all other treatments for this store, and perform at least as well across all other observed stores. Intuitively, it can be understood that the correlation of store inventory and display inventory tends to exhibit behavior similar across buyer-family combinations and stores. Therefore, by running regressions on larger sets of data for multiple buyer-family combinations and stores, model performance can be improved.

For number of unique store articles, a different multiple time series treatment performance hierarchy is observed. Store/section cluster regressions perform the worst, followed by regressions by section and by buyer-family. Store aggregate and buyer-family cluster regressions perform the best, although the margin of improvement is at most half a percentage point from $\sim$4% to $\sim$3.5%. The same hierarchy is not consistently observed across all stores, but store aggregate and buyer-family cluster regressions tend to perform well, and store/section cluster regressions at times depending on the stores. This result implies that the number of unique articles in stores do not tend to have strong linear correlations with store inventory along section or

Figure 4-9: Linear regression test results for % of store inventory on display for 5 multiple time series treatments



individual buyer-family lines. Instead, patterns are better learned for individual stores, or groups of buyer-family combinations, or at times groups of store sections.

The behavior observed in % of unique articles in store on display mirrors that in % of store inventory on display, and is observed consistently across stores. Regressions by store/section clusters usually perform the best, and as well as if not slightly better than regressions by section, by buyer-family clusters, and store aggregate regressions, while regressions by buyer-family combination consistently perform the worse.

Overall, this exercise shows that various treatments of time series can have a profound impact on linear regression performances, and their relative accuracy sheds light on how the types of patterns in the data can be similar along store, section and buyer-family boundaries. Accuracy results in this section will be compared to those in Section 4.5.2 to determine whether linear regressions or tree models are more suitable for the project.

Figure 4-10: Linear regression test results for number of unique store articles for 5 multiple time series treatments

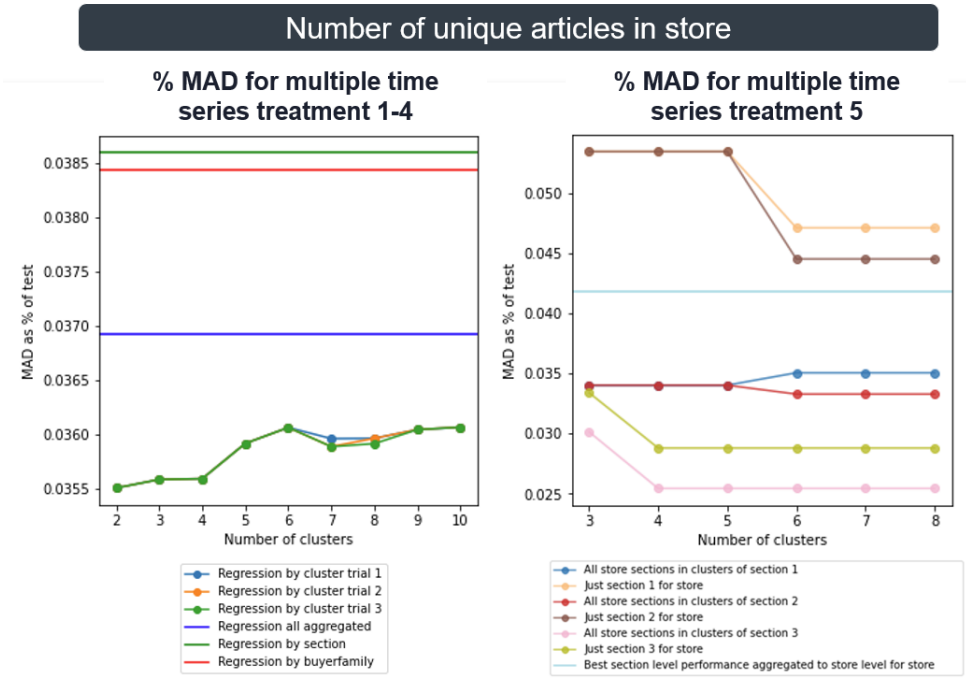
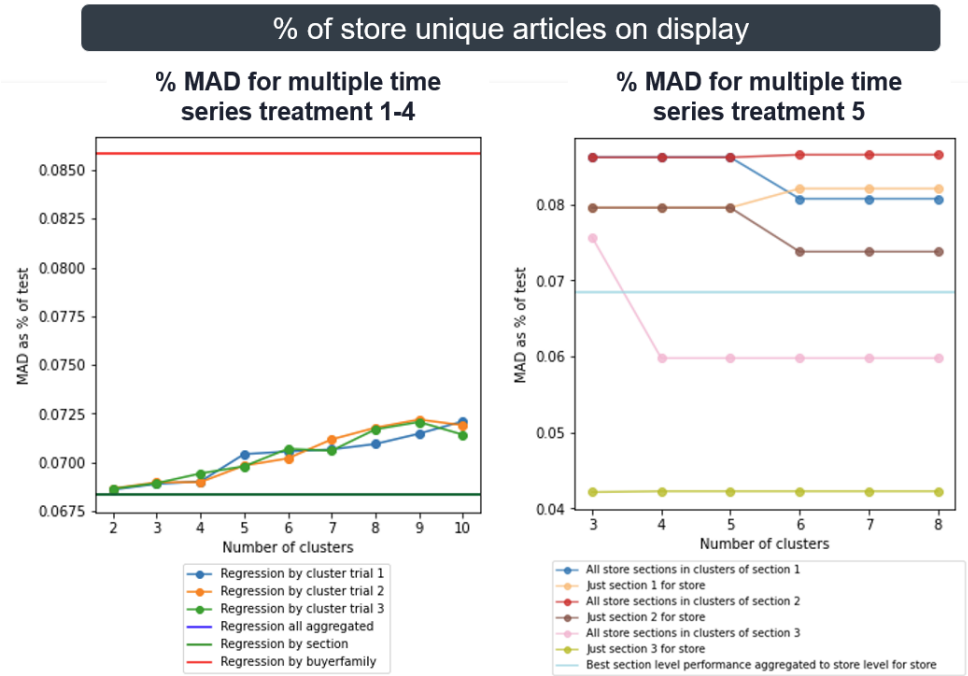


Figure 4-11: Linear regression test results for % of unique articles in store on display for 5 multiple time series treatments

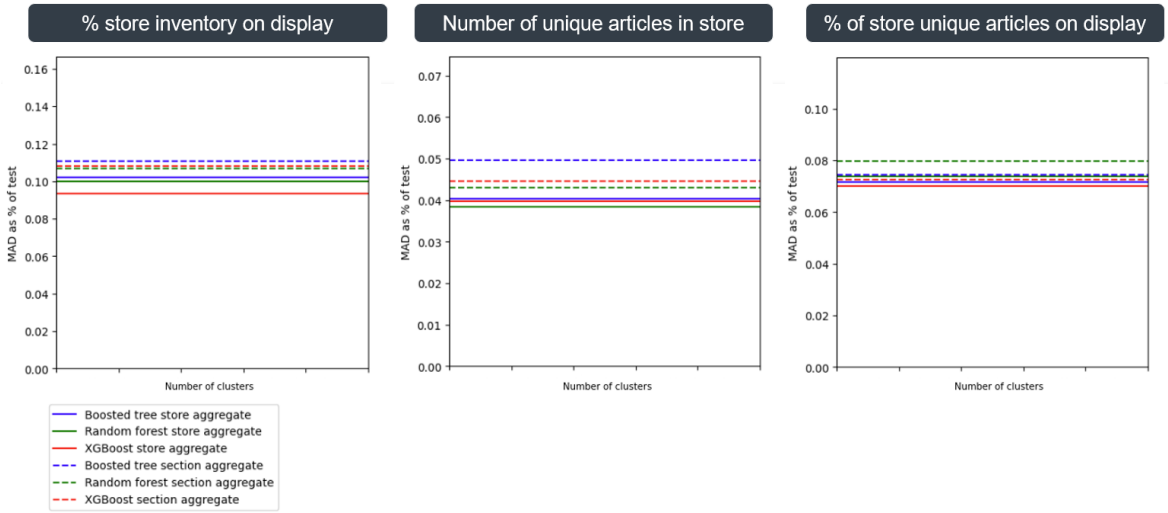


4.5.2 Tree models training and testing

Comparison of random forests, gradient boost trees and XGBoost models

First of all, a comparison of the various tree models is performed. Section 4.2.1 discussed three architectures based on decision trees, namely random forests, gradient boost trees and regularized gradient boost trees. The performances of the three models are benchmarked against each other for store aggregate and section aggregate and for for all three machine learning prediction tasks for a specific store in Figure 4-12.

Figure 4-12: Comparison of store aggregate and section aggregate random forests, gradient boosted trees, and XGBoost models for all three machine learning prediction tasks for a specific store



The figure is plotted against number of clusters in the x-axes for consistency with Figure 4-11, but neither multiple time series treatments require clustering. From the figure, we can see that the performance differences between the three tree models are not significant, an observation seen across all other stores examined. Because the particular training package used for XGBoost models has much faster runtime, it is adopted as the default tree model to examine in this project. Similar to linear regressions, the validation data set is used to tune the model hyperparameters, including the learning rate (the amount each incremental tree corrects regression results from the previous), number of boost stages (number of trees in series), max depth for each decision tree (max number of "if-else" statements for each branch of each tree), and the regularization coefficient (strength of regularization). XGBoost models have more

than four possible hyperparameters, but in the interest of computational resources, these four are selected as the most important ones.

Shapley values

XGBoost models, due to their complexity like many other machine learning models, are often difficult to interpret, yet statistical measures like P value are not applicable, because the models themselves are not linear and thus not suitable for traditional statistical techniques. To help us understand the factors that drive prediction for non-linear machine learning models, one commonly adopted method is the Shapley value. By treating the models as black boxes, Shapley values look at the model predictions with and without each parameter and derive a linear approximation for the impact of each parameter on the predictions.

Shapley values can be either negative or positive, with signs indicating the direction of impact. The mean absolute Shapley value therefore indicates the relative importance of each parameter, which would hopefully help us better understand the model mechanisms and select the most important parameters to prevent overfitting, like in the exercise with P values and multi-colinearity. Results for all three prediction tasks and six specific stores are presented in Figure 4-13, 4-14, and 4-15.

Figure 4-13: Shapley value for % of store inventory on display for six specific stores

Features with consistently high Shapley values are circled and selected as final explanatory variables in the model training. The features ranked with high importance are typically very similar across stores, making the selection task relatively straight-

Figure 4-14: Shapley value for number of unique articles in store for six specific stores

Figure 4-15: Shapley value for % of unique articles in store on display for six specific stores

forward. Historicals of the prediction value at time $t-1$ (and $t-2$ and $t-3$) and store stock at time t (and $t-1$) are some of most important features according to their Shapley values, which agree with the general intuition that more recent time series values are more important for predictions.

Comparison of various multiple time series treatments

Comparisons of various multiple time series treatments for XGBoost models, similar to those in Figure 4-9, 4-10, and 4-11, are presented in Figure 4-16, 4-17, and 4-18.

Figure 4-16: XGBoost test results for % of store inventory on display for 4 multiple time series treatments

Figure 4-17: XGBoost test results for number of unique store articles for 4 multiple time series treatments

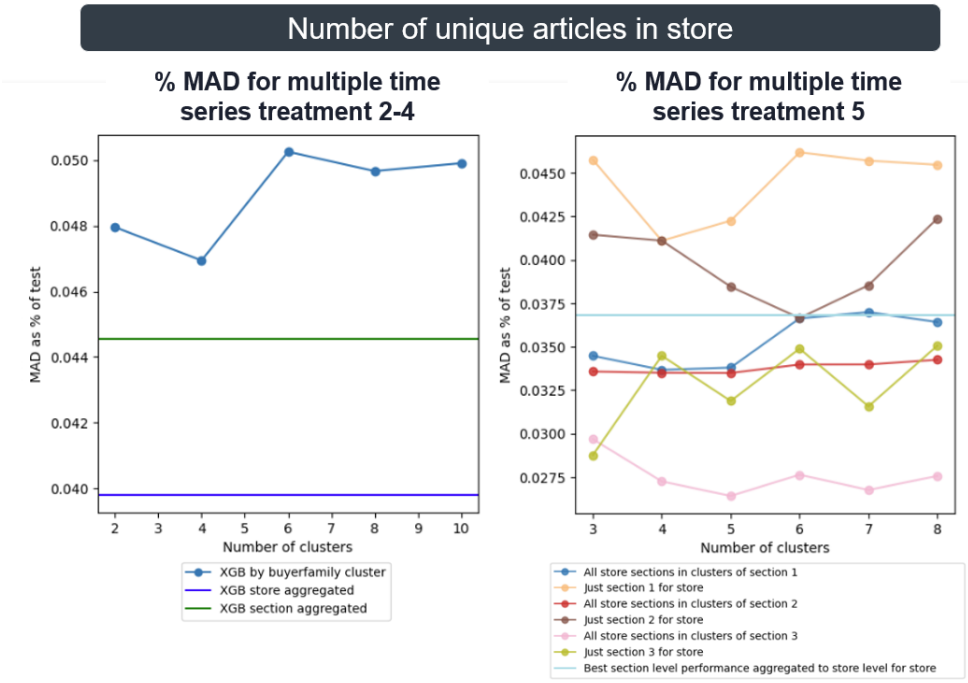
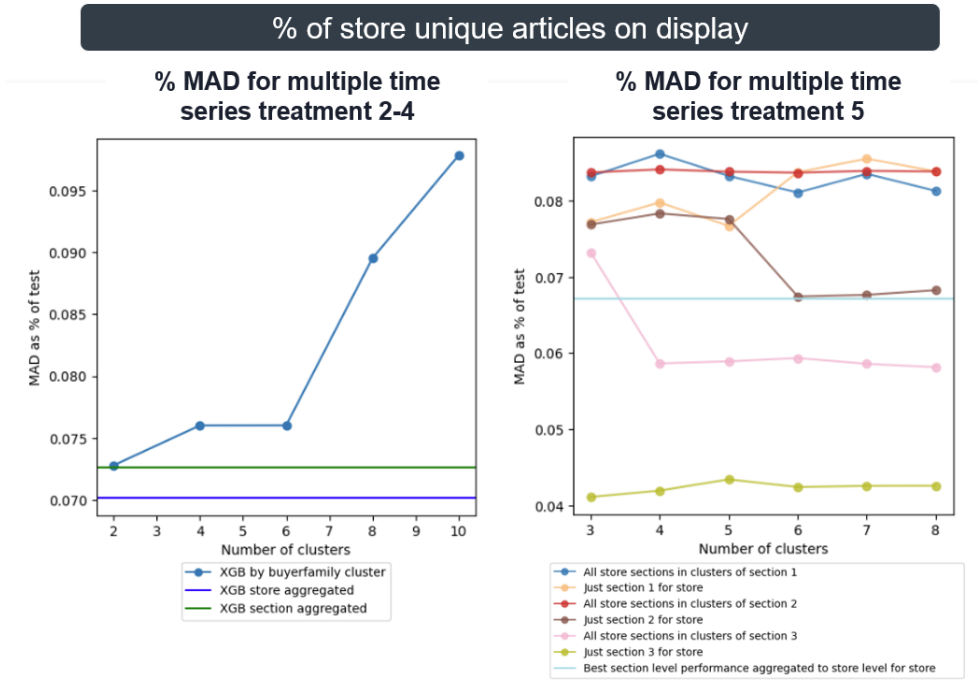


Figure 4-18: XGBoost test results for % of unique articles in store on display for 4 multiple time series treatments



Across all three prediction tasks as well as other stores, it is observed that XGBoost with store section clusters performs better than store aggregate, which is better than section aggregate. In other words, the model consistently performs better with a larger training data set, regardless of the prediction task or store. Intuitively, because XGBoost models are much more flexible than linear regressions and can learn non-linear patterns in the data, even if behaviors across buyer-family combinations and store sections don't demonstrate strong similarities, the models can still parse patterns more accurately in the presence of more data.

Comparing XGBoost model performance with store section clusters with linear regression performance, XGBoost tends to perform better, but is not guaranteed, as illustrated by the specific store examined in Figure 4-9, 4-10, 4-11, 4-16, 4-17, and 4-18. Predicting display inventory, and store and display unique articles, using historical time series, store inventory and store attributes, is not an easy task itself. Machine learning models are able to learn patterns, only if the right explanatory features are fed into the model. Therefore, even with more complicated tree models, better performance is not guaranteed compared to carefully constructed linear regressions for our purposes. The advantage of tree models is that performance can be consistently

improved with more data, and the same cannot be said about linear regressions. Therefore, considering this benefit, XGBoost models trained on store section cluster data are selected over XGBoost models with other multiple time series treatments and linear regression models as a short-list candidate for the final simulation.

4.5.3 Fully connected neural network models training and testing

As laid out in Table 4.1, the advantage of fully connected neural network models is that they can be trained to produce multiple outputs and optimize the predictions over all of them at the same time. The outputs leverage the same inputs, input layer operations and hidden layer operations, but receive their own output layer operations, which differentiate one output from another. In other words, for each buyer-family combination, 28 days worth of display inventory, store unique articles and display unique articles are generated based on 7 days of store inventory and historical time series of the prediction quantity.

The challenge of training FCNNs lies in the flexibility of the model. Generally speaking, FCNNs with a specified input and output size are defined by the learning rate (the numerical impact of each epoch of training on the model weights), the number of hidden layers, the size of hidden layers and the activation function for each layer in the network. Unlike regressions or tree models, there isn't always a simple understandings of how each hyperparameter impacts the final outcome of the prediction, nor is the tuning process necessarily rigorous. Based on trial and error, the following parameters are selected for each prediction task:

- For % of store inventory on display, learning rate of 0.000001, 2 hidden layer of 150 nodes, ReLU activation for the input and hidden layers and softmax activation for the output layer are chosen. ReLU activation tends to be the most versatile and suitable for many purposes, and softmax activation produces an output between 0 and 1, which matches the range of the prediction quantity.

- For number of unique articles in store, learning rate of 0.00001, 2 hidden layer of 200 nodes, ReLU activation for the input and hidden layers and no activation for the output layer are chosen. ReLU activation maps any negative input value to 0 and can lead to meaningless outputs for a subset of predictions.

- For % of store unique articles on display, learning rate of 0.00001, 2 hidden

layer of 200 nodes, ReLU activation for the input and hidden layers and sigmoid activation for the output layer are chosen. Like softmax, sigmoid also maps the input values to a number between 0 and 1, but follows a different shape. Sigmoid is found to outperform softmax in this application in practice and therefore selected.

Because FCNNs optimize the loss not over individual predictions but the entire time series, the prediction accuracy metrics are not comparable to linear regression or tree model metrics (MAD). Instead, their accuracies are compared after integration with the store inventory simulation in the following section.

4.6 Integrating machine learning models with store inventory simulation

4.6.1 The problem with standard tree models

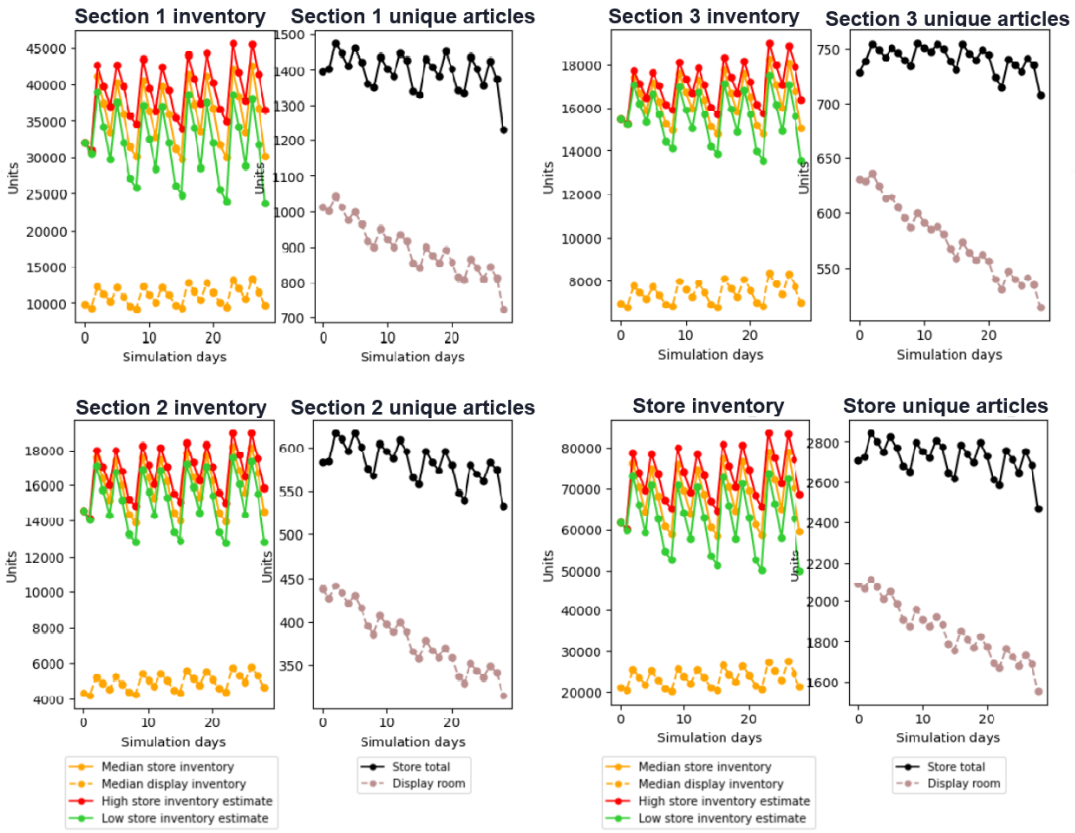
One of the drawbacks of standard tree models, as discussed in Section 4.3.4 and 4.3.5, is that they make time series predictions incrementally and are trained to optimize each prediction individually. This means that if each prediction has some systematic biases, longer term deviation is possible as the biases compound.

Unfortunately, this drawback has become a reality the three prediction tasks here. Figure 4-19 shows the display inventory, number of store and display unique articles for each section and store aggregate, under the scenario that store inventory is expected to remain flat from the Monte Carlo simulation. For a store with no inventory build-up or depletion, one would expect display inventory, store unique articles and display unique articles to remain relatively flat. Contrary to expectations, the XGBoost models result in slight increase in display inventory and significant decrease in store and display unique articles.

The decrease in store and display unique articles is likely driven by inventory behavior throughout each week. Figure 4-14 and 4-15 show that by far the most important driver for prediction at time t is the historical time series value at time $t-1$ and store inventory in general have relatively weak influence, especially for % of unique articles in store on display. In other words, what the model does for the most part is using historical data to make future predictions, more so than using store inventory as an important predictor. For unique articles in store and display, they tend to decrease

throughout the week as articles are sold out, except on Mondays and Thursdays when shipments of new articles are received. As a result, the data set has more entries where unique store inventory is lower at time t than at $t-1$. Even though day of week is a categorical explanatory variable used in the training process, the model does not seem to pick up on the correlation between day of week and increasing or decreasing number of unique articles. It is reasonable to suspect that this asymmetry in data has led to downward biases on the unique article predictions for the entire simulation time horizon.

Figure 4-19: Display inventory and number of store and display unique articles for a specific store assuming store inventory is flat over simulation period; predictions made using XGBoost models with standard architecture



4.6.2 Workaround for tree models

The downward biases of tree models with standard architecture mean that the simulation is not able to use these machine learning models as is. That said, the problem we are facing here is fundamental to tree models. Architecturally, they are not designed to accommodate multiple time series inputs even as scalars, so when asked to do so,

they are not able to take both of them into account with similar importance. Instead, they simply find the parameters that minimize the loss functions, even if it means to take most of the input from one time series and put a lot less emphasis on the other, which is what took places here.

Unfortunately, with the same model architecture, there is not a simple fix to the systematic biases. However, there is a practical workaround that is at least capable of producing intuitive results. The Achilles heel of tree models here lies within its inability to intelligently take multiple time series as inputs. Historical time series of the prediction value is provided to ensure temporal continuity. Meanwhile, store inventory is provided because it is the Monte Carlo store simulation output, which needs to have influence over prediction values. A natural next step is to observe the model behavior with store inventory as the only time series input, while temporal continuity can be enforced afterwards. This way, the model can focus on learning just the correlation between store inventory and the predicted quantities, which can be adjusted up and down afterwards to ensure agreement with historical time series.

Figure 4-20: Display inventory and number of store and display unique articles for a specific store assuming store inventory is flat over simulation period; predictions made using XGBoost models with standard architecture without historical time series of the prediction quantities

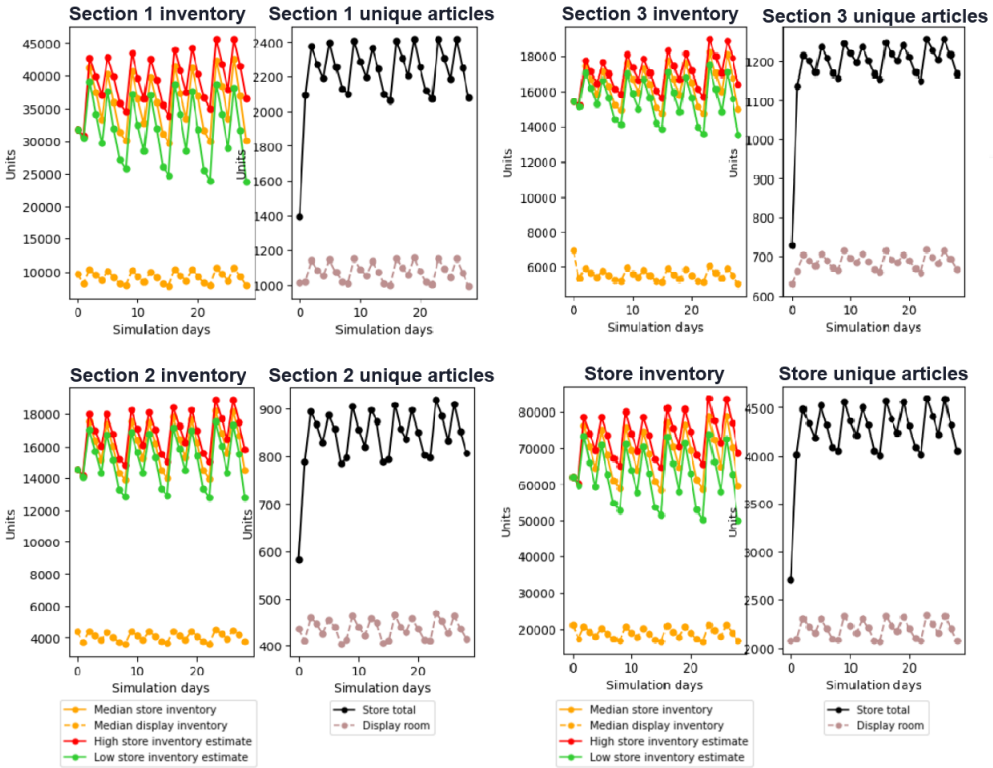
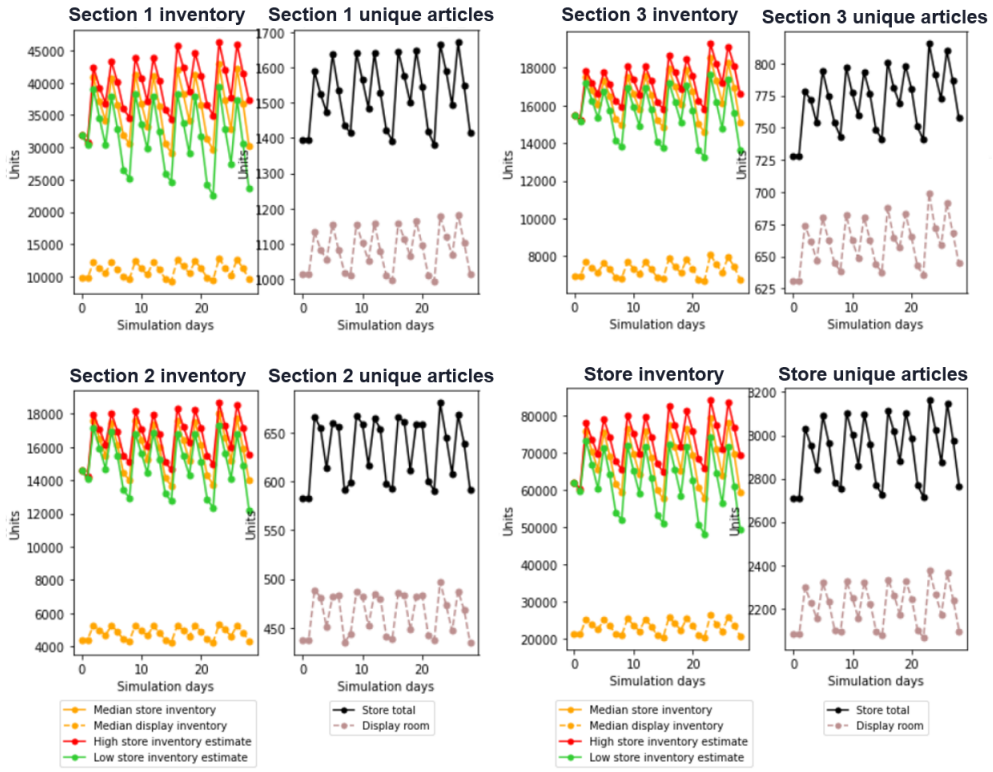


Figure 4-21: Display inventory and number of store and display unique articles for a specific store assuming store inventory is flat over simulation period; predictions made using XGBoost models with standard architecture without historical time series of the prediction quantities and scaled such that day 1 matches day 0



Therefore, the same XBGoost models are trained without historical time series of the prediction values, and the results are presented in Figure 4-20. As expected, the systematic biases no longer exists, but huge jumps in values between day 0 and day 1 are observed. Although not a rigorous approach rooted in machine learning theory, the temporal discontinuity can be addressed by simply scaling up or down the predicted values from day 1 to day 28 for each buyer-family combination by common factors, such that the values on day 1 is the same as day 0. By doing so, both objectives of maintaining temporal continuity and exploring the correlation with store inventory are achieved. The final results are shown in Figure 4-21. Lastly, the approach is applied to 6 stores during a historical time period and store level (aggregated across buyer-family combinations) accuracy metrics, consistent with those in the store inventory simulation in Chapter 3, are calculated and presented in Table 4.2. Compared to Table 3.1, the accuracy here is overall lower with more systematic biases, which is inevitable given the difficulty of the prediction tasks as well as the compounding of prediction errors due to the simulated store aggregate inventory being inputs to the machine learning

predictions.

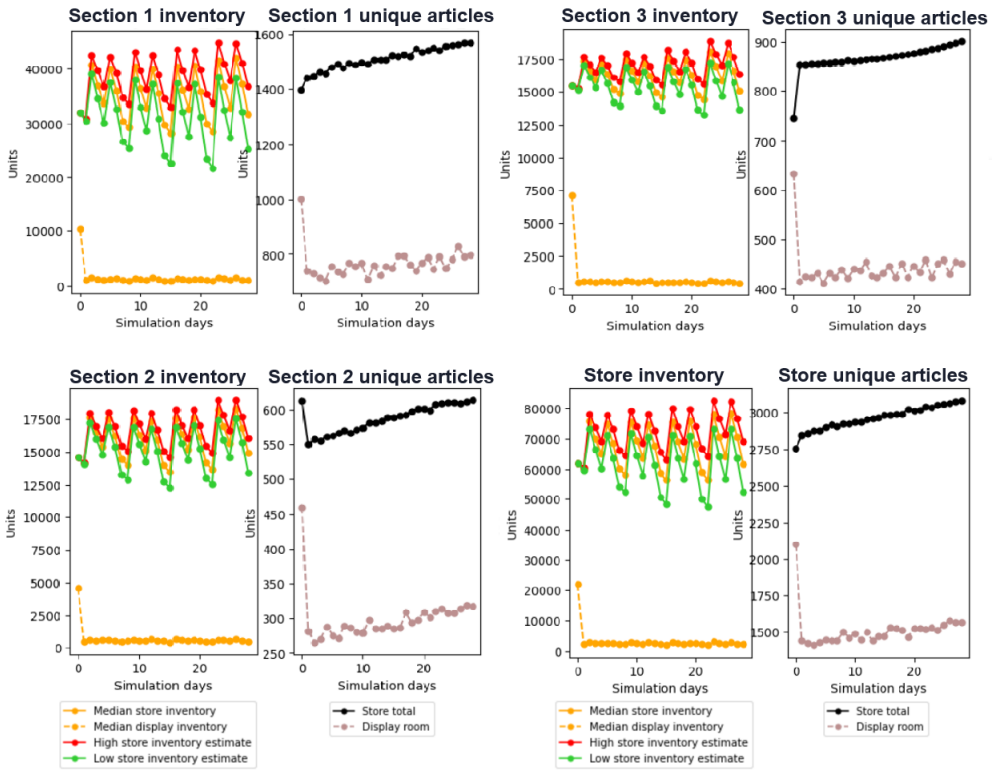
	Display inventory	Store unique articles	Display unique articles
Daily % MAD	3 to 11%	3 to 20%	3 to 10%
Daily % MD	-4 to 11%	-10 to 20%	-9 to -1%

Table 4.2: Ranges of display inventory and store and display unique articles accuracy metrics for the six specific stores examined

4.6.3 Performance of fully connected neural network models

The prediction results of the FCNNs for a specific store, trained on store section cluster data, are presented in Figure 4-22.

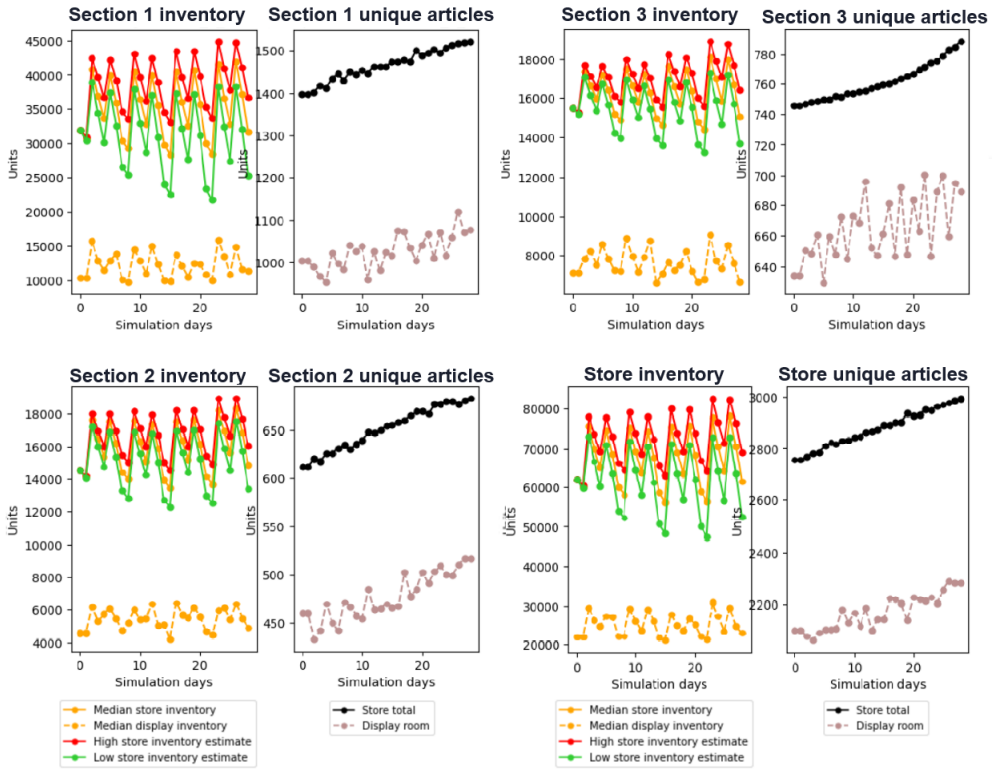
Figure 4-22: Display inventory and number of store and display unique articles for a specific store assuming store inventory is flat over simulation period; predictions made using FCNNs



Despite the model being able to generate predictions for the entire series and optimize over them at the same time, both temporal discontinuities and systematic biases can still be observed. As powerful as the model can be, the prediction task itself here has

proven to be challenging. For instance, predictions for display inventory in week 4 is generated based on store inventory and display inventory in week 0, meaning the model would have to make predictions almost an entire month out. Therefore, it is not surprising that the accuracy could be lacking. In addition, FCNNs have the same problem as other models when it comes to learning features simultaneously from two time series inputs. Both inputs are treated equally and the model is only incentivized to learn features that minimize training loss, which does not always translate to treating the two time series inputs with equal importance.

Figure 4-23: Display inventory and number of store and display unique articles for a specific store assuming store inventory is flat over simulation period; predictions made using FCNNs and scaled such that day 1 matches day 0



For a fair comparison with tree models, the same practical workaround to force temporal continuity as described in Section 4.6.2 is applied to the output of the neural network models and the results are shown in Figure 4-23. Despite store inventory being flat over the simulation horizon, store and display unique articles continue to increase, which is not an intuitive result. Therefore, it is fair to conclude that FCNNs are not suitable for the time series prediction tasks here.

Last but not least, it is important to mention that even if FCNNs are able to produce

sensible results, they can be computationally intensive to train. While each XGBoost model takes less than a minute to train, an FCNN for our prediction tasks can take up to 20 minutes. Given the need to train a separate FCNN for each section in a store and each prediction task, the run time can easily be multiple hours, which is not feasible for a real time simulation, unless the models are pre-trained and stored beforehand ready to be called upon.

4.6.4 Final model selection for store simulation

In Chapter 4, a number of different machine learning models and architectures are explored and contrasted. Models that make incremental predictions and optimize over them individually do not perform well for our purposes as systematic biases can lead to significant errors further into the prediction time horizon. On the other hand, models that are able to make many predictions and optimize over all of them at once also do not perform well, likely due to the difficulty of the tasks and lack of explanatory features. Upon evaluating all the options available, XGBoost model, trained on store section cluster data without historical time series of the prediction value, manually scaled to maintain temporal continuity, is adopted. Despite lack of theoretical machine learning underpinnings, the approach produces consistently intuitive results with slight systematic biases across stores and sections, at a speed reasonable for the simulation purposes, and is hence selected for the store simulation model.

Chapter 5

Dashboard visualization and conclusions

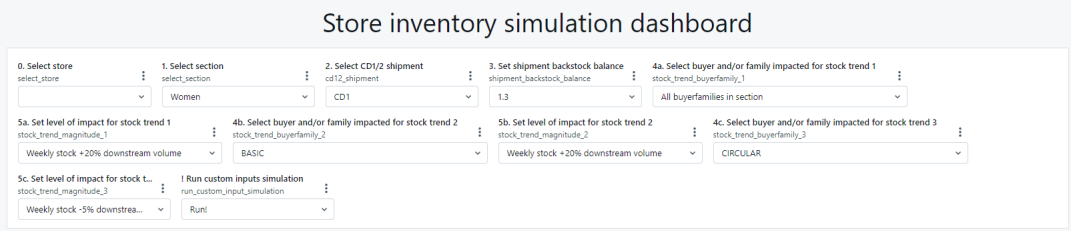
5.1 Dashboard visualization

With the Monte Carlo and machine learning model mechanism and architecture finalized, the simulation is complete with features that the project has set out to achieve. Nonetheless, the simulation itself is not exactly user-friendly, both its interface and its outputs. Therefore, to adapt the simulation to the business context, a dashboard is designed and built atop the models for better usability.

The design of the dashboard is based on a store by store comparison of a baseline scenario and a custom scenario that the user specifies. The baseline scenario uses a pre-specified set of inputs, including no inventory build-up or depletion trends and the status quo of CD1/2 connection. In contrast, the custom scenario allows three custom inventory trends (additive) for select buyer-family combinations or all of them in the selected section, CD1/2 connection, and shipment and backstock balance. Inventory trends only impact one section at a time, because the way business units are structured within Zara means that each team is only focused on one single section and not the other two. Additionally, CD1/2 selection not only affects day of week shipment volume, but it also impacts the machine learning model. If a store section has only been connected to a CD1 and we are interested in its behavior with CD2 replenishment, the machine learning models are trained using the closest store connected to a CD2 in terms of store section clustering distance (metrics laid out in Section 4.3). The layout

of the input interface is presented in Figure 5-1.

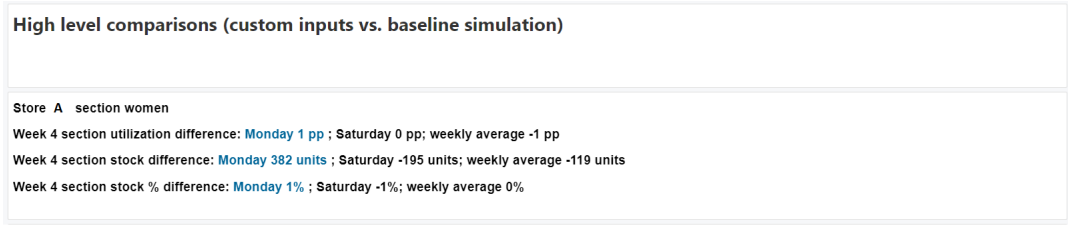
Figure 5-1: Dashboard input interface



Once the inputs are specific, the differences of the baseline and custom scenarios of the selected store section are calculated and first displayed as high level summaries. Because the simulation is not perfect and may be biased in some areas, the differences between two scenario are likely to be even more accurate, as the biases could cancel each other out. The differences are also easy to interpret, as they directly result from the user specified inputs in the custom scenario.

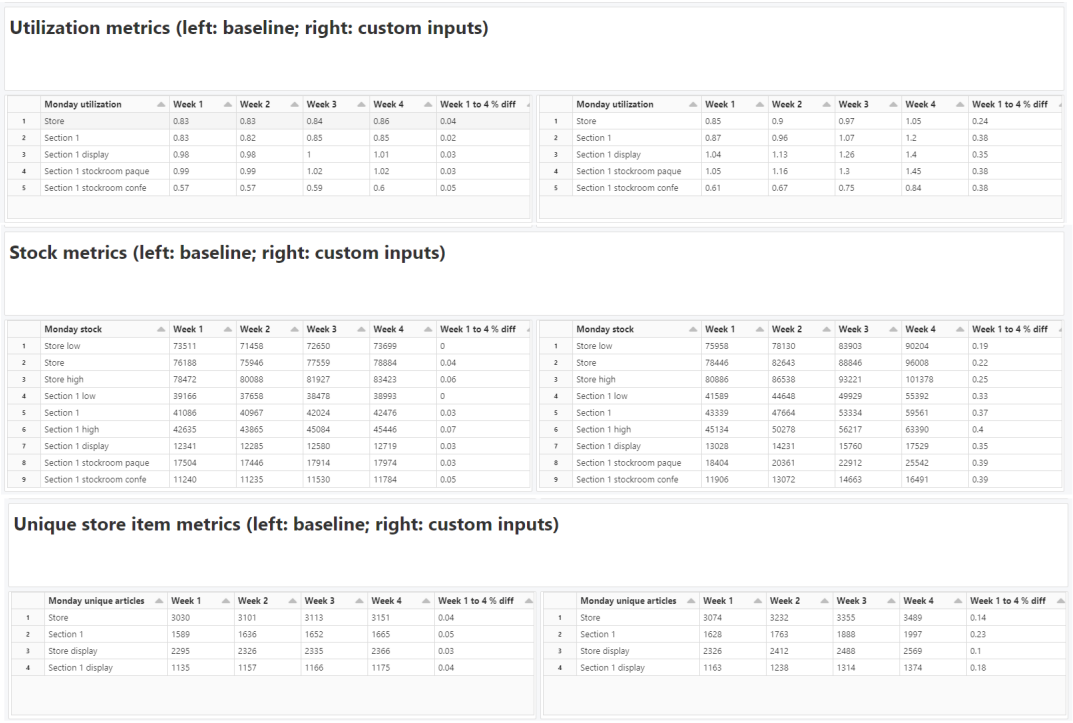
For high level summary, section utilization (% of section capacity), inventory differences (number of units) and % inventory differences between the baseline and custom scenarios are displayed, for Mondays and Saturdays end of day and weekly average. Mondays and Saturdays are specifically chosen, because they are typically the highest and lowest points of inventory in each week due to new article shipment schedules and demand profiles. A screenshot of the summary view is shown in Figure 5-2.

Figure 5-2: Summary view of the dashboard



In addition to the summary view, specific tables are provided in the detailed view for quantities in the summary view for both the baseline and custom scenarios. Select tables in the detailed view are shown in Figure 5-3.

Figure 5-3: Detailed view of the dashboard



5.2 Project conclusions

This project sets out to build a store simulation tool that provides a 4-week forward-looking view of store inventory and assortment complexity, for both a daily and a twice a week store shipment model. It first uses the Monte Carlo method to simulate store inventory by buyer-family, based on demand forecasts, quantified demand stochasticity, and upstream inputs. It then takes the store inventory, along with historical time series data and store attributes data, and predicts display inventory, and number of unique articles in store and display room.

To evaluate model performance, the simulation is performed on historical periods to calculate accuracy metrics that are custom defined specifically for the project. We see that the simulation can achieve a daily mean absolute deviation of 2-4% for store aggregate inventory, 2-8% for section inventory and 10-15% for median buyer-family combination, all with little systematic biases. The prediction accuracy for display inventory and assortment complexity is lower, due to the nature of the prediction tasks. For store total (aggregated across buyer-family combinations), MAD of 3-10% can be expected in general, but can be as high as 20% for some stores, with potential systematic biases depending on the store.

In the process of building the store aggregate inventory simulation, the project examined various methods to model inventory trends, developed a method to quantify daily demand stochasticity, and explored possibilities to control the stochasticity further into the simulation time horizon.

- The project examined possibilities to extract inventory trends using historical data and linear regressions. However, the attempt has proven to be difficult, as inventory trends on buyer-family levels for the business tend to be noisy and not precisely replicated from year to year.

- The project looked at how sales and returns vary from week to week and between days of the week across buyer-family combinations, and was able to calculate the variance and covariance of the expected sales and returns. By sampling from these variance-covariance matrices, the simulation is able to quantitatively model demand stochasticity and the results have been shown to accurately represent historical sales and returns.

- The project also explored options to prevent the simulations to continuously diverge from the median further into the simulation time horizon due to the modeled demand stochasticity. Practical inventory minimums and maximums are calculated from historical data, and are enforced in the simulation with considerations to avoid systematic biases. Additionally, various shipment adjustments mechanisms that react to the specific demand stochasticity in each Monte Carlo trial are tested, including both inputs manually specified and calculated from historical data. Calculated inputs do not contribute to higher model accuracies and thus manual inputs are adopted in the simulation.

In the process of building machine learning models to predict display inventory and product portfolio complexity, the project systematically examined the efficacy of various machine learning models and architectures at making time series predictions using two time series as inputs.

- Tree models perform better when the models are trained on more time series data across different categories (stores and buyer-family combinations). The same conclusion is not consistently observed for linear regressions, presumably due to them being less flexible.

- Random forest, gradient boost trees and XGBoost do not have any material

performance differences and tend to outperform linear regressions, although not always guaranteed. Because limited explanatory features are available due to the nature of forward-looking simulations, the differentiation between the tree and linear regression models is not so significant.

- Time series models where predictions are made incrementally and the loss functions are minimized over each individual prediction can lead to considerable systematic biases, which accumulate and result in large inaccuracies further into the prediction horizon.

- Making time series predictions with two different time series of equal significance as inputs simultaneously is difficult for linear regressions, tree models and FCNNs. Because of the black box nature of machine learning models, it is difficult to control how the models will learn the features from the two time series inputs. As a result, the models tend favor one time series over the other, since the training objective is simply to minimize loss of the training data. Therefore, it can be more advantageous to use the time series inputs incrementally in two steps, even if it means to deviate from rigorous machine learning techniques.

5.3 Potential for future work

From a usability perspective, in order for the tool to be put into production and fully deployed as a standard tool for the business, a number of hurdles need to be overcome.

- The tool currently sits in a data platform that is typically only accessed by the technology and business analytics teams. For store operations managers that are potential users, it can difficult for them to use the tool logistically or understand the coding interface.

- Additionally, the tool can only be used one store at a time only for stores with calculated parameters (e.g. variance, covariance), and the data pipeline to calculate these parameters requires multiple hours per store. For store operations managers that oversee not a single store but a large number of them, the tool can only provide a narrow snapshot of select stores within reasonable time.

Therefore, for the tool to be usable in the day to day work store operations managers, the data preparation pipeline needs to be streamlined and the final tool needs to be

built with a more accessible software front end.

From a modeling perspective, three options come to mind as the most meaningful improvements.

- First of all, downstream forecasts are currently provided on section levels and allocated to buyer-family combinations based on historical proportions. If the forecasts are available on more granular levels, buyer-family level accuracy could be significantly improved, and as a result, section and store aggregate accuracy.

- Furthermore, inventory build-up and depletion trends are currently only modeled linearly. In reality, inventory trends, especially on buyer-family combination levels, can demonstrate much more complicated behaviors. Therefore, more sophisticated inventory trend mechanisms, informed by historical inventory behaviors, could meaningfully improve model fidelity.

- Third, recurrent neural network models as a potential candidate to predict display inventory and number of unique items in store and display are not explored in this project as a potential candidate, due to the difficulty presented to accommodate multiple time series as inputs. Nonetheless, the model does have its advantages for time series predictions. If the model can be customized to fit our prediction needs, improvements to prediction accuracy are certainly possible.

With all the possibilities in mind, it is also important to remember that the room to improve model fidelity is endless, and the model after all is only an approximation. Any improvement to model must also be examined practically in conjunction with model complexity and usability, especially when the end users may not have the same level of technically knowledge as the developers.

Bibliography

[1] Nuridawati Baharom and Pa'ezah Hamzah. "Inventory Optimization Using Simulation Approach". In: *Journal of Computing Research Innovation* (2018).

[2] R. Brits and J. Bekker. "A Multi-objective Coal Inventory Management Model Using Monte Carlo Computer Simulation". In: *South African Journal of Industrial Engineering* (2016).

[3] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[4] Scott Douglas Foster. "Fulfillment Algorithm for Integrating Stock between Brick and Mortar and E-commerce". Master's Thesis. Massachusetts Institute of Technology, 2018.

[5] Lila Fridley. "Improving Online Demand Forecast using Novel Features in Website Data: A Case Study at Zara". Master's Thesis. Massachusetts Institute of Technology, 2018.

[6] *Inditex finance*. URL: www.inditex.com/itxcomweb/en/investors/finance.

[7] *Inditex history*. URL: www.inditex.com/itxcomweb/en/group/history.

[8] Joos Korstanje. *Advanced forecasting with Python*. Apress, 2021.

[9] Helmut Luetkepohl. *Introduction to Multiple Time Series Analysis*. Springer, 1991.

[10] Helmut Luetkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

[11] Elizabeth A. Maharaj, Pierpaolo D'Urso, and Jorge Caiado. *Time Series Clustering and Classification*. CRC Press, 2019.

[12] A Mansur, FI Mar'ah, and P Amalia. "Platelet Inventory Management System Using Monte Carlo Simulation". In: IOP Conf. Series: Materials Science and Engineering. IOP, 2020. DOI: 10.1088/1757-899X/722/1/012004.

[13] Yuri Delano Regent Montororing and Murwan Widyantoro. "Model of Inventory Planning Using Monte Carlo Simulation in Retail Supermarket with Consider to Competitors and Stimulus Strategies". In: *Journal of Applied Engineering and Technological Science* (2022).

[14] Manuel Martinez Puppo. "Replenishment in an Integrated Stock World". Master's Thesis. Massachusetts Institute of Technology, 2019.

[15] Gianpaolo Luciano Rivera. "Data-driven clustering for new garment forecasting". Master's Thesis. Massachusetts Institute of Technology, 2023.

[16] Nick T. Thomopoulos. *Essentials of Monte Carlo Simulation.* Springer, 2013.

[17] Lampros Tsontzos. "Dynamic Algorithm for Target Inventory and the Impact on Replenishment Strategy". Master's Thesis. Massachusetts Institute of Technology, 2022.

[18] I Gede Agus Widyadana, Alan Darmasaputra Tanudireja, and Hui-Ming Teng. "Optimal Inventory Policy for Stochastic Demand Using Monte Carlo Simulation and Evolutionary Algorithm". In: A Missing Link on Entrepreneurship Education Curricula. JIRAE, 2017. DOI: 10.9744/JIRAE.2.1.8-11.

[19] Wayne A. Woodward, Bivin P. Sadler, and Stephen D. Robertson. *Time Series for Data Science.* CRC Press, 2022.