

Demand Forecasting and Markdown Optimization
Application at a Fashion Retailer

by

Nazlı Oğuz

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Industrial Engineering

Koç University

July, 2014

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Nazlı Oğuz

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Dr. Yalçın Akçay

Prof. Dr. A. Gürhan Kök

Asst. Prof. Dr. Pelin Gülşah Canbolat

Assoc. Prof. Dr. Skander Esseghaier

Prof. Dr. Zeynep Akşin Karaesmen

Date:

ABSTRACT

Coping with uncertainty and price decisions are two significant factors in revenue maximization for the retailers who sell products with short life cycles. In this study, we develop a methodology to forecast discount season sales and generate optimal pricing policies for selling a finite amount of perishable products over a finite horizon. First, we establish three regression-based forecasting models and compare their performances. In order to generate optimal pricing policies, we then use a dynamic pricing model and analyze its extension under a maximum final inventory constraint. We implement our proposed methodology using real data from a fashion retailer in Turkey. Our simulations indicate that, using our method, the retailer could expect a 9.7% increase in its revenues. As an additional study, the retailer's stores are clustered into two sets according to their price sensitivities and simulation is run for these two clusters separately. Clustering stores achieved a 10.7% improvement in revenues.

ÖZET

Kısa ömürlü ürünler satan perakendecilerin mümkün olan en iyi geliri elde etmelerinde talepteki belirsizlikle başa çıkabilmeleri ve aldıkları fiyatlandırma kararları iki önemli unsurdur. Bu çalışmada, perakendecilerin kısıtlı sayıda ürünün satışı için indirim dönemi satışlarını tahmin edebilmek ve yine bu dönem için en iyi fiyat politikalarını belirlemek amacıyla bir yöntem geliştirilmiştir. Öncelikle, üç tane regresyon tabanlı talep tahmin modelinin performansları karşılaştırılmıştır. En iyi fiyat politikalarının üretilmesinde kısıtlı sayıda indirim opsiyonu içeren bir dinamik fiyatlandırma modeli kullanılmış ve bu modelin azami bitiş envanter seviyesi kısıtı altında da performansı değerlendirilmiştir. Devamında, Türkiye’den bir moda perakendecisine ait satış verileri kullanılarak geliştirilen yöntem ile kapsayıcı bir benzetim uygulaması yapılmıştır. Simülasyon uygulamalarının sonucunda, önerilen yöntemin uygulanması ile elde edilen gelir miktarında %9,7’lik bir artış beklenebileceği gösterilmiştir. Ek bir çalışma olarak, perakendecinin mağazaları, mağazaların fiyat duyarlılıkları gözetilerek kümelenmiş ve iki ayrı kümeye bağımsız benzetim uygulaması yapılmıştır. Bu uygulamayla, elde edilen gelirde ortalama %10,7’lik bir artış sağlanmıştır.

to the best brother, Oğuzcan and my invaluable family...

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Literature Review	3
1.3 Contributions	9
1.4 Outline	10
Chapter 2: Sales Forecasting Model	11
2.1 Objectives	11
2.2 Models	12
2.2.1 Caro-Gallien Model (CG)	12
2.2.2 Caro-Gallien Model with Reference Price (CGwRP)	19
2.2.3 Forecasting from Decomposed Data (FDD)	20
2.3 Performance Criteria	22
2.4 Bias Correction Methods	23
2.4.1 Miller Method	24
2.4.2 Smearing Method	24
Chapter 3: Revenue Optimization Model	26
3.1 Aim of the Model	26
3.2 Optimization Model	27
Chapter 4: Implementation	35
4.1 Data	35
4.2 Forecasting Model Application	40
4.2.1 Training and Validation Data Sets	40
4.2.2 Comparison of Forecasting Models' Result	41
4.2.3 Best Model's Parameter Results	41
4.3 Clustering	42
4.4 Simulation Application	44
4.4.1 Aim of Simulation Application	44
4.4.2 Simulation Application Steps	45

4.4.3	Results of Simulation Application	50
4.4.4	Results of Simulation Application with Clustering	59
4.4.5	Economic Impact, Practicability and Drawbacks of Application	61
Chapter 5: Conclusion.....		63
Bibliography		65
APPENDICES		69
APPENDIX A: Forecasting Codes - R		69
APPENDIX B: Simulation Codes – R		74
APPENDIX C: Optimization Codes – GAMS.....		78

LIST OF TABLES

Table 4.1: MD and MAD Values of Forecasting Models	41
Table 4.2: Linear Regression Coefficient Values of the Best Model.....	42
Table 4.3: Store Lists	43
Table 4.4: MD and MAD Values of Two Store Sets	44
Table 4.5: Revenue Improvement Percentages of Product-Models.....	52
Table 4.6: Final Inventory Level Ratios.....	54
Table 4.7: Revenue Improvement Percentages of Product-Models with the Additional Constraint	57
Table 4.8: First Part of Simulation Results with Clustering	60
Table 4.9: Second Part of Simulation Results with Clustering.....	61

LIST OF FIGURES

Figure 2.2: Two-Step Procedure for Calculating Linear Regression Parameters by Model CG	18
Figure 2.3: Price Percentage, Real and Decomposed Sales Data.....	22
Figure 4.2: Screenshot from the Data.....	36
Figure 4.4: Screenshot from the Data.....	36
Figure 4.5: Weekly Sales and Average Price Percentage	39
Figure 4.6: Inventory Level Aggregated to Weeks	39
Figure 4.8: Average Price Percentage Amounts to Weeks	50
Figure 4.9: Average Price Percentages of Selected Product-Models.....	51
Figure 4.10: Real and Optimal Price Percentage & Sales Values.....	55
Figure 4.12: Real and Optimal Price Percentage & Sales Values with the Additional Constraint	58

Chapter 1

Introduction

1.1 Motivation

Retailing is one of the promising industries in the world. Having 16 trillion-dollar volume today, global retailing industry is estimated to reach 22 trillion-dollar value by 2015. According to the Deloitte Turkey Corporate Finance Department's projections, Turkish retail sector is expected to grow by 10% a year till 2016. As the volume of industry increases, keeping or increasing the market share is getting harder. Moreover, with the developments in information sharing and mobile technology, consumers are now more sensitive to the companies' moves. Therefore, the decisions made in sales planning and selling phases have become more critical.

Retailers who sell products with short life cycles place their orders months before the selling season starts due to long lead times. Electronics and fashion retailers can be considered among these retailers. Moreover, these retailers face high levels of demand uncertainty for each of their products during the selling season. One of the reasons of this uncertainty is absence of a reliable forecasting procedure. Even when the retailer is good at estimating demand due to the changes in needs and interests of customers, competitors' actions and technology development uncertainty still exists. This uncertainty may hurt the retailer's financials in two ways. For one, if retailer's sales expectations are below the actual sales, the retailer runs out of inventory before the season ends and loses both sales and potential customers. Second, the retailer might order too much and has excess inventory on hand at the end of the season. Inventory holding cost is one of the major

expenses of retailers, and in this case the excess inventory is a dead investment for the retailer.

At this point the need for a good forecasting methodology arises. A good methodology should consider distinctive requirements of the company, characteristic of its consumer group and competitive environment where the company takes place. If the retailer understands the major determiners of demand and reflects them on its forecasting tools successfully, it will be safe especially in case of excess demand.

Even though having a successful forecasting model reduces uncertainty in demand, it cannot eliminate uncertainty completely. Thus, the actions to manipulate demand throughout the selling season have significant importance for the retailers. One of the main methods used in boosting sales and clearing the inventory of products with high remaining inventories is cutting prices. Price cuts are required not only to boost sales, but also to stay competitive. This method is also known as markdown pricing or clearance pricing and the period when prices are cut is referred as discount or clearance season. On the other hand, the period before the discount season is called the regular season. The revenue obtained during discount season constitutes a significant part of the total revenue obtained throughout the whole selling season. Thus, a good decision policy making methodology is required together with a reliable forecasting model in order to be able to cope with uncertainty in demand.

Our focus for constructing a model to generate optimal clearance period pricing policies is on products with short life cycles. Generally, these types of products are not discounted during the regular season. Typically, when the discount season starts, the retailer does not have any price response data for the current season's products which causes one of the main challenges in building this model. Further, different customer segments have different sensitivities to price changes and thus a general pricing policy may not perform well for all segments. Finally, incorporating real-life conditions and business constraints in the problem can easily make the problem intractable. While the second reason above suggests that a successful markdown optimization algorithm should provide solutions with

potentially different price levels for each store segment, the state of the art methodology and software (and the current practice in retailing in Turkey) follow a one-size fits all approach: markdown estimation is performed by treating all customers as a single pool and for each product, and markdown price is set the same at all stores of the retailer.

1.2 Literature Review

The papers by McGill and van Ryzin (1999), Elmaghraby and Keskinocak (2003), Bitran and Caldentey (2003) together with the book written by Talluri and van Ryzin (2004a) are comprehensive studies in the areas of dynamic pricing and revenue management. In this context, our focus is on the dynamic pricing models which are designed for environments in which the inventory amount is restricted and the time for sales is finite. The studies for those models can be grouped into two as single-product and multi-product studies.

Within the single-product studies Gallego and van Ryzin (1994) and Zhao and Zheng (2000) studied dynamic pricing for continuous time. In these studies, they showed that optimal prices increase with increasing inventory at a constant time and decrease with decreasing time at a constant inventory level. In most of the product management studies, the demand from customers was assumed to be independent from price and is managed by accepting or refusing customer requests rather than price changes. However, in Gallego and van Ryzin's study, Poisson arrival process was used to define demand. It was assumed that demand is sensitive to price changes and price increases affect demand negatively. Differently from the similar studies in this area, in the study of Smith and Achabal (1998) considering a finite horizon it was observed that sales rates are affected from inventory level in one direction. Toward these observations, the revenue maximizing price and inventory policies were determined by dynamic programming. For the cases where continuous price changes are not applicable, Feng and Gallego (1995), Bitran and Mondschein (1997) and Feng and Xiao (2000) studied models where the number of price changes are limited. Çelik, Muharremoğlu and Savin (2009) defined a cost for the price changes and Netessine (2006) studied dynamic pricing with infrequent price changes. In a different study in this area, Talluri and van Ryzin (2004) investigated the problem of

selling a product to different customer groups at different prices. The remarkable point here is, in this study contrary to the common use of *pricing* in literature the intention from *pricing* is not dynamic pricing. In this study, the price decision is made for every period separately. Chen and Farias (2013) showed that the optimal pricing policies obtained from the classical single-product dynamic pricing problem compensate for forecast imperfections by frequent reoptimization.

In the studies conducted for multi-product cases, price decisions of different products cannot be made independently due to the relations between demands and common capacity constraints. Maglaras and Meissner (2006) developed a one-dimensional formulation to the well studied revenue management problems where multiple products are available for customers. Gallego and van Ryzin (1997) developed a model for multi-product pricing problem with given beginning inventory level and finite horizon for sale. Since the developed model is hard to solve, in this application heuristic solution methods are developed and the solutions to these methods are shown to be asymptotically optimal. Lin and Li (2004) presented problem of multi-product dynamic pricing as a continuous time model and explained demand with a linear model considering cross flexibilities. However, they cannot reach a structural property or understanding related to the solution of the problem. They measured the performance of a heuristic method that they proposed instead of an optimal pricing strategy. Finally, Zhu and Thonemann (2003) studied on joint dynamic pricing and inventory decisions. In the study they assumed that the company sells two products and they modeled the demand as a linear equation reflecting cross-price elasticity. Using Karush-Kuhn-Tucker conditions they found the optimal base-stock. Moreover, they showed that joint strategies are better than independent ones and dynamic pricing is more profitable than stationary. However, rather than a general case the used model is special to a particular case. Thus, it seems far away from providing an insight to managers and being practical.

Raman and Chatterjee (1995) incorporated dynamic effects of demand and supply in the market to their pricing problem. However, different from the prior studies by Dolan and Jeuland (1981), Kalish (1983), Jeuland and Dolan (1982), Robinson and Lakhani (1975),

Raman and Chatterjee (1995) added demand uncertainty in their problem and developed both a theoretical characterization of the optimal policy for the general model and specific solutions to popular special cases. Besbes and Saure (2014) studied a dynamic pricing problem over a finite horizon for a retailer who does not have the opportunity to replenish its inventory and the demand function or the retailer is changing. They observed that optimal pricing policies follow a monotone path up to the change in demand. Araman and Caldentey (2012) conducted a dynamic pricing study to maximize the expected discounted cumulative revenue of a nonperishable product whose demand is driven by a price-sensitive Poisson process with an unknown parameter. They formulated the problem as an intensity control problem and derived structural properties of an optimal solution. Moreover, a simple and efficient approximated solution was suggested. Wernerfelt (1986) studied the effects of brand loyalty and experience curves on optimal dynamic pricing policy. Besbes and Maglaras (2012) developed a revenue-maximizing dynamic pricing policy under the settlement of revenue and sales targets for predetermined time points during a finite selling horizon. Feng (2010) integrated the supply capacity uncertainty to the traditional dynamic pricing problems. Levin, McGill and Nediak (2007) extended the traditional dynamic pricing problem by incorporating price guarantees in the model. The price guarantees provide customer with coverage if, prior to a predetermined time in the future, the price of the product drops below a certain level.

Pricing decisions are closely related to production, inventory and assortment decisions. Among the studies which focus on these decisions together, Dong, Kouvelis and Tian (2009) studied dynamic pricing and inventory control problem of substitute products where the retailer has a short time to sell its products, and supply lead time of the products are long. They proved that dynamic pricing solution converges to static pricing when inventory levels are close to the number of remaining selling periods, assuming one customer arrives in each period. Talebian, Boland and Salevsbergh (2014) conducted a study on making simultaneous decisions about assortment and price using a stochastic dynamic programming model for the cases of new product launches. They incorporated demand learning by Bayesian updates to their model. The profitability of the early price reductions

to accelerate demand learning is shown analytically. Chen, Wu and Yao (2010) indicated that dynamic pricing is most effective when it is jointly optimized with inventory replenishment decisions. Jorgensen, Kort and Zaccour (1999) brought production, inventory and pricing decisions together by incorporating demand learning and obtained closed-form characterizations of optimal policies.

Among the fashion industry-specific studies, Aviv and Pazgal (2005) studied the dynamic pricing problem faced by a retailer who plans to sell a given stock of fashion-like items during a finite season. They developed a rigorous upper bound for the seller's optimal dynamic decision problem and used it to propose an active-learning heuristic pricing policy which exhibits an impressive performance in general. In another similar study by Şen and Zhang (2009) a Bayesian model was developed to summarize past sales information which is expected to resolve the uncertainty about the magnitude and price sensitivity of demand. Incorporating this model, an optimization model was developed and findings were empirically validated using data from U.S. apparel industry.

Competition is a significant parameter in pricing decisions and revenue management studies. Kachani and Shmatov (2011) studied the problem of pricing complementary and substitutable products which have several distinct price differentiation classes (e.g., customer segments and different distribution channels) in both monopolistic and oligopolistic selling environments. Studying a real-world case-study example they provided an iterative relaxation algorithm, and state sufficient conditions for the convergence of the algorithm. Sato and Sawaki (2013) established a continuous-time dynamic pricing model for a duopoly where one of the sellers knows that its competitor adopts a static pricing strategy. They tried to observe whether dynamic pricing improves total expected revenue in the market and derived an explicit closed form expression for an optimal pricing policy. It is shown that adopting a fixed price policy rather than a dynamic one might be more advantageous for some cases in such an environment. Martínez-de-Albéniz and Talluri (2011) studied on pricing decisions for an oligopoly where the sellers have fixed capacity and finite time to sell. In their environment, demand is stochastic and sellers change their prices considering changing conditions in the environment such as estimated demand,

inventory levels and competitors' prices. They developed a closed-form solution to the equilibrium price paths for a duopoly and extend their results to an n-firm oligopoly. In a similar study, Gallego and Hu (2014) studied dynamic price competition in an oligopolistic market where substitutable and complementary perishable assets are sold. They showed any equilibrium strategy has a simple structure assuming deterministic customer arrival rates. Moreover, the equilibrium solutions from the deterministic case provide asymptotically optimal heuristic policies for the stochastic case. Sibdari and Pyke (2010) studied the problem of pricing in a market where two firms are trying to sell substitutable and nonperishable products with different quality levels over a finite horizon. In this study, they refer the relationship between stockpiling and consumption together with the relationship between price and demand. They showed that a unique Nash equilibrium exists on unit sales prices when the market stockpile information is assumed to be available to the public. Finally, they reviewed the effects of customer population composition on optimal revenue, consumer surplus and social welfare.

Su (2007) studied dynamic pricing problem of a monopolist who sells a finite inventory over a finite horizon to strategic consumers who are aware of the possible price changes and may time their purchases accordingly. It is found that, contrary to intuition, strategic waiting by customers may benefit the seller. Akçay, Natarajan and Xu (2010) studied the pricing problem of multiple substitutable and perishable products over a finite selling horizon under consumer choice. In another study by Su (2010), in a similar selling environment with Su (2007), it was shown that presence of a speculator in the market increases the firm's expected profits. Levin, McGill and (2009) and Liu and Zhang (2013) studied dynamic pricing policies in the presence of strategic consumers in the markets where more than one firm are active and concluded that if the firms ignore the strategic behavior of the customer they lose money together. Cachon and Swinney (2009) conducted a comprehensive study on purchasing and pricing decisions and the value of quick response in a selling environment with uncertain demand over a finite selling horizon and in the presence of strategic consumers.

Among the applied studies one of the most comprehensive ones is by Mantrala and Rao (2001). They developed a decision support system to determine optimal order quantities and markdown prices. With their study, they improved profitability and provided managers a better understanding of the decision problems. An experimental study from Sigurdsson, Foxall and Saevarsson (2010) was established about how price decisions affect choices of customers. The results they reached underline the need to examine all of the factors in a competitive environment, not only price. A similar study was conducted in a toy retailer by Gaur and Fisher (2004). An unexpected result arose from this experiment: the demand increases with price in some cases. Kesavan, Gaur and Raman (2010) showed that historical inventory and gross margin data involves valuable information which can improve sales forecasts using data from U.S. retailing industry. Caro and Gallien (2012) developed a decision support system to decide optimal clearance season prices and this system was employed by the Spanish apparel retailer Zara. The decision support system includes demand forecasts for various clearance prices. With this application they managed to improve the clearance sales revenues by 6%. Another impressive study by Natter et al. (2007) a decision-support system for dynamic retail pricing and promotion planning was developed. The implemented system increased gross profit on average by 8.1% and sales by 2.1%.

Koenig and Meissner (2010) investigated the risk and applicability of a fixed-price capacity control strategy instead of dynamic pricing for the cases in which steady price changes are costly and unachievable in practice. They consider the case of selling multiple products with fixed capacity over a finite horizon. They showed that, list pricing can be useful when dynamic pricing is costly or impractical.

Finally, Dana (2008) investigated how the changes in selling environment due to the Internet, the adoption of new information technologies and other market forces affect the needs in revenue management research. In this study, more sophisticated models of consumer behavior, more general models and understanding of contest and more general pricing mechanisms are the promising research directions in revenue management.

1.3 Contributions

In this research we develop a methodology to maximize clearance season revenue by generating a pricing policy for the discount season. In the first step of this methodology a forecasting model for the future sales using historical data is established. Three alternative forecasting models are evaluated and their performances are compared. The first model is a similar application to the forecasting model used in the paper from Caro and Gallien (2012). The second model is an extension of the first model using sales and price information from historical data as moving averages in the forecasting model. In the third model we developed a new approach and estimate explanatory variables from the decomposed data. The forecasting model to be used within this methodology is chosen among these three models.

In the second step, we study the dynamic pricing problem of perishable products. We assume a finite-horizon to sell a finite-amount of inventory and determine optimal pricing policy for the discount season. An optimization model is constructed and a simulation application is designed to generate optimal pricing policy. Output of the forecasting model is used as an input to the simulation. Within the simulation application, the forecasts are adjusted according to forecast errors. The methodology is implemented on a retailer's sales data and 9.74% improvement in revenue is expected. Moreover, the optimization model is run under a final inventory value restricting constraint. Another extension to the simulation model is clustering the stores of the retailer according to their discount season price sensitivity and implementation of simulation application to these clusters separately. Thus, different price policies are obtained for different set of stores and average expected revenue improvement increased to 10.66%.

1.4 Outline

In this study a sales estimation and markdown optimization methodology for retailers of seasonal goods is developed.

In Chapter 2, three linear regression models are developed to generate sales estimates during discount season. The forecasting models use regular season data to estimate the coefficients of explanatory variables. The performance criterion chosen to compare these models is explained. Finally, the two techniques which are used to reduce transformation bias are discussed.

In Chapter 3, the mixed integer programming optimization model which is built in order to generate the optimal price policy for discount season is discussed. The inputs, decision variables, objective function and constraints of the model are explained in detail.

In Chapter 4, the data which is used in implementation of forecasting and optimization models is introduced. Afterwards, the results of the forecasting model application are presented. Then, the simulation model which is established to imitate real life reactions to price decisions is explained and the results of the simulation application are presented.

In Chapter 5, the whole study is summarized and insights for future research opportunities are provided.

Chapter 2

Sales Forecasting Model

2.1 Objectives

The sales forecasting model is the first step of our study and has a great significance for the success of the project. In order to build a robust pricing strategy, we need to express our expectations for the future sales in a systematic manner. Therefore, the aim of the sales forecasting model is to have the ability to produce accurate forecasts for the future sales by utilizing the past sales data.

The model is built so as to produce periodic forecasts for products. Thus, there are two dimensions of aggregation decisions: time and product representation level.

Firstly we divide the sales season into equal intervals. These intervals are the timeframes to produce forecasts and can be days, weeks, months and even seasons. In our study, we use calendar weeks as our timeframes. Moreover, with single season data, the entire sales season is divided into two, as the regular season and the discount season. Regular season consists of periods starting from the beginning of the sales season until a significant discount is observed. From the end of the regular season till the end of the sales season is the discount season. The break point between two seasons can be determined by the company or identified by monitoring the price changes throughout the season.

For the second dimension of aggregation, we generate forecasts in model level by ignoring the size and color differences. Since the model is built to be applied in fashion retailing industry, different aggregation levels can be performed. In Chapter 2, the following indices and index sets will be used to demonstrate periods and product-models.

t : Periods

$$t \in \{1, \dots, T\}$$

r : Product-Models

$$r \in \{1, \dots, R\}$$

2.2 Models

Sales forecasting model is built with the purpose of generating accurate forecasts for the future selling seasons so that the company will not face with stock-outs while demand for a product still exists, or end up with excess inventory at the end of the season due to overestimation of demand. With these purposes, a similar study “Clearance Pricing Optimization for a Fast-Fashion Retailer” (Caro & Gallien, 2012) has been taken as a starting point and will be referred to as “CG” in the following sections.

2.2.1 Caro-Gallien Model (CG)

Model CG generates weekly sales forecasts for each product-model using the following variables:

λ_r^t : Sales amount of product-model r in week t

Purchase _{r} : Purchase amount of product-model r at the beginning of the sales season

Age _{r} ^{t} : Age of product-model r in week t

λ_r^{t-1} : Sales amount of product-model r in week $t-1$

InventoryPos _{r} ^{t} : Broken assortment effect for product-model r in week t

PricePerc _{r} ^{t} : Average price percentage of product-model r during week t

Forecasts are generated by running a linear regression model with the explanatory variables mentioned above. The exponential functional form of the forecasting model is as follows:

$$F(x_1, \dots, x_n) = e^{\beta_1 x_1} * \dots * e^{\beta_n x_n} \quad (2.1)$$

$$\lambda_r^t = F(\text{Purchase}_r, \text{Age}_r^t, \lambda_r^{t-1}, \text{InventoryPos}_r^t, \text{PricePerc}_r^t) \quad (2.2)$$

2.2.1.1 Explanatory Variables

We now explain these variables in more detail:

Purchase (Purchase_r): Purchase_r indicates the total purchase amount from supplier of each product-model at the beginning of the season. It is calculated for a particular product-model as the sum of the sales throughout the whole season in all stores plus the remaining inventory on hand at the end of the season. The purchase amount can be considered as the managers' anticipation about the sales potential of a product-model, and it is expected to have positive correlation with demand forecast level.

Age (Age_r^t): Age_r^t indicates the number of weeks between the the first sale of a product-model and the current week. For instance, if the first sale of a product-model occurred at week 5, then at the 6th week the age variable of that product-model takes the value "1". The sale volume of a product-model is expected to increase early, and then reach maturity. As the model ages, the sales of that product-model is expected to decrease. In other words, a negative correlation is anticipated between age and demand level of a product-model.

Sales Amount of the Previous Week (λ_r^{t-1}): λ_r^{t-1} is the sales amount of product-model r one week before the week for which we generate forecast. No significant changes from one week to the other are expected for this variable during the regular season. The change in the form of this variable for the discount season will be discussed later in section 2.3.1.1.

Broken Assortment Effect (InventoryPos_r^t): *Broken Assortment Effect* captures the situations when willing-to-pay customers are available, but they cannot find what they are looking for (i.e. their size/color). In order to measure the effect of broken assortment, the

first step is determining a threshold level for the inventory position under which the stock level will be considered to be insufficient for a healthy selling environment. The threshold level is calculated by multiplying the number of colors that the product-model is available (C_r) by the number of sizes that the product-model is available (S_r). This number represents the requirement of the availability of at least each size and color of a product-model in each store. Afterwards, this value is multiplied by a product-model-specific number (St_r) which indicates the number of stores that the product-model is available. Final value computed by the multiplication of the three factors equals to threshold level and means that minimum one from each size-color combination is required at each store so that the sales aren't affected by the inventory level. In order to reflect this effect to the forecasts, the minimum of current inventory level over threshold level and 1 is chosen, so that when the inventory level drops under the threshold level the forecast for demand would be cut down referred as in Equation 2.4.

C_r : The number of colors of product-model r on sale

S_r : The number of sizes of product-model r on sale

St_r : Number of stores where product-model r is sold

$Threshold_r$: Threshold amount of product-model r

$InventoryLevel_r^t$: Inventory amount of product-model r available on week t

$$Threshold_r = St_r * C_r * S_r \quad (2.3)$$

$$InventoryPos_r^t = \min \left(1, \frac{InventoryLevel_r^t}{Threshold_r} \right) \quad (2.4)$$

There is an inherent assumption while reflecting the broken assortment effect to forecasting model. Actually, the inventory level can affect the sales in both directions. Specifically, excess inventory can boost and insufficient inventory can damage the sales. We assume that excess inventory which is not on display does not affect the assortment and availability

image of the company. However, insufficient inventory would result in empty spaces and if the customer cannot find what s/he is looking for this situation can damage the customer loyalty. In brief, the effect of insufficient inventory is more direct and visible in shortage times. Thus, in this study only the negative effect of insufficient inventory is considered.

Price Percentage ($PricePerc_r^t$): Price percentage variable is calculated as the ratio of the average sales price for a week of a model to its full price as shown in Equation 2.5. Reflecting the effect of price changes to sales accurately is one of the main objectives of this study. Thus, this variable has an extra importance for us. Instinctively, the price and sales are expected to have a negative correlation.

p_r^t : Average price of product-model r in week t

FullPrice_r: Full price of product-model r without any discount

$$PricePerc_r^t = \frac{P_r^t}{FullPrice_r} \quad (2.5)$$

2.2.1.2 Forecasting Functional Form

Our forecasting model is of the form:

$$\lambda_r^t = e^{\beta_{0r}} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 LN(Age_r^t))} * e^{(\beta_3 LN(\lambda_r^{t-1}))} \\ * e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))} \quad (2.6)$$

Model Based β_0 Calculation

The forecasting model generates forecasts at the product-model level, thus a general intercept calculated for all product-models is insufficient for product-model level forecasts. Here, the need of calculating an individual intercept for each product-model arises. Two alternatives are introduced to calculate individual intercept points. The first option is

adding a sixth parameter depending on the product-model number, to the linear regression as in Equation 2.7.

$$\lambda_r^t = e^{\beta_0} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 LN(Age_r^t))} * e^{(\beta_3 LN(\lambda_r^{t-1}))} \quad (2.7)$$

$$* e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))} * e^{(\beta_6(r))}$$

The second option is firstly running a linear regression with parameters β_0 through β_5 (Equation 2.8) and then calculating residuals between real sales and the estimation from the linear regression. Afterwards, a linear regression is run on these residuals with one variable depending on product-model number (Equation 2.10). The intercept points with both alternatives are calculated using regular season data. Since the R^2 (coefficient of determination) of the second option is higher, it is used to calculate individual intercept points for product-models.

$$\lambda_r^t = e^{\beta_0} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 LN(Age_r^t))} * e^{(\beta_3 LN(\lambda_r^{t-1}))} \quad (2.8)$$

$$* e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))}$$

$$\hat{\lambda}_r^t = e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 LN(Age_r^t))} * e^{(\beta_3 LN(\lambda_r^{t-1}))} \quad (2.9)$$

$$* e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))}$$

$$\lambda_r^t - \hat{\lambda}_r^t = e^{(\beta'_{0r}(r))} \quad (2.10)$$

$$\beta_{0r} = \beta'_{0r} + \beta_0 \quad (2.11)$$

Two Step Procedure for Calculating Linear Regression Parameters

It is hard to see the effect of broken assortment and price cut in the regular season. The reason is, in the regular season, the major discounts have not introduced yet and price percentages are mostly close to 1. This case can be seen in Figure 2.1 which represents sales and average price percentage data of a fashion retailer. Similarly, without a significant price cut we cannot say that the sales are boosted and the stores suffer from insufficient inventory, so the broken assortment effect variables take mostly the value “1”. On the other hand, the effect of purchase, age and sales level from the previous week can be accurately estimated from the regular season data and these effects do not change much in the discount season. Thus, the parameters β_0 through β_5 are calculated in two steps.

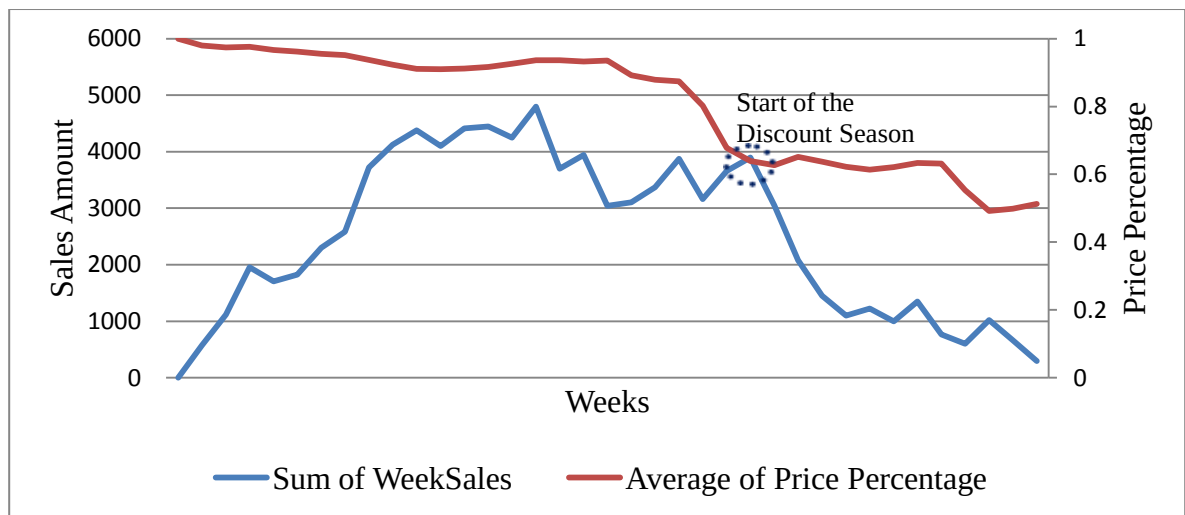


Figure 2.1: Weekly Sales and Average Price Percentage of a Fashion Retailer

The flowchart of the two-step procedure is shown in Figure 2.2.

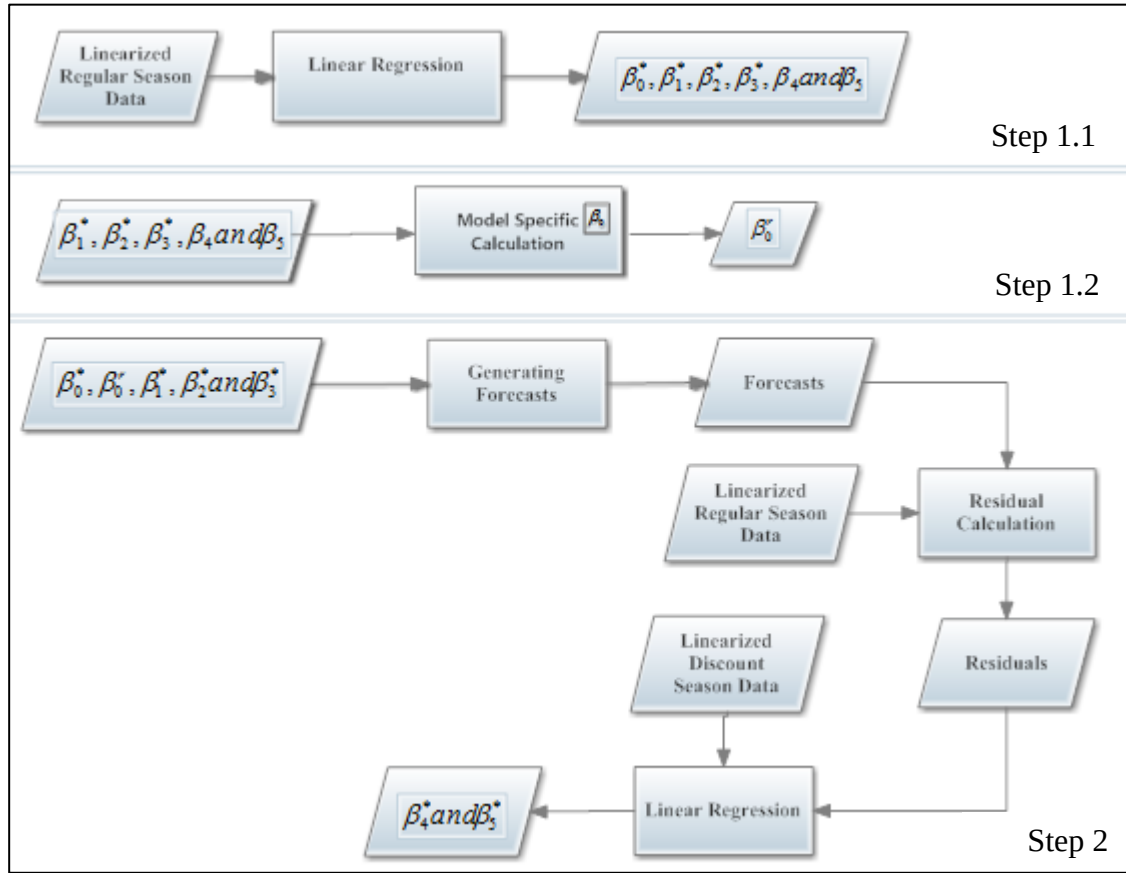


Figure 2.2: Two-Step Procedure for Calculating Linear Regression Parameters by Model CG

First, all parameters β_0 through β_5 are estimated using the regular season data. Afterwards, an estimation $(\hat{\lambda}_r^t)$ ignoring broken assortment and price percentage variables are calculated using the coefficients $\beta_0, \beta_1, \beta_2$ and β_3 (Equation 2.12). However, in this step the “previous period sales” term which is multiplied by β_3 is taken as the mean of the regular season sales for each product-model $(\overline{\lambda}_r^{RS})$.

$\overline{\lambda}_r^{RS}$: Mean sales of product-model r during regular season

The reason for this change is that, in the discount period due to the price cuts there may be unexpected fluctuations in sales from week to week and this is not desired. Next step is, calculating residual values which mean the differences between the realized sales amount and the estimations which are calculated with four variables (Equation 2.13).

$$\widehat{\lambda}_r^t = e^{(\beta_{0r})} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 LN(Age_r^t))} * e^{(\beta_3 LN(\overline{\lambda}_r^{RS}))} \quad (2.12)$$

$$\varphi_r^t = \lambda_r^t - \widehat{\lambda}_r^t \quad (2.13)$$

Finally, we run a linear regression on these residual values considering only broken assortment and price percentage effects. For this run, the discount season data is used. With the run, the calculation of the six variables in total is completed in two steps (Equation 2.14).

$$\varphi_r^w = e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{f * C_r * S_r})))} * e^{(\beta_5 LN(\frac{P_r^w}{P_r}))} \quad (2.14)$$

2.2.2 Caro-Gallien Model with Reference Price (CGwRP)

In this model, we aim to focus more on the effect of *Price Percentage* and build a more reactive model. In order to better see the effect of *Price Percentage* there are two different applications from the model CG. The first one is, instead of using the mean of the regular season sales for the *Sales Amount of the Previous Week* parameter while calculating residual values, the 4-week moving average of sales is used (Equation 2.15). Secondly, instead of using price percentage a different price tracking procedure is used. The new procedure aims to catch only the significant changes in price so that it takes the proportion of the average sales price to the moving average of the average price of the last two weeks (Equation 2.17). In this model, the β parameters are again calculated with the two-step procedure as was done in the model CG.

$$\hat{\lambda}_r^t = e^{(\beta_{0r})} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 (Age_r^t))} \quad (2.15)$$

$$* e^{\left(\beta_3 LN \left(\lambda_r^{t-1} + \lambda_r^{t-2} + \lambda_r^{t-3} + \lambda_r^{t-4} / 4 \right) \right)}$$

$$\varphi_r^w = \lambda_r^w - \lambda_r^t \quad (2.16)$$

$$\varphi_r^t = e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{(P_r^{t-1} + P_r^{t-2})/2}))} \quad (2.17)$$

2.2.3 Forecasting from Decomposed Data (FDD)

In this model, our aim is observing the effect of price and broken assortment without the calendar effects. With this aim, we normalize data in two steps. Firstly, we need to know the value of sales weights of the periods throughout the whole selling season. However, the important issue here is, we want to calculate the value of sales weights without the effect of price. Thus, before calculating weights of the weeks we eliminate the effect of price. Afterwards, we normalize the data using sales weights. Finally we obtain decomposed data to calculate targeted explanatory variables.

This model estimates the effect of broken assortment and price from the decomposed data. The seasonality is calculated from the data which is isolated from the effect of price and the parameters are estimated using only the discount season data.

The first step in decomposing is eliminating the price effect from the data. For this process we need to know the full price and average sales price of a product-model in a week and also determine a value for the price elasticity. Let's say the proportion of the average sales price of product-model r in week t to its full price is 0.5, in other words 50% discount and the price elasticity is -2. The price elasticity is our estimate for β_5 which is demonstrated by $\widehat{\beta}_5$. If the recorded sales amount of model r in week t is 40 then using the formula $\left(\frac{P_r^t}{FullPrice_r}\right)^{\widehat{\beta}_5}$ the sales without the price cut would be one fourth of the recorded one. The price effect is eliminated by Equation 2.18.

$$\bar{\lambda}_r^t = \lambda_r^t / \left(\frac{P_r^t}{FullPrice_r} \right)^{\beta_5} \quad (2.18)$$

After eliminating the price effect using this procedure the seasonality for each week is calculated using the whole season data. In other words, the sales data with the price correction is aggregated to weeks and the sales weight of each week is calculated. The sales weight of each week takes values between 0 and 1 (Equation 2.19).

WeekEffect^t: Sales weight of week t

$$WeekEffect^t = \frac{\sum_r \bar{\lambda}_r^t}{\sum_{t,r} \bar{\lambda}_r^t} \quad (2.19)$$

Afterwards, the real sales records are corrected with these weights. For instance, let's say the 1st week comprises the 20% of the total sales, and then all the sales occurred in the 1st week were multiplied by 5 and divided by the total number of weeks in our one season. With this procedure, the seasonality, which is calculated from the data with price correction, is eliminated.

We examined the result with the same sample data which is used in Figure 2.1. As it can be seen in Figure 2.3 the sales data exhibits desired increase from the week that the major price cut occurs. Eventually, we expect to observe a more reliable price effect coefficient.

$$\bar{\lambda}_r^t = (\lambda_r^t / WeekEffect^t) / \text{Number of Weeks Throughout the Season} \quad (2.20)$$

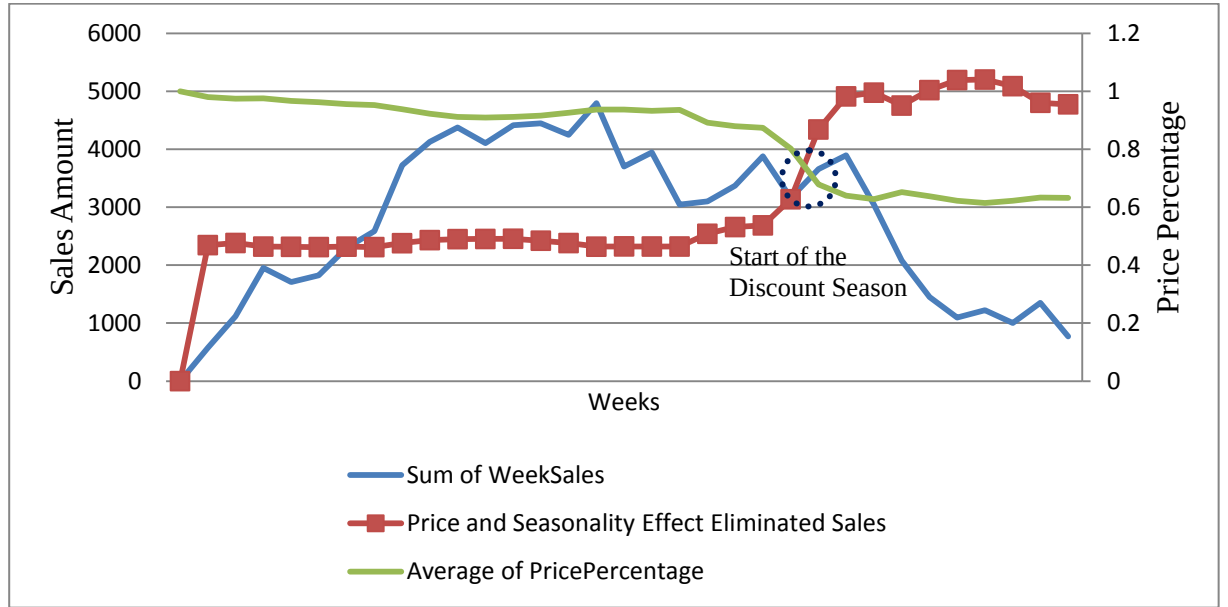


Figure 2.3: Price Percentage, Real and Decomposed Sales Data

After the price effect and seasonality are eliminated from the data, the parameters for the broken assortment and price percentage were calculated with linear regression using only the discount period data.

$$\tilde{\lambda}_r^t = e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))} \quad (2.21)$$

2.3 Performance Criteria

While evaluating the performance of these three models our first attempt is comparing the R-square values of the models. In order to do this comparison, we needed to calculate the R-square values by hand because of the use of the two-step procedures in the first two models. However, it was found that the sum of SSR and SST values which are used in the calculation of R-square is not equal to 1 by contrast with the expectations. It is believed that the two-step procedure does not meet the linearity requirement of R-square calculation.

For this reason, the mean absolute deviation and mean deviation are used as the performance criteria for the comparison of the success of models. These two values are calculated only for the discount season. The reason is that, β_4 and β_5 values work better for the discount season and it is more appropriate to test their performance on discount season data.

$\overline{\lambda_r^{DS}}$: Mean of all sales throughout the discount season

$$Mean\ Absolute\ Deviation(MAD) = \left(\sum_{r,t} |\lambda_r^t - \lambda_r^{t_{Forecast}}| / R * T \right) / \overline{\lambda_r^{DS}} \quad (2.22)$$

$$Mean\ Deviation(MD) = \left(\sum_{r,t} (\lambda_r^t - \lambda_r^{t_{Forecast}}) / R * T \right) / \overline{\lambda_r^{DS}} \quad (2.23)$$

2.4 Bias Correction Methods

In order to linearize the very first forecasting equation (Equation 2.6) we did take the natural logarithm of the dependent variable and independent variables. When applying reverse transformation from the natural logarithm environment to original values, as stated in the paper of Miller (1984) “it is a common problem that the de-transformed fitted linear

model sometimes produces a severely biased model”. With the aim of preventing this bias, we found two alternative bias correction methods in literature.

2.4.1 Miller Method

The first alternative is from the article of Miller (1984). Miller (1984) suggests preventing this bias by multiplying the de-transformed estimations by a term $e^{\hat{\theta}^2/2}$. This term calculated by taking the exponential of the squared standard error of residuals divided by two (Equation 2.24). According to Don M. Miller although multiplying the detransformed estimation with adjustment factor does not eliminate the bias in the model completely, it eliminates the major part of bias.

$$\lambda_r^{t^{Forecast}} = e^{\beta_{0r}} * e^{(\beta_1 LN(Purchase_r))} * e^{(\beta_2 * (Age_r^W))} * e^{(\beta_3 LN(\lambda_r^{W-1}))} * e^{(\beta_4 LN(\min(1, \frac{InventoryLevel_r^t}{Threshold_r})))} * e^{(\beta_5 LN(\frac{P_r^t}{FullPrice_r}))} * e^{\hat{\theta}^2/2} \quad (2.24)$$

2.4.2 Smearing Method

The second alternative is applying smearing estimate method which is proposed in N. Duan’s article (Duan, 1983). To explain this method with an example, let’s take the estimation generated by sales forecasting model of product-model r' for week t' in the natural logarithm form as $\lambda_{r'}^{t'}$. In order to obtain the smearing estimate of model r' in week t' , we add all residual values (Equation 2.25) in the data set to $\lambda_{r'}^{t'}$ one by one and take the mean of the exponential values of these summations. This mean value is the smearing estimate (*Smearing Estimate* $\lambda_{r'}^{t'}$) of product-model r' in week t' . The same calculation is applied to all models and weeks in the data set using all residual values (Equation 2.26). By this model, we distribute the total deviation from the real sales to our estimates.

$$Residual_r^t = \lambda_r^t - \lambda_r^{t, Forecast} \quad (2.25)$$

$$Smearing\ Estimate_{r'}^{t'} = \frac{1}{R * T} \sum_{r,t} \exp (\lambda_{r'}^{t'} + Residual_r^t) \quad (2.26)$$

Chapter 3

Revenue Optimization Model

3.1 Aim of the Model

The aim of the price optimization model is to make the best pricing decisions for a fast-fashion retailer during the discount period so that the total revenue is maximized at the end of the season. Since the project is planned as an applied study, we keep our model environment and assumptions as close as possible to real life conditions.

In our optimization model the discount period is divided into equal intervals and the products are aggregated into product-model-based groups. These groups ignore the size and color differentiation between the products which have the same model code. The price decisions are made for each time interval and each group of products. Moreover, the discounts applied can take values from a discrete and finite set of numbers changing between 0 and 1.

The optimization model gets the following product-model based parameters as input and by solving a mixed-integer optimization problem determines the optimal price and expected sales amount at that price for each period and each product-model throughout the discount season.

Input list:

- Beginning inventory at the beginning of the discount season;
- Full price of the product-model;

- Threshold value of the product-model which is the multiplication of the number of stores across the country where the product-model is sold, the number of sizes and the number of colors that the product is available;
- Forecasted sales table for each discount value and interval throughout the discount season. The forecast table is generated by the forecasting model. The forecasts ignore the broken assortment effect since this effect is included in the optimization model. The reason is that, in the optimization model optimal discount amount will be determined and this decision will affect the expected sales amount and thus next period's inventory level. The changed inventory level will change the *Broken Assortment Effect*. The effect of *Broken Assortment* is calculated by the formula: $\min(1, \frac{InventoryLevel_r^t}{Threshold_r})$. However, this formula cannot be incorporated in our linear optimization model since it is a nonlinear term. Thus, we use the solution of a quadratic formula which minimizes the distance between the non-linear and linear equations.
- The amount of discount applied in real life just before the start of discount season as a lower constraint for the beginning interval of discount season;
- Maximum expected inventory level at the end of the discount season;
- A restrictive factor (U_c) to calculation of expected sales at optimal price. The effect of broken assortment on expected sales is represented by U_c . This factor is calculated using the coefficient of broken assortment effect from the Forecasting Model. The calculation is based on the close form solution to the problem of minimizing the distance between the functional forms for the forecasting and optimization Models. (Caro & Gallien, 2008)

3.2 Optimization Model

Indices and Index Sets:

k: Discount amount index

$$k \in \{1, \dots, K\}$$

r: Product-Models

$$r \in \{1, \dots, R\}$$

w: Discount season periods

$$w \in \{1, \dots, W\}$$

Parameters:

Inv0_r: Inventory level of product-model *r* at the beginning of the discount season

$$Inv0_r = \left(\sum_1^{W-1} \lambda_r^w \right) + Inventory_r^W \text{ for } \forall r \quad (3.1)$$

λ_r^w : The sales amount of product-model *r* in period *w* from the real data

$Inventory_r^W$: The inventory level of product-model *r* at the beginning of the last period of discount season from the real data

d_k: Discount amount, as index increases the discount amount decreases

$$d_1 > d_2 > \dots > d_k > d_{k+1} > \dots > d_{K-1} > d_K$$

$$0 \leq d_K \leq 1$$

DiscBound_r: Discount constraint for the beginning of discount season. Since there may be small discounts applied in real life just before the start of the discount season, this applied discount amount is used as a lower bound for the beginning period of the discount season

SuccDiscCons: Maximum allowed discount amount change between two successive weeks

FullPrice_r: Full price of product-model r

Threshold_r: Inventory threshold value for product-model r

$$Threshold_r = \#Stores_r * \#Colors_r * \#Sizes_r \text{ for } \forall r \quad (3.2)$$

#Stores_r: The number of stores across the country where the product-model r is sold

#Colors_r: The number of colors that product-model r is available

#Sizes_r: The number of sizes that product-model r is available

F_r: Forecasted sales table of product-model r for each discount value and period throughout the discount season. The forecast table is generated by the forecasting model. The forecasts ignore the broken assortment effect since this effect is included in the optimization model

$$F_r = e^{\beta_0 r} * e^{\beta_1 * Purchase_r} * e^{\beta_2 * Age_r^w} * e^{\beta_3 * \overline{\lambda_r^{RS}}} * e^{\beta_4 * (1 - d_k)} \text{ for } \forall w, \forall k \quad (3.3)$$

f_{r,k}^w: Forecasted sales amount for product-model r at period w and discount k from the forecasting table F_r

MaxFinInv_r: Maximum expected inventory level at the end of the discount season which is computed by the purchase amount and a positive ratio less than 1

$$MaxFinInv_r = ExpectedRatio * Purchase_r \text{ for } \forall r \quad (3.4)$$

$$ExpectedRatio \in (0,1)$$

$$Purchase_r = \lambda_r^{RS} + \left(\sum_{w=1}^{W-1} \lambda_r^w \right) + Inventory_r^W \text{ for } \forall r \quad (3.5)$$

λ_r^{RS} : Sum of sales of product-model r during the regular season

U_c : The adjustment factor capturing broken assortment effect for expected sales from the forecasting model. This parameter converts broken assortment effect into a sensitivity parameter on expected sales. The parameter is calculated directly from the β_4 parameter based on the close form solution to the minimization of the distance between the functional forms for the Forecasting Model and the Optimization Model.

$$U_c = \frac{3*\beta_4^2 + 9*\beta_4}{2*\beta_4^2 + 6*\beta_4 + 4} \quad (3.6)$$

Decision Variables:

$x_{rk}^w \in \{0, 1\}$: Whether product-model r should be sold with discount amount d_k or a larger discount amount during period w

$y_{rk}^w \in \{0, 1\}$: Whether product-model r should be sold at discount amount d_k during period w

$Sales_{rk}^w$: Expected sales amount of product-model r at discount amount d_k at period w

Objective Function:

$$\max \left(\sum_r \sum_w \sum_k Sales_{rk}^w * (1 - d_k) * FullPrice_r \right) \quad (3.7)$$

Constraints:

BegDiscCons_r: The beginning period's discount amount should be larger than or equal to the beginning period discount constraint

$$\sum_k y_{rk}^1 \geq DiscCons_r \text{ for } \forall r \quad (3.8)$$

SuccDiscCons^w: The amount of discounts between two successive weeks should be lower than or equal to the determined amount SuccDiscCons

$$\sum_k y_{rk}^1 * d_k - DiscCons_r \leq SuccDiscCons \text{ for } \forall r \quad (3.9)$$

$$\sum_k y_{rk}^w * d_k - \sum_k y_{rk}^{w-1} * d_k \leq SuccDiscCons \text{ for } w \in \{2, \dots, W\}, \forall r \quad (3.10)$$

SuccPeriodDiscCons^w: The price cannot increase as periods proceed through the discount season. If discount amount is lower than or equal to d_k at one period, the discount amount should be larger than or equal to d_k during the following week too.

$$x_{rk}^{w-1} \leq x_{rk}^w \text{ for } w \in \{2, \dots, W\}, \forall r, \forall k \quad (3.11)$$

SamePeriodDiscInc^w: If discount amount d_k is active in period w for product-model r then all the x variables until x_{rk-1}^w is active and take the value “1”

$$x_{rk-1}^w \leq x_{rk}^w \text{ for } \forall w, \forall r, k \in \{2, \dots, K\} \quad (3.12)$$

y_x_relation^w: Ensures the relation between y and x . When the x_{rk}^w variable is active and takes the value “1” it ensures a discount larger than or equal to d_k for period w and product-model r . If the next x variable (x_{rk-1}^w) is active too and takes the value “1”, y_{rk}^w takes value “0”. When the x variable stops being active for the same period and product-model, y variable gets active and determines the discount amount for that period and product-model.

$$y_{rk}^w = x_{rk}^w - x_{rk-1}^w \text{ for } \forall w, \forall r, k \in \{2, \dots, K\} \quad (3.13)$$

$$y_{r1}^w = x_{r1}^w \text{ for } \forall w, \forall r \quad (3.14)$$

OneDiscPerPeriod^w_r: Ensures only one discount is active for each product-model per discount period.

$$\sum_k y_{rk}^w = 1, \text{ for } \forall w, \forall r \quad (3.15)$$

Beg_Inv^w_r: Ensures the beginning inventory of the first period is equal the parameter $Inv0_r$

I_r^w : Inventory amount of product-model r at the beginning of period w

$$Inv_r^1 = Inv0_r \text{ for } \forall r \quad (3.16)$$

PeriodicInv^w_r: Calculates each periods beginning inventory level by getting previous period's inventory amount and previous week's sales amount of each product-model

$$Inv_r^w = Inv_r^{w-1} - \sum_k Sales_{rk}^{w-1} \text{ for } w \in \{2, \dots, W\}, \forall r \quad (3.17)$$

FinInv_r: Keeps track of final inventory at the end of the discount season for each product-model

$FinalInv_r$: Inventory amount of product-model r at the end of the season

$$Final_Inv_r = Inv_r^W - \sum_k Sales_{rk}^W \text{ for } \forall r \quad (3.18)$$

FinInvBond_r: Ensures the final inventory level is lower than or equal to the determined final inventory bond $MaxFinInv_r$

$$Final_Inv_r \leq MaxFinInv_r \text{ for } \forall r \quad (3.19)$$

Sales1_{rk}^w: The expected sales amount is constrained by the forecast of active discount amount at period w for product-model r.

$$Sales_{rk}^w \leq f_{rk} * y_{rk}^w \text{ for } \forall w, \forall r, \forall k \quad (3.20)$$

Sales2_{rk}^w: The expected sales amount is constrained also by the broken assortment effect which is determined by the current inventory level at period w, threshold value of the product-model r and the U_c value

$$Sales_{rk}^w \leq f_{rk} * \left(1 - U_c + U_c * \frac{Inv_r^w}{Threshold_r}\right) \text{ for } \forall w, \forall r, \forall k \quad (3.21)$$

NonnegativeSales_{rk}^w: The sales variable cannot take negative values

$$Sales_{rk}^w \geq 0 \text{ for } \forall w, \forall r, \forall k \quad (3.22)$$

NonnegativeInv_r^w: The inventory cannot take negative values

$$Inv_r^w \geq 0 \text{ for } \forall w, \forall r \quad (3.23)$$

NonnegativeFinInv_r: The final inventory levels cannot take negative values.

$$Final_Inv_r \geq 0 \text{ for } \forall r \quad (3.24)$$

Outputs:

OptimalDiscount_r^w: The optimal discount amount for product-model r in period w

$$d_r^{w*} = \sum_k y_{rk}^w * d_k \quad (3.25)$$

OptimalSales_r^w: The amount of expected sales at the optimal discount for product-model r in week w

We incorporate the company's pricing policies to our optimization model formulation. The number of price changes is not restricted; however, the amount of price change between two successive weeks is restricted and sharp decreases in price are not allowed. The ending inventory can be forced to be less than or equal to some portion of purchase amount thus the aimed ending inventory level can be incorporated into the model. Moreover, the effect of changing inventory level affects the expected sales and thus price decisions so that, the effects of optimal price decisions on expected sales are better reflected and the model produces more realistic results.

Chapter 4

Implementation

4.1 Data

With the purpose of developing a forecasting model, a dataset was obtained from a leading fashion retailer which has 150 stores in Turkey and introduces approximately 5000 different products to the market each year with 500 million TL annual turnover. The data is crucial for generating models, methods and solutions which are applicable in practice. We planned to produce our forecasts for every *product-model* which is the basic code for every design on sale regardless of color and size and for every week. The details of the data are explained in the next section.

The data has the weekly information between the 4th – 40th weeks of 2011 on SKU level; in other words, from the end of January to the beginning of October. This time interval corresponds to spring-summer sales season for the firm. The total data consists of 142 columns and 45827 rows.

The sample screenshots from the available data can be seen in Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4.

	A	B	C	D	E	F	G	H	I
1	Model Kodu	Renk	SKU	durum	Beden	UrUn Grubu	Marka	Kategori	Stok Grubu
2	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl
3	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl
4	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl
5	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl
6	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl
7	1PIMG01C473	KAHVE	1PIMG01C473KAHVE		36	Merdane	Oxide	Casual	Ayakkabl

Figure 4.1: Screenshot from the Data

	J	K	L	M	N	O	P	Q
1	Beden Grubu	Alt Kategori	Saya malzemesi	Uretici	Tedarik	Kavala	Fiyat	Depo Tipi
2	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	BEKLEME DEPOLARI
3	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	MERKEZ STOK DEPO
4	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	MERKEZ-MAGAZALAR
5	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	MERKEZ-MAGAZALAR
6	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	MERKEZ-MAGAZALAR
7	Garson	0 DERI		NazIm Can	Kesin Allm - IC Piyasa	5	59.99	MERKEZ-MAGAZALAR

Figure 4.2: Screenshot from the Data

	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	MaGaza	satis 4	satis TL 4	stok 4	satis 5	satis TL 5	stok 5	satis 6	satis TL 6	stok 6	satis 7	satis TL 7	stok 7
2				3			1			1			1
3				-3			-27			-27			1
4	212 CARREFOURSA						1			1			1
5	ANKAMALL						1			1			1
6	ANKARA IZMIR CD.						1			1			1
7	ANKARA OPTIMUM						1			1	1	59.91	

Figure 4.3: Screenshot From the Data

	EE	EF	EG	EH	EI	EJ	EK	EL
1	satis 39	Full Price 39	satis TL 39	stok 39	satis 40	Full Price 40	satis TL 40	stok 40
2	0	0	0		0	0	0	
3	0	0	0		0	0	0	
4	0	0	0		0	0	0	
5	0	0	0	1	1	59.99	27.77	
6	0	0	0		0	0	0	
7	0	0	0		0	0	0	

Figure 4.4: Screenshot from the Data

The details of the column contents are as follow:

(A) Model Code: The unique code of a model regardless of its color/size. 112 different models exist in the data set.

(B) Color: Indicates the color of the entry. 20 different colors are available in total; however, every model has a different color set.

- (C) SKU: Stock keeping unit of the entry. It includes both the model number and the color information of the entry. There are 216 SKUs in this data set.
- (D) Durum: Not used in the analysis.
- (E) Size: Indicates the size information of the entry. The size range in total is from 36 to 47; however, every model has a different size set.
- (F) Product Group: Indicates the product group that the entry belongs to.
- (G) Brand: Indicates the brand of the entry. In this data set only one brand is available as "Oxide".
- (H) Category: Indicates the category that the entry belongs to. In this data set only one category is available as "Casual".
- (I) Stock Group: Indicates the stock group that the entry belongs to. In this data set only one stock group is available as "Shoe".
- (J) Size Group: Indicates the size group that the entry belongs to. In this data set two size groups are available as "Men" and "Lad".
- (K) Sub Category: Not used in the analysis.
- (L) Upper-Shoe Material: Indicates the material that forms the shoe except the footbed. In this data set there are four leather kinds used which are "leather", "split leather", "patent leather", "leather (suede)".
- (M) Producer: Indicates the supplier of the product. There are 15 different suppliers.
- (N) Supply: Indicates the market that the product is bought from. All products in this data set are from domestic market.
- (O) Kavala: Indicates the value of the price of the model in modulus 10.
- (P) Price: Indicates the full price of the product. The price range for the products in the data set is from 49.99 to 119.99.

- (Q) Warehouse Type: Indicates the warehouse type that the product is kept. There are three different warehouses which are “waiting warehouses”, “central warehouse” and “central stores”.
- (R) Store: Indicates the store that the product is kept or sold. In this data set 100 stores are available.
- (S) Sales w: Indicates the amount sold of the mentioned SKU in the mentioned store in week w.
- (T) Sales TL w: Indicates the revenue gained from the “Sales w”.
- (U) Stock w: Indicates the total amount of stock of the mentioned SKU in the mentioned store in week w.

Before starting to analyze data, some product-models which we think that will not add any meaningful information to our analysis were eliminated. The maximum inventory position during the season was the first determiner.

To begin with, the SKUs whose maximum inventory levels in the season are less than 100 were removed from the data and this removal dropped the number of SKUs from 216 to 172. Afterwards, the maximum inventory levels were screened on a higher level of detail. This time the product-models were ordered according to their maximum inventory positions and the ones which have less than 50 units of maximum inventory during the season were eliminated. Consequently total number of product-models dropped to 97.

Second determiner was the number of stores where the product-model is sold. This time, the product-models, which are on sale in less than 10 stores, were eliminated and the number of product-models decreased by five. Eventually, the analyses were done using the remaining 92 product-models.

By aggregating the data to weeks; the following total sales and price percentage (Figure 4.5) and total stock (Figure 4.6) graphs were obtained. The price percentage was obtained

as the ratio of average sales price over average full price. Average sales price is calculated as the total revenue divided by total number of unit sales.

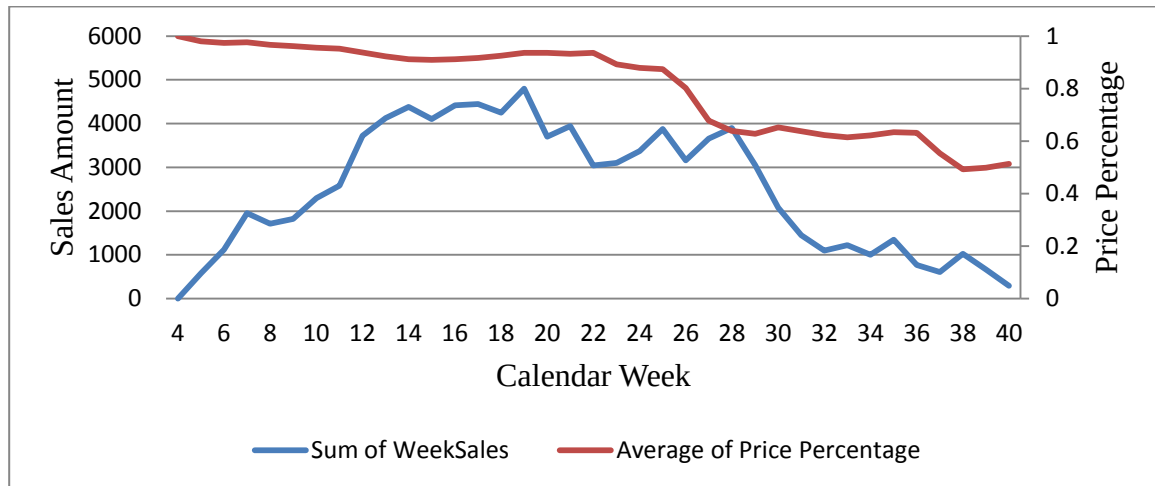


Figure 4.5: Weekly Sales and Average Price Percentage

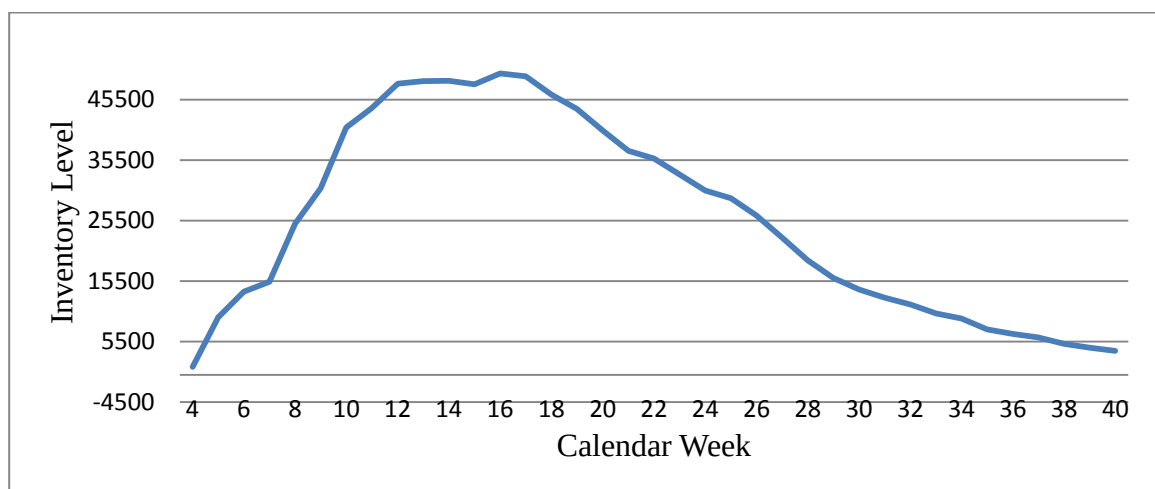


Figure 4.6: Inventory Level Aggregated to Weeks

The company does not have a fixed annual schedule for price cuts. Thus, we had to investigate the starting time of the discount period from the data we have. According to Figure 4.5, 24th week was determined as the beginning of the major discount season since the average sales price over average full price ratio decreases from 80% to 60%. For this reason, the period before week 24 will be called as the regular season, and the period after 24th week to the end of the season will be called as discount season. In the last four weeks of the season a drastic decline in the inventory levels is observed and the stock-outs arising from this decline would prevent us to observe a robust sales-price relationship. Thus, the interval between 37th – 40th weeks was removed from the data. The 4th week was also removed since the total sales volume for all products in this week was zero.

In the final situation, we had the table which includes number of colors, sizes and stores offered, sales amount, inventory level, revenue from the sales and price of sales record information which are aggregated to weeks and 92 product-models from 5th to 36th week of 2011 which corresponds to the timeframe between end of January and beginning of September. After the removals, the time interval of the data still covers the same season with the data and will be sufficient to make healthy deductions about the season characteristics. In addition to this, we keep the majority of the product-models and only the ones which wouldn't add great value to sales information were eliminated. To sum up, after the reductions without losing any valuable information we now have a clear and the most essential part of the data.

4.2 Forecasting Model Application

4.2.1 *Training and Validation Data Sets*

It is crucial for our studies that forecasting model to be applicable to different data sets and cases. In order to test this property of our model we decided to separate the data into two sets which are fit and holdout samples. The parameters of the forecasting model are

estimated using the fit data and the performance of them was measured both on fit and holdout data sets.

In the partitioning process, the first step is sorting the product-models in ascending total sales volume order. If we split the 92 ordered product-models into groups of three, the first product-model of each group is selected for holdout sample and the remaining two belong to the fit sample. In the end, fit set contains 61 and holdout set contains 31 product-models.

4.2.2 Comparison of Forecasting Models' Result

The mean deviation and mean absolute deviation for both Ln and real values of fit and holdout samples from our three models and bias correction results are presented in Table 4.1. The Miller Method did not improve our MD and MAD results, thus we continue with Smearing Estimate Correction. It can be summarized that the performance of “CG Model with Reference Price” competes with “Model CG”’s performance over all models both with fit and holdout samples. Moreover, smearing estimate correction successfully decreased the MD and MAD values, especially in fit sample and MD values in holdout sample.

Table 4.1: MD and MAD Values of Forecasting Models

SalesType	MeasureType	Fit Sample			Holdout Sample		
		CG	CGwRP	FDD	CG	CGwRP	FDD
<i>LN</i>	<i>MAD</i>	19.50%	20.71%	30.87%	23.39%	26.13%	32.16%
<i>LN</i>	<i>MD</i>	0.00%	0.00%	0.00%	-1.40%	-1.43%	-3.44%
<i>Real</i>	<i>MAD</i>	37.42%	38.74%	72.41%	43.32%	43.44%	75.52%
<i>Real</i>	<i>MD</i>	14.83%	8.64%	39.06%	16.67%	10.02%	41.89%
<i>SmearingCorr</i>	<i>MAD</i>	35.10%	42.18%	75.36%	41.20%	46.35%	82.93%
<i>SmearingCorr</i>	<i>MD</i>	-0.54%	-11.21%	-11.29%	1.63%	-9.52%	-3.25%

4.2.3 Best Model's Parameter Results

According to performance comparison performed in Section 4.2.2 Caro & Gallien model is the most successful model according to our performance criteria. The resulting linear regression model parameters are presented in Table 4.2.

Table 4.2: Linear Regression Coefficient Values of the Best Model

Explanatory Variable	Coefficient	Value
Common Intercept	β_0	-1.59631
Purchase	β_1	0.498786
Age	β_2	-0.01793
Previous Week Sales	β_3	0.600418
Broken Assortment Effect	β_4	1.179975
Price Percentage	β_5	-1.23922

The coefficients of explanatory variables are significant according to model results and the values are parallel to our expectations mentioned in Section 2.3.1.1. *Purchase*, *Previous Week Sales* and *Broken Assortment Effect* have positive correlation with sales forecasts whereas *Age* and *Price Percentage* seem to affect predictions negatively.

4.3 Clustering

While studying with the forecasting and simulation models we used all the available sales information of all product-models without making any clustering. However, we want to identify whether customers of different stores have similar shopping behaviors from each other and the sales patterns of stores are parallel or not. If the shopping behaviors are dissimilar, this means the sales patterns' responses to the changes in sales forecasting model parameters will be different in different stores.

With the purpose of identifying these possible differences we summed sales to stores. The sales amounts of discount season (*Discount Season Sales (DSS)*, between 24th - 36th weeks) and regular season (*Regular Season Sales (RSS)*, between 4th - 23rd weeks) are summed separately for each store. Afterwards, stores' ratios of discount season sales to regular season sales are calculated and sorted from largest to smallest. We call this ratio as discount sensitivity.

The sorted store list is divided into two. The first 48 stores in the list are called "high score stores" and expected to be more sensitive to price changes than the remaining 47 stores.

The store list can be seen in Table 4.3. Store names are changed with representative letters due to privacy issues.

Table 4.3: Store Lists

Store	DSS/RSS	Store	DSS/RSS	Store	DSS/RSS	Store	DSS/RSS
HS1	2.011	HS25	0.624	LS1	0.497	LS25	0.371
HS2	1.552	HS26	0.623	LS2	0.494	LS26	0.370
HS3	1.143	HS27	0.620	LS3	0.493	LS27	0.366
HS4	1.087	HS28	0.618	LS4	0.489	LS28	0.364
HS5	1.010	HS29	0.616	LS5	0.484	LS29	0.359
HS6	1.000	HS30	0.614	LS6	0.481	LS30	0.357
HS7	0.981	HS31	0.597	LS7	0.478	LS31	0.351
HS8	0.959	HS32	0.589	LS8	0.473	LS32	0.351
HS9	0.957	HS33	0.584	LS9	0.469	LS33	0.348
HS10	0.928	HS34	0.580	LS10	0.462	LS34	0.346
HS11	0.903	HS35	0.576	LS11	0.459	LS35	0.338
HS12	0.902	HS36	0.575	LS12	0.456	LS36	0.332
HS13	0.839	HS37	0.574	LS13	0.453	LS37	0.330
HS14	0.831	HS38	0.555	LS14	0.448	LS38	0.322
HS15	0.724	HS39	0.550	LS15	0.441	LS39	0.312
HS16	0.722	HS40	0.548	LS16	0.437	LS40	0.305
HS17	0.706	HS41	0.544	LS17	0.432	LS41	0.303
HS18	0.700	HS42	0.541	LS18	0.417	LS42	0.302
HS19	0.690	HS43	0.529	LS19	0.410	LS43	0.299
HS20	0.689	HS44	0.521	LS20	0.400	LS44	0.295
HS21	0.643	HS45	0.519	LS21	0.390	LS45	0.293
HS22	0.642	HS46	0.499	LS22	0.386	LS46	0.285
HS23	0.634	HS47	0.497	LS23	0.381	LS47	0.271
HS24	0.624			LS24	0.374	LS48	0.243

After identifying stores' sensitivity to discount season, the data is divided into two. One part of the data has only the data of high score stores and the other part has the data of low score stores. The forecasting model, which is explained in Section 3.2, is applied to two data sets separately and two beta sets together with mean and mean absolute deviation values are obtained for different store groups.

Mean deviation and mean absolute deviation values for two store sets can be seen in Table 4.4. The deviation values are similar to the results obtained from the analysis done for all stores and data samples.

Table 4.4: MD and MAD Values of Two Store Sets

SalesType	Measure Type	High Score Stores		Low Score Stores	
		Fit	Holdout	Fit	Holdout
<i>LN</i>	<i>MAD</i>	27.63%	34.52%	35.07%	37.22%
LN	MD	0.00%	-2.41%	0.00%	-2.83%
<i>Real</i>	<i>MAD</i>	38.59%	43.83%	49.21%	50.84%
Real	MD	16.80%	21.04%	15.89%	19.07%
<i>SmearingCorr</i>	<i>MAD</i>	37.19%	41.43%	48.48%	49.70%
SmearingCorr	MD	-1.85%	3.34%	-6.94%	-2.91%

4.4 Simulation Application

4.4.1 Aim of Simulation Application

The constructed optimization model gives the price decisions of all periods of clearance season in one go. In order to convert this optimization model to a dynamic one, a simulation application is designed.

By the simulation application, instead of deciding all price levels of clearance season at point 0 only the very first period's price decision from the optimization model is used. The forecast at that discount amount for the current period is adjusted according to the real sales amount of that period, inventory level is updated with that adjusted forecast and forecast table is adjusted according to the difference between forecast and real sales. Afterwards, optimization model is run for a one-period shorter horizon with the updated parameters.

4.4.2 Simulation Application Steps

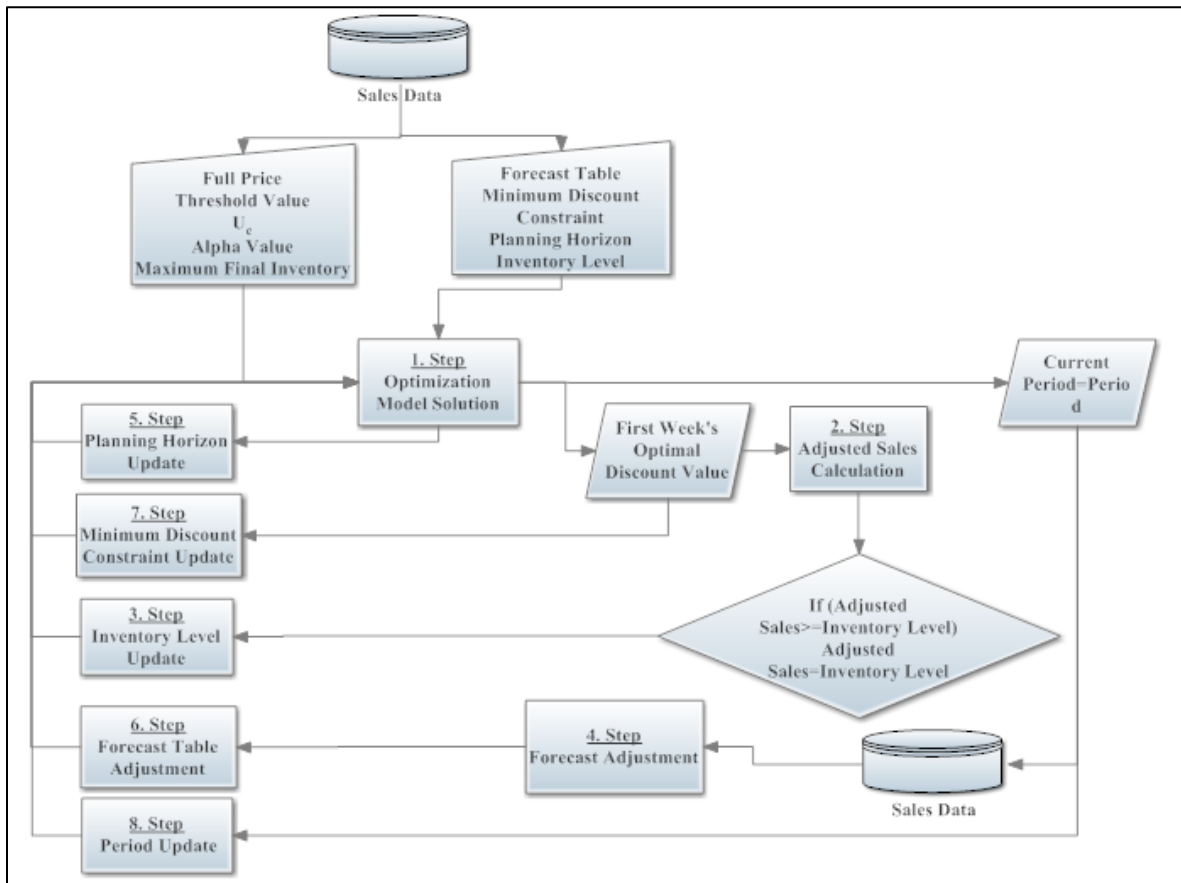


Figure 4.7: Flowchart of Simulation Application

Simulation model works for each product-model separately. Two groups of inputs are given to the optimization model: Constant inputs and dynamic inputs.

Constant Inputs:

- $FullPrice_r$
- $Threshold_r$
- U_c
- α : The update weight for the forecasting table

Dynamic Inputs:

- F_r
- $DiscBound_r$
- $Horizon_r^w$: Length of planning horizon
- *Current Period*: The variable to keep track of the real time location of the product-model
- I_r^w

1. Step: Solving Optimization Model

Optimization model which was explained in Section 3.2 gets the mentioned inputs and produce 16-week price decision set. The results are written on an excel file and the file is read by R. R takes only the beginning week's optimal price value (d_r^{1*}) and proceeds to step 2.

2. Step: Calculating Adjusted Sales Amount

The first period of the discount season is 24th week of the year. So, initially R looks for the amount of sales in week 24 for a product-model. Adjusted sales amount is calculated in order to keep close to the real life sales. We have a reliable estimation about the effect of change in inventory level (b_4) and price (b_5). The optimal discount amount is chosen by the optimization model and we calculate what would be the amount of sales at that price and inventory level. The calculated sales amount

$Adjusted\ Sales_r^w$ is accepted as the realized sales amount with the chosen discount level.

$$Adjusted\ Sales_r^w = Sales_r^w * InventoryCorrection_r^w * PriceCorrection_r^w \quad (4.1)$$

$$InventoryCorrection_r^w = \left[\min\left(1, \frac{I_r^w}{Threshold_r}\right) / \min\left(1, \frac{InventoryLevel_r^w}{Threshold_r}\right) \right]^{b_4} \quad (4.2)$$

$$PriceCorrection_r^w = \left[(1 - d_r^{w*}) / (1 - d_r^{w^{real}}) \right]^{b_5} \quad (4.3)$$

$d_r^{w^{real}}$: Real discount amount applied in week w for product-model r

After the calculation, we check whether the value of $Adjusted\ Sales_r^w$ exceeds the present inventory level (I_r^w). If our expectation is larger than we have on hand (I_r^w) for that product-model, then we can sell at most as much as the inventory level I_r^w .

3. Step: Updating the Inventory Level

In the step 1 the $Adjusted\ Sales_r^w$ is calculated and accepted as realized sales amount of product-model r with the optimal discount amount at week w . The next step is updating the inventory level accordingly.

$$I_r^{w+1} = I_r^w - Adjusted\ Sales_r^w \quad (4.4)$$

4. Step: Adjusting Forecast

One of our aims in constructing a dynamic simulation model is to update our sales forecasts considering the deviation from real sales on each cycle. In order to calculate this deviation, we look for the real discount amount applied in the current week ($d_r^{w^{real}}$) and what the closest index in our discount index table to that discount. Afterwards, we find what our forecast for that index ($f_{r,k}^w$) is.

Our forecast table includes forecasts which ignore the broken assortment effect. However, we can calculate the deviation of our forecast from real sales only if we consider the effect of inventory level. For that purpose, we multiply our forecast with the broken assortment equation which is used in the optimization model.

$$Adjusted\ Forecast_r^w = f_{r,k}^w * \left(1 - U_c + U_c * \min \left(1, \frac{I_r^w}{Threshold_r} \right) \right) \quad (4.5)$$

5. Step: Updating Planning Horizon

The optimization model is going to solve the problem starting from week $w + 1$, thus the next planning horizon is going to be one week shorter than the current one.

$$Planning\ Horizon_r^{w+1} = Planning\ Horizon_r^w - 1 \quad (4.6)$$

6. Step: Adjusting Forecast Table

In order to adjust the forecast table for the next week, we determine an alpha (α) value which controls the weight of change due to the deviation of forecast from real sales value in adjustment process. Adjusted forecast table for week w and product-model r is represented as F_r^w .

$$F_r^{w+1} = (1 - \alpha) * F_r^w + \alpha * F_r^w * (Sales_r^w / Adjusted\ Forecast_r^w) \quad (4.7)$$

The next step of adjusting forecast table is excluding the week for which we already calculated optimal discount amount. By excluding the forecast line for week w the dimensions of the forecast table is going to be consistent with the span of *Planning Horizon* $_r^{w+1}$.

$$F_r^{w+1} = F_r^{w+1}[(w + 1): W * K] \quad (4.8)$$

7. Step: Updating Minimum Discount Constraint

At the beginning of the season the minimum discount constraint is the real discount applied just before the start of the discount season. As discount season weeks proceed, the minimum discount constraint is updated as the optimal discount amount of the previous week.

$$DiscBound_r = d_r^{w*} \quad (4.9)$$

8. Step: Updating the Current Week

As weeks proceed throughout the discount season the variable which keeps the current week information is updated. We need this variable to keep in touch with real life and pull the required real data from the database.

$$w = w + 1 \quad (4.10)$$

With the first week of the discount season the simulation process starts by the first step and the optimal discount amount is found for each product-model. The eight steps are completed in each cycle till the last week of the discount season.

4.4.3 Results of Simulation Application

In Figure 4.8 the average price percentage computed for weeks are presented. As can be seen from the table, the simulation application does not propose a significant discount (more than 30% discount) until the last 3 weeks of the discount season.

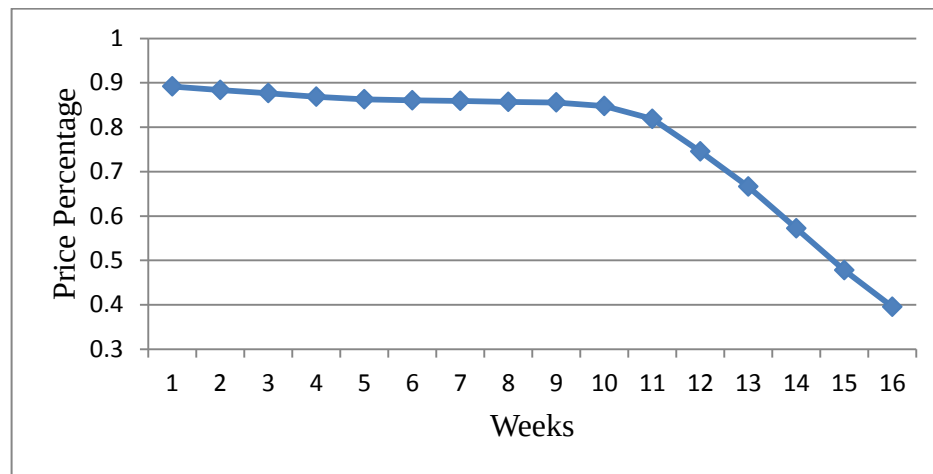


Figure 4.8: Average Price Percentage Amounts to Weeks

Some of the good price policies generated by the model are presented in Figure 4.9. In these examples, one major discount is observed at the beginning of the discount season or two gradual discounts are applied throughout the season.

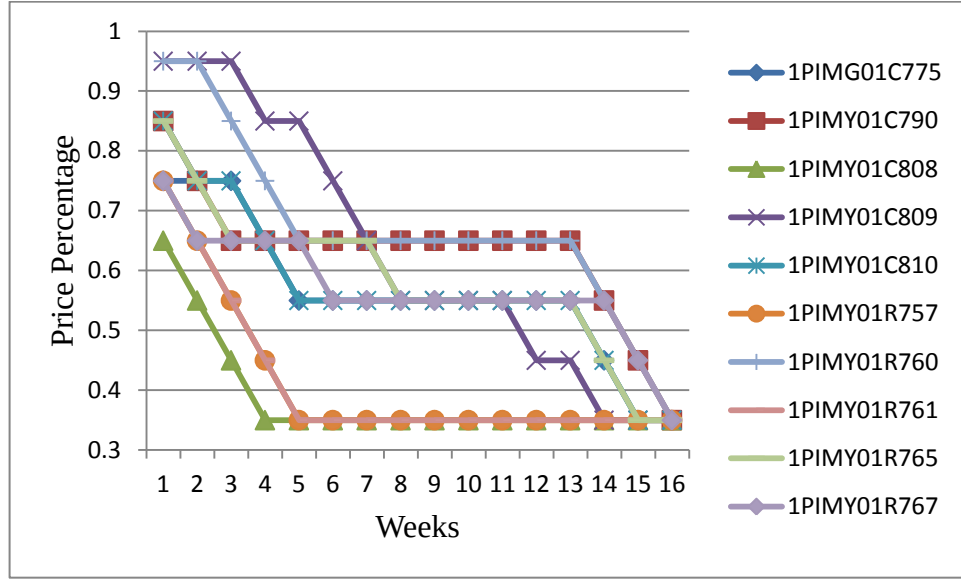


Figure 4.9: Average Price Percentages of Selected Product-Models

In Table 4.5 “Improvement %” column is calculated by the following formulas:

$$Cumulative\ Optimal\ Revenue_r = \sum_{w=1}^w Adjusted\ Sales_r^w * FullPrice_r * (1 - d_r^{w*}) \quad (4.11)$$

$$Real\ Revenue_r = \sum_{w=1}^w Sales_r^w * FullPrice_r * (1 - d_r^{w^{real}}) \quad (4.12)$$

$$Revenue\ Difference_r = Cumulative\ Optimal\ Revenue_r - Real\ Revenue_r \quad (4.13)$$

$$Revenue\ Improvement\ \% = Revenue\ Difference_r / Real\ Revenue_r \quad (4.14)$$

Within the 86 product-models, optimal revenue falls behind the real revenue for only 10 product-models with an average of 4.31%. With these losses the average revenue improvement percentage is 9.74 and without the negative values the average revenue improvement is 11.59%. In both cases, we observe a significant revenue improvement in total.

Table 4.5: Revenue Improvement Percentages of Product-Models

Model#	Improve. %	Model#	Improve. %	Model#	Improve. %
A82	27.81%	A50	12.39%	A2	5.44%
A64	27.66%	A11	12.35%	A18	5.15%
A16	25.53%	A48	11.97%	A3	4.87%
A68	25.33%	A59	11.91%	A21	4.51%
A67	24.64%	A29	11.87%	A65	4.18%
A57	22.92%	A39	11.86%	A34	3.25%
A56	22.36%	A4	11.63%	A44	3.24%
A79	21.54%	A31	11.54%	A19	3.19%
A40	21.28%	A17	11.33%	A45	2.70%
A58	19.99%	A10	11.11%	A5	2.44%
A66	19.91%	A55	11.08%	A8	2.22%
A72	18.77%	A41	11.01%	A80	1.79%
A49	18.72%	A53	10.96%	A70	1.60%
A52	18.09%	A14	10.80%	A84	1.47%
A12	16.31%	A28	10.70%	A71	0.80%
A37	14.94%	A77	10.43%	A47	0.62%
A32	14.41%	A38	10.33%	A86	-0.24%
A73	14.22%	A42	10.09%	A74	-0.48%
A24	13.55%	A62	10.03%	A75	-0.51%
A22	13.54%	A33	9.84%	A63	-0.52%
A54	13.44%	A7	9.71%	A36	-2.50%
A30	13.42%	A23	9.56%	A46	-2.55%
A60	13.22%	A25	9.01%	A13	-3.86%
A15	13.20%	A20	8.76%	A78	-7.01%
A83	12.93%	A43	8.16%	A69	-12.14%
A51	12.83%	A27	7.91%	A81	-13.30%
A26	12.78%	A76	7.64%	Average= 9.74%	
A6	12.77%	A61	6.83%		
A35	12.76%	A1	5.61%		
A9	12.63%	A85	5.46%		

In Table 4.7 the ratio of final inventory levels to the beginning inventory amounts are presented. The displayed values are calculated as follows:

$$\text{Optimal Final Inventory}(OFI) = I_r^W - \text{Adjusted Sales}_r^W \quad (4.15)$$

$$\text{Real Final Inventory}(RFI) = \text{RealInventory}_r^W - \text{Sales}_r^W \quad (4.16)$$

$\text{Real Inventory}_r^W$: Amount of real inventory of product-model r from the sales data in the last week of discount season

$$\text{Beginning Inventory}(BI) = I_r^1 \quad (4.17)$$

$$\text{Optimal Final Inventory \%} = \frac{\text{Optimal Final Inventory}}{\text{Beginning Inventory}} \quad (4.18)$$

$$\text{Real Final Inventory \%} = \frac{\text{Real Final Inventory}}{\text{Beginning Inventory}} \quad (4.19)$$

A drawback observed from these results is our simulation application together with optimization model performs worse than real life in finishing the on hand inventory. For most of the product-models, our model is late in applying a significant discount in comparison by applied discount policy. This case can be seen in Figure 4.10. On the other hand, this late discount ensures the observed revenue improvement. As can be seen in Figure 4.11 optimal and real life revenues go together till the last six weeks of discount season.

Table 4.6: Final Inventory Level Ratios

Model#	O/I/BI	R/I/BI	Model#	O/I/BI	R/I/BI	Model#	O/I/BI	R/I/BI
A1	42.49%	41.86%	A31	11.95%	10.55%	A61	8.23%	5.83%
A2	31.37%	30.59%	A32	7.4%	6.34%	A62	14.35%	9.07%
A3	49.90%	44.05%	A33	23.34%	18.18%	A63	20.71%	10.66%
A4	15.44%	17.11%	A34	28.22%	24.53%	A64	17.29%	8.10%
A5	39.58%	43.81%	A35	24.10%	14.86%	A65	22.76%	8.87%
A6	18.99%	17.43%	A36	22.41%	11.11%	A66	13.69%	6.51%
A7	34.30%	30.86%	A37	9.89%	7.45%	A67	13.28%	5.56%
A8	39.24%	34.07%	A38	12.30%	10.47%	A68	15.58%	10.19%
A9	18.84%	12.01%	A39	14.46%	6.98%	A69	19.26%	38.82%
A10	25.52%	20.60%	A40	14.17%	6.87%	A70	30.90%	36.23%
A11	24.98%	15.61%	A41	5.72%	4.84%	A71	38.36%	41.18%
A12	15.31%	9.62%	A42	14.79%	11.45%	A72	17.38%	9.75%
A13	29.32%	8.64%	A43	7.47%	7.26%	A73	12.43%	7.45%
A14	11.60%	7.53%	A44	22.77%	26.04%	A74	33.75%	38.13%
A15	18.25%	8.97%	A45	23.13%	22.61%	A75	24.04%	23.27%
A16	17.35%	6.67%	A46	28.63%	19.65%	A76	22.19%	20.45%
A17	11.13%	5.91%	A47	20.69%	18.37%	A77	39.45%	25.20%
A18	12.90%	6.06%	A48	17.09%	9.34%	A78	26.05%	47.00%
A19	10.03%	8.91%	A49	6.42%	10.81%	A79	17.60%	10.00%
A20	11.03%	4.55%	A50	22.02%	13.85%	A80	39.70%	32.56%
A21	13.65%	8.49%	A51	18.06%	14.54%	A81	17.64%	34.88%
A22	9.16%	4.53%	A52	16.76%	10.54%	A82	13.91%	8.33%
A23	15.44%	13.71%	A53	11.27%	9.84%	A83	24.68%	21.26%
A24	13.94%	8.26%	A54	15.81%	11.66%	A84	32.70%	38.69%
A25	16.65%	8.41%	A55	20.33%	11.17%	A85	14.20%	12.12%
A26	16.30%	9.04%	A56	15.94%	8.24%	A86	21.70%	21.88%
A27	9.89%	4.75%	A57	14.91%	8.97%	Average	19.47%	15.65%
A28	18.36%	9.48%	A58	15.27%	8.64%			
A29	8.23%	6.60%	A59	10.66%	7.85%			
A30	10.61%	4.68%	A60	11.33%	8.15%			

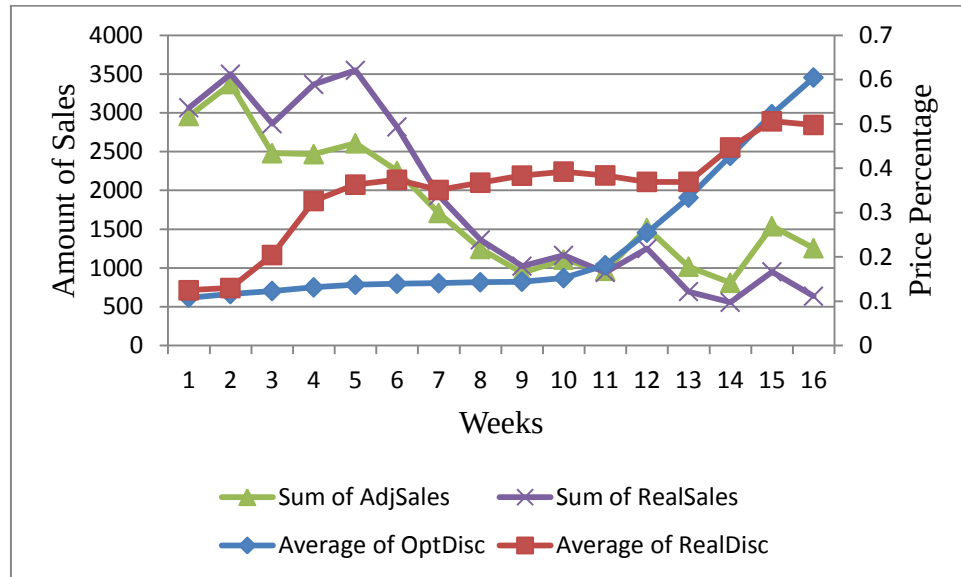


Figure 4.10: Real and Optimal Price Percentage & Sales Values

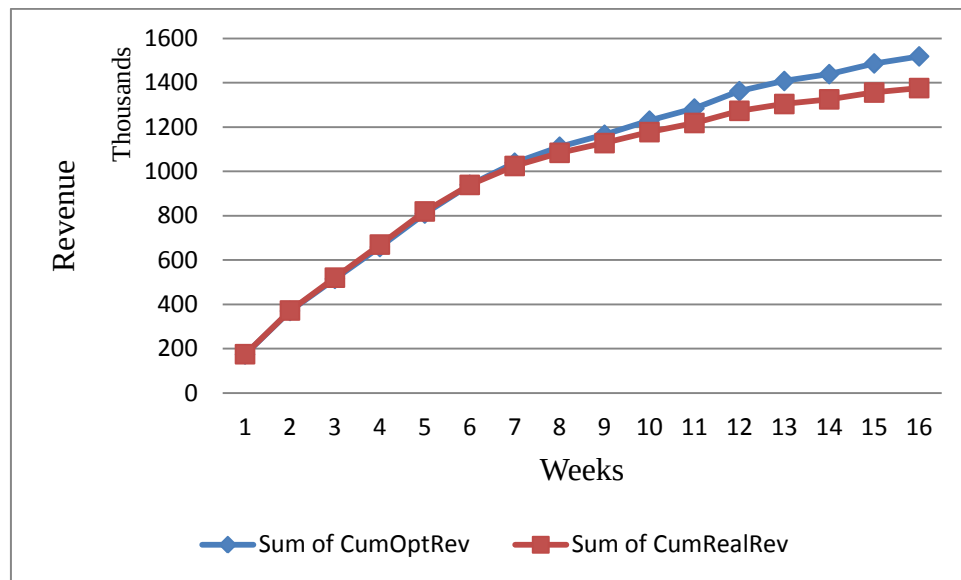


Figure 4.11: Cumulative Real and Optimal Revenues

Limited Ending Inventory Extension

Simulation application results show that ending the use of optimal pricing policies causes higher end-up inventory levels than the current real-life pricing decisions do. In order to reduce inventory on hand at the end of the season, we add an additional constraint to the optimization model. This constraint forces optimization model to have inventory level on hand at the end of the season less than 10% of the purchase amount. The additional constraint is as follows:

$$Final_Inv_r \leq Purchase_r * \alpha \quad (4.20)$$

Running the simulation application with this additional constraint causes infeasibility especially toward the end of the discount season. Thus, we present results from the first 9 weeks of the discount season and for 70 product-models. As can be seen in Table 4.7, at the 9th week of the discount season revenues of 14 product-models are behind the real revenue which results in a improvement of 3.2% on average. Without the product-models which are behind the real revenue the average improvement is 4.68%. In both cases the average improvement is far less than the improvement in the application which does not have a constraint on final inventory. There are two reasons of this worsening. The first one is the new model is a more constrained model with an additional constraint which affects maximization performance negatively. The second reason is that we evaluate the revenue improvement performance before the end of the season and before the product-model reaches planned sales amount due to infeasibility.

Table 4.7: Revenue Improvement Percentages of Product-Models with the Additional Constraint

Model#	Improve. %	Model#	Improve. %
A66	10.94%	A26	4.05%
A63	10.24%	A58	3.68%
A40	8.80%	A38	3.57%
A55	8.14%	A60	3.08%
A69	7.73%	A48	2.97%
A11	7.63%	A52	2.71%
A68	7.39%	A33	2.66%
A57	7.34%	A10	2.65%
A20	7.32%	A42	2.31%
A32	7.28%	A43	2.07%
A8	7.13%	A22	1.89%
A37	7.06%	A18	1.58%
A51	7.06%	A15	1.49%
A13	6.66%	A24	1.22%
A30	6.55%	A35	1.09%
A27	6.54%	A45	0.56%
A29	6.17%	A16	0.52%
A31	6.10%	A19	0.50%
A56	6.09%	A47	0.50%
A23	5.86%	A65	0.50%
A14	5.85%	A41	0.08%
A67	5.69%	A44	-0.10%
A59	5.65%	A34	-0.32%
A61	5.51%	A2	-0.82%
A6	5.48%	A17	-1.00%
A49	5.43%	A70	-1.73%
A50	5.37%	A3	-1.87%
A54	5.03%	A1	-2.16%
A39	4.62%	A62	-2.54%
A53	4.45%	A4	-2.68%
A7	4.41%	A46	-3.31%
A21	4.32%	A36	-3.45%
A28	4.27%	A9	-5.42%
A5	4.17%	A12	-5.76%
A25	4.10%	A64	-6.86%
		Average=	3.20%

According to Figure 4.12 our optimal discount amounts are less than the applied discount amounts and our expected sales are below the real sales. This case is in contrast with our expectations. We expect harsher discount to increase sales and decrease ending inventory level. By adding a constraint on the ending inventory amount, we aim the product-model to hurry in order to boost sales and cut prices earlier; however, it does not work.

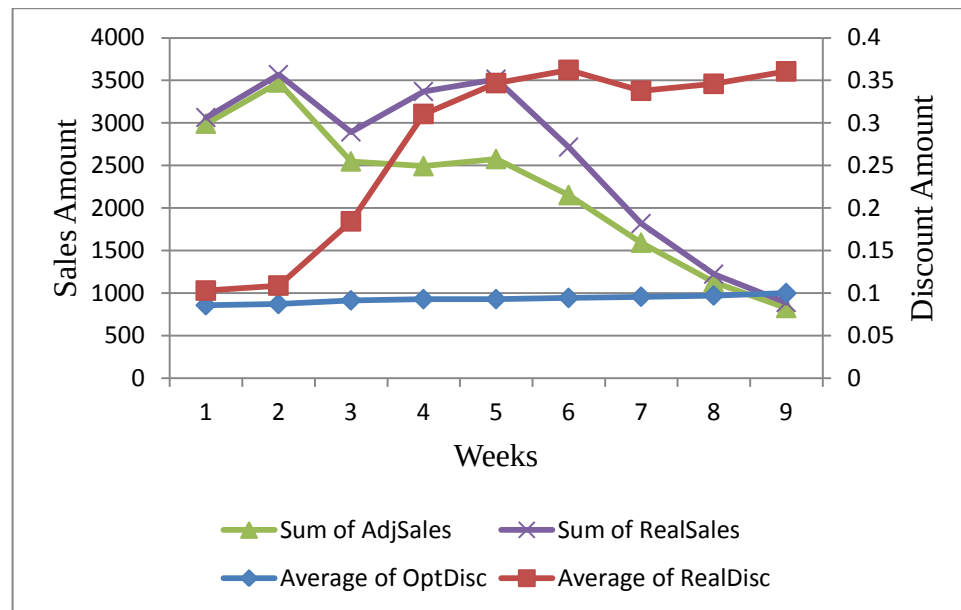


Figure 4.12: Real and Optimal Price Percentage & Sales Values with the Additional Constraint

4.4.4 Results of Simulation Application with Clustering

The simulation application is run for different store sets which are formed in Section 4.3. Clustering the stores and applying different price policies to each cluster increased the average expected revenue improvement percentage. The percentage improvement values in revenue can be seen in Table 4.8 and Table 4.9.

Table 4.8: First Part of Simulation Results with Clustering

Model #	Improvement %			Model #	Improvement %		
	HS	LS	Avg.		HS	LS	Avg.
A73	29.71%	25.57%	28.40%	A64	20.14%	9.63%	13.36%
A55	30.00%	25.86%	27.60%	A51	15.71%	10.59%	13.33%
A59	25.97%	24.86%	25.20%	A61	2.16%	26.21%	13.18%
A48	30.78%	15.77%	23.87%	A18	16.28%	9.16%	13.00%
A13	32.15%	12.30%	23.32%	A6	15.46%	11.40%	12.67%
A47	28.93%	9.91%	23.07%	A41	16.19%	8.05%	12.66%
A70	24.37%	9.80%	20.90%	A20	12.43%	12.65%	12.56%
A49	23.01%	15.89%	20.33%	A8	18.27%	9.29%	12.48%
A58	18.26%	20.06%	19.25%	A22	14.16%	10.51%	12.26%
A40	26.83%	13.19%	18.86%	A44	12.03%	11.98%	12.01%
A43	18.85%	18.38%	18.59%	A30	13.59%	8.96%	11.77%
A63	20.44%	16.94%	18.07%	A46	19.73%	5.02%	11.73%
A31	21.82%	13.29%	17.72%	A14	14.98%	7.19%	11.44%
A57	13.79%	20.21%	17.28%	A11	13.47%	8.85%	11.09%
A9	20.28%	13.60%	15.73%	A29	11.47%	10.47%	11.02%
A68	12.15%	19.15%	15.09%	A32	10.29%	11.21%	10.80%
A3	19.64%	8.79%	15.08%	A21	14.25%	8.57%	10.72%
A28	14.33%	15.43%	14.90%	A12	12.02%	9.11%	10.62%
A23	8.87%	18.87%	14.88%	A7	11.03%	9.63%	10.15%
A39	20.72%	10.67%	14.69%	A4	15.69%	6.84%	9.80%
A19	17.50%	10.44%	14.58%	A26	8.04%	11.35%	9.77%
A50	16.51%	11.17%	14.56%	A53	10.69%	8.99%	9.67%
A74	14.40%	13.42%	14.10%	A34	8.98%	9.13%	9.05%
A45	13.46%	14.25%	13.92%	A77	7.98%	8.83%	8.29%
A17	16.22%	9.61%	13.75%	A67	7.13%	10.15%	8.03%
A42	13.92%	12.93%	13.43%	A72	4.58%	11.92%	7.42%
A24	18.86%	7.15%	13.38%	A62	1.31%	12.51%	6.75%

Table 4.9: Second Part of Simulation Results with Clustering

Model #	Improvement %		
	HS	LS	Avg.
A33	6.64%	6.36%	6.54%
A1	3.99%	6.33%	5.54%
A52	5.85%	5.05%	5.45%
A15	6.17%	4.32%	5.28%
A75	5.69%	4.30%	5.01%
A71	4.03%	6.05%	4.48%
A16	9.19%	0.23%	4.39%
A65	0.62%	5.42%	3.65%
A5	4.62%	2.52%	3.47%
A56	12.11%	-3.03%	3.45%
A35	3.22%	3.08%	3.15%
A2	5.08%	2.39%	3.15%
A36	3.48%	1.61%	2.58%
A38	1.54%	0.88%	1.26%
A69	2.27%	-2.65%	0.66%
A60	0.68%	0.29%	0.46%
A25	-2.48%	4.03%	0.00%
A66	0.80%	-1.30%	-0.28%
A37	0.68%	-3.33%	-1.41%
A27	0.90%	-3.64%	-2.05%
A54	2.09%	-5.00%	-2.61%
A10	2.69%	-7.56%	-3.81%
A76	-16.53%	4.38%	-13.91%
Average=			10.66%

4.4.5 Economic Impact, Practicability and Drawbacks of Application

Simulation application is performed with the aim of imitating real life effects of optimal price decisions on the reactions of customers. According to the results presented in Section 3.4.3 if the optimal price decisions are applied there is an additional 9.74% potential revenue gain potential for the company. Moreover, generating a 16-week price policy for a product-model takes 1 minute on average. So, for 92 product-models that we study with we needed one and a half hour at the beginning of the discount season to produce a price policy for the whole discount season. In the following periods due to the planning horizon

gets shorter, it will take less time. So, the model is practicable enough to make weekly pricing decisions. At the same time, as optimal price decisions are applied and real reactions from customer to these prices are reflected to the model, it will give healthier proposals.

However, as it is the model generates a price policy which does not propose a major discount until the middle of the discount season which may contradict with the company's price policy. Besides, the model does not include competitive factors. For example, a price decision requested by the model for a week may not be suitable to apply considering the competitive environment where the company operates.

Chapter 5

Conclusion

Retailing is one of the most promising and growing industries. Among retailing activities having a reliable forecasting tool has a significant importance. The reason is that, a successful tool for forecasting provides company the advantage of coping with uncertainty more effectively and thus opportunity to decrease its risks of stock-out or excess inventory at the end of the season. The ability and power of using price to manipulate sales is another important factor that provides potential for improvement in revenues and competitive position within the market for the company.

In this study we developed a methodology to decide optimal discount amounts during the clearance season for a fast-fashion retailer. The first step of this methodology is generating a reliable forecasting model. Three linear regression models are evaluated to choose the most applicable forecasting model. In the main model, annual total purchase amount of the product, the time passed since the product is first sold, sales amount from the previous period, and current period's inventory position and price percentage are used as the explanatory variables to estimate sales. Among these five variables inventory position and price percentage are believed to perform well during the discount season. Thus, the coefficients of explanatory variables are generated by a two-step procedure. The first three variables' coefficients are computed from the regular season data and coefficients of inventory position and price percentage are computed from the discount season data. The three models are extensions of the main model.

In the second step of the methodology we developed an optimization model to generate a price policy for the use of price decisions during the clearance period for each product. This optimization model consists of 30912 decision variables and 65964 constraints. By getting forecasting model's outputs and product-specific information from the database, the optimization model generates an optimal price policy and expected sales amounts at optimal prices.

In Section 4, we implemented our findings to a sample weekly sales data of a fast-fashion retailer. The forecasting model generates forecasts with less than 40% Mean Absolute Deviation from the real sales data. This MAD value is acceptable for a reliable forecasting method. Afterwards, a simulation application is designed to use a dynamic pricing model and imitate real life responses to the applied prices. According to the results obtained from the simulation application, with the use of optimal prices from the pricing model a 9.74% revenue improvement is possible.

The solution process of simulation application takes an acceptable time. This supports the applicability of the methodology by the company in real-life decisions.

During the building process of these models the competitive effects of the market, the cross-price effects between the substitute products and effect of possible reactions of competitors are disregarded. As a future work, these kinds of competitive factors can be reflected in the models.

Bibliography

- Akçay, Y., Natarajan, H. P. & Xu, S. H., 2010. Joint Dynamic Pricing of Multiple Perishable Products Unders Consumer Choice. *Management Science*, 56(8), pp. 1345-1361.
- Araman, V. F. & Caldentey, R., 2009. Dynamic Pricing for Nonperishable Products with Demand Learning. *Operations Research*, 57(5), pp. 1169-1188.
- Aviv, Y. & Pazgal, A., 2005. A Partially Observed Markov Decision Process for Dynamic Pricing. *Management Science*, 51(9), pp. 1400-1416.
- Besbes, O. & Maglaras, C., 2012. Dynamic Pricing with Financial Milestones: Feedback-Form Policies. *Management Science*, 58(9), pp. 1715-1731.
- Besbes, O. & Saure, D., 2014. Dynamic Pricing Strategies in the Presence of Demand Shifts. *Manufacturing and Service Operations Management*, pp. 1-16.
- Bitran, G. & Caldentey, R., 2003. An Overview of Pricing Models for Revenue Management. *Manufacturing & Service Operations Management*, 5(3), pp. 203-209.
- Bitran, G. R. & Mondschein, S. V., 1997. Periodic Pricing of Seasonal Products in Retailing. *Management Science*, 43(1), pp. 64-79.
- Cachon, G. P. & Swinney, R., 2009. Purchasing, Pricing, and Quick Response in the Presence of Strategic Consumers. *Management Science*, 55(3), pp. 497-511.
- Caro, F. & Gallien, J., 2012. Clearance Pricing Optimization for a Fast-Fashion Retailer. *Operations Research*, 60(6), pp. 1404-1422.
- Chen, H., Wu, O. Q. & Yao, D. D., 2010. On the Benefit of Inventory-Based Dynamic Pricing Strategies. *Production and Operations Management*, 19(3), pp. 249-260.
- Chen, Y. & Farias, V. F., 2013. Simple Policies for Dynamic Pricing with Imperfect Forecasts. *Operations Research*, 61(3), pp. 612-624.
- Çelik, S., Muharremoğlu, A. & Savin, S., 2009. Revenue Management with Costly Price Adjustments. *Operations Research*, 57(5), pp. 1206-1219.

- Dana, J. D., 2008. New Directions in Revenue Management Research. *Production and Operations Management*, 17(4), pp. 399-401.
- Dong, L., Kouvelis, P. & Tian, Z., 2009. Dynamic Pricing and Inventory Control of Substitute Products. *Manufacturing and Service Operations Management*, 11(2), pp. 317-339.
- Duan, N., 1983. Smearing Estimate: A Nonparametric Retransformation Method. *Journal of the American Statistical Association*, September, 78(383), pp. 605-610.
- Elmaghraby, W. & Keskinocak, P., 2003. Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices, and Future Directions. *Management Science*, 49(10), pp. 1287-1309.
- Feng, Q., 2010. Integrating Dynamic Pricing and Replenishment Decisions Under Supply Capacity Uncertainty. *Management Science*, 56(12), pp. 2154-2171.
- Feng, Y. & Gallego, G., 1995. Optimal Starting Times for End-of-Season Sales and Optimal Stopping Times for Promotional Fares. *Management Science*, 41(8), pp. 1371-1391.
- Feng, Y. & Xiao, B., 2000. A Continuous-Time Yield Management Model with Multiple Prices and Reversible Price Changes. *Management Science*, 46(5), pp. 644-657.
- Gallego, G. & Hu, M., 2014. Dynamic Pricing of Perishable Assets Under Competition. *Management Science*, 60(5), pp. 1241-1259.
- Gallego, G. & van Ryzin, G., 1997. A Multiproduct Dynamic Pricing Problem and Its Applications to Network Yield Management. *Operations Research*, 45(1), pp. 24-41.
- Gaur, V. & Fisher, M. L., 2005. In-Store Experiments to Determine the Impact of Price on Sales. *Production and Operations Management*, 14(4), pp. 377-387.
- Jorgensen, S., Kort, P. M. & Zaccour, G., 1999. Production, Inventory, and Pricing Under Cost and Demand Learning Effects. *European Journal of Operational Research*, Volume 117, pp. 382-395.
- Kachani, S. & Shmatov, K., 2011. Competitive Pricing in a Multi-Product Multi-Attribute Environment. *Production and Operations Management*, 20(5), pp. 668-680.
- Kesavan, S., Gaur, V. & Raman, A., 2010. Do Inventory and Gross Margin Data Improve Sales. *Management Science*, 56(9), pp. 1519-1533.
- Koenig, M. & Meissner, J., 2010. List Pricing versus Dynamic Pricing: Impact on the Revenue Risk. *European Journal of Operational Research*, Volume 204, pp. 505-512.

- Levin, Y., McGill, J. & Nediak, M., 2007. Price Guarantees in Dynamic Pricing and Revenue Management. *Operations Research*, 55(1), pp. 75-97.
- Levin, Y., McGill, J. & Nediak, M., 2009. Dynamic Pricing in the Presence of Strategic Consumers and Oligopolistic Competition. *Management Science*, 55(1), pp. 32-46.
- Lin, K. & Li, F., 2004. *Optimal dynamic pricing for a line of substitutable products*, s.l.: Grado Department of Industrial Engineering.
- Liu, Q. & Zhang, D., 2013. Dynamic Pricing Competition with Strategic Customer Under Vertical Product Differentiation. *Management Science*, 59(1), pp. 84-101.
- Maglaras, C. & Meissner, J., 2006. Dynamic Pricing Strategies for Multiproduct Revenue Management Problems. *Manufacturing & Service Operations Management*, 8(2), pp. 136-148.
- Mantrala, M. K. & Rao, S., 2001. A Decision-Support System that Helps Retailers Decide. *Interfaces*, 31(3), pp. 146-165.
- Martinez-de-Albeniz, V. & Talluri, K., 2011. Dynamic Price Competition with Fixed Capacities. *Management Scienc*, 57(6), pp. 1078-1093.
- McGill, J. I. & van Ryzin, G. J., 1999. Revenue management: Research overview and prospects. *Transportation Science*, 33(2), pp. 233-256.
- Miller, D. M., 1984. Reducing Transformation Bias in Curve Fitting. *The American Statistician*, pp. 124-126.
- Musalem, A. et al., 2010. Structural Estimation of the Effect of Out-of-Stocks. *Management Science*, 56(7), pp. 1180-1197.
- Natter, M., Reutterer, T., Mild, A. & Taudes, A., 2007. An Assortmentwide Decision-Support System for Dynamic Pricing and Promotion Plannin in DIY Retailing. *Marketing Science*, 26(4), pp. 576-583.
- Netessine, S., 2006. Dynamic Pricing of Inventory/Capacity with Infrequent Price Changes. *European Journal of Operational Research*, Volume 174, pp. 553-580.
- Raman, K. & Chatterjee, R., 1995. Optimal Monopolist Pricing Under Demand Uncertainty in Dynamic Markets. *Management Science*, 41(1), pp. 144-162.
- Sato, K. & Sawaki, K., 2013. A Continuous-Time Dynamic Pricing Model Knowing the Competitor's Pricing Strategy. *European Journal of Operational Research*, Volume 229, pp. 223-229.

Sibdari, S. & Pyke, D. F., 2010. A Competitive Dynamic Pricing Model When Demand is Independent Over Time. *European Journal of Operational Research*, Volume 207, pp. 330-338.

Sigurdsson, V., Foxall, G. & Saevarsson, H., 2010. In-Store Experimental Approach to Pricing and Consumer Behavior. *Journal of Organizational Behavior Management*, 30(3), pp. 234-246.

Smith, S. A. & Achabal, D. D., 1998. Clearance pricing and inventory policies for retail chains. *Management Science*, 44(3), pp. 285-300.

Smith, S. A. & Achabal, D. D., 1998. Clearance Pricing and Inventory Policies for Retail Chains. *INFORMS*, pp. 285-300.

Su, X., 2007. Intertemporal Pricing with Strategic Customer Behavior. *Management Science*, 53(5), pp. 726-741.

Su, X., 2010. Optimal Pricing with Speculators and Strategic Consumer. *Management Science*, 56(1), pp. 25-40.

Şen, A. & Zhang, A. X., 2009. Style Goods Pricing with Demand Learning. *European Journal of Operational Research*, Volume 196, pp. 1058-1075.

Talebian, M., Boland, N. & Savelsbergh, M., 2014. Pricing to Accelerate Demand Learning in Dynamic Assortment Planning for Perishable Products. *European Journal of Operational Research*, Volume 237, pp. 555-565.

Talluri, K. & van Ryzin, G., 2004. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1), pp. 15-33.

Wernerfelt, B., 1986. A Special Case of Dynamic Pricing Policy. *Management Science*, 32(12).

Zhao, W. & Zheng, Y.-s., 2000. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science*, 46(3), pp. 375-388.

Zhu, K. & Thonemann, U. W., 2009. Coordination of Pricing and Inventory Control Across Products. *Naval Research Logistics*, 56(2), pp. 175-190.

APPENDICES

APPENDIX A: Forecasting Codes - R

#Data separation into fit and holdout sets

```
rawd_fit=rawd[rawd$ModelNo %in% fit,]
```

```
rawd_ho=rawd[rawd$ModelNo %in% holdout,]
```

#1st model on random sample, Caro & Gallien

```
cg_rs_fit=rawd_fit
```

```
cg_rs_holdout=rawd_ho
```

```
cg_rs_fitR<-cg_rs_fit[,c(1,2)]
```

```
cg_rs_fitR$LnWeekSales=log(cg_rs_fit$WeekSales)
```

```
cg_rs_fitR$LnPurchase=log(cg_rs_fit$Purchase)
```

```
cg_rs_fitR$Age=cg_rs_fit$Age
```

```
cg_rs_fitR$LnPreWeekSales=log(cg_rs_fit$PreWeekSales)
```

```
cg_rs_fitR$LnBrokenAssortmentEffect=log(cg_rs_fit$BrokenAssortmentEffect)
```

```
cg_rs_fitR$LnPricePercentage=log(cg_rs_fit$PricePercentage)
```

```
cg_rs_holdoutR<-cg_rs_holdout[,c(1,2)]
```

```
cg_rs_holdoutR$LnWeekSales=log(cg_rs_holdout$WeekSales)
```

```
cg_rs_holdoutR$LnPurchase=log(cg_rs_holdout$Purchase)
```

```
cg_rs_holdoutR$Age=cg_rs_holdout$Age
```

```
cg_rs_holdoutR$LnPreWeekSales=log(cg_rs_holdout$PreWeekSales)
```

```
cg_rs_holdoutR$LnBrokenAssortmentEffect=log(cg_rs_holdout$BrokenAssortmentEffect)
```

```
cg_rs_holdoutR$LnPricePercentage=log(cg_rs_holdout$PricePercentage)
```

```
cg_rs_fitR=lm(formula=LnWeekSales ~ LnPurchase + Age + LnPreWeekSales +  
LnBrokenAssortmentEffect + LnPricePercentage, data=cg_rs_fitR[cg_rs_fitR$Week < 24,])  
summary(cg_rs_fitR)
```

```
cg_rs_fitR$Est1=predict(cg_rs_fitR,cg_rs_fitR)
```

```
cg_rs_fitR$Res1=cg_rs_fitR$LnWeekSales-cg_rs_fitR$Est1
```

```
cg_rs_holdoutR$Est1=predict(cg_rs_fitR,cg_rs_holdoutR)
```

```
cg_rs_holdoutR$Res1=cg_rs_holdoutR$LnWeekSales-cg_rs_holdoutR$Est1
```

```
cg_rs_fitR2=lm(formula=Res1 ~ ModelNo, data=cg_rs_fitR[cg_rs_fitR$Week %in% c(5:23),])  
summary(cg_rs_fitR2)
```

```
cg_rs_holdoutR2=lm(formula=Res1 ~ ModelNo, data=cg_rs_holdoutR[cg_rs_holdoutR$Week %in%  
c(5:23),])
```

```
summary(cg_rs_holdoutR2)
```

```
cg_rs_fitR2=cg_rs_fitR[,c(1:8)]
```

```
cg_rs_fitR2$LnBrokenAssortmentEffect=0
```

```

cg_rs_fitR2$LnPricePercentage=0
cg_rs_fitR$Est2=predict(cg_rs_fitR2, cg_rs_fitR)+predict(cg_rs_fitR,cg_rs_fitR2)

cg_rs_holdoutR2=cg_rs_holdoutR[,c(1:8)]
cg_rs_holdoutR2$LnBrokenAssortmentEffect=0
cg_rs_holdoutR2$LnPricePercentage=0
cg_rs_holdoutR$Est2=predict(cg_rs_holdoutR2,
cg_rs_holdoutR)+predict(cg_rs_fitR,cg_rs_holdoutR2)

cg_rs_fitR$Res2=cg_rs_fitR$LnWeekSales-cg_rs_fitR$Est2

cg_rs_fitR3=lm(formula=Res2 ~ LnBrokenAssortmentEffect + LnPricePercentage,
data=cg_rs_fitR[cg_rs_fitR$Week %in% c(24:36),])
summary(cg_rs_fitR3)

cg_rs_fitR$Est3=cg_rs_fitR$Est2+predict(cg_rs_fitR3,cg_rs_fitR)
cg_rs_holdoutR$Est3=cg_rs_holdoutR$Est2+predict(cg_rs_fitR3,cg_rs_holdoutR)

cg_rs_fitR$Res3=cg_rs_fitR$LnWeekSales-cg_rs_fitR$Est3
cg_rs_holdoutR$Res3=cg_rs_holdoutR$LnWeekSales-cg_rs_holdoutR$Est3

MADfit=mean(abs(cg_rs_fitR$Res3[cg_rs_fitR$Week %in%
c(24:36)]))/mean(cg_rs_fitR$LnWeekSales[cg_rs_fitR$Week %in% c(24:36)])
MADMDtable[1,4]=MADfit
MDfit=mean(cg_rs_fitR$Res3[cg_rs_fitR$Week %in%
c(24:36)])/mean(cg_rs_fitR$LnWeekSales[cg_rs_fitR$Week %in% c(24:36)])
MADMDtable[2,4]=MDfit

MADholdout=mean(abs(cg_rs_holdoutR$Res3[cg_rs_holdoutR$Week %in%
c(24:36)]))/mean(cg_rs_holdoutR$LnWeekSales[cg_rs_holdoutR$Week %in% c(24:36)])
MADMDtable[7,4]=MADholdout
MDholdout=mean(cg_rs_holdoutR$Res3[cg_rs_holdoutR$Week %in%
c(24:36)])/mean(cg_rs_holdoutR$LnWeekSales[cg_rs_holdoutR$Week %in% c(24:36)])
MADMDtable[8,4]=MDholdout

#MAD with real sales
cg_rs_fit$Estimation=exp(predict(cg_rs_fitR2,
cg_rs_fitR)+summary(cg_rs_fitR)$coefficients[1,1]+summary(cg_rs_fitR3)$coefficients[1,1])*exp(sum
mary(cg_rs_fitR)$coefficients[2,1]*cg_rs_fitR$LnPurchase)*exp(summary(cg_rs_fitR)$coefficients[3,1
]*cg_rs_fitR$Age)*exp(summary(cg_rs_fitR)$coefficients[4,1]*cg_rs_fitR$LnPreWeekSales)*exp(su
mmmary(cg_rs_fitR3)$coefficients[2,1]*cg_rs_fitR$LnBrokenAssortmentEffect)*exp(summary(cg_rs_fi
tr3)$coefficients[3,1]*cg_rs_fitR$LnPricePercentage)
cg_rs_fit$Residuals=cg_rs_fit$WeekSales-cg_rs_fit$Estimation
cg_rs_fitMAD=mean(abs(cg_rs_fit$Residuals[cg_rs_fit$Week %in%
c(24:36)]))/mean(cg_rs_fit$WeekSales[cg_rs_fit$Week %in% c(24:36)])
MADMDtable[3,4]=cg_rs_fitMAD
cg_rs_fitMD=mean(cg_rs_fit$Residuals[cg_rs_fit$Week %in%
c(24:36)])/mean(cg_rs_fit$WeekSales[cg_rs_fit$Week %in% c(24:36)])
MADMDtable[4,4]=cg_rs_fitMD

cg_rs_holdout$Estimation=exp((predict(cg_rs_holdoutR2,
cg_rs_holdoutR)+summary(cg_rs_fitR)$coefficients[1,1]+summary(cg_rs_fitR3)$coefficients[1,1])*ex
p(summary(cg_rs_fitR)$coefficients[2,1]*cg_rs_holdoutR$LnPurchase)*exp(summary(cg_rs_fitR)$co
efficients[3,1]*cg_rs_holdoutR$Age)*exp(summary(cg_rs_fitR)$coefficients[4,1]*cg_rs_holdoutR$Ln

```

```

PreWeekSales)*exp(summary(cg_rs_fit3)$coefficients[2,1]*cg_rs_holdoutR$LnBrokenAssortment
Effect)*exp(summary(cg_rs_fit3)$coefficients[3,1]*cg_rs_holdoutR$LnPricePercentage)
cg_rs_holdout$Residuals=cg_rs_holdout$WeekSales-cg_rs_holdout$Estimation
cg_rs_holdoutMAD=mean(abs(cg_rs_holdout$Residuals[cg_rs_holdout$Week %in%
c(24:36)]))/mean(cg_rs_holdout$WeekSales[cg_rs_holdout$Week %in% c(24:36)])
MADMDtable[9,4]=cg_rs_holdoutMAD
cg_rs_holdoutMD=mean(cg_rs_holdout$Residuals[cg_rs_holdout$Week %in%
c(24:36)])/mean(cg_rs_holdout$WeekSales[cg_rs_holdout$Week %in% c(24:36)])
MADMDtable[10,4]=cg_rs_holdoutMD

cg_rs_fit$SmearingEstimation=cg_rs_fit$Estimation*mean(exp(cg_rs_fitR$Res3[cg_rs_fit$Week
%in% c(24:36)]))
cg_rs_fit$SmearingResiduals=cg_rs_fit$WeekSales-cg_rs_fit$SmearingEstimation
cg_rs_fit_SmearingMAD=mean(abs(cg_rs_fit$SmearingResiduals[cg_rs_fit$Week %in%
c(24:36)]))/mean(cg_rs_fit$WeekSales[cg_rs_fit$Week %in% c(24:36)])
MADMDtable[5,4]=cg_rs_fit_SmearingMAD
cg_rs_fit_SmearingMD=mean(cg_rs_fit$SmearingResiduals[cg_rs_fit$Week %in%
c(24:36)])/mean(cg_rs_fit$WeekSales[cg_rs_fit$Week %in% c(24:36)])
MADMDtable[6,4]=cg_rs_fit_SmearingMD

cg_rs_holdout$SmearingEstimation=cg_rs_holdout$Estimation*mean(exp(cg_rs_fitR$Res3[cg_rs_f
itR$Week %in% c(24:36)]))
cg_rs_holdout$SmearingResiduals=cg_rs_holdout$WeekSales-
cg_rs_holdout$SmearingEstimation
cg_rs_holdout_SmearingMAD=mean(abs(cg_rs_holdout$SmearingResiduals[cg_rs_holdout$Week
%in% c(24:36)]))/mean(cg_rs_holdout$WeekSales[cg_rs_holdout$Week %in% c(24:36)])
MADMDtable[11,4]=cg_rs_holdout_SmearingMAD
cg_rs_holdout_SmearingMD=mean(cg_rs_holdout$SmearingResiduals[cg_rs_holdout$Week
%in% c(24:36)])/mean(cg_rs_holdout$WeekSales[cg_rs_holdout$Week %in% c(24:36)])
MADMDtable[12,4]=cg_rs_holdout_SmearingMD

write.xlsx(MADMDtable, "MADMDTableNoSizes2.xlsx")

#Extracting Beta0 values
beta0r_fit=data.frame(matrix(nrow=61, ncol=2))
names(beta0r_fit)=c("ModelNo", "Beta0r")

for(i in 1:61)
{
temp1="ModelNo"
temp2=as.character(fit[i])
beta0r_fit$ModelNo[i]=as.character(fit[i])
temp2=paste(temp1, temp2, sep = "")
beta0r_fit$Beta0r[i]=cg_rs_fit2$coefficients[temp2]
}

beta0r_fit$Beta0r[beta0r_fit$ModelNo=="1PIMG01C473"]=cg_rs_fit2$coefficients[1]

beta0r_holdout=data.frame(matrix(nrow=31, ncol=2))
names(beta0r_holdout)=c("ModelNo", "Beta0r")

for(i in 1:31)
{
temp1="ModelNo"

```

```

temp2=as.character(holdout[i])
beta0r_holdout$ModelNo[i]=as.character(holdout[i])
temp2=paste(temp1, temp2, sep = "")
beta0r_holdout$Beta0r[i]=cg_rs_holdoutr2$coefficients[temp2]
}

beta0r_holdout$Beta0r[beta0r_holdout$ModelNo=="1PIMG01C475"]=cg_rs_holdoutr2$coefficients[
1]

beta0r=rbind(beta0r_fit,beta0r_holdout)

#Threshold Table
threshold_table=data.frame(matrix(nrow=92, ncol=2))
names(threshold_table)=c("ModelNo","ThresholdValue")

#i=1
#for (j in modelno2)
#{
#threshold_table$ModelNo[i]=j
#v= min(which(RegressionTable$ModelNo==j))
#threshold_table$ThresholdValue[i]=RegressionTable[v,"NumberStores"]*RegressionTable[v,"NoC
olorsOffered"]*RegressionTable[v,"AdjNoSizesOffered"]
#i=i+1
#}

i=1
for (j in modelno2)
{
threshold_table$ModelNo[i]=j
v= min(which(RegressionTable$ModelNo==j))
threshold_table$ThresholdValue[i]=RegressionTable[v,"NumberStores"]*RegressionTable[v,"NoCol
orsOffered"]*RegressionTable[v,"NoSizesOffered"]
i=i+1
}

#Extracting Beta Values
b1=as.numeric(cg_rs_fit$coefficients["LnPurchase"])
b2=as.numeric(cg_rs_fit$coefficients["Age"])
b3=as.numeric(cg_rs_fit$coefficients["LnPreWeekSales"])
b4=as.numeric(cg_rs_fit3$coefficients["LnBrokenAssortmentEffect"])
b5=as.numeric(cg_rs_fit3$coefficients["LnPricePercentage"])

b0r_forall=(cg_rs_fit$coefficients[1]+ cg_rs_fit3$coefficients[1])

betas=data.frame(matrix(nrow=6, ncol=4))
colnames(betas)=c("Beta#", "All", "HighScoreStores", "LowScoreStores")

betas[1,1]="Beta0"
betas[2,1]="Beta1"
betas[3,1]="Beta2"
betas[4,1]="Beta3"
betas[5,1]="Beta4"
betas[6,1]="Beta5"

```

```

i=2
betas[1,i]=b0r_forall
betas[2,i]=b1
betas[3,i]=b2
betas[4,i]=b3
betas[5,i]=b4
betas[6,i]=b5

MADMDtable2=MADMDtable

colnames(MADMDtable2)[c(4:6)]=c("All","HighScoreStores","LowScoreStores")
MADMDtable2[,i+2]=MADMDtable[,4]

#Beginning Inv Table
beg_inv=data.frame(matrix(nrow=92, ncol=2))
names(beg_inv)=c("ModelNo","Sevk")
l=1
for (j in modelno2)
{
  beg_inv$ModelNo[l]=j
  beg_inv$Sevk[l]=RegressionTable$InitialStock[RegressionTable$ModelNo==j &
  RegressionTable$Week=="37"]+sum(RegressionTable$WeekSales[RegressionTable$ModelNo==j
  & RegressionTable$Week %in% c(24:36)])
  l=l+1
}

write.xlsx(MADMDtable2, "MADMDtable2.xlsx")
write.xlsx(MADMDtable, "MADMDtable_Ls.xlsx")
write.xlsx(betas, "betas.xlsx")

#Real Policy Table With 7 Discounts
CG2.2=RegressionTable[,c(1,2,3,4,6,13,14)]
colnames(CG2.2)[6]="BrokenAssortmentEffect"
colnames(CG2.2)[7]="PricePercentage"

for (j in modelno2)
{
  CG2.2$PreWeekSales[CG2.2$Week %in% c(24:40) &
  CG2.2$ModelNo==j]=mean(CG2.2[CG2.2$Week %in% c(5:23) & CG2.2$ModelNo==j &
  CG2.2$WeekSales!=0,"WeekSales"])
}

real_policy7=CG2.2[CG2.2$Week %in% c(23:39),c("ModelNo","Week","PricePercentage")]

for(k in 1:nrow(real_policy7) )
{
  real_policy7$RealDiscount[k]=1-real_policy7$PricePercentage[k]
  real_policy7$ClosestIndex[k]=discountindex2[which.min(abs(discountindex2 -
  real_policy7[k,"RealDiscount"]))]
  real_policy7$NoClosestIndex[k]=which.min(abs(discountindex2 - real_policy7[k,"RealDiscount"]))
}

```

APPENDIX B: Simulation Codes – R

#Uc Calculation

```
Uc=(3*((b4)^2)+9*b4)/(2*((b4)^2)+6*b4+4)
```

#Smearing Parameter Calculation

```
SmearingParameter=mean(exp(cg_rs_fitR$Res3[cg_rs_fitR$Week %in% c(24:36)]))
w2=16
```

```
sim_runmip_7disc<-function(){
cmd=paste("C:/GAMS/win64/24.0/gams.exe","simulation_7disc.gms")
system(cmd,intern=TRUE)
}
```

```
sim_runmip2<-function(){
cmd=paste("C:/GAMS/win64/24.0/gams.exe","simulation_realprice.gms")
system(cmd,intern=TRUE)
}
```

```
setwd("C:/GAMS/win64/24.0")
getwd()
```

```
#simulation_results=data.frame(matrix(nrow=length(mlist)*w2,ncol=11))
#colnames(simulation_results)=c("ModelNo","Week","BegInv","OptDisc","OptPrice","OptSales","Adj
Sales","RealDisc","RealPrice","RealSales","RealInv")
```

```
m1ist2= tot_real_revenue[order(-tot_real_revenue$SumRevenue),][c(1:10,41:50),1]
m1ist=m1ist2[c(1:5,11:15)]
```

```
v=1
```

```
discountindex=c(seq(0.95,0.05,-0.1))
```

```
discountindex2=discountindex[4:10]
d_i2=length(discountindex2)
```

```
simulation_smeared_with_2nd_method_7disc<-function(mlist){
```

```
alpha=1/3
```

```
simulation_results=data.frame(matrix(nrow=w2,ncol=15))
colnames(simulation_results)=c("ModelNo","Week","BegInv","OptDisc","OptPrice","OptSales","AdjS
ales","RealDisc","RealPrice","RealSales","RealInv","CumOptRev","CumRealRev","ForecastedSales
RealPrice","CorrectionFactor")
```

```
v=1
```

```
forecast_table=data.frame(matrix(nrow=w2, ncol=d_i2))
p=1
age=CG2[CG2$Week==24 & CG2$ModelNo==j,"Age"]
b0r=b0r_forall+beta0r$Beta0r[beta0r$ModelNo==j]
purch=b1*log(RegressionTable$Purchase[min(which(RegressionTable$ModelNo==j))])
```

```

pws=b3*log(CG2$PreWeekSales[CG2$Week=="24" & CG2$ModelNo==j])
  for(i in 1:w2)
  {
    for(k in 1:d_i2)
    {
      disc=b5*log(1-discountindex2[k])
      forecast_table[p,k]=exp(b0r)*exp(purch)*exp(b2*age)*exp(pws)*exp(disc)
    }
    p=p+1
    age=age+1
  }

colnames(forecast_table)=c(1:d_i2)
forecast_table=SmearingParameter*forecast_table

first_ft=forecast_table

fp <- fullprice_df[fullprice_df$ModelNo==j,2]
tsh <- threshold_table[threshold_table$ModelNo==j,2]

maximum_final_inv=purchase[purchase$ModelNo==j,4]*1
dynamic_beginv <- beg_inv[beg_inv$ModelNo==j,2]
dynamic_beginv2 <- dynamic_beginv
dynamic_ft <- first_ft
opt_disc=real_policy7[real_policy7$ModelNo==j & real_policy7$Week==23,"ClosestIndex"]
wmax=16

  for (i in 1:w2)
  {

    setwd("C:/GAMS/win64/24.0")
    sim_wb <- createWorkbook(type="xls")

    sheet1 <- createSheet(sim_wb, sheetName="forecast_table")
    addDataFrame(dynamic_ft, sheet1)

    sheet2<- createSheet(sim_wb, sheetName="beg_inv")
    addDataFrame(dynamic_beginv, sheet2)

    sheet3<- createSheet(sim_wb, sheetName="fullprice")
    addDataFrame(fp, sheet3)

    sheet4<- createSheet(sim_wb, sheetName="threshold")
    addDataFrame(tsh, sheet4)

    sheet5<- createSheet(sim_wb, sheetName="discount_cons")
    addDataFrame(opt_disc, sheet5)

    sheet6<- createSheet(sim_wb, sheetName="wmax")
    addDataFrame(wmax, sheet6)

    sheet7<- createSheet(sim_wb, sheetName="final_inv_cons")
    addDataFrame(maximum_final_inv, sheet7)
  }

```

```

sheet8<- createSheet(sim_wb, sheetName="Uc")
addDataFrame(Uc, sheet8)

saveWorkbook(sim_wb, "sim_opt_table.xls")

sim_runmip_7disc()
#sim_runmip_7disc_inf()

opt_sales <- as.numeric(read.xlsx("sim_results_hs.xlsx",1,as.data.frame=FALSE))
opt_disc <- as.numeric(read.xlsx("sim_results_hs.xlsx",2,as.data.frame=FALSE))

simulation_results[v,1]=j
simulation_results[v,2]=i
simulation_results[v,3]=dynamic_beginv
simulation_results[v,4]=opt_disc
simulation_results[v,5]=(1-opt_disc)*fp
simulation_results[v,6]=opt_sales

#####real price
real_disc=1-RegressionTable[RegressionTable$ModelNo==j &
RegressionTable$Week==i+23,"Discount"]

simulation_results[v,11]=dynamic_beginv2
rs=RegressionTable[RegressionTable$ModelNo==j &
RegressionTable$Week==i+23,"WeekSales"]
simulation_results[v,10]=rs
dynamic_beginv2=dynamic_beginv2-rs
simulation_results[v,8]=real_disc
simulation_results[v,9]=(1-real_disc)*fp

p_p=((1-opt_disc)/RegressionTable[RegressionTable$ModelNo==j &
RegressionTable$Week==i+23,"Discount"])^b5
baep=(min(dynamic_beginv/tsh,1)/min(simulation_results[v,"RealInv"]/tsh,1))^b4

if(rs*p_p*baep>dynamic_beginv){adj_sales=dynamic_beginv} else
{adj_sales=rs*p_p*baep}

simulation_results[v,7]=adj_sales

#Simulating Real Price Policy
#real price policy for week w
di=which(discountindex2==real_policy7[real_policy7==j &
real_policy7$Week==i+23,"ClosestIndex"])

opt_sales_rp=forecast_table[i,di]*(1-Uc+Uc*min((dynamic_beginv/tsh),1))

simulation_results[v,14]=opt_sales_rp
simulation_results[v,15]=rs/opt_sales_rp

```

```

        forecast_table=((1-alpha)*forecast_table)+(alpha*forecast_table*(rs/opt_sales_rp))

        dynamic_beginv=dynamic_beginv-adj_sales

        simulation_results[v,12]=crossprod(simulation_results[1:v,"OptPrice"],simulation_results[1:v
,"AdjSales"])

        simulation_results[v,13]=crossprod(simulation_results[1:v,"RealPrice"],simulation_results[1:
v,"RealSales"])

        wmax=wmax-1
        dynamic_ft=forecast_table[(17-wmax):16,]
        rownames(dynamic_ft)=c(1:wmax)

        v=v+1

        setwd("C:/Users/Nazli/Dropbox/Nazli Compt/Flo Sums/R
Codes/tezsimsimhighscore_ind")
        tempname=paste("SimResults",j,"FTrevised_highscorestores_ind",".xlsx",sep="")
        write.xlsx(simulation_results, tempname)

    }

}

simulation_smeared_with_2nd_method_7disc(mlist[10])

for(j in modelno2)
{
  try(simulation_smeared_with_2nd_method_7disc(j))
}

```

APPENDIX C: Optimization Codes – GAMS

```

Sets
k discount amount /1*7/
w week number /1*16/
;

$onecho > sim_opt_table.txt
i=%system.fp%sim_opt_table.xls
r1=beg_inv!b2
o1=parInv0.inc
r2=fullprice!b2
o2=parfullprice.inc
r3=threshold!b2
o3=parthreshold.inc
r4=forecast_table!a1:k17
o4=parforecast.inc
r5=discount_cons!b2
o5=pardisc_cons.inc
r6=wmax!b2
o6=parWmax.inc
r7=final_inv_cons!b2
o7=parMaxFin_inv.inc
r8=Uc!b2
o8=parUc.inc

$offecho
$call =xls2gms @sim_opt_table.txt
parameter Inv0
/
$include parInv0.inc
/
;

parameter fullprice
/
$include parfullprice.inc
/
;

parameter threshold
/
$include parthreshold.inc
/
;

table forecast(w,k)
$include parforecast.inc
;

```

```
parameter disc_cons
/
$include pardisc_cons.inc
/
;
```

```
parameter Wmax
/
$include parWmax.inc
/
;
```

```
parameter MaxFin_inv
/
$include parMaxFin_inv.inc
/
;
```

```
parameter Uc
/
$include parUc.inc
/
;
```

Parameter

$d(k)$ discount amount
/

1 0.65

2 0.55

3 0.45

4 0.35

5 0.25

6 0.15

7 0.05

/

Positive variable

sales(w,k) exp sales for model r when disc is k in week w ;

Binary variables

x(w,k) whether model r should be sold with a discount k or lower during week w

y(w,k) whether model r should be sold with a discount k during week w ;

Variables

z total revenue

l(w) available inventory level of model r at the beginning of week w

;

Positive variables

FinInv final inventory

revenue(w)

fwd first week discount

fws first week sales

;

Equations

discount_cons beginning disc cons
discount_cons2 beginning disc cons'le ilk hafta disc arasindaki fark 0.1'den küçük olacak
less_than_tp(w) the difference between discount amount for successive weeks should be less than ten percent

periodic_rev(w)

sales_eq(w,k)

sales_eq2(w,k)

xry(w,k) relates y and x

xry2(w) for the first week

xrx(w,k) definition of x parameter

ysum(w) sum of y parameters for a week 1

pricegoesdown(w,k) prevents price increase as time passes

inventory1 first week inventory of model r

inventory2(w) inventory balance equation

finalinventory(w)

firstweekdisc

firstweeksales

obj defines the total revenue at the end of the clearance season;

discount_cons.. sum(k,y("1",k)*d(k))=g=disc_cons;

discount_cons2.. sum(k,y("1",k)*d(k))-disc_cons=l=0.1;

less_than_tp(w)\$(ord(w) gt 1).. sum(k,y(w,k)*d(k))-sum(k,y(w-1,k)*d(k))=l=0.1;

periodic_rev(w)\$(ord(w) le Wmax).. revenue(w)=e=sum(k,sales(w,k)*(1-d(k))*fullprice);

sales_eq(w,k)\$(ord(w) le Wmax).. sales(w,k)=l=forecast(w,k)*y(w,k);

sales_eq2(w,k)\$(ord(w) le Wmax).. sales(w,k)=l=forecast(w,k)*(1-Uc+Uc*l(w)/threshold) ;

```
xry(w,k)$ (ord(k) gt 1 and ord(w) le Wmax) .. y(w,k)=e=x(w,k)-x(w,k-1) ;
xry2(w)$ (ord(w) le Wmax) .. y(w,"1")=e=x(w,"1") ;
xrx(w,k)$ (ord(k) gt 1 and ord(w) le Wmax) .. x(w,k-1)=l=x(w,k) ;
ysum(w)$ (ord(w) le Wmax) .. sum(k, y(w,k))=e=1;
pricergoesdown(w,k)$ (ord(w) gt 1 and ord(w) le Wmax) .. x(w-1,k)=l=x(w,k) ;
inventory1 .. l("1")=e=Inv0 ;
inventory2(w)$ (ord(w) gt 1 and ord(w) le Wmax) .. l(w)=e=l(w-1)-sum(k,sales(w-1,k)) ;
finalinventory(w)$ (ord(w) eq Wmax) .. FinInv=e=l(w)-sum(k,sales(w,k)) ;
```

```
firstweekdisc.. fwd=e=sum(k,y("1",k)*d(k));
firstweeksales.. fws=e=sum(k,sales("1",k)) ;
obj .. z=e=sum((w,k)$ (ord(w) le Wmax), sales(w,k)*(1-d(k))*fullprice) ;
```

```
Model simpleopt /all/ ;
```

```
option optcr=0;
OPTION RESLIM = 50000;
Solve simpleopt using mip maximizing z ;
```

```
Display x.l, x.m, y.l, z.l , sales.l, l.l , fininv.l, revenue.l ;
```

```
execute_unload "sim_results_hs.gdx" fws.L fwd.l
```

```
execute 'gdxxrw.exe sim_results_hs.gdx var=fws.L rng=first_week_sales!'
execute 'gdxxrw.exe sim_results_hs.gdx var=fwd.L rng=first_week_disc!'
```