An Intelligent Fashion Replenishment System Based on Data Analytics and Expert Judgment

..

An intelligent fashion replenishment system based on data analytics and experts judgement

Retail stock allocation is crucial but challenging. The authors developed an innovative solution, successfully tested in the context of high-end fashion: collaboration between articial intelligence and human intuition. Each week, stores are assigned a budget based on current stock levels versus potential sales, and offered to "spend" this budget with an initial data-driven recommendation on which SKU/sizes order and release. Each store manager is then given a time window, so she can modify the proposal while respecting budget constraints; and finally the articial intelligence optimally allocates available stock to requests based on the expected likelihood of sale minus cost of logistics, subject to management-dened constraints. Our test showed how this system outperformed the control group of stores, relying on a traditional headoce-driven allocation without direct human input. The retailer boosted sales, demand cover, and stock rotation performance: an estimated 1M EUR margin/month positive impact. Moreover, the new system improved store managers morale through non-monetary incentive-driven empowerment.

Key words: Retail, artificial intelligence, constrained optimization, forecasting, dynamic markets, innovation, luxury, fashion

1. Introduction

Inventory allocation is crucial to retail, even more so to those verticals whose sales are difficult to accurately predict. Case in point is fashion retail: the SKU lifecycle is short, residual end-of-life values are either very low or zero, SKU performance is volatile, and the unit values at stake are high. Inventory allocation has to match finite stock resources with unknown demands, to optimally accelerate rotation, avoid missed sales opportunities, and therefore ultimately increase profits. The topic is even more challenging in the case of high-end brands, due to the concetration of value in a small amount of pieces.

Inventory decisions influence sales in many ways: different assortments can be assigned to different stores, due to local trends and to the store dimension; the allocation on sizes can be fundamental for customer satisfaction. At the same time, the retailer has to avoid missing sales coming from stock-out as possible. The fragmented and highly local item/size nature of the demand made it difficult to accurately predict sales, creating challenges to centrally-driven stock allocation process.

A uniform stock allocation across stores and seasons can be perceived by the management as sub-optimal; however, a more differentiated allocation needs to catch each store peculiarities and to guarantee at the same time the optimal performance of the whole network.

The purpose of this work is to show the implementation and results of an original intelligent system able to improve sales by reducing stock-outs and missed opportunities, thanks to improvements in forecast accuracy, even at a high granular level such as store/SKU/size. This approach can leverage the store managers' insight and experience, by directly feeding their input into a new allocation process. Additionally, the proposed system enables the direct trans-shipment across stores, to correct any errors in the initial stock allocation, and therefore increase the liquidity of the store network.

With the new approach, inventory allocation becomes part of a retail-integrated management process, from merchandise planning and open-to-buy management, to in-season sales forecasting, to the design and measurement of pricing and promotion initiatives. This approach achieved multiple ambitious goals: reduce unsold stock, increase rotation, empower store managers, increase relevance to local consumer demand, and increase profits.

This article is structured as follows: Section 2 presents the literature review. The data sets used are detailed in Section 3; Section 4 presents methodology and process macro-steps: forecast, proposal, internal marketplace, allocation, and shipment optimization. Section 5 reviews the main results and, finally, Section 6 provides ideas for further potential improvements.

2. Literature review

Multiple authors covered the various components of fashion inventory management: how to buy inventory before the season starts, how to allocate stock to stores, how to update the merchandise plan during the season.

A dynamic programming approach is often used to solve such problems: [1] formulate a dynamic stochastic optimization model to determine total order size and optimal inventory allocation across non-identical stores for each period. They use Bayesian inference to model partially correlated demands across stores and time periods, and then dynamic programming to find an optimal allocation. [6] apply linear programming to estimate the season forecast for the chain by testing a subset of the chain. However, these approaches do not factor in any human input.

Macy's reported (WSJ, 12th August 2010) that considering stores as homogeneous is suboptimal: differences in store demands create opportunities to increase relevance and therefore ultimately grow profits.

[13] highlights the diversity of demand across stores and provides a framework to apply *micromerchandising*, a practice followed by a significant number of retailers: each store has a unique assortment, that maximizes its consumer appeal. The author attempts to infer store managers' behavior

in order to adapt the automatic replenishments. However, in practice it is operationally challenging to manage a dynamic programming model across multiple time periods and multiple stores.

Attempts at developing a traditional store-level forecast are generelly hinded by the idiosyncracies of the fashion industry, that make it complex and characterized by unsatisfactory results. [11] shows how fashion sales are highly volatile and fragmented, due to multiple intrinsically noisy and volatile factors: seasonality, fashion trends, promotions, and others.

The allocation system is often crucial in the relation between the company and the single stores, and any changes in the replenishment process reflects the company thinking. There are two common alternatives to manage the store replenishment: some models centralize the shipment decisions to achieve the best result for the entire network of stores, other models make shipment decisions based primarily on stores'input. Some company experimented both automated as well as non-automated allocation systems. In the non-automated system, store managers are allowed to freely request inventory, and central oversight is limited to applying simple stock availability constraints and manually prioritizing requests. This system is relatively manual, time-consuming and sometimes not yielding the desired sales performance. Moreover, store managers often attempt to order more than necessary in order to receive a number of pieces that is close to what they truly want (as it is pointed out in [3] for a similar framework). When dealing with scarse resource in a competitive environment, as pointed out in [7], an optimal (from the retailer's point of view) allocation can be reached giving the stores truth-inducing mechanisms. On the other side, fully-automated system managed centrally can lead smaller stores to consistently carry a limited range, while larger stores systematically received the best picks from the available inventory.

Multiple authors explored the possibility to combine an automatic store forecast with human expert judgement to leverage the impartiality of the former with the expertise of the latter. Among these, [2] concluded that the combined forecast generally outperforms automatic-only and expert-only forecast. In the case of high-end fashion, the strong influence of local weather and consumer preferences on sales, combined with relatively deep knowledge of individual customers that store managers have, led us to choose them as the key providers of human judgement.

[4] implemented a similar model for Zara that combines a forecast with store managers' demands. The core model is stochastic and predicts SKU-store level sales during a replenishment period as a function of: expected demand, available inventory, and store stockout policy. They then formulate a mixed-integer program that embeds a piecewise-linear approximation of the first model applied to every store in the network; the output store shipment quantities maximize overall predicted sales subject to inventory availability and other constraints. The model is based on weekly forecasts, best suited to fast-moving items (such as those of Zara), and is optimal for the particular merchandising policy in which SKUs with size-level stockouts are removed from the shelves. Human judgement is

applied by multiple experts working in a competitive environment where total stock is constrained. Store managers would tend to over-order if no balancing mechanism was provided, so Caro and Gallien introduced weights for the automatic forecast and store managers. Separately, the same authors [8] provided an implementation of an automatic allocation algorithm of initial shipments from the central warehouse, based on forecast updating and dynamic optimization.

We propose a replenishment model combining automatic forecast and stores' input. In the high-end fashion segment, multiple differences make the mentioned approach of [4] less directly appliable: a greater emphasis on increasing stock rotation and reducing slow-moving items to avoid a very significant cash depreciation; and the high item value makes trans-shipment across stores economically feasible. Therefore we adapted Caro and Gallien's model, allowing for stock releases (negative shipments to a store) and creating an internal cross-store marketplace that allows store managers to trade their stock. We do not approximate the optimization model; thus, we obtain an integer programming problem that is easier and faster to solve.

Additionally, we introduce a store budget to address the different levels of stock and sales potential across the stores. In case the store manager was not able to provide his forecast, we provide an initial proposal that approximaties human behavior and tries to capture the local tendencies.

Creating a sort of internal marketplace among stores, the proposed system allows to differentiate each store's offer, according to the real local demand. Finally, the system allows to implement a defragmentation mechanism to minimize store-level stockouts, by optimally matching missing sizes to over-stocked sizes in other stores, and attempting at completing the SKU-size plans of each store to the greatest feasible extent through an extension to our approach.

3. Methodology and implementation

The whole process consists of 4 main steps (Fig.1) as follows:

1. Headoffice-driven *replenishment suggestion* and *proposal to the stores*. SKU/size/store allocated quantity, calculated based on past sales, category seasonality and size-level allocation. The output is a list of item/sizes to order and release. This list tries to approximate the store manager's expected contribution to the forecast in case he does not edit it. Moreover, each store is assigned a budget based on the current balance between current stock and expected (potential) future sales. Suggested SKU/size orders consumes the budget, releases free it up.

2. Store-driven *internal marketplace*. During a given time window, store managers can modify the proposed non-mandatory suggestions freely, while always operating within the budget constraints (e. g. quantities of new items that can be ordered are capped by the budget, but additional budget can be freed up by releasing items currently in stock).

3. Headoffice-driven *optimal allocation*. The expected profit is maximized through the optimization of stock movements. The problem formulation contemplates expected demand, warehouse availability, logistic costs and stock availability constraints. The output of the optimization includes both deliveries from the central warehouses as well as direct trans-shipments across stores.

4. *Logistics*. Finally, orders and releases are optimally matched (headoffice-driven, minimizing shipments) and the parcels are shipped.

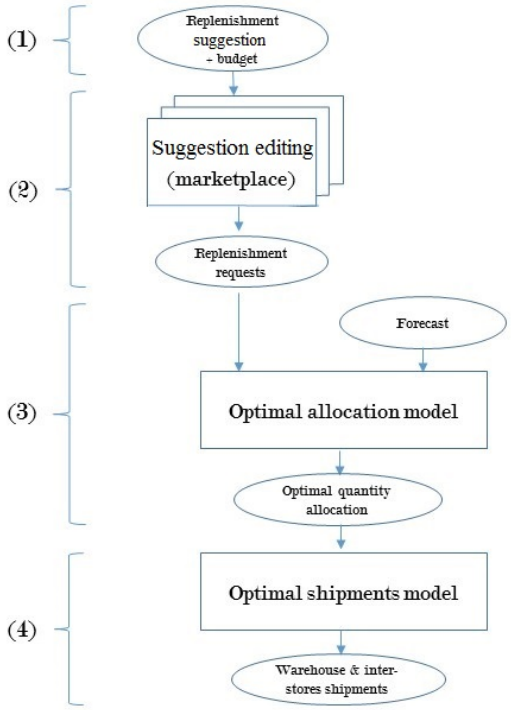

Figure 1 **Process chart.**

3.1. Notation

The notation used hereafter is as follows:

- J is the set of stores, and j is a specific store
- I is the set of items, i is a specific item, and I_C is the set of items of a specific category C;
- S is the set of sizes, and s is a specific size; d is a specific sizing system (eg. EU or US system), and S_d is the subset of sizes of sizing system d;
- w is a generic week, and $\tilde{w}$ is the current week; the set of weeks is denoted as W;
- P_{ij}^w is the full price for item i in store j at week w, $\tilde{P}_{ij}^w$ is the pocket price paid;
- E_{ijs}^w the quantity in stock for item i and size s at the store j at the end of week w;

- H_{is}^w is the quantity available in the warehouse for item i and size s at the end of week w;

- n_f is the number of future weeks for which the replenishment is planned , and n_p is the number of past weeks to use as a reference for a first replenishment suggestion;

- Y_{ijs}^w is the quantity of item i and size s sold in store j at week w;

- $\tilde{Y}_{ijs}^*$ is the sales forecast (n_f weeks) for item i and size s in store j at week w

- the omission of indexes i or j or s indicates a sum (eg Y_j^w is the quantity sold in store j at week w, all sizes and items)

Available data includes point of sales (POS) transactions and their related master data: item categories, unit costs, store coordinates, name and type, daily store traffic, stock levels in the central warehouse and at each store.

Historical data is netted of returns and sales aggregated at different levels (e.g. category, store, and SKU level), calculating total quantity sold, average price, average markdown, total traffic, and total tickets, at each level of aggregation.

3.1.1. Replenishment Suggestion The replenishment suggestion together with the budget serves as a starting point for store managers and also provides an automated contribution in case the store managers decide not to modify the suggestions (or are not able to). Hence the rationale is to imitate the reasoning that could lead an average (imaginary) store manager to a request of items to be replenished. A roughly estimated sale potential by SKU/store is calculated multiplying the past weeks sales by a future seasonality coefficient.

The number of future weeks for which the replenishment is planned n_f can vary in time and depend on the retailer's supply chain policy. It can be low (even 1 or 2 weeks) for fast rotation retailers, higher (eg. 6-8 weeks) for low rotation or high-end retailers. The number of past weeks to use as a reference for a first replenishment suggestion n_p has to be fixed in order to account for recent item mix and seasonality and to guarantee at the same time a reliable baseline for future forecast.

3.1.2. SKU-store level allocation At the current week $\tilde{w}$, the total sales of item i in store j for the last n_p past weeks is given as

$$Y_{ij}^* = \sum_{w=\tilde{w}-4}^{\tilde{w}-1} \sum_{s \in S} Y_{ijs}^w.$$

The seasonality coefficient is calculated using weekly seasonalities z_C^w for each item category C (see Appendix A for the details). The estimated sales potential at the current week $\tilde{w}$ $\tilde{Y}_{ij}^*$ for the next $n_f - 1$ following weeks is then calculated as

$$\tilde{Y}_{ij}^* = Y_{ij}^* \frac{\sum_{k=0}^{n_f} z_C^{\tilde{w}+k}}{\sum_{k=1}^{n_p} z_C^{\tilde{w}-k}}.$$

3.1.3. Size-level allocation As we need size-level quantities, $\tilde{Y}^*_{ij}$ is allocated by size following the method described in Section 3.1.3 , yielding $\tilde{Y}^*_{ijs}$. Item size-level allocation is typically performed based on the assumption of historical consistency and homogeneous store distribution of sales by size ([12, p. 87]). The proposed systems aims to realize a more flexible allocation of sizes, adapted to the store characteristics and to the distribution of sales by size for a particular item. This aspect is often neglected, due to the complexity introduced by multiple sizing systems and to the high fragmentation of the information.

In order to distribute the SKU/store quantity forecast by size, we compute the relative frequency of a certain size s for a certain item and for a certain store. The relative frequency of sales of size s for item i is denoted by $p_i(s)$ and is defined as total sales of item i and size s versus the total sales of item i in the whole store network:

$$p_i(s) = \frac{\sum_{w\in W}\sum_{j\in J} Y^w_{ijs}}{\sum_{w\in W}\sum_{j\in J}\sum_{s\in S} Y^w_{ijs}} \quad \forall s \in S.$$

In order to define the relative frequency of size s for each store p_j, we first need to consider the sizing system of size s, denoted by d_s. Then, we compute for each store j the ratio between sales of size s and total sales of all sizes of the same sizing system d_s:

$$p_j(s) = \frac{\sum_{w\in W}\sum_{i\in I} Y^w_{ijs}}{\sum_{w\in W}\sum_{i\in I}\sum_{s\in S_{d_s}} Y^w_{ijs}} \quad \forall s \in S.$$

Notice that using the frequency across sizes of sales of item i in store j would not be accurate, as volumes can be low.

Figure 2 shows example distribution of sizes for different stores and items.

Both factors are equally weighted, in order to take into account both the item and the store characteristics separately:

$$p_{ij} = 0.5 \cdot p_i + 0.5 p_j. \tag{1}$$

To forecast future sales $\tilde{Y}^*_{ijs}$ for each item and size, we generate a random sample according to the distribution p_{ij} of sizes, to match the total forecast $\tilde{Y}^*_{ij}$.

The initial proposal dramatically reduces the workload for store managers, thanks to automated allocations to sizes and balancing of demand across items.

3.1.4. Proposal to the stores The store-level budget is a simple measure of the gap between the current stock levels and the future expected sales potential, it gives store managers a transparent, non-monetary incentive to release non-performing stock so they can order new items; it also increases empowerment and direct ownership of stock allocation decisions: all stock received will always have been requested, even if not all the requests can be fulfilled due to system constraints.

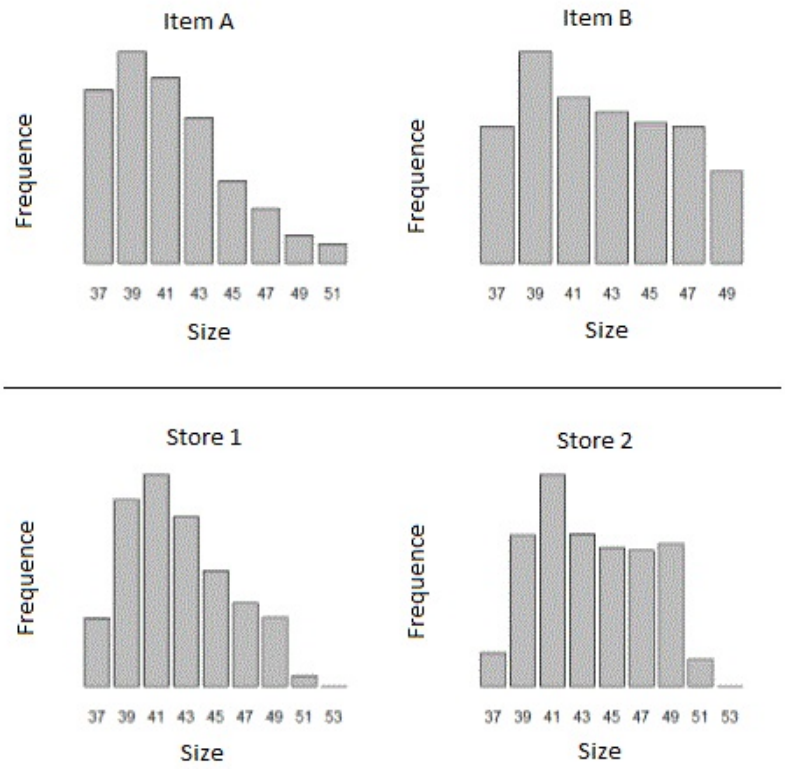

Figure 2 **Frequencies of pieces sold by size, for two items (all stores) and for two stores (all items)**

For each store j, the budget b_j at current week $\tilde{w}$ is calculated as the sales expected overall potential for future n_f weeks plus a safety extra coverage of 20%, minus the total potential value of current stock:

$$b_j = (1.2) \cdot \sum_i \tilde{Y}^*_{ij} P^{\tilde{w}}_{ij} - \sum_i \left(\sum_{s \in S_i} E^{\tilde{w}-1}_{ijs} \right) P^{\tilde{w}}_{ij},$$

where $P^{\tilde{w}}_{ij}$ is the selling price of item i in store j and $E^{\tilde{w}}_{ijs}$ is the current stock of item i, size s in store j. Then $\tilde{b}_j = b_j - \frac{\sum_j b_j}{|J|}$, calibrating all budgets so that the median across stores is zero.

If the budget is positive, then the store is *relatively understocked* and therefore needs additional stock. Otherwise, the store is *relatively overstocked* and should therefore release stock; Table 1 shows a simple illustrative example.

	Item 1	Item 2	Item 3
Selling price $P^{\tilde{w}}_{ij}$	100	200	150
Pieces forecast $\tilde{Y}^*_{ij}$	5	3	3
Stock E^w_{ij}	2	4	2

Expected revenues= $5 \times 100 + 3 \times 200 + 3 \times 150 = 1550$
Stock potential value= $2 \times 100 + 4 \times 200 + 2 \times 150 = 1300$
Extra-coverage= $20\% \times 1550 = 310$
Budget= $1550 - 1300 + 310 = +560$

Table 1 Example of store budget definition for store j with 3 items

Δ_{ijs}% quartile	$\Delta_{ijs} > 0$ (high stock)	$\Delta_{ijs} < 0$ (low stock)
1 or 2 (low)	No action	No action
3 (medium)	Recommended release	Replenishment
4 (high)	Mandatory release	Urgent replenishment

Table 2 Recommended actions based on quartiles

The budget is a measure of the store stocking level. However, it does not indicate how the stock level is distributed over the SKUs and sizes. Thus, we call Δ_{ijs} the difference between stock and potential sales at store/SKU/size level:

$$\Delta_{ijs} = E_{ijs}^{\tilde{w}-1} - \tilde{Y}_{ijs}^{*}.$$

we rank orders and releases based on their relative priority, through the ratio:

$$\Delta_{ijs}^{\%} = \frac{|\Delta_{ijs}|}{E_{ijs}^{\tilde{w}-1} + \tilde{Y}_{ijs}^{*}}.$$

The recommendation for each item is based on its Δ_{ijs} and $\Delta_{ijs}^{\%}$ quartiles for each store:

- if $\Delta\%$ is low, stocking of item i, size j in store s can be considered sufficient and no action is needed

- is $\Delta\%$ is very high, the stock is not balanced with potential sales; a urgent replenishment (if $\Delta < 0$) or forcing the release of the item (if $\Delta > 0$) is needed

- for mid levels of $\Delta\%$, stock and potential sales are quite unbalanced; a replenishment (if $\Delta < 0$) or a release of the item (if $\Delta > 0$) is suggested

A summary of recommended actions is given in Table 2, while an example of item-level analysis is in Table 3.

3.2. Extra-features of the proposal

If all releases and replenishment would be actually done, the stock level would be balanced across the whole store network. However, after considering recommended actions, some further adjustments to the *proposal* can be made, in order to account for additional business requests:

- store demands can be capped at the total quantity available from warehouse inventory and other stores (releases)

Store	Item	Stock	Potential	Delta	Ratio	Quartile	Action
Store 1	Item 1	10	1	9	9/11= 0.82	4	Mandatory release
Store 1	Item 2	1	10	-9	9/11= 0.82	4	Urgent replenishment
Store 2	Item 1	3	1	2	2/4= 0.5	3	Recommended release
Store 2	Item 2	6	5	1	1/11= 0.09	1	No action
Store 3	Item 1	7	8	-1	1/15= 0.07	1	No action
Store 3	Item 2	2	5	-3	3/7= 0.43	2	Replenishment

Table 3 Example recommended actions

- items introduced for the first time in the last (eg. two) weeks can be excluded from release recommendations

- a minimum quantity (eg. one piece) for each item/size can be left in each store

- orders for sizes that are currently out of stock can be prioritized

The proposed system can be also used to facilitate logistic and merchandising activies by accounting for orders and requests that are due to supplementary criteria, not directly related to demand and stock level. For example, it is possible to use the system to reduce the fragmentation of the stock, ie to reduce the number of stores where stockout occurs for most of the sizes of a certain SKU.

In order to take into account this, fragmentation has to be definined measured at the store/SKU level. The definition takes into account the number of different sizes in which the SKU is sold. If the percentage of sizes of SKU i in store s that are not available is greater than a percentage threshold t_{f1}, item i is defined *fragmented at the store level* in store s. We define the variable φ_{is} as one if i is fragmented at the store level in s, otherwise zero. Then, for all SKUs i we define:

$$F_i = \{s \in S | \exists (i,s), E^{\tilde{w}}_{ijs} > 0, \varphi_{is} = 1\}$$

$$T_i = \{s \in S | \exists (i,s), E^{\tilde{w}}_{ijs} > 0\}$$

Finally, we compute for each SKU the ratio between the number of stores in which i is fragmented and the number of stores in which i is available, that is

$$R_i = \frac{|F_i|}{|T_i|}$$

And we define the SKU i *globally fragmented* if R_i is higher than a threshold t_{f2} (eg. 40%). It is possible to modify store requests in order to reduce the fragmentation in the whole retail network. One possibility is that stores with highest potential order one piece of each size of globally fragmented SKUs, and stores with lowest potential release the same SKUs.

SKU/Size	Pcs Sold	Stock	Sugg Action	Budget (sugg)	Req Action	Budget (req)
Item1/37	0	2	Rep 1pc	+129	Rep 2pc	+258
Item1/41	2	3	Rep 1pc	+129	Rep 2pcs	+258
Item1/43	0	2	MRel 1pc	+129	MRel 1pc	+129
Item1/47	0	2	MRel 1pc	+129	MRel 1pc	+129
Item1/49	1	1	Order 1pc	-129	Order 1pc	-129
Item1/51	0	0	-	-	Order 1pc	-129
Item1/53	0	0	-	-	Order 1pc	-129
Item2/37	2	2	Rep 1pc	+99	Rep 1pc	+99
Item2/39	2	2	Rep 1pc	+99	Rep 1pc	+99

Table 4 Extract of proposal and edited proposal for a store (example).

3.3. Internal marketplace

Store managers can edit the initial proposal as they see fit, based on their relevant local knowledge, except for mandatory instructions. The stores can request items and sizes outside the initial proposal, and even if not yet sent to the store; they can also change the quantity of pieces ordered.

However, they must respect their budget constraint. This force the store manager to accurately choose orders, and facilitates the release of unsold items, that can be shipped to stores in which they are considered more attractive. This way, the system gives store managers the possibility to decide what to offer, but also guarantees that the allocation is fair and transparent.

At the end of the assigned time window to make changes, the modified proposals are collected. Tab. 4 shows an extract of a Edited proposal example. For each SKU/size, the store manager can see the *suggested action* (Orders/Releases/Mandatory Releases) and their contribution to the budget. Then, the store manager edits the suggestion writing *requested actions*. In the example, the store manager has made new orders but also increased the releases so to match the budget constraint.

3.3.1. Popularity index Additionally to direct monetary benefits, the new system provided the central merchandising team with a frequent "survey-like" review of customers' preferences, based on the requests of store managers (as a proxy). This can help to predict slow and fast movers, both locally as well as globally.

In order to measure and control store managers' preferences, we defined a *popularity index* that takes into account the intensity of orders or releases for each SKU and their inventory level . For each item i, we define

$$\phi_i^w = \min\left(1, \frac{\sum_{s,j} R_{ijs}^{\tilde{w}}}{\sum_{s,j} E_{ijs}^{w}}\right) \tag{2}$$

The numerator in 2 is the net total request for item i, on all stores and sizes; the denominator is the total inventory in the stores involved in the replenishment program. The resulting quantity is bounded, ie $\phi_i^w \in [-1, 1]$. Ranking all the SKUs of the brand at a fixed week w, an S-shape curve in figure 3 is found.

Figure 3 Ranked popularity index of SKUs of the brand in the test and recommended actions

It is then possible to classify the SKUs in groups according to their popularity. The management can then consider strategic actions. For example, SKUs with highest popularity can be urgently restocked from central warehouse or other sources; other SKUs with positive index can be recommended for a price increase if they won't be restocked. On the other side, SKUs with negative index can be recommended for promotional activities, even during the season. If $\phi_i^w \sim -1$, all stores would prefere to replace the SKU with others, so the retailer can consider to remove it from the assortment.

The central part of the S-curve in 3 represents the group of SKUs for which $\sum_{s,j} R_{ijs}$ is almost null. Those items are likely to be the most profitable to reallocate, due to opposite requests made by different stores.

3.4. Optimal allocation

Once proposals of every store managers are compiled, the allocation need to be optimized. We base the allocation process on 3 calculation steps:

1. Demand forecast for week $\tilde{w}$ at item/size level for each store (adapted from [5])
2. Expected sales as a function of stock constraints, for each item in each store
3. Optimal allocation, maximizing the expected future profit

At week $\tilde{w}$, let $R_{ijs}^{\tilde{w}}$ be the number of pieces ordered by the store managers, if positive, or released, if negative. Moreover, let $x_{ijs}^{\tilde{w}}$ be the quantity (item i, size s, and store j) to be allocated . If $x_{ijs}^{\tilde{w}} > 0$ the store will receive the corresponding articles; otherwise, if $x_{ijs}^{\tilde{w}} < 0$ the store is expected to release the goods and ship them to other stores.

3.4.1. Demand forecast It is possible to consider the number of sold pieces of a certain SKU/size/store as the count of occurrencies of a random event that happens at a certain rate. In this framework, we can model sales as a Poisson process. Let $N_{ijs}^{w}(t)$ be the Poisson process that counts the number of items sold in the time interval $(w, w+t]$ with intensity λ_{ijs}^{w}. The parameter λ_{ijs}^{w} is estimated as the mean number of articles expected to be sold in week w, calculated as follows.

For a given category C, let $\tilde{Y}_{C}^{\tilde{w}}$ be the sales forecast (aggregated by sizes, stores and items of the category C) of week $\tilde{w}$ calculated as described in Appendix A . To derive the sales forecast for a single item/size in a given store the allocation is computed:

- by store, according to the ratio between past store sales for items in C and the total past sales for items in C across all stores, during the previous year

- by item, according to the proportion of store managers' positive requests for the item versus the total positive requests for items in C

- by size, using the relative frequency for size s of item i already derived in 3.1.3

So, we obtain:

$$\tilde{Y}_{ijs}^{\tilde{w}} = \tilde{Y}_{C}^{\tilde{w}} \cdot \frac{\sum_{w \in W^*} \sum_{i \in C} \sum_{s \in S} Y_{ijs}^{w}}{\sum_{j \in J} \sum_{w \in W^*} \sum_{i \in C} \sum_{s \in S} Y_{ijs}^{w}} \cdot \frac{\sum_{s \in S} \max(R_{ijs}^{\tilde{w}}, 0)}{\sum_{i \in C} \sum_{s \in S} \max(R_{ijs}^{\tilde{w}}, 0)} \cdot p_{ij}(s), \tag{3}$$

where R_{ijs}^{w} are the store managers' requests, W^* is the set of weeks corresponding to the previous year , $p_{ij}(s)$ is the size frequency described in section 3.1.3. Finally, the estimated mean expected sales are based on a combination of forecast and store requests, as follows:

$$\lambda_{ijs}^{\tilde{w}} = \alpha \cdot \left[E_{ijs}^{\tilde{w}-1} + R_{ijs}^{\tilde{w}} \right] + \beta \cdot \tilde{Y}_{ijs}^{w}, \tag{4}$$

where α and β are two coefficients such that $\alpha + \beta = 1$. The weights assigned to the manager's experience and the statistical forecasting can be updated based on the relative accuracy observed. The better the measured performance of store managers at predicting future sales, the larger the α.

3.4.2. Expected sales as a function of stock constraints The demand forecast so far does not account for stock constraints, however the maximum potential sales depend on the store stock. Thus, the simple Poisson process is not enough to model sales. Assuming mutual independence across the sales of different sizes and items, and that the average intensity of the process is equal

to the demand forecast $\lambda_{ijs}^{\tilde{w}}$ computed in 3.4.1, it is possible to show (see [10]) that the expected sales in a time period of T weeks until stock-out equals

$$\sum_{k=1}^{E_{ijs}^{\tilde{w}-1}+x_{ijs}^{\tilde{w}}} \mathbb{P}(N_{ijs}^{\tilde{w}}(T) \geq k),$$

which yields the expected sales as a function of forecast demand (see Section 3.4.1) and new stock allocated.

3.4.3. Logistics The total expected income can be expressed as a function of the items' allocation among stores $x_{ijs}^{\tilde{w}}$.

First, the expected income for item i, size s, across all stores, is calculated by multiplying the expected sales and the price $P_{ij}^{\tilde{w}}$ of the item in each store

$$\sum_{j \in J} P_{ij}^{\tilde{w}} \sum_{k=1}^{E_{ijs}^{\tilde{w}-1}+x_{ijs}^{\tilde{w}}} \mathbb{P}(N_{ijs}^{\tilde{w}}(T) \geq k).$$

Then, the potential profit from items remaining in the warehouse after the shipments is:

$$K \cdot \left(H_{is}^{\tilde{w}-1} - \sum_{j \in J} x_{ijs}^{\tilde{w}} \right).$$

These items can still be used for future shipments, so the value K corresponds to their expected future income. K is larger at the beginning of the season and of the item's life cycle, then progressively smaller as the remaining shelf time left gets shorter.

Finally, the unknown $x_{ijs}^{\tilde{w}}$ are calculated to maximize, for each i and s, the total expected net income:

$$\sum_{j \in J} P_{ij}^{\tilde{w}} \cdot \left(\sum_{k=1}^{E_{ijs}^{\tilde{w}-1}+x_{ijs}^{\tilde{w}}} \mathbb{P}\left(N_{ijs}^{\tilde{w}}(T) \geq k\right) \right) + K \cdot \left(H_{is}^{\tilde{w}-1} - \sum_{j \in J} x_{ijs}^{\tilde{w}} \right), \tag{5}$$

where T is the time horizon of the procedure. The optimization is subject to the following constraints:

$$\begin{aligned} (a) \qquad & x_{ijs}^{\tilde{w}} \leq \max\left(R_{ijs}^{\tilde{w}}, 0\right) \\ (b) \qquad & x_{ijs}^{\tilde{w}} \geq \min\left(R_{ijs}^{\tilde{w}}, 0\right) \\ (c) \qquad & 0 < \sum_{j \in J} x_{ijs}^{\tilde{w}} < H_{is}^{\tilde{w}-1}, \end{aligned} \tag{6}$$

where (a) and (b) ensure that the store requests are not exceeded (if $R_{ijs}^{\tilde{w}} > 0$) and that the maximum deliveries from the store do not exceed the released quantity (if $R_{ijs}^{\tilde{w}} < 0$). The last

Store	Budget [EUR]	Sugg O	Sugg R	Req O	Req R	Fulf O	Fulf R	% proc
WH	-7653	0	0	0	0	0	0	0 %
1	-7386	155	198	155	198	102	175	77 %
2	-2827	0	0	30	230	11	10	13 %
3	-2069	54	59	70	64	43	49	70 %
4	43095	356	0	383	22	219	0	56 %
5	26149	229	0	240	12	151	0	61 %
6	13037	184	62	203	134	95	32	41 %
7	-30356	0	202	0	74	0	61	84 %
8	-3762	0	0	1	0	0	0	70 %
9	-2916	0	0	52	144	23	2	19 %

Table 5 Summary of optimal allocation output (extract example)

constraint (c) ensures that no items are sent back to the warehouse from the stores, and that (trivially) every piece is available from the stores or the central warehouse.

The optimization problem is an integer programming formulation, solved with the GENOUD algorithm ([9]). Example summary of optimal allocation output is given in Tab.5, showing Orders and Releases for each store and for the warehouse (WH). The initial proposal (suggested), the store-reviewed proposal (requested), and the final output allocation (fulfilled) is shown. In this example, over 50% of the store managers' requests are satisfied.

Notice that the store budget b_j directly influences the store requests, but is not directly involved in eq., where individual stock levels by SKU and store are involved. The optimization process attempts to rebalance stock levels only if profitable extra-shipment are possible. Thus, after the optimization the stock levels across the stores will be more balanced, but still not exactly balanced, avoiding non profitable extra costs.

4. Pilot Study and Results

Our case study company, an italian retailer, is a leader in the Italian premium curvy fashion retail, with approximately 100 stores and an average item selling price of 150 EUR.

The retail footprint of the retailer in analysis spans across ∼100 stores in Italy, collectively selling ∼400k pieces a year. Two collections alternate every year, summer and winter, each consisting of ∼1300 SKUs grouped in 20 categories; as is typical of the luxury segment, stores carry a low inventory of each item, but guaranteeing availability of all sizes in the item size plan at each store. Sales are highly seasonal. The company switched to a fully-automated system managed centrally, that assumed a homogenous demand distribution across stores (with rare exceptions for special categories such as beachwear), with a simple scaling based on the overall level of sales for each store (rounded to the nearest integer).

This new process has been subject to a rigorous pilot experiment during the Spring-Summer 2016 season to estimate impact on sales and revenues, but also to collect feedback from the store managers.

We set up a classical test-control study: the retailer's management selected test and control stores based on their experience, as a representative sample of the store universe in terms of geographical location, sales volume, and historical seasonality. Test and control groups had respectively 18 and 46 stores, and the entire product range was included in the study.

During the 13-week test phase, the coefficients in 1 have been set to $\alpha = \beta = 1/2$ as it is the natural initial choice when the store manager's predictive skill is not evaluable.

The test group used the new replenishment process, while the control group kept the automatic headquarter-driven process. The number of future weeks for replenishment n_f was set to 8. Moreover, the number of past weeks n_p to use as a reference for the replenishment suggestion and budget computation was set to 4.

The stores actively took part to the test modifying the first suggestions. The average partecipation ratio (in terms of ratio of edited suggestions over total suggestions) was 72%. As shown in Fig.4, store managers edited from 0% to 33% of total pieces in the proposal, and the median of the percentage of suggestion edited is 4%.

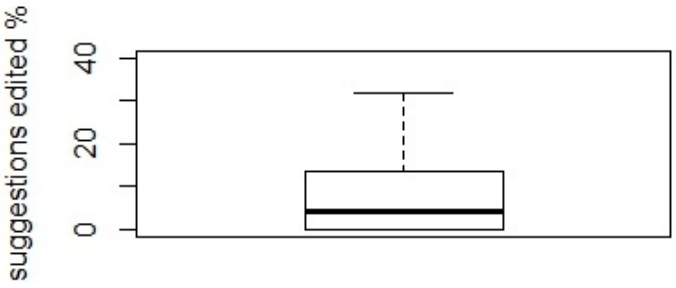

Figure 4 Boxplot of the ratio of the suggestion modified, for all the test stores at week 8 of the test.

4.1. Test impact evaluation

Indeed, we compared test and control stores in terms of operational performance of allocation process. We used 4 performance metrics: 3 overstock measures (stock to the stores not to exceed demand) and 1 of understock (shipments sufficient to cover consumers' demand), explained in the following.

Shipment Success Ratio (SSR): the proportion of delivered items (positive replenishments) actually sold after delivery. For each positively delivered item, the ratio between gained and sold articles is calculated. As we want to measure the shipped items actually sold, the sales are capped to the

total number of delivered items. This quantity is then cumulated over time, since items can be sold for few weeks after delivery. So finally, at week $\tilde{w}$ the SSR is given by

$$\mathrm{SSR}_{\tilde{w}} = \frac{\sum_{w=1}^{\tilde{w}} \mathrm{shipments}_w}{\sum_{w=1}^{\tilde{w}} \min\left(\mathrm{sales}_w, \mathrm{shipments}_w\right)}.$$

Sales to Shipment Ratio (StSR): the proportion of total sales over the number of delivered pieces. The StSR at week $\tilde{w}$ is then

$$\mathrm{StSR}_{\tilde{w}} = \frac{\sum_{w=1}^{\tilde{w}} \mathrm{sales}_w}{\sum_{w=1}^{\tilde{w}} \mathrm{shipments}_w}.$$

If the number of sold items is larger than the number of injected items, this metric is larger than one and means that the stores have high stock levels, that should be monitored. Every unsold stock item is a potential sale lost.

Stock Velocity Ratio (SVR): the ratio between total sales and potential sales. The latter are evaluated as residual stock plus cumulated sales. The SVR at week $\tilde{w}$ is given by

$$\mathrm{SVR}_{\tilde{w}} = \frac{\sum_{w=1}^{\tilde{w}} \mathrm{Sales}_w}{\mathrm{Stock}_{\tilde{w}} + \sum_{w=1}^{\tilde{w}} \mathrm{Sales}_w}.$$

To complete the picture we need a metric to evaluate wether the stores are better fulfilling the clients' expected demand. We call it *Demand Cover Ratio (DCR)*: the ratio between sold items and expected demand, calculated as in [4]. At each week w, if sales are positive and the number of stockout days is smaller than 7, the expected demand is:

$$\mathrm{demand}_w = \mathrm{sales}_w \cdot \left(\frac{7}{7 - \mathrm{stockout\ days}_w}\right).$$

Otherwise, expected demand is set to the most recent nonnegative demand (or zero). The rationale is that potential demand is equal to actual sales plus missed sales due to stockouts. So finally DCR at week $\tilde{w}$ is given as

$$\mathrm{DCR}_{\tilde{w}} = \frac{\sum_{w=1}^{\tilde{w}} \mathrm{sales}_w}{\sum_{w=1}^{\tilde{w}} \mathrm{demand}_w}.$$

The metrics have been calculated for test and control groups, results are illustrated in Table 6.

Figure 5 shows how SSR and SVR increase for Test stores since the beginning of the test, and that this increase remains stable along the prosecution of the test.

At the same time, DCR Figure 5 shows how DCR of Test and Control stores remained similar (ie extra sales coming from shipments do not lead to higher occurrence of stock-out in the shipping store). StSR sharply increased especially in the first weeks.

On average, the pilot demonstrated the following:

- up to 7% increase in SSR, and up to 19% increase in StSR, a significant improvement in the relevance of shipments compared to the past, resulting in more sales;

Week	(T) SSR %	(C) SSR %	(T) StSR%	(C) StSR%	(T) SVR%	(C) SVR%	(T) DCR%	(C) DCR%
1	12%	12%	166%	161%	6%	5%	58%	57 %
2	19%	17%	111%	112%	12%	9%	59%	56 %
3	22%	19%	86%	76%	17%	13%	60%	58 %
4	30%	24%	93%	76%	24%	19%	60%	60 %
5	31%	27%	88%	68%	27%	21%	61%	62 %
6	35%	30%	90%	73%	28%	23%	61%	61 %
7	40%	33%	91%	74%	31%	26%	61%	61 %
8	43%	36%	100%	81%	34%	28%	60%	61 %
9	43%	39%	92%	89%	36%	31%	60%	60 %
10	47%	42%	99%	94%	38%	33%	60%	60 %
11	51%	46%	106%	101%	41%	36%	60%	59 %
12	53%	50%	112%	110%	44%	39%	60%	59 %
13	58%	53%	132%	123%	49%	43%	61%	60 %

Table 6 Monitoring metrics: test (T) vs control (C) KPIs

Figure 5 Weekly difference between Test and Control indexes SSR and SVR

Figure 6 Weekly difference between Test and Control indexes StSR and DCR

- up to 6% increase in SVR; greater stock rotation leading to a reduction in unsold stock;

- <1% greater DCR, suggesting further opportunity to improve the coverage of latent demand and further reduce the risk of stock-outs for items with additional potential demand.

These results, in monetary terms, translated into an estimated 280k EUR margin impact during this pilot. The positive impact on sales of the test stores is shown in Fig. 7, where control sales were scaled to test group sales before the test. The estimated impact was a significant +16%

revenues, worth 280k EUR margin impact just during this pilot (over three months and 18 stores). We expect a margin of 1M/month over the entire network for the entire year, considering a 40% lower expected impact after the end of the test phase.

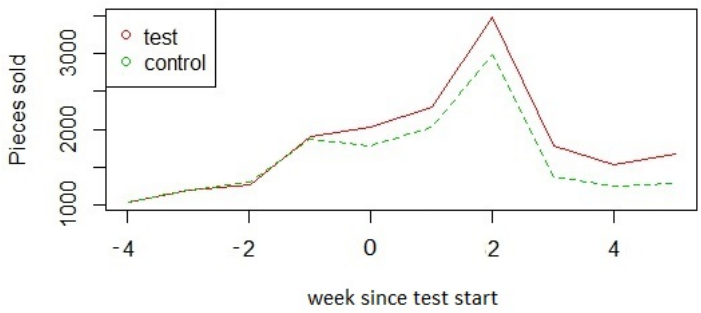

Figure 7 Impact on sales: test vs scaled control.

As non-monetary benefits, store managers' positive feedback, sense of ownership and empowerment, resulted in a boost of staff morale. The store managers became more motivated to sell the items they had specifically requested.

As it was mentioned before, the requests of the store managers provide the retailer central office a survey on the quality of each item. Fig. 8 shows a correlation between SKU level popularity index and sell-through for each SKU, meaning that store managers'orders have a good predictive power.

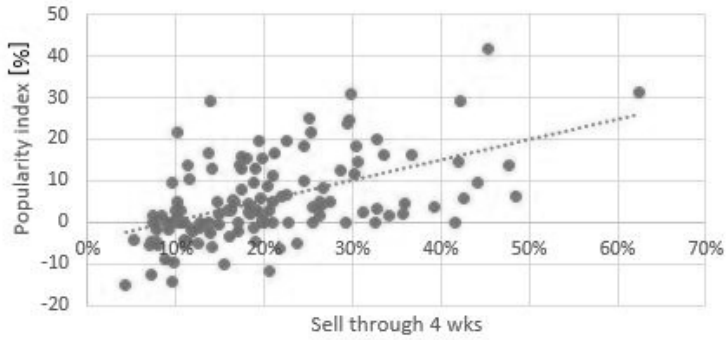

Figure 8 SKU level popularity index and sell-through for each SKU. The chart shows a correlation between the two, meaning that store managers'orders have a good predictive power.

5. Conclusions

The purpose of this work was to show the implementation and results of an original intelligent system able to improve the replenishment process of fashion retail companies, in order to limit missing sales due to stock-out, enrich and adapt the assortment of the single stores to local preferences. We combined a traditional forecast-driven approach to the requests of store managers, in order to achieve a greater level of store-customization and ultimately greater local relevance to the specific demand of each store.

Each store manager receives an initial proposal they can modify, within certain budget constraints, creating a non-monetary incentive to accurately assess her expectations on the relative expected performance of the current stock, and decide what to release and what to request from other stores, thus creating an internal marketplace.

All requests are matched to the available inventory across all stores, and deliveries are optimized to minimize the number of parcels (and logistic costs) while maximizing the expected profit based on a greater expected likelihood of sale.

We successfully tested the system in a 13-weeks pilot. Test stores outperformed the control group on all the metrics and also increased sales relative to control.

However, this new process is open to many further improvements. Transhipments across stores could be optimized in order to further reduce logistic costs and store managers' workload. For example, a minimum shipment value can be imposed, or a cap to the maximum number of parcels that can be delivered by a store might be enforced.

Furthermore, the pilot store managers personally recommended some further additional improvements, mainly to reduce the size-level fragmentation and probability of stock-outs, for example, by penalizing releases that would lead to a stock-out, or by modifying the system so it would re-compact SKUs across their entire size plan.

Appendix. Forecasting model

Category sales $\tilde{Y}_C^{\tilde{w}}$ (aggregate: all sizes, all stores, and items in C, with one year time horizon) are forecasted through multivariate regression on seasonality $\bar{z}$, average full price p, average markdown m, and units per tickets ratio u. The forecasting procedure consists in these steps:

1. historical data are aggregated at the category, year and week level
2. for every category C and week of the year w, weekly seasonality index $\bar{z}_C^w$ is computed
3. a linear model $\tilde{Y} = f(z, p, m, u)$ is fitted for every category
4. an estimation of the KPI p, m, and u is given for the following 52 weeks
5. one year horizon sales forecast is computed $\tilde{Y}_C^{\tilde{w}}$ for each category and week by using the estimated models

In the following, details on seasonality index and estimation of the regressors for the future year are provided.

A. Category seasonality

From the weekly aggregated sales, for each category, we remove the effect of markdowns by estimating the model

$$Y_C^w = \alpha_C + \beta_C md_C^w.$$

Then for each week we calculate normalized sales

$$\bar{Y}_C^w = Y_C^w - \beta_C md_C^w.$$

We then average normalized sales $\bar{Y}_C^w$ over the years, obtaining $\bar{\bar{Y}}_C^w$, and we smooth this using the moving average of 3 weeks:

$$z_C^w = \frac{\bar{\bar{Y}}_C^{w+1} + 3\bar{\bar{Y}}_C^w + \bar{\bar{Y}}_C^{w-1}}{5}.$$

Finally, we normalize seasonality (every category sums to 52)

$$\bar{\bar{z}}_C^w = z_C^w \times \frac{52}{\sum_w z_C^w}.$$

B. Prediction of business indicators

Future average values of full price p, markdown m, and units per tickets ratio u are estimated by moving averages. As an example, starting from category level average full price for a given week and year, p_C^w, we average price over the years obtaining $\bar{p}_C^w$. Then we smooth this using the moving average of 3 weeks:

$$\bar{\bar{p}}_C^w = \frac{\bar{p}_C^{w+1} + 3\bar{p}_C^w + \bar{p}_C^{w-1}}{5}.$$

$\bar{\bar{m}}$ and $\bar{\bar{u}}$ are then computed in the same way and then used with in (5) as regressors together with $\bar{\bar{p}}$ and $\bar{\bar{z}}$. It is also possible to manually change these values according to future management strategy.

References

[1] Agrawal, N., & Smith, S. A. (2013). Optimal inventory management for a retail chain with diverse store demands. European Journal of Operational Research, 225(3), 393-403.

[2] Blattberg, R.C., and Hoch, J.S. Database models and managerial intuition: 50% model+ 50% manager. Management Science 36.8 (1990): 887-899

[3] Cachon, G. P. and Lariviere, M. A. (1999). An equilibrium analysis of linear, proportional and uniform allocation of scarce capacity, IIE Transactions, 1999 Sep, 31(9), 835-849.

[4] Caro, F. and Gallien, J. (2010). Inventory management of a fast-fashion retail network. Operations Research, 58(2): 257-273.

[5] Correa, J. (2007). optimization of a fast-response distribution network. M. S. thesis, LFM, MIT, Cambridge, MA.

[6] Fisher, M., & Rajaram, K. (2000). Accurate retail testing of fashion merchandise: Methodology and application. Marketing Science, 19(3), 266-278.

[7] Furuhata M., Zhang D. Capacity Allocation with Competitive Retailers.

[8] Gallien, J., Mersereau, A. J., Garro, A., Mora, A. D., & Vidal, M. N. (2015). Initial shipment decisions for new products at Zara. Operations Research, 63(2), 269-286.

[9] Mebane, W. R. Jr. and Sekhon, J. S. (2011). Genetic Optimization Using Derivatives: The rgenoud package for R. Journal of Statistical Software,42(11): 1-26.

[10] Sirovich, Marocco, Craparotta, A woman's touch in fashion forecasting: combining analytics & experts'judgement, in preparation

[11] Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. International Journal of Production Economics, 128(2): 470-483.

[12] Thomassey, S., Happiette, M., & Castelain, J. M. (2005). A global forecasting support system adapted to textile distribution. International Journal of Production Economics, 96(1), 81-95.

[13] Van Donselaar, K. H., Gaur, V., Van Woensel, T., Broekmeulen, R. A., & Fransoo, J. C. (2010). Ordering behavior in retail stores and implications for automated replenishment. Management Science, 56(5), 766-784

[14] WSJ, 12th August 2010.