THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

HEURISTIC BASED CLEARANCE-MARKDOWN OPTIMIZATION FOR A FAST-FASHION RETAILER

Master Thesis

TUFAN BAYDEMİR

İSTANBUL, 2017

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

THE GRADUATE SCHOOL OF NATURE AND APPLIED SCIENCES
INDUSTRIAL ENGINEERING

HEURISTIC BASED CLEARANCE-MARKDOWN OPTIMIZATION FOR A FAST-FASHION RETAILER

Master Thesis

TUFAN BAYDEMİR

Advisor: Asst. Prof. Dr. ETHEM ÇANAKOĞLU

İSTANBUL, 2017

T.C.
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURE AND APPLIED SCIENCES

INDUSTRIAL ENGINEERING

Name of the Thesis: Heuristic Based Clearance-Markdown Optimization for a Fast Fashion Retailer

Name of the Student: Tufan Baydemir

Date of Thesis Defense: 10.05.2017

This thesis has been approved by the Graduate School of Natural and Applied Sciences

Prof. Dr. Nafiz ARICA
Graduate School Director

…………………..

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Sciences

Asst. Prof. Dr. Adnan ÇORUM
Program Coordinator

……………………….......

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Sciences.

Examining Committee Members Signature	Signature
Thesis Supervisor Asst. Prof. Dr. Ethem ÇANAKOĞLU	……………………….
Member Assoc. Prof. Dr. Dilek Tüzün AKSU	……………………….
Member Assoc. Prof. Dr. Mehmet Alper TUNGA	……………………….

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Asst. Prof. Dr. ETHEM ÇANAKOĞLU for his useful advices and contributions to my research activities. I also would like to thank to my committee member, Assoc. Prof. Dr. DİLEK TÜZÜN AKSU for her valuable contributions and advices during the research study. I would also like to thank my committee member Assoc. Prof. Dr. Mehmet Alper TUNGA, for his suggestions to improve my study in the future.

Also, I'd like to thank to my colleague MERVE AKSÖZ for her precious time and useful advices during my study. I would like to thank to my family members, my mother and father, my wife SİBEL BAYDEMİR and my son, who supported me during the whole study.

İstanbul, 2017 Tufan BAYDEMİR

ABSTRACT

HEURISTIC BASED CLEARANCE-MARKDOWN OPTIMIZATION FOR A FAST-FASHION RETAILER

BAYDEMİR, TUFAN

Industrial Engineering

Thesis Supervisor: Asst. Prof. Dr. ETHEM ÇANAKOĞLU

May 2017, 48 pages

In apparel retail business, the main objective is to gain the best possible revenue in short selling season. Retailers have to cope with many uncertainties to achieve this objective. The most important tool to overcome the uncertainties is usually price discounts and campaigns. Merchandising planners generally apply price discounts based on their experience intuitively. The aim of this thesis is finding a practically applicable and good solution to help managers to decide what time and to which products to apply mark-downs in clearance period.

In this study, a decision support system which decides the timing and amounts of discounts is derived. A forecasting model and a dynamic pricing methodology is proposed which decides when to make discounts and which price to be used for each individual SKU's under uncertainties. The proposed model is applied to a reputable fashion retailer's historical data and results are evaluated.

Keywords: Apparel, Retail, Mark-Down, Optimization, Dynamic Pricing

ÖZET

BİR HIZLI MODA PERAKENDECİSİ İÇİN SEZGİSEL TABANLI İNDİRİM ENİYİLEMESİ

BAYDEMİR, TUFAN

Endüstri Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. ETHEM ÇANAKOĞLU

Mayıs 2017, 48 sayfa

Hazır giyim perakendesinde ana hedef kısa satış sezonunda mümkün olan en iyi geliri elde etmektir. Perakendeciler bu amaca ulaşmak için talepteki belirsizliklerle baş etmek zorundadırlar. Belirsizliklerin üstesinden gelebilmenin en önemli aracı genellikle fiyat indirimleri ve kampanyalardır. Mağazacılık planlayıcıları genel olarak fiyat indirim kararlarını sezgisel deneyimlerine dayanarak verirler. Bu tezin amacı, planlayıcıların, indirim döneminde hangi ürünler için ne zaman ve ne kadar indirim uygulayacaklarına karar vermelerine yardımcı olacak pratik ve en iyiye yakın bir çözüm bulmaktır.

Bu çalışmada, indirimlerin zamanlaması ve miktarını belirleyen bir karar destek sistemi sunulmaktadır. Bir tahmin modeli kurulmuş ve her bir SKU için ne zaman indirim yapılacağına ve bu indirimin hangi fiyat ile yapılacağına dinamik programlama ile karar veren bir eniyileme algoritması önerilmektedir. Önerilen model, Türkiye'de saygın bir moda perakendecisinin verileri üzerinde tatbik edilerek sonuçları değerlendirilmiştir.

Anahtar Kelimeler: Hazır Giyim, Perakende, İndirim, Eniyileme, Dinamik Fiyatlandırma

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

ADP:	Approximate Dynamic Programming
DP:	Dynamic Programming
SKU:	Stock Keeping Unit

SYMBOLS

Demand rate	:	λ
Error term of forecasts	:	ε
Elasticity of price	:	$\gamma, e^{\gamma p}$
Vector of sample path	:	ω
Age of a product	:	A
Capacity	:	C
Demand	:	d, D
Expectation of a function	:	$\mathbb{E}$
Special days dummy variable	:	G
Inventory level	:	I
Seasonal dummy variable	:	K
Individual product-model	:	m
Iteration	:	n, N
Initial order quantity of a product	:	O
Price	:	p
Probability distribution	:	$\mathbb{P}$
Sales Quantity	:	Q
Reward or revenue gained	:	r, R
Interval of time	:	t
Total time in selling horizon	:	T
Value function	:	V
New information	:	W
Output	:	$y, \hat{y}, \bar{y}$

1. INTRODUCTION

Fashion retailers plan their budgets for the upcoming seasons in a year advance. Because of the longer lead times, they have to decide which product and how much to be produced months before the selling season starts. Merchandising managers plan their budgets under some uncertainties like "which products the customers will likely to buy?"," which color will be popular?". Besides, in apparel retail business, products are changed dramatically in every selling season. Generally, many of the products sold during a selling season has no historical information. Lack of information about customer's tastes and not existence of historical sales data cause great uncertainty about demand planning. Therefore, many retailers start with higher prices at the beginning of each season. Even though retailers have accurate forecasts on demand, competitors' actions in the selling season causes deviations from the budget. For these reasons, managing product portfolio in the fashion industry is similar to managing a bunch of stocks in the stock market.

When the selling season starts, retailers collect first signals of demand, then they evaluate the performance of each product. When a product performs poor, planning and operations managers have to control the intensity of demand. Either they decide to reduce the price or cancel the remaining orders which have not been delivered yet. Canceling the remaining orders depend on the contracts with the suppliers. And generally, it's not possible to cancel these orders. It is common that they prefer to mark-down to manage the demand of the products.

Pashigan (1988) showed that number of colorful and printed products increase in fashion industry which causes that total amount of revenue gained by markdowns as a percentage of total sales increases over time. Generally, many of the fashion retailers have adopted the policy that start with premium (or full) prices for the season.

Another important dynamic in apparel retail is the behavior of the customers. Customers will not be willing to buy products from retailers whose prices frequently increase and decrease dramatically in the season. That's why in apparel retail, prices monotonically decrease over time. Therefore, most retailers do not increase the prices even though product is being sold successfully.

For these reasons, accurate sales forecasting in apparel industry is the most important input for the markdown optimization models. To generate better forecasting algorithms some should well understand the dynamics behind the purchasing decision in apparel. Purchasing decision of a customer is generally related with the price of the product. Usually there is a reference price that the customers are willing to pay. This price is called reservation or utility price. Whenever the price of a product is less then this reservation price, it is believed that customer decides to buy. And generally these reservation prices decrease over time which means that customers are willing to pay much in early stages of the season than later stages.

Generally, there are two types of customer related with the purchasing behavior. The first type of customers buys the products whenever they see them on the shelves. They don't consider future markdowns as a purchasing driver. This types of customers are called myopic customers. Other type of customers decides to wait for the future markdowns even though they like the product. They take the risk of not to find the product on the shelves in the future. These types of customers are called strategic customer. From the modeling perspective, companies assume that either they have same types of customers (homogeneous environment) or customers behave differently based on their income, age or purchasing decisions (heterogeneous environment).

Another important case in apparel retail is the perishability of the products. Even though they don't perish physically, companies do not want to stand for the holding costs of remaining inventories till the next season starts. Because of high inventory costs and frequent changes in customer tastes in fashion, they apply liquidation discounts to liquidate the unsold inventory at the end of the season.

Under these types of difficulties, apparel retailers confront with shortage or excess inventory at the end of the selling season. Inventory shortage effects the product availability and causes unhappy customers. In contrary, excess inventory causes revenue loss and additional operational costs like carrying inventory till the next selling season (holding costs) or handling and transfer cost to outlets. That's why some retailers sell out remaining inventories to the big discounters or give them to the charity at the end of the season.

Since ordinary apparel retailers have thousands of products, manually deciding the discounts is not an easy job. Besides, characteristics of the demand is very complex in apparel retail business. To deal with this sophisticated problem merchandising planners need a decision support tool to decide the timing of the mark downs. By using revenue optimization tools Friend & Walker (2001) has reported that potential increase in gross margins are between 5% to 15%.

In consideration of difficulties mentioned in this section, aim of this study is to develop a revenue optimization tool to help merchandising planners to decide in which time to apply markdowns to which products. Because of the complexity of the problem, we divided our study into two main phase which are demand forecasting and optimization. We took considerable amount of time to generate an accurate forecasting method. Next, we focused on solving the optimization model. Because of the objective function is non-linear, we used dynamic programming approach. In dynamic programming, we especially take much attention on approximating the value functions and value iteration. We have taken into account of the uncertainty in demand and proposed a solution. We applied the solution to the historical data and evaluated the results.

In Chapter 2, we reviewed the related literature, in Chapter 3 we give some brief information about the data and the business rules of the retail company. We developed a forecasting methodology and a procedure periodically updating the model parameters. And results are evaluated. For the remainder of the Chapter 3, we explained our optimization methodology and how to solve the curse of dimensionality by using approximate dynamic programming approach. In Chapter 4, we evaluated our findings and in Chapter 5 we shared our ideas about the related future works.

2. LITERATURE REVIEW

In this section, the literature review was given about the revenue management and its application in apparel industry.

Selling products or services to the individual customers is called retail business. In today's competitive environment, retailers have to operate their business in an efficient way to become competitive in the market. Generally, focus of the retailers is to satisfy the customer demand in the right time. In particular, apparel industry has short selling season in comparison with the longer lead times. For that reasons many apparel retailers have to plan their product assortments long before the selling season starts. Besides, most of the products which presented in stores are the products that are put on sale for the first time. Therefore, demand is highly uncertain in apparel retail and customers' willingness to purchase decision change over time. For these reasons, generally plans might largely deviate in the selling season.

Under these circumstances, companies' main target is to gain maximum amount of revenue in this short selling horizon. In a finite amount of time, ensuring product availability and optimizing the prices which leads to maximize the total expected revenue is called revenue or yield management. The first applications of revenue management were implemented on airlines industry and hotels booking in 1970's.

In their book, Talluri & Van Ryzin (2004) have addressed dynamic pricing and revenue management problems in broad terms. Elmaghraby & Keskinocak (2003) reviewed the dynamic pricing problems and proposed new directions to the future research. In their study, they also interviewed with the managers of dynamic pricing software companies and reported the challenges of implementation of dynamic pricing applications.

Determination of the products initial price is one of the first important decision in apparel retail. In two stage selling problem, Lazear (1986) assumed that all customer's utility prices are identical to each other in a finite horizon. With a fixed inventory on hand at the beginning of the season and without any replenishment allowed, Lazear showed that, starting with a higher initial price at first stage and if product is not sold then reducing the price at the second stage causes higher revenues rather than using optimal price at both stages. With a fixed amount of inventory, as T increases then the initial price also increases and the price drops accordingly over time.

Another important decision in apparel retail is “When and how much the product will be discounted?”. In their study, Gallego & Van Ryzin (1994) offered a customer choice model with a stochastic demand to find optimal prices for the products. In their model, customer arrivals are constant over time and every customer’s reservation price comes from a continuous Poisson distribution and demand rate $\lambda(p)$ is a function of current price. With assumption of the company is monopolist in the market, by using the price, they control the intensity of the demand. They consider a model where prices can only be selected from a predefined set which ensures that $p_1 > p_2 > \ldots > p_T$. They developed a stochastic model and some heuristic algorithms which assume that the demand is deterministic. They showed that their fixed price heuristic is asymptotically optimal with the stochastic solution as the sales amount goes to infinity.

Many studies assumed that customer arrival process is independent over time. But Bitran and et al. (1998) modeled demand as a function of time. Considering special days such as New Year's Day, Mother's Day and Ramadan Feast, it is expected that the demand will vary according to the time. Namely, arrival of a customer is related with their shopping pattern rather than price discounts. In their model, arrival rate of customer’s $\lambda_i(t)$ for store i, changes over time t and demand function comes from joint distribution of customer arrivals and customer’s utility prices at time t. Because of the state space is very large, they offered a heuristic algorithm for the case both with inventory replenishment allowed or not. They compared the total revenue earned by merchandising manager with their model and reported that in each case their model obtained 7-12% higher revenues.

Warner & Barsky (1995) examined daily prices of some goods at different retailers from November to February and identified that in special days like pre-Christmas season, customers are more sensitive to the price discounts than in ordinary days. This is because, in such days, customers go for shopping trip and have much time to search for the best prices. For example, before the Ramadan Feast, customers visit the stores to buy some new products for the family members. But at that time they are searching for the best possible offer from the retailers.

Zhao & Zheng (2000) modeled customer arrivals as a Poisson process and assumed that it changes over time, demand $\lambda(p,t)$ is sensitive to both price and time. Instead of continuous prices they studied discrete price sets. In their model, product replenishment is not allowed and unsold inventory is sunk. They showed that willingness of customers to pay full price decreases over time which is a realistic determination in apparel retail. For example, customers wishing to buy winter coats are more eager to pay full price at the beginning of the season than at the end of the season. They numerically showed that optimal dynamic pricing strategy gains 2.3-7.4% more revenue than single optimum pricing strategy.

Bitran & Mondschein (1997) modeled an optimal dynamic pricing model, assumed that time horizon is fixed. They assumed customer arrivals as a continuous stochastic process and arrival of customers is a function of customers shopping habits, not a function of individual price discounts. They use Poisson distribution to represent the customer arrivals and Weibull distribution to represent the customer's reservation prices. They offered a model to update prices periodically in such a way that prices monotonically decrease during the selling season. They also add a constraint on number of discounts applied in the selling horizon which limits the number of allowed price changes in the clearance period. In practice this limitation is followed by many retailers. This policy is important if the frequent price discounts cause customer regret. They showed that profit loss is small when appropriate number of reviews are applied. Another study to support this result was also done by Gupta and et al. (2004). They consider a discrete-time model and assuming that the demand is stochastic. They generate a heuristic procedure to decide optimal prices under the limiting the number of applied discounts constraint. They offered that multiple markdown decisions are better than one shot bigger markdown decisions. And they concluded that whenever the demand curve is smoother, retailers' expected revenue gets bigger.

In their study, Smith & Achabal (1994) pointed out that if some colors or sizes of a product was not on display that would cause decrease in sales. In reality, during the times very close to the end of the selling season, some products' color or sizes are not available. This is called broken assortment effect and this leads to a loss of sale. Low inventory under a threshold may slow down the sales but high level of inventory which is higher than certain threshold doesn't cause any increase in sales. Therefore, they formulated demand as a function of price, time and inventory. They proposed a log-linear model which estimates the demand as a function of seasonal variations, price and inventory levels. In this study, demand at time t is given by $k(t)y(I)e^{-\gamma p}$ where $k(t)$ is the seasonal demand at time t , $y(I)$ is the inventory effect when inventory level is at I and $e^{-\gamma p}$ is the sensitivity of demand to the price p. Their model assumed that demand is deterministic. They implemented their models in three different retailers (Smith & Achabal 1998) and reported that the percentage of inventory sold in each week was 15-20% higher than previous years.

Maglaras & Meissner (2006) offered a multiproduct dynamic pricing model in an imperfect competition environment with a finite horizon. They showed that resolving deterministic problem periodically performs better than a fixed pricing policy. Sen (2012) offered some heuristic algorithms that continuously review the optimal prices and update the prices based on inventory on hand and remaining time to the end of season. He showed with a numerical experiment that in case of periodically price updating policy the potential revenue is higher than fixed pricing policy. He concluded that dynamic pricing solutions should be preferred in practice.

Heching, Galleo & Van Ryzin (2002), formulated the demand as a function of price, seasonal variations and age factor. Age represents the number of weeks passed from the time which product is being sold. They analyzed the effectiveness of companies' decisions such as applied discounts and timings of the discounts. They reported that their model causes significant increase on revenue at major retailers in clearance period. They found that price and seasonal variations affect the demand significantly. But for the age dependent factor, they reported additional data was needed. They found that it is better for the companies to apply earlier markdowns in the selling season. They conclude that if all demand information is known in advance than company's potential revenue increase is 13%, but adaptive policies can increase revenue by only around 3%. This means that there is a gap potentially to be closed by better algorithms. Xu & Hopp (2005) find a similar result. In addition to Heching's findings, they concluded that pricing policies which take into account uncertainty in demand are better than policies without caution.

Up to now all of these studies focused on the optimal pricing policies for selling out the products, Mantrala & Rao (2001) developed a decision support system based on stochastic dynamic programming model called MARK, which determines discrete prices and pre-season purchase order levels based on a time varying price elasticity demand model.

Chew & Lee (2014) came up with the idea that demand for the products with different ages are dependent on the prices of others. For the products that have longer lifetime with more than one season, a heuristic based optimal pricing algorithm was proposed. The computational results showed that the total profit significantly increases when the demand transition between products of different ages were considered. In the case of discounting on one product might affect the demand of other product, Akcay and et al. (2010) proposed a dynamic pricing model over multiple products which are substitutable and perishable.

In their empirical study, Caro & Gallien (2012) modeled a deterministic multi-product markdown optimization solution and reported large scale implementation of their model at fast-fashion retailer Zara. They used a multiplicative demand model as function of seasonal variations, initial purchase, broken assortment effect and elasticity of price. They updated the coefficient of parameters when the new information arises in each week to make forecasting system more responsive. They solved the problem with integer programming approach. Their deterministic model run every week with the updated parameters. The solution was implemented as a pilot in two countries. They showed empirically that the revenue gained by the new system was 6% higher than existing planning system.

Yao, Sanli & Gulcu (2015) developed a model with a log-linear demand function which uses the price and remaining inventory as a decision variable and solved revenue maximization problem with an additional budget constraint to limit the discounts based on the company’s markdown budget. In their study, they used dynamic programming approach. They reported that with the periodically updating optimal prices, their model achieved 1-3% revenue increase and lowered the end of season inventories in test stores over a control group.

Soysal & Krishnamurthi (2012) offered a model that incorporates price, limited inventory and strategic customers in an imperfect competition. They estimate the demand using historical data from an apparel goods retailer. Results of the study indicate that ignoring customer's strategic decisions causes biased estimates on demand and has negative impact on retailer's revenues by 9-35%. They stated that in order to achieve as highest as possible profit in the selling season, retailers should apply smaller markdowns early in the season. Zhao and et al. (2012) also studied the effects of strategic customer behavior. They formulated this problem using the finite-horizon dynamic programming approach and concluded that if your customers are strategic than this causes negative impact on company's expected profit. Levin and et al. (2010) proposed a stochastic game-theoretical dynamic pricing model which considers customer's strategic buying decisions. In their study, they stated that in a competitive market, companies who ignore the customer's strategic behaviors might go in trouble with their pricing decisions and loose profit. Dasu & Tong (2010) stated that when consumers are strategic, the differences between the highest and the lowest prices are narrowed which means that if a company ignores strategic behavior, then it's initial prices will be too high and most customers will decide to wait for a discount.

Koening & Meissner (2009) tried to find an answer to the question of "How much riskier it is applying a list pricing policy rather than a dynamic pricing policy". After several numerical experiments they compared the two policies by examining the expected revenue, standard deviation, and conditional-value-at-risk. They concluded that if changing prices are costly or impractical, list pricing can be a useful strategy to apply. In 2012, department store JC Penney introduced it's "Everyday low prices" strategy. With this new strategy, the company has stopped making promotion campaigns, sending cash coupons to the customers and offered the lowest price in the market. Ofek & Avery (2012) reported that in the first quarter, this strategy caused $163 million revenue loss to the company. After 3 months, number of customers visiting the stores dropped 10%. One year later the company turned back to its previous pricing strategy.

Up to now many studies assumed that the demand distribution or it's parameters is known in advance which is not a true assumption in real life. If historical demand information was under censored environment, parameters will be underestimated. Recently demand learning was used to deal with the uncertainty. Demand learning assumes that form of demand function is known in advance but the parameters are unknown and retailers try to learn the parameters when the new information arrive.

Maglaras & Eren (2015) proposed a data driven forecasting approach to accurately forecast the future demand when historical information might had censored. They stated that uncensoring techniques might decrease the optimal revenues to a suboptimal level. Therefore, they used the maximum entropy distribution and showed that their forecasting algorithm converges to the real demand even there exists any censored information.

Lim & Shanthikumar (2007) proposed a robust pricing model which considers forecasting errors as a risk measure. Levin and et al. (2008) proposed a model which includes risk measure to assure that total revenue is greater than it's expected minimum. Perakis & Sood (2006) studied a robust dynamic optimization policy with demand uncertainty in a competitive environment that has more than one seller. All of these studies state that considering forecasting errors in revenue optimization models has positive impacts on company's total profits.

Chen & Chen (2015) summarized the dynamic pricing literature and stated that many of the studies in literature are highly stylized and less paper incorporates the business rules into the models. In their study, they offered future researchers to consider the business rules in their models. Consideration of business rules and constraints will make the model more practical. And these new model has potential to generate better results and more useful insights.

3. DATA AND METHODOLOGY

In this section, the data that collected from a well-known apparel retailer in Turkey was used and the research methodology of the thesis study was introduced. Some early research studies and literature reviews were provided in previous sections. This section provides the research subjects, data analysis, solution approach and application of the methodology for the markdown optimization in a fashion retailer.

3.1 DATA DESCRIPTION

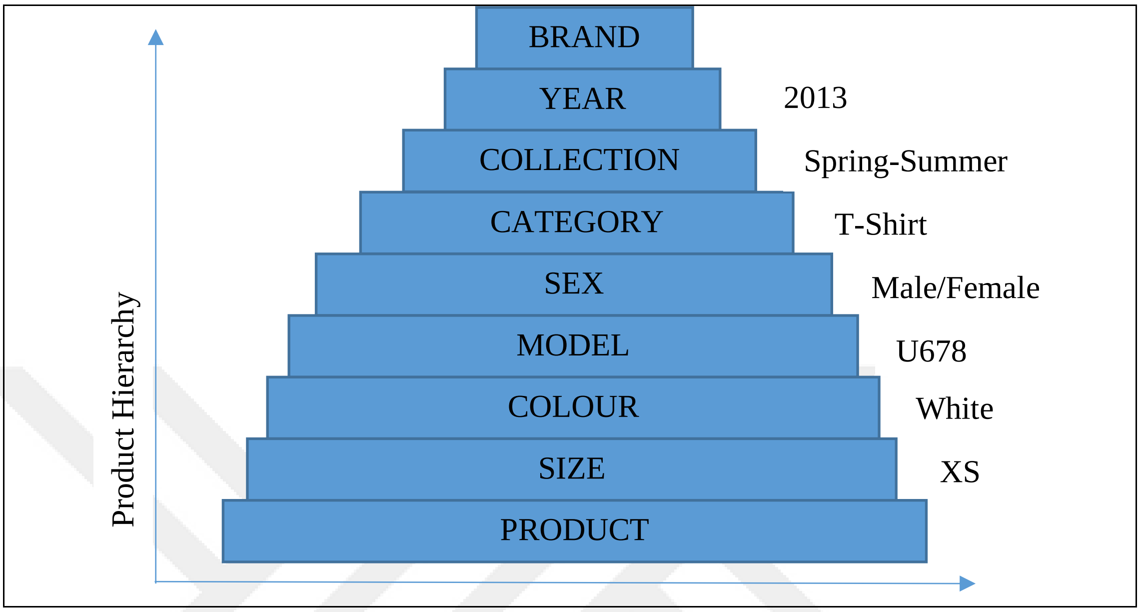
Sales forecasting is the first step of our study which has great importance on optimization results. The aim of sales forecasting is to predict the future periods demand as accurate as possible. In this study historical sales data from an apparel retailer in Turkey (which will be called as The Company) was used to model the behavior of demand. To understand the demand characteristics and important features which affect the demand we used past historical observations.

The Company mainly sells on women’s and child’s wear and has 21 product groups. In this study, according to the past historical sales, the group that has the most product variety and performed most of the sales were selected. Two years of historical data which belongs to this group was collected. The company has more than 30 stores in the big cities of the country and mainly sells the products by two sales campaigns which are called Autumn-Winter and Spring-Summer.

In our examinations on historical data, we figured out that sales amount varies from day to day within weeks and week to week within the entire sales season. Because markdown decisions are usually taken weekly, we decided to divide historical data into weekly time frames and focused on the monthly seasonality. This also helps us to increase our forecasting accuracy. Another seasonal effect on demand is special days like pre-Christmas, Valentine’s day or Ramadan Fest. Although special days such as Christmas are fixed, Ramadan is celebrated in different dates which shifts back every year. That’s why we focused on Spring-Summer campaign that includes Ramadan Fest in 2013 and 2014.

The company uses hierarchical structure to organize its product portfolio. This hierarchy is depicted in Figure 3.1.

Figure 3.1: Product Hierarchy



Based on this hierarchy, product resides at the lowest level and generally it is called as SKU (Stock Keeping Unit). Another variable that is not found in the hierarchy is the Reference. The product models that has specific color is called Reference.

The examinations made from the historical sales data indicated that there are 3 different sales characteristics in the selling season. At first stage of the season sales increase depending on the time. At next stage sales start decreasing and the company tries to increase the sales level by applying planned markdowns. At last stage of the season, even though the company increases the discount rates, sales amount decrease and disappear. The first stage of the season is called Regular Season which the products are being sold with full or premium prices. The second period is called Planned Markdown period and the last period is called Clearance period. These different periods are depicted in Figure 3.2.

Figure 3.2: Typical Seasonal Sales

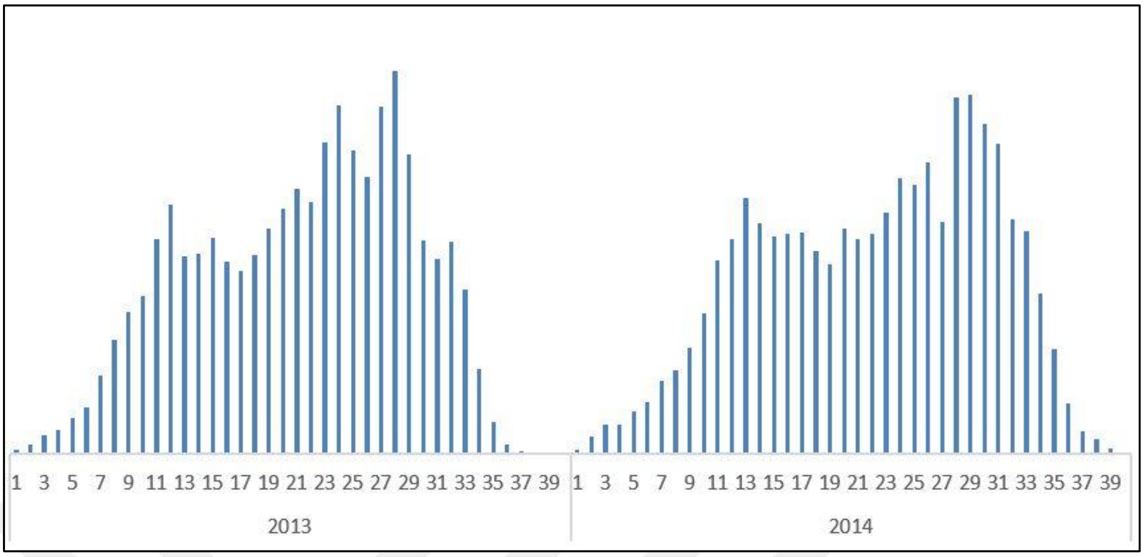
3.2 EXAMINATION OF MODEL PARAMETERS

Since many factor affects the demand in apparel retail, in this section possible effects of decision variables on demand were examined. The correlations between the decision variables and the sales quantities were examined based on descriptive statistics such as linear relationship, kurtosis and skewedness.

3.2.1 Sales Quantity

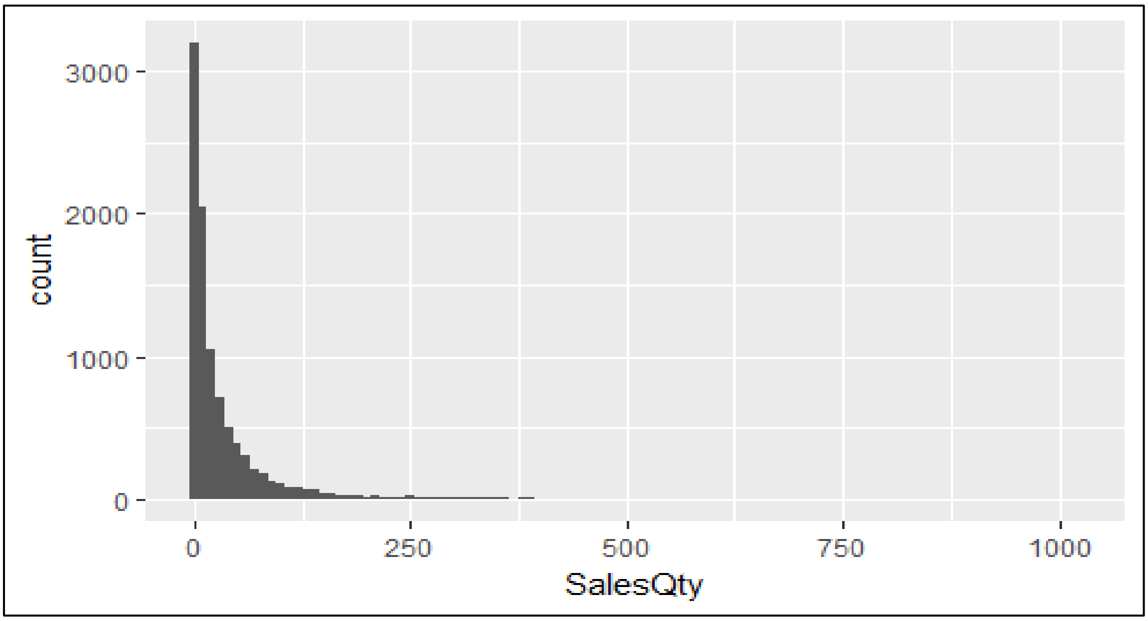
Time dependent variations in sales amounts were examined based on weekly framed historical data. Trends and variations over sales amount by time is depicted in Figure 3.3. It seems that sales distributions for the years 2013 and 2014 are similar. It also shows that sales characteristics are identical. In both years, sales distributions have more than one peaks. It is assumed that the last and the highest peak is related with Ramadan Fest.

Figure 3.3: Sales variations in season in 2013 and 2014



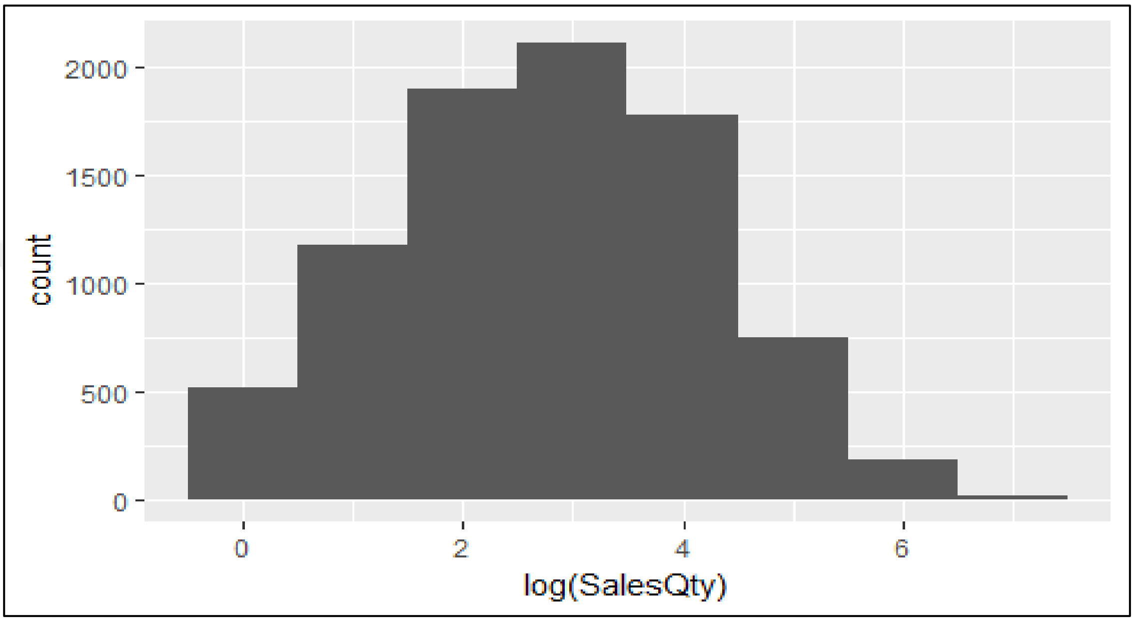
Histogram of the sales quantities of products which is given in Figure 3.4 indicates that distribution of the sales quantities is non-normal and highly right skewed. In the forecasting model a log transformation is needed to deal with this problem.

Figure 3.4: Histogram of Sales Quantity



After logarithmic transformation of the sales quantities, the results which is given in Figure 3.5 seem much better than it was used to be.

Figure 3.5: Histogram of Sales Quantity after logarithmic transformation



3.2.2 Price Elasticity

Many studies which explained in Section 2 specify that main driver of sales in any retail product is the price. As seen in Figure 3.2, retailers generally use the price (or discounts) to control the level of the demand. Our historical data shows that during the planned markdown period, when discounts increase sales quantities also increase. But during the clearance period, even though discount rates increase, sales quantities seem to negatively affected. It is thought that lack of product availability and decreasing of customers' intention to purchase is the reason for this negative effect.

In our study, we expect that price and demand are negatively correlated. As Caro & Gallien (2012) offered, instead of using the price directly in demand function, price rate is used.

$$PR_{t,m} = LN\left(\frac{p_{t,m}}{p_m^F}\right) \tag{3.1}$$

$PR_{t,m}$ indicates the price rate of a product-model m at week t.
p_m^F: Full price (the selling price at the beginning of the season) of a product-model m.
$p_{t,m}$: Price of a product-model m at week t.

3.2.3 Age

Age is another variable which is assumed to affect the demand. Zhao & Zheng (2000) showed that willingness of customers to pay full price decreases over time. To explain this behavior, age is used in the forecasting model. The Age of a product-model indicates the number of weeks that has passed from the beginning of the sales season.

3.2.4 Initial Purchase

Another important factor to explain the demand in Caro & Gallien's (2012) model is the initial purchase amount of a product. It is the total number of units ordered to the supplier before the selling season starts. Since we don't have this information, we calculated the initial purchase amount by summing all of the sales amounts in all seasons for a product.

3.2.5 Broken Assortment

In apparel industry, assortment means the quantity distribution of a product sizes. It usually is related with the distribution of the sizes of potential customers. Sometimes customers who are looking for a product which they would like to buy cannot find the appropriate size, even though other sizes of the product exist in the shelves. This is called broken assortment. Zhang & Fitzsimons (1999) indicated that broken assortment makes customer unhappy. Smith & Achabal (1994) found that broken assortment leads to loss of sales.

In order to measure the effect of the broken assortment, a threshold level was determined. The threshold level was formulated by multiplying the number of stores which the product was being sold in, the number of colors and the number of sizes which the product-model had in the season. Then, inventory level of the product-model is divided by this threshold.

As Smith & Achabal (1994) stated, inventory levels that is greater than the threshold was assumed as 1.

Broken Assortment effect is formulated as in equation 3.2.

$$BA_{t,m} = min\left(1, \frac{I_{t,m}}{C_m * S_m * N_{t,m}}\right) \quad \textbf{(3.2)}$$

$BA_{t,m}$: Broken assortment of a product-model m at week t.
$I_{t,m}$: Inventory level of a product-model m at week t.
C_m: Number of colors of a product-model m.
S_m: Number of sizes of a product-model m.
$N_{t,m}$: Number of stores of a product-model m is displayed at week t.

3.3 FORECASTING MODEL

In order to achieve better markdown decisions, it is necessary to forecast the future demand accurately. There are many demand models which was summarized in Section 2. In this study, we developed a new model with the guidance of Caro's model. The least square method of multivariate linear regression analysis was used. Unlike Caro's demand model, we didn't use the previous week's demand as a decision variable in our model. Because demand of this week is highly correlated with the demand of previous week. Putting this variable into the model caused high amount of variance to be explained only by this decision variable. This reduced the importance of the other factors.

As previously described, there exists a non-linear relationship between independent variables and dependent variable. To deal with the non-linearity, it seemed that additional transformations was needed. Log transformations have potential to correct the heteroscedasticity. And also in many studies (e.g. Smith & Achabal 1994, Caro & Gallien 2012) log-linear models were used. Based on our historical data, log-linear model explained variability on demand much better than many other models. Unlike Caro's model, we used month and special day dummy variables to explain the variability on sales amount during the season. Therefore, log-linear model was selected in this study. The notations of the variables used in the model are given in Table 3.1.

Table 3.1: Notations Used in the Model

t	Week index $t \in \{1,2,..T\}$,
m	Product-Model index $m \in \{1,2,..M\}$,
$D_{t,m}$	Demand of product-model m at week t,
$A_{t,m}$	Age of product-model m at week t,
O_m	Initial order quantity of product-model m given to the supplier,
K_i	Seasonal dummy of month i, $K \in \{0,1\}$ $i \in \{1,2,..12\}$,
G^r	Special days dummy of Ramadan Fest (r) $G \in \{0,1\}$
$\mathcal{E}_{t,m}$	Error term

Model is formulated in equation 3.3.

$$
\begin{aligned}
LN(D_{t,m}) = \beta_{0,m} + \sum_{i=1}^{12} \beta_i K_i \ + \beta_{13}(A_{t,m}) + \beta_{14}(O_m) \\
+ \beta_{15} LN\left(min\left(1, \frac{I_{t,m}}{C_m S_m N_{t,m}}\right)\right) + \beta_{16} LN\left(\frac{p_{t,m}}{p_m^F}\right) + \beta_{17} G^r \\
+ \mathcal{E}_{t,m}
\end{aligned}
\qquad \mathbf{(3.3)}
$$

Based on regression equation in 3.3, all β coefficients are constant including regression intercept. Directly using this regression intercept will cause poor forecasts since forecasting model generates prediction at product-model level using a constant intercept for all products. To deal with this problem, we need to estimate β_0 at each product-model level separately. To do this, we subtract constant β_0 from the left-hand side of the equation and reach the new estimates $LN(\widehat{D}_{t,m})$. Formulation of new variable is given in equation 3.4.

$$LN(\widehat{D}_{t,m}) = LN(D_{t,m}) - \beta_0 \qquad \mathbf{(3.4)}$$

We apply additional linear regression between $LN(\widehat{D}_{t,m})$ and actual sales data for each product-model and use the intercept of new linear regression as our $\beta_{0,m}$. We calculated the R^2 of the regression models to see if the results were good. By using the coefficient of determination formulation in equation 3.5, R^2 values were calculated.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad \textbf{(3.5)}$$

Table 3.2: Comparing the coefficient of determinations of the two model

$R^2$of general model	0.78
$R^2$of individually $\beta_0$updated model	0.80

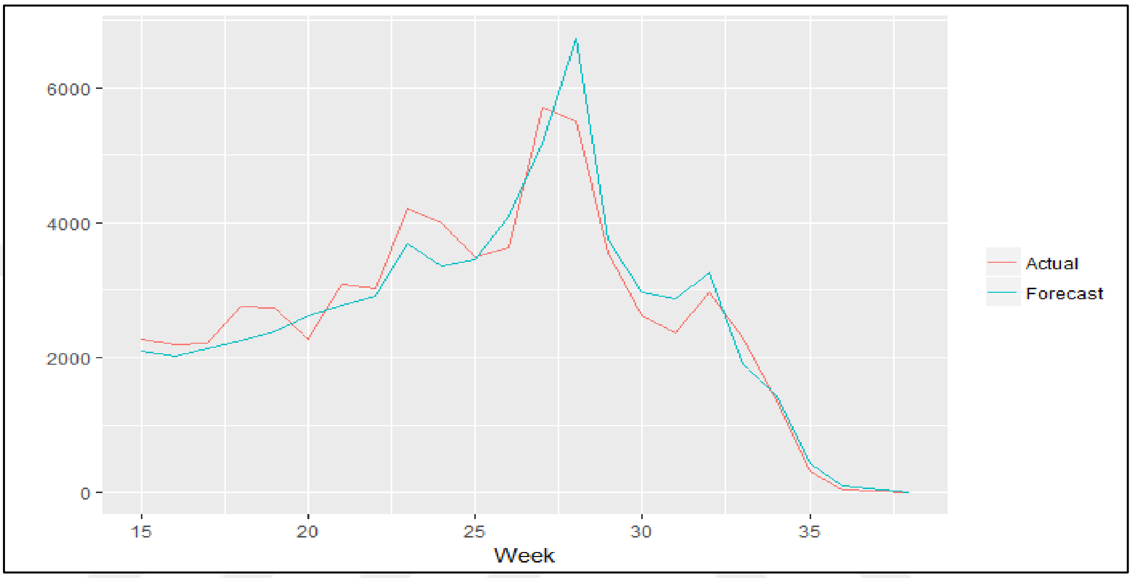
It seemed that using the individual product-model's regression intercept was increased the variance explained by all variables. Regression model statistics are given in Table 3.3.

Table 3.3: Regression Model Statistics

Independent Variable	**Estimate**	**Std. Error**	**t-value**	**p-value**	**Sig.**
Initial Purchase	1.12697	0.02456	45.878	< 2e-16	***
Broken Assortment	0.52016	0.04439	11.717	< 2e-16	***
Age	0.05278	0.01729	3.053	< 2e-16	**
Price Rate	-0.75758	0.15514	-7.819	1.80e-14	***
Ramadan Fest	0.28205	0.07164	3.937	9.01e-05	***
Month 4 (April)	3.71204	0.67634	5.488	5.54e-08	***
Month 5 (May)	3.51505	0.64718	5.431	7.55e-08	***
Month 6 (June)	3.56376	0.61586	5.787	1.05e-08	***
Month 7 (July)	3.39456	0.59102	5.744	1.34e-08	***
Month 8 (August)	2.77631	0.56883	4.881	1.29e-06	***
Month 9 (September)	2.21320	0.55933	3.957	8.31e-05	***
R^2: 0.80	DF: 755				

Figure 3.6 shows the forecasting results. Figure 3.6 indicates that model fits well at the clearance period. But it seems that model underestimates the demand at the first planned markdown period and overestimates the demand at high season which consists of both Ramadan Fest and price discounts.

Figure 3.6: Forecasting results



Results of the forecasting model indicates that β coefficients should be updated when the new observations arrived.

3.3.1 Updating Parameter Estimates

When the new observations arrive, updating the parameters have potential to increase the forecasting accuracy. Figure 3.2 indicates that in the regular season period no major discounts introduced. Therefore, Price rate is almost 1 and it is thought that all products are fully displayed in the stores. Hence, it is thought that price rate and broken assortment has no effect on demand in regular period. Since initial purchase quantity is constant over the season and seasonal pattern does not change frequently, only the base demand rate β_0 is effective in regular season. That's why, only updating the β_0 for each product-model is enough in regular season. After week 13, retailer starts applying price discounts.

Our updating methodology is twofold. Regular season is the learning period for the β_0, rest of the season is the learning period for Price Rate coefficient. We assume that updating the seasonal and the special day coefficients would not contribute to the accuracy of forecasts in the current season. It would contribute to the next years' predictions.

We tried two different methods to update β_0 coefficient. In each of these methods, since it is a learning process, system waits for the first signals of the new season, system does not generate any forecast at the first three weeks. After 4th week system tries to estimate the β_0 parameters for each product-model. First method assumes that initial intercept level of the demand is stable and does not oscillate too much. Therefore, the system tries to estimate next week's parameter by taking the average of the last three week's β_0 parameter. The second method assumes that initial intercept level might change over time and there might be significant trend on that parameter. The system tries to predict the next week's parameter by applying regression analysis of historical intercept levels. By using this estimated parameters, we forecasted the next week's demand. We calculated the percentage errors to compare the methods reliability.

$$Percentage\ Error\ (PE) = \frac{(Forecast - \ Actual)}{Actual} \quad \textbf{(3.6)}$$

Figure 3.7: Comparison of Two Estimator Method

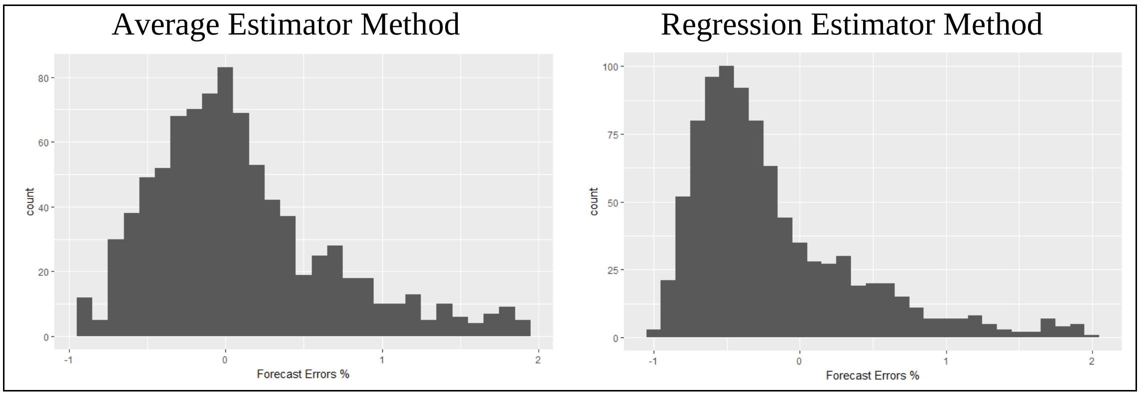


Figure 3.7 indicates that estimating the intercept levels with Average Estimator Method is better than Regression Estimator Method. In Average Estimator Method, forecast errors are normally distributed around zero percentage. Second method usually underestimates the demand. That's why we update intercept levels of the product-model's by using the Average Estimator Method.

After week 14th we start updating both intercept level and Price Rate coefficients. We first update intercept levels by using previously explained Average Estimator Method. After that we combine last year's historical data with the current year's data and subtract all the effects except Price Rate by using the following formulation.

$$\mathrm{LN}(\breve{D}_{t,m}) = LN(D_{t,m}) - \beta_{0,m} - \sum_{i=1}^{12} \beta_i K_i - \beta_{13}(A_{t,m}) - \beta_{14}(O_m) - \beta_{15} LN\left(min\left(1, \frac{I_{t,m}}{C_m S_m N_{t,m}}\right)\right) - \beta_{17} d^r \qquad \textbf{(3.7)}$$

After elimination of all the effects except the Price Rate, updated Price Rate coefficient is calculated using formulation in equation 3.8 with linear regression.

$$\breve{D}_{t,m} = e^{\beta_{16} LN\left(\frac{p_{t,m}}{p_m^F}\right)} \qquad \textbf{(3.8)}$$

Starting from week 14th we forecasted the next week's demand at product-model level by using the updated initial intercept and Price Rate coefficients. Summary statistics are given in Table 3.4.

Table 3.4: Comparison of Percentage Errors of Forecasting Methods

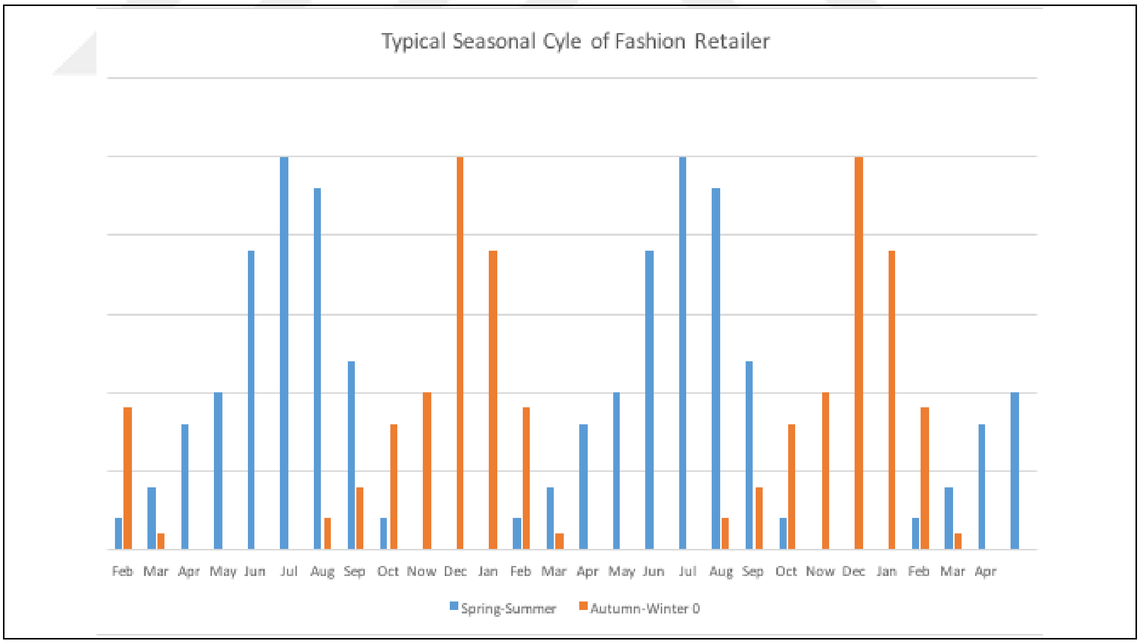
Statistics	β_0 parameter updated	Both β_0 and Price Rate updated
Min.	-0.94420	-0.94590
1st Quantile	-0.30019	-0.30646
Median	-0.01022	-0.02182
Mean	0.09384	0.07679
3rd Quantile	0.36529	0.34725
Max.	1.93943	1.93217

From Table 3.4, we conclude that updating both the initial intercept and the Price rate parameters is slightly better than only updating the initial intercept parameter.

3.4 BUSINESS RULES

The company has 30 stores and mainly sells women's and child's wear. The company has 21 product groups and majority of its products are manufactured abroad. Because of the long lead times, the next season's products are ordered one year in advance. In almost all products the company ordered fixed amounts of purchase orders. The company have two main selling seasons which are Autumn-Winter and Spring-Summer. Autumn-Winter season starts at the beginning of August and ends at the end of January. Spring-Summer season starts at the beginning of February and ends at the end of October. A typical seasonal transitions for the company is depicted at Figure 3.8.

Figure 3.8: Typical Seasonal Transitions for the Company



At the beginning of each selling season, merchandising planners agreed on budgets at product category level. When the selling season starts they try to realize the targeted budgets. Sometimes they need to revise their budgets based on the current season's performance. Generally, they focus on achieving the total budget rather than individual category level.

Company’s main driver to manage the demand is planned and liquidation discounts and generally these discounts are not announced. Since frequent discounts have negative effects on company’s image, they prefer not to introduce any discount at regular season. Merchandising managers pointed out that weather conditions, ads on TV’s and billboards and in store promotions have significant impact on sales. The timing of these activities are unknown in advance and they do not keep track of these information systematically. Hence, we faced some unidentified outlier sales amounts in historical sales data.

The company periodically review the performance of each individual product-model and apply discounts to the products that have higher risk in revenue. They can make discounts to clean the inventories or sometimes to increase the demand level. Sometimes they apply discounts just to make extra room in the stores for the upcoming season. In some cases, they prefer to make promotions rather than applying discounts. For example, making “buy 3 pay for 2” promotion is more effective on reducing the inventory levels than price discounts. Unfortunately, they do not put this promotion information in their ERP system.

Merchandising planners periodically review the sales and they update the prices based on the performances of the products. They use the rules generated by their experiences based on “Week Cover” metric to decide the markdowns. Week cover indicates that how many weeks of inventory left on hand. If week cover is greater than the remaining weeks to the end of the season, they decide to apply markdown. Generally, during the planned markdown period they apply 30%-50% discounts based on the initial sales price. But in the clearance period, discounts become more aggressive. The company uses predefined price sets for each of their products. These prices are psychological prices that is reduced by 5 cents (i.e. 19.95, 29.95 etc.). And each product has a minimum price which the company does not want to sell below this price.

At the end of the season, the company either send unsold inventory to the outlet stores or sell to the wholesalers. In the worst case scenario, they give the unsold inventory to charity. But this final decision is not known in advance.

In meetings with the merchandising planners, they indicated not to prefer to use an automated pricing system, rather they prefer to use scenario based decision support system to indicate the riskier products to be discounted.

In this study, with the business rules mentioned above we intended to develop a decision support tool helping merchandising planners to decide when and how much discount to apply for the products. We decided to develop a solution with a dynamic programming approach so that the state space is large and expected solution time is limited.

3.5 OPTIMIZATION MODEL

Markdown optimization models aim to achieve the maximum expected revenue by managing the demand rate with discounts that will be applied during the whole sales season. In this study, we assumed that demand of one product is independent of other products. And in a finite horizon, we tried to maximize the total expected revenue. We assumed that customers do not act strategically and whenever the price is less than their expectation they decide to purchase. Our next assumption was that the company is a monopolist and we ignored the competitor's actions. And there is no replenishment allowed during the sales season. As described in the previous chapter, our demand model is a function of time, price and inventory level at a given time. Our goal is to find the best feasible pricing policy of a product-model over the finite horizon.

We assumed that the demand rate of a product-model is independent of others and there is no substitution or cannibalization effect. And we can decide only one price at each time t. Our demand $D_t = D(p_t, I_t)$ is a function of price (p) and inventory level (I) at week t. Thus, we expect to gain $r_t = p_t \min(D(p_t, I_t), I_t)$ amount of revenue in each week t.

The simplest form of a deterministic optimization problem was offered by (Talluri & Van Ryzin 2006). This basic model is given in equation 3.9.

$$\max \sum_{t=1}^{T} p_t d(p_t) \qquad \textbf{(3.9)}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} d(p_t) \leq C$$

$$d(p_t) \geq 0$$

In this form, the model tries to maximize the total revenue in a finite horizon $t \in \{1..T\}$. Capacity C is the limiting constraint which is the initial purchase at the beginning of the selling season. We adapted this basic model to our case. Formulation of our optimization model as follows;

$$\max \sum_{i}^{n} \sum_{t}^{T} p_{i,t} \alpha_{i,t} \min\left(I_t, D\left(p_{i,t}, I_t\right)\right) - cI_T \tag{3.10}$$

s.t.

$$\sum_{i}^{n} \alpha_{i,t} = 1, \qquad \forall\, t \in \{1 \ldots T\} \tag{3.11}$$

$$I_{t+1} = I_t - \sum_{i}^{n} \mathrm{\alpha_{i,t}} \min(I_t, D(p_{i,t}, I_t)) \tag{3.12}$$

$$p_{i,t} \in \Omega \tag{3.13}$$

$$\alpha_{i,t} \in \{0,1\} \tag{3.14}$$

In this model, our objective is to maximize the total revenue. We penalize the remaining inventory (I_T) with a predetermined cost c at the end of the season. In this notation, $p_{i,t}$is the set of possible prices and $\alpha_{i,t}$ is the binary decision variable (0 or 1) which represents the selected price in the price set Ω at time t. Equation 3.11 assures that only one price can be selected at a time t. Equation 3.12 gives the inventory levels in the rolling horizon. $D\left(p_{i,t}, I_t\right)$ is the demand at price $p_{i,t}$ with an inventory level I_t at week t.

Since the objective function is in non-linear form we preferred to use dynamic programming approach instead of using a non-linear solver.

3.6 DYNAMIC PROGRAMMING

Dynamic programming is a method for solving a large problem by separating it into smaller sub-problems, solving them just once and stores them into the memory. Whenever we need to solve a sub-problem which has already been solved, instead of solving it again, we only get the solution from the memory. The idea in dynamic programming is to divide our big problem into smaller sub-problems in such a way that we can achieve the optimal solution of the larger problem by summing up the optimal solutions in the sub-problems.

These are the elements of a dynamic programming model;

Stage: Smaller subset of a large problem which is independent over others.

State variable: A state includes the information to make a decision. We solve the problem moving from one state to another between stages by searching the most effective decisions.

Decision variable: Set of possible actions that we control the whole process.

Transition function: The function which describes how system evolves from one stage to another.

Objective function: This function indicates the cost or reward to be achieved.

To model a problem in dynamic programming, one should divide the problem into the smaller and independent stages. The aim of dynamic programming algorithm is jumping from current stage to another based on a decision in such a way that this transaction gives the maximum reward (or minimum cost). If transitions between each state is known with certainty, then the model is called deterministic otherwise it is called stochastic. Basic notation of a deterministic dynamic programming model is of the form in equation 3.15.

$$V_t = \max(or \min)\ \{reward\ (or\ cost) during\ current\ stage + V_{t+1}\} \qquad \textbf{(3.15)}$$

Since our time intervals were weekly basis, we specified weeks as our stages and the inventory level in each stage as our states. The price that we choose in each state was our decision variables. Demand $D(p_t, I_t)$ was our transition function. Our objective was to reach the maximum total revenue.

3.6.1 Deterministic Model

In our deterministic model, we assumed that we knew the demand with certainty. I_t is the number of units of inventory on hand at week t. Value function V_t is the revenue gained at week t. For each individual product-model our dynamic programming model is of the form in equation 3.16.

$$\begin{aligned} V_t(I_t) = \max_{\substack{p_t \in \Omega, \\ p_t \le p_{t+1},}} \ & \{p_t \min(I_t, D(p_t, I_t)) \\ & + V_{t+1}(max\{0, I_t - D(p_t, I_t)\})\}, \qquad \forall t, t = 1, \dots, T, \\ V_T(I_T) = -cI_T & \end{aligned} \qquad \textbf{(3.16)}$$

In equation (3.18), p_t represents the price at week t, Ω represents the available price sets. $D(p_t, I_t)$ accounts for the demand at week t if we have I_t units of inventory and the price p_t is selected. In each stage our possible next move depends on the weekly demand D_t and inventory level I_t at the beginning of the week t.

We assumed that at the end of the season (at week T) remaining inventories has constant cost of c per unit and each inventory left have negative impact on total revenue. The aim of this model is to maximize the total income achieved during the whole selling season.

3.6.2 Stochastic Model

Since our forecasting algorithm has errors, the next state that we will go to from the current state is not known with certainty. Instead, all we know is the probability of moving to the next states. Wayne L. Winston (2004) formulated stochastic formulation of dynamic programming into the form in equation 3.17.

$$V_t(I_t) = \max_{p_t \in \Omega} \left\{ \mathbb{E}(R_t | I_t, p_t) + \sum_j \mathbb{P}(j | I_t, p_t, t) V_{t+1}(\max(0, I_t - (j | I_t, p_t, t))) \right\} \tag{3.17}$$

In this formulation, we calculate the expected revenue in each stages using the probability distribution of $\mathbb{P}(j|I_t, p_t, t)$. These are our transition probabilities. The aim is to calculate a pricing policy that shows how to act optimally when we face of an uncertainty. We use forward dynamic programming which visits any of the possible states at each stage. Solution methodology of a dynamic programming is as follows;

Step 0: Initialize the contribution of last week $T. V_T(I_T)$

Set $t = T - 1$.

Step 1: Calculate the following

$$V_t(I_t) = \begin{matrix} max \\ p_t \in \Omega \end{matrix} \left\{ \mathbb{E}(R(I_t, p_t)) + \sum_j \mathbb{P}(j | I_t, p_t, t) V_{t+1} \right\} \tag{3.18}$$

For all possible I_t

Step 2: If $t > 0 \; then \; t = t - 1$ and GOTO Step 1 Else STOP.

We start with stage T (at the end of the selling season), initialize the contribution of Week T. After that we go to stage $T - 1$ and calculate the V_t for all possible I_t. $\mathbb{P}(j|I_t, p_t, t)$ is our transition probabilities from moving stage t to $t + 1$. In deterministic solution we have only one transition probability and it is always 1. $\mathbb{E}(R_t|I_t, p_t)$ is our reward at current stage t.

3.6.3 The Curse of Dimensionality

We need to decide the optimal price p_t at each state I_t. In our example data, inventory level of a product-model goes up to thousands of units and each product-model have a possible price set Ω which consist of tens of possible prices. For example, for a product-model which has initial price of 69.95, all possible prices are 69.95, 64.95, 59.95, 54.95, 49.95 and so on. Limiting price that limits us to sell a product is 9.95.

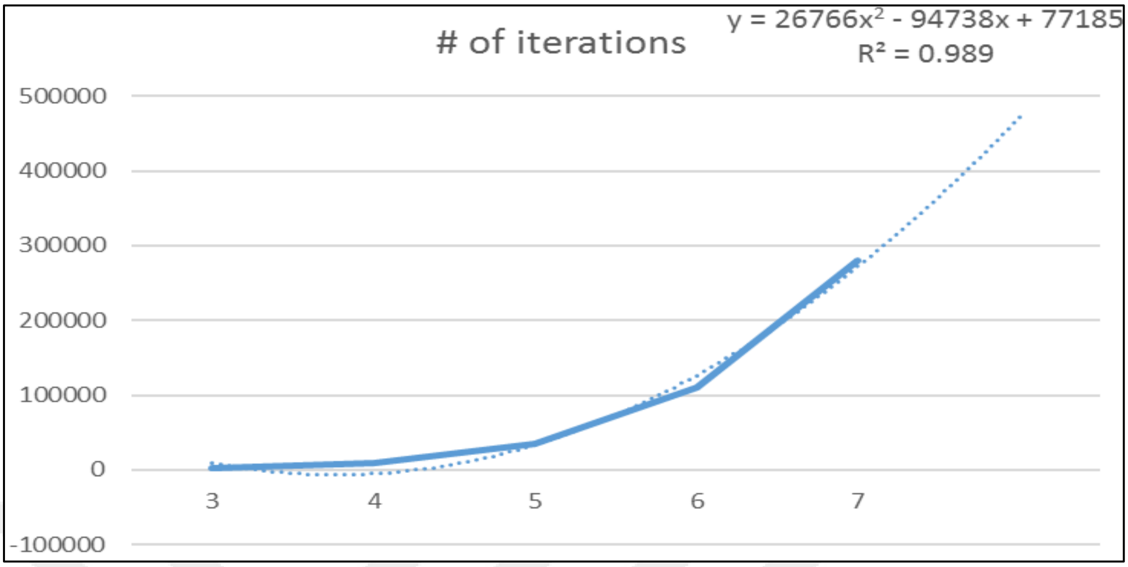
Because of "the curse of dimensionality", number of iterations (states visited) to reach the optimal results even in deterministic case increases dramatically with the number of prices in price set. Number of iterations needed to reach the optimal solution can be seen at Table 3.5.

Table 3.5: Number of iterations needed to reach optimal solution

Price Set	**3**	**4**	**5**	**6**	**7**
initial price	69.95	69.95	69.95	69.95	69.95
possible discounted prices	34.95	34.95	44.95	49.95	54.95
	14.95	24.95	34.95	39.95	49.95
		14.95	24.95	24.95	39.95
			14.95	14.95	24.95
				9.95	14.95
					9.95
# of iterations	**1.784**	**9.104**	**35.441**	**110.054**	**280.597**

When we add an additional new price to our price set, the number of iterations needed to solve the problem optimally increases as can be seen in Figure 3.9. Even in deterministic case with 8 price available, it is expected to make nearly 500.000 iterations to reach the optimal solution.

Figure 3.9: Number of iterations needed to reach optimal solution with 7 price



In our case, we have to decide the discount timings and optimal prices when the clearance period starts. Clearance period starts around week 20th and season ends at week 35th. This means we have 16 weeks, and nearly 5 to 8 prices and thousands of discrete inventory levels to reach the optimal solution. In practice, it is not possible to solve our problem by applying stochastic dynamic programming directly within a feasible time frame. By the way, we know that assuming the demand as deterministic and ignoring the uncertainty might affect the model output negatively. Considering these problems, we decided to apply some approximations to solve our stochastic model. To solve the problems that have large state space with stochastic dynamic programming, Powell (2007) offered some approximations.

3.6.4 Approximate Value Iteration

The idea in approximate value iteration is choosing the decision variable at the current stage without the need to visit all next possible stages, instead we use approximate expected value of the next stage. By using this methodology of course, we accept the risk of choosing the sub optimal solution.

Let $\bar{V}_t(I_t)$ be our approximate value in state I_t. We can decide our decision variable using formulation in equation 3.19.

$$p_t = \arg\max_{p_t \in \Omega}(\mathbb{E}(R(I_t, p_t)) + \mathbb{E}(\bar{V}_{t+1}(I_{t+1}))) \quad \textbf{(3.19)}$$

The challenge of using this approximation technique is finding good approximation of $\bar{V}_t(I_t)$. Powell (2007) offered many different versions of approximate value iterations by using lookup tables methodology. Lookup table methodology uses the basic idea of estimating the value of each states based on randomly selected paths. For our problem we used the post-decision state variable approach. The post-decision state is the state of the system after that we have made a decision but before any new information has arrived. In another words, we firstly decide our decision variable (in our case "which price to go") then using the new unknown information (in our case "how many items will be sold?") we determine which state to go. Basic steps of this algorithm is as follows;

Step 0: Initialization

a) Initialize $\bar{V}_t^0(I_t)$ for all states I_t.
b) Choose an initial state I_t^1.
c) Set $n = 1$.

Step 1: For $\mathrm{t} = 0,1,2 \ldots . \mathrm{T}$ DO

a) Solve equation 3.20.

$$\hat{v}_t^n = \max_{p_t \in \Omega} \left\{ \mathbb{E}(\sum_j \mathbb{P}(I' | I_t^n, p_t^n)(R(I_t^n, p_t^n) + \bar{V}_{t+1}^{n-1}(I')) \right\} \quad \textbf{(3.20)}$$

b) Update $\bar{V}_t^{n-1}(I_t)$ using equation 3.21.

$$\bar{V}_t^n(I_t) = \begin{cases} (1-\alpha_{n-1})\bar{V}_t^{n-1}(I_t) + \alpha_{n-1}\hat{v}_t^n, & I_t = I_t^n \\ \bar{V}_t^{n-1}(I_t), & otherwise \end{cases} \quad \textbf{(3.21)}$$

c) Choose a random path ω^n

d) Compute $I_{t+1}^n = I^M(I_t^n, p_t^n, W_{t+1}(\omega^n))$.

Step 2: $n = n + 1$. If $n < N$ GOTO Step 1.

In this model, at the time we are in a state I_t, we make our decision on p_t and then observe the new information W_{t+1}, than we move on to the next state I_{t+1} randomly. This is called the transition function I^M. Our transition function is $I_{t+1} = I^M(I_t, p_t, W_{t+1})$. In our case W_{t+1} consists of demand uncertainty or in another words unknown fluctuations in demand which is a random number. After N iterations, our expectation is to reach the realistic approximations by updating the value of each states which belongs to the sample paths. In addition to this flow, we add additional random component to make the model more realistic. In each iteration at Step 1, algorithm decides the optimum price which is equal or less than the previous stage's price. We forced algorithm to decide unexpected pricing decisions which leads the algorithm to visit as much possible as states in our state space. Thus, our lookup table have some ideas when merchandising planners decide an unexpected pricing decision in real life scenario. We changed equation 3.20 into the form at equation 3.22.

$$\hat{v}_t^n = \begin{cases} \max\limits_{p_t \in \Omega} \left\{ \mathbb{E}\left(\sum_j \mathbb{P}(I' | I_t^n, p_t^n)(R(I_t^n, p_t^n) + \bar{V}_{t+1}^{n-1}(I') \right) \right\} if \rho = 1 \\ \mathbb{E}\left(\sum_j \mathbb{P}(I' | I_t^n, \dot{p}_t^n)(R(I_t^n, \dot{p}_t^n) + \bar{V}_{t+1}^{n-1}(I') \right) otherwise \end{cases} \quad \textbf{(3.22)}$$

In equation 3.22, ρ is a Bernoulli random variable which has success probability 0.7 and $\dot{p}_t^n$ is randomly selected price from price set Ω.

In this simulation-like methodology, it is important to find the realistic initial values for $\bar{V}_t^0$ to achieve the better result and reduce the time needed to converge values to realistic values. To fill the initial values more intelligently, we tried to find a value function which represents our revenue function.

3.6.5 Value Function Approximation

The idea in approximating value functions is to find a good function and estimators to explain the behavior of our revenue maximization model. In lookup table methodology, we tried to estimate the values of all possible states in our state space. We updated the values whenever we visited each states. Naturally, if we fill the initial values randomly then it will take more time that our values converge to realistic values.

In this application, we used least square method to reach an approximate value function from the data. Week, price and stock level variables are selected to explain the value of each states. Our deterministic model already searched for the global optimal solution by visiting all possible states. These log records of our deterministic solution was given to the least square method.

Our regression model can be seen in equation 3.23.

$$\bar{V}_{t,p,I} = \beta_0 + \beta_1 t + \beta_2 p + \beta_3 I + \beta_4 (t * I) + \beta_5 (t * p) + \beta_6 (I * p) + \beta_7 (t * p * I) \quad \textbf{(3.23)}$$

In equation 3.23, t represents weeks, p represents the price and I represents inventory levels at week t. For an example product-model, statistics of regression results can be seen in Table 3.6.

Table 3.6: Regression results of value function for an example product-model

	Estimate	Std. Error	t value	Pr(>|t|)	Sig.
(Intercept)	-5.95E+02	2.32E+02	-2.567	0.010596	*
t	2.37E+01	9.01E+00	2.631	0.008834	**
I	1.51E+01	1.55E+00	9.757	< 2e-16	***
p	6.41E+01	1.15E+01	5.567	4.67E-08	***
$t * I$	-3.83E-01	6.01E-02	-6.385	4.63E-10	***
$t * p$	-1.82E+00	4.48E-01	-4.062	5.83E-05	***
$I * p$	3.00E-01	7.68E-02	3.911	0.000108	***
$t * p * I$	-8.47E-03	2.99E-03	-2.836	0.004796	**
R^2: 0.9507	R^2_{adj} : 0.9498				

We do this approximation for all product-model pairs and calculated the initial values of $\bar{V}_t^0(I_t)$.

3.7 APPLICATION

In the application phase, a sample product-model was selected and both deterministic and approximate value iteration methodologies were applied. From the guidance of historical data, we found out that the company applied its first markdown to the category at week 22nd and also deterministic solution didn't offer any discount until week 22. For these reasons, initial week for the markdown application was determined as week 21st. And end of the season was defined as week 35th.

Inventory left at the end of the season causes additional operations to be completed. First, the company have to decide the channel which the inventories to be sent. If this channel is a wholesaler than they have to interview with the wholesalers and agree on the terms of conditions. If the channel is outlet stores or warehouse than there will be additional transportation, handling and holding costs. For these reasons, inventory leftover is a headache for the company. It is not only the money paid for the products, there are extra invisible operational costs. The cost of inventory left over at the end of season should be defined as high as possible. That's why it was defined half of the initial (full) price of each product-model.

For the stochastic model it is needed to define the iteration size n. Because of the high computational time N was defined as 100. 100 iteration seemed not enough for the lookup table values to converge the actual ones, in each week 100 new iteration was repeated from the starting week to week 35th and previously updated lookup table updated again. Thus, at the end of the season, lookup table values were updated 1500 times. Smoothing parameter α was set to 1 for the lookup table values to be converged as fast as possible.

We needed to fix some issues while implementing the model. For example, by nature of the algorithm, it is not possible to visit all states during the iterations. This can lead to some meaningless results. In example given at Table 3.7, values of some unvisited stages might become higher than expected.

Table 3.7: Meaningless values of unvisited stages

Inventory	Price	Week	Expected Revenue	# of visit
1450	**14.95**	**33**	**10,178**	**4**
1449	14.95	33	11,233	0
1448	14.95	33	11,233	0
1447	14.95	33	11,233	0
1446	14.95	33	11,233	0
1445	**14.95**	**33**	**9,700**	**3**

In Table 3.7, there are two states visited and rest of the state's values have not been updated. But unvisited states seem to have higher contributions even though they have less inventories. To correct this situation, unvisited state values updated based on the closest two visited state's values. In this example, $v_{t,p,1450}$ and $v_{t,p,1445}$ were used to correct unvisited state values. Unit contribution of each values ($\bar{v}$) are calculated by dividing values to their inventory levels as seen in equation 3.24.

$$\bar{v}_{t,p,I} = \frac{v_{t,p,I}}{I} \quad \textbf{(3.24)}$$

The average of two closest and visited state's unit contribution is multiplied by unvisited state's inventory level. The results obtained were accepted as the new values of the unvisited states.

Table 3.8: Updated values of unvisited states

Inventory	Price	Week	Expected Revenue	# of update
1450	**14.95**	**33**	**10,178**	**4**
1449	14.95	33	9,949	1
1448	14.95	33	9,942	1
1447	14.95	33	9,935	1
1446	14.95	33	9,928	1
1445	**14.95**	**33**	**9,700**	**3**

Deterministic optimal policy was found by using the formulation in equation 3.18. This model was called as Model 1. Very similarly, stochastic optimal policy was found by using the formulation in equation 3.25. The only difference from the deterministic solution is that the expected value of the next state is taken directly from the lookup table. Stochastic model was called as Model 2.

$$V_t(I_t) = \max_{\substack{p_t \in \Omega, \\ p_t \leq p_{t+1},}} \{p_t D(p_t, I_t) + \bar{V}_{t+1}(min\{I_t, I_t - D(p_t, I_t)\})\}, \quad \forall t, t = 1, \ldots, T, \qquad \textbf{(3.25)}$$

In order to compare the two pricing policies fairly, we used the historical data as our base. We excluded the effect of pricing and broken assortment from the sales history by using the coefficients of both parameters in forecasting model. Equation 3.26 gives the formulation of the adjustment using historical sales quantity.

$$Q_t^{adj} = e^{LN(Q_t) - \beta_{15} * LN\left(\min(1, \frac{I_t}{Inv.Threshold})\right) - \beta_{16} * LN(\frac{p_t}{p^F})} \qquad \textbf{(3.26)}$$

After calculating the adjusted sales quantity, we add up the optimal pricing decisions on top of the adjusted sales quantity to reach the sales quantities under the optimal pricing policy by using the equation 3.27.

$$Q_t' = e^{LN\left(Q_t^{adj}\right) + \beta_{15} * LN\left(\min\left(1, \frac{I_t'}{Inv.Threshold}\right)\right) + \beta_{16} * LN(\frac{p_t'}{p^F})} \qquad \textbf{(3.27)}$$

In this notation superscript $'$ represents the values under the optimal pricing policy.

Both Model 1 and Model 2 run and results were calculated based on their optimal pricing policy. Results are given at Table 3.9. Net Revenue values were calculated by using the formulation 3.28.

$$Net\ Revenue = \sum_{t}^{T} R_t - cI_T \tag{3.28}$$

In this notation R_t represents revenue gained at week t, cI_T represents the cost of remaining inventory at the end of the season.

Table 3.9: Comparison of optimal policies for individual product-model pairs

	Actual		Model 1 (Deterministic)		Model 2 (Stochastic)	
Product	**% of Inv. Left**	**Net Revenue**	**% of Inv. Left**	**Net Revenue**	**% of Inv. Left**	**Net Revenue**
E1L13	3.0%	32,301	4.1%	**36,092**	2.5%	33,135
E1095	0.8%	4,704	0.0%	**4,884**	2.3%	4,835
E1090	0.2%	8,160	8.9%	7,794	1.2%	**8,441**
J1Z50	1.7%	25,579	4.7%	**27,512**	5.9%	26,464
J4Z63	1.5%	34,949	5.1%	**37,720**	3.7%	35,883
J1L02	2.9%	**4,326**	1.0%	4,200	29.5%	2,805
J4398	6.1%	5,600	0.0%	**6,029**	15.2%	5,602
E1069	3.7%	7,548	6.6%	**8,498**	11.3%	7,820
J7182	7.7%	5,753	0.0%	5,727	4.6%	**6,733**
E1081	0.6%	6,269	0.0%	**6,479**	0.3%	5,974
E8009	6.8%	**3,352**	14.1%	2,963	11.7%	2,998
E1L34	1.5%	14,400	0.0%	15,250	5.1%	**15,522**
J1041	12.2%	10,272	3.6%	**11,739**	10.8%	11,091
J1059	2.3%	13,931	7.6%	**14,909**	17.5%	13,574
E1067	13.8%	4,690	8.2%	**5,173**	9.9%	5,014
E1085	0.9%	5,293	2.7%	**6,124**	0.0%	3,863
E1102	0.5%	8,146	2.2%	8,012	0.1%	**8,290**
E1110	3.0%	8,035	9.8%	**8,727**	16.7%	7,988
J1047	3.4%	8,155	3.0%	**8,700**	5.6%	8,266
J1048	7.8%	**2,787**	0.0%	2,690	7.1%	2,721
J1L49	2.2%	6,407	0.0%	**6,537**	4.7%	6,373
J1Z29	27.4%	5,185	9.1%	**6,143**	21.7%	5,678
E1063	0.4%	15,865	8.6%	**16,595**	6.2%	16,218
E1078	1.1%	9,803	1.3%	**10,131**	5.9%	9,708
E1099	0.3%	8,121	0.9%	**9,246**	0.0%	7,676
E1106	0.3%	**23,046**	4.9%	22,541	0.0%	21,411
E1089	1.6%	8,412	0.0%	**9,011**	2.2%	8,587
J1Z90	2.9%	30,796	0.0%	29,713	0.0%	**31,128**
J4Z23	2.8%	**43,346**	0.0%	41,762	0.0%	43,212

J7Z80	3.2%	19,816	0.0%	20,483	7.2%	**21,900**
E1112	0.8%	**1,917**	5.8%	1,758	0.0%	1,857
E1L22	0.0%	13,235	0.0%	**13,455**	1.5%	13,380
J3Z56	1.5%	90,689	1.8%	**95,779**	0.0%	91,511
TOTAL	**2.8%**	**490,887**	**2.9%**	**512,375**	**3.9%**	**495,656**
				4.4%		**1.0%**

In Table 3.9, maximum revenue achieved among all models are labeled in bold for each product-model. Model 1 reached the highest revenues among the 22 out of 34 products. Percentage of inventory left over is very close to the actual policy. Overall, Model 1 gained 4.4% higher revenue than actual pricing policy. Model 2 gained 1.0% higher revenue than actual policy but achieved the worst performance based on percentage of inventory left over. To compare models individually with the actual pricing policy "% of Revenue Difference" is calculated by using equation 3.29.

$$\% \, Difference \; in \; Revenue = (R_{model} - R_{actual})/R_{actual} \tag{3.29}$$

Figure 3.10 and 3.11 indicates that in many cases both Model 1 and Model 2 performs better than actual policy.

Figure 3.10: Comparison of the Model 1 with the Actual Revenues

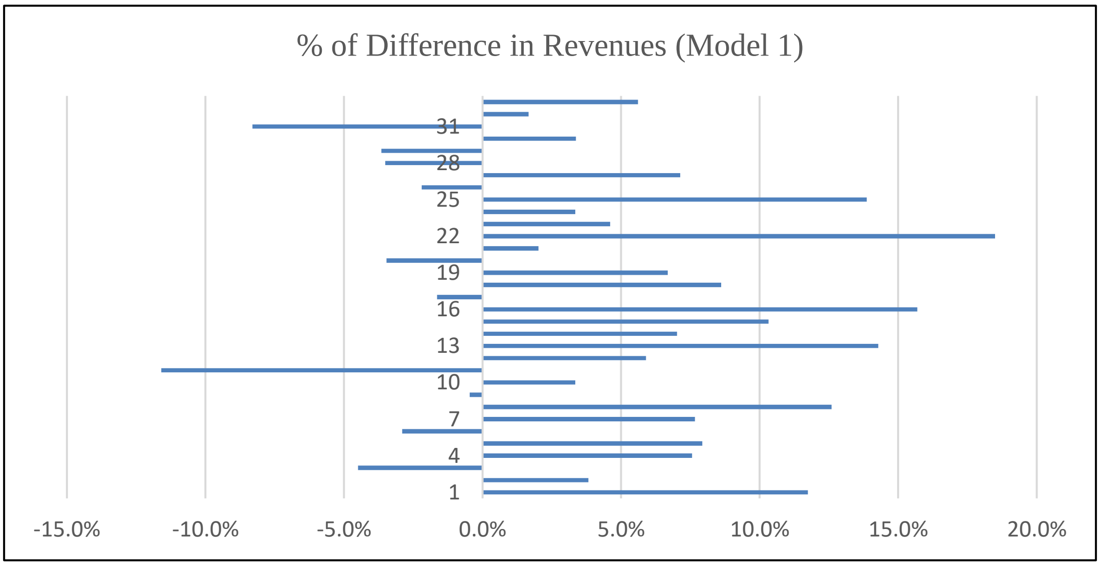
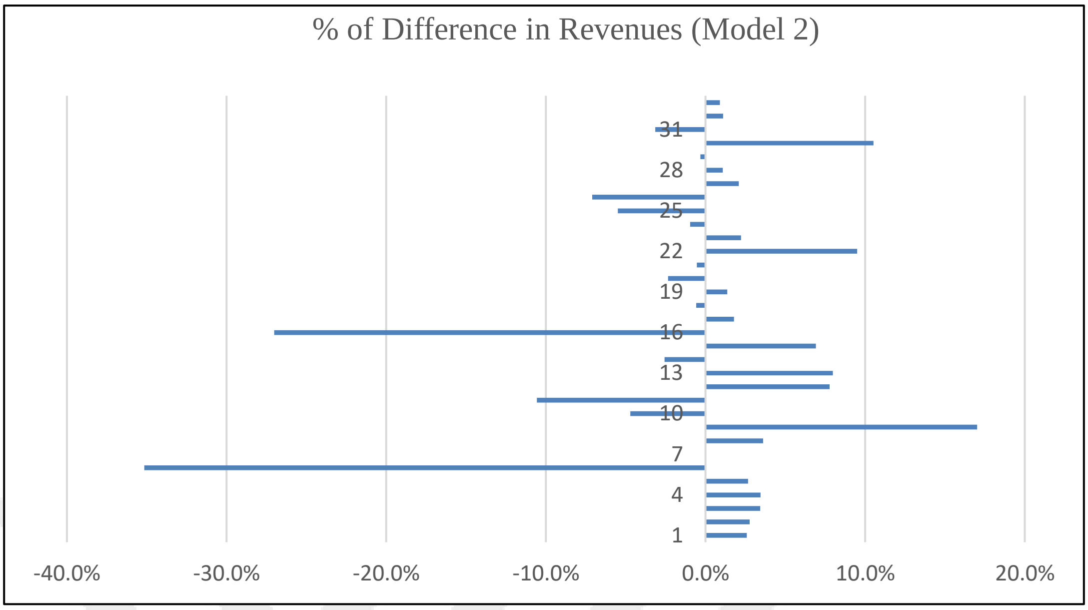
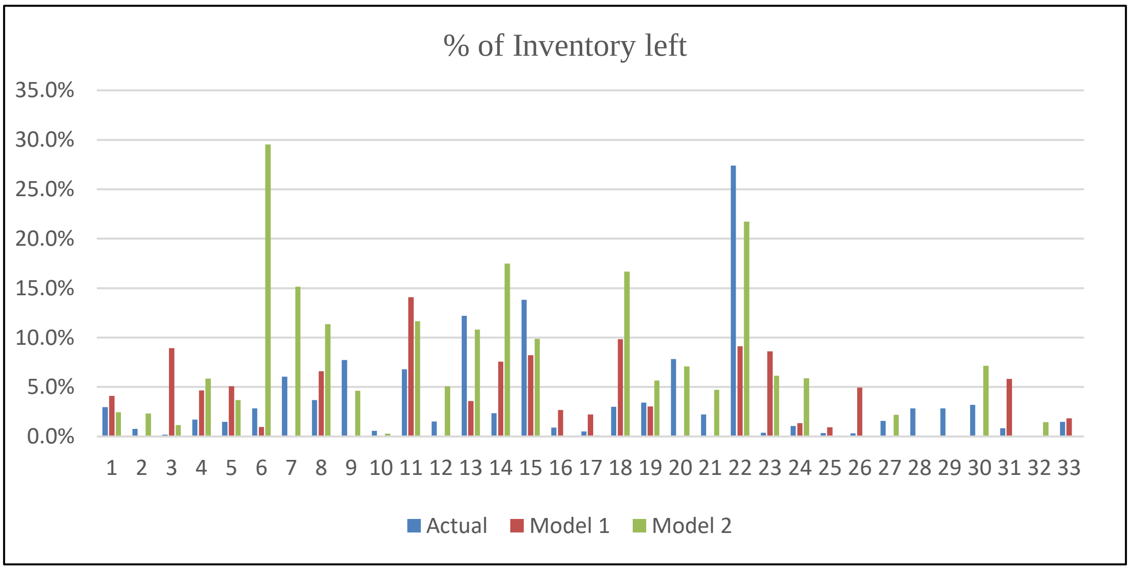


Figure 3.11: Comparison of the Model 2 with the Actual Revenues



“% of Inventory Left” is calculated by dividing the remaining inventory at the end of the season to the initial purchase quantity. As seen in Figure 3.12, with the exception of only one product, Model 1 seems to have sold more than 90% of each product.

Figure 3.12: Remaining inventory performance comparison



With the results of all models, we concluded that Model 1 performs better than both actual and Model 2.

Figure 3.13: Cumulative Revenues by Model

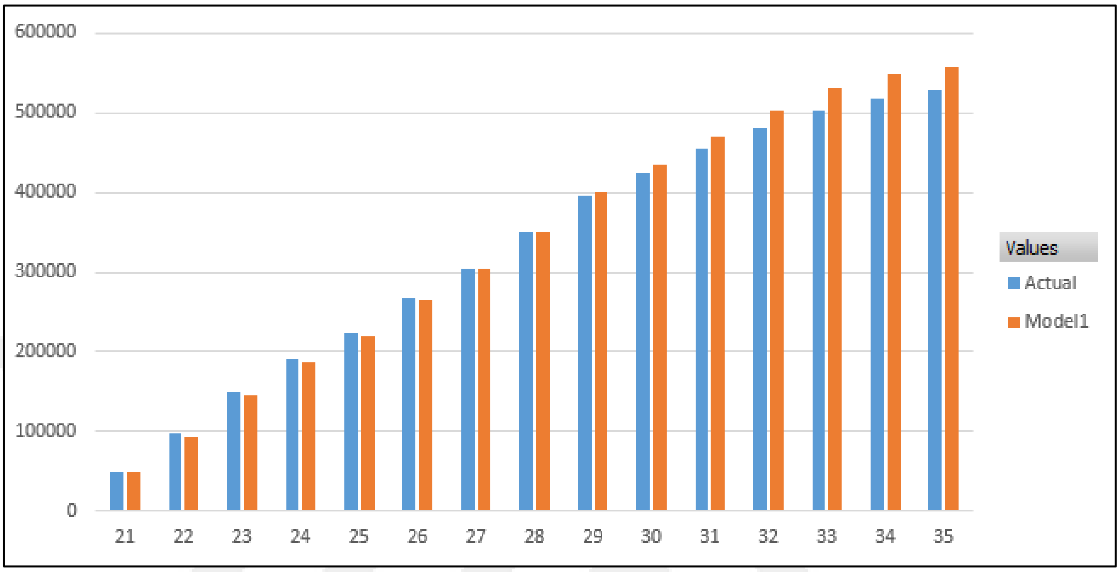


Figure 3.13 indicates that until week 28th revenues are identical. After week 28th Model 1 gains more revenue than actual policy.

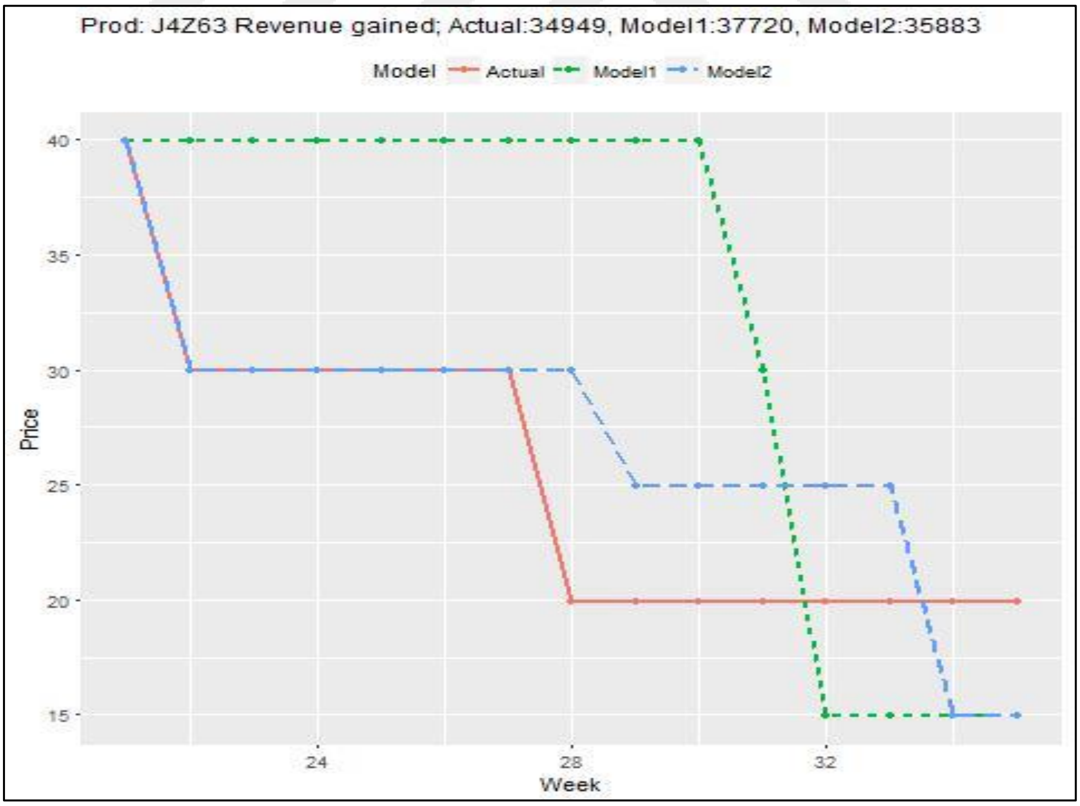
Under the results based on Table 3.9, the company has the potential to earn 4.4% extra revenue by using the Model 1. Compared with the computational time, it is more effective than Model 2. Model 2 needs 10 times higher computational time than Model 1. Under these circumstances Model 1 was selected for our base model.

The weakest part of the Model 1 was that because of our forecasting model and parameters, it assumed that customer's reservation prices were independent of special days. As mentioned in chapter 2, it is known that customers are more price sensitive during the special days than usual. That's why additional constraint was added to the model. According to this new constraint, if product has premium (full) price during the special days (Ramadan Feast at week 28th in our case) model was forced to apply a minimum discount.

When the results of the Model 1 examined, it was observed that Model 1 offered discounts two weeks in a row. Since the company do not announce the discounts, system should wait at least 2 weeks so that customers can notice the discounts. This business rule will also positively affect the company's public image against discounts.

Sample optimal pricing policy offered by Model 1 can be seen at Figure 3.14. In this example, Model 1 violated the business rules. Firstly, Model 1 insisted on staying at full price at Ramadan Feast at week 28. Second, it applied 2 consecutive discounts between week 29 and 30. As Chen and Chen (2015) offered future researchers to consider the business rules in their models, a new heuristic model developed which includes the business rules. Based on the optimal deterministic pricing policy, we applied following heuristic algorithm which solves the violations of consecutive discounts.

Figure 3.14: Sample pricing policy that was offered by Model 1



Step 0: Get the optimal pricing policy from Model 1

a) Set $t = T$.

b) Set counter $n = 1$.

c) Set $p = p_T$

Step 1: For t = T..2,1,0 DO

a) Check if $p \neq p_t$.

  a. IF TRUE THEN

    i. Check $n \geq 2$

      1. IF TRUE THEN, Set $n = 1$

      2. IF FALSE THEN Set $p_t = p$

    ii. Set $n = \mathrm{n} + 1$

  b. ELSE , Set $n = \mathrm{n} + 1$

Step 2: $p = p_t$

Step 3: IF $t = 0$ THEN STOP ELSE GOTO Step 1.

The new model called as Model 3. Figure 3.15 showed that new model considers the new business rules. The results at Table 3.10 showed that Model 3 gain more revenue than the Model 1 which was impossible. Actually, Model 1 always gain the best revenues but since we penalized the inventory left over at the end of the season in our simulation approach, that's why the Net Revenue of Model 3 seemed greater than Model 1.

Figure 3.15: Sample pricing policy that was offered by Model 3

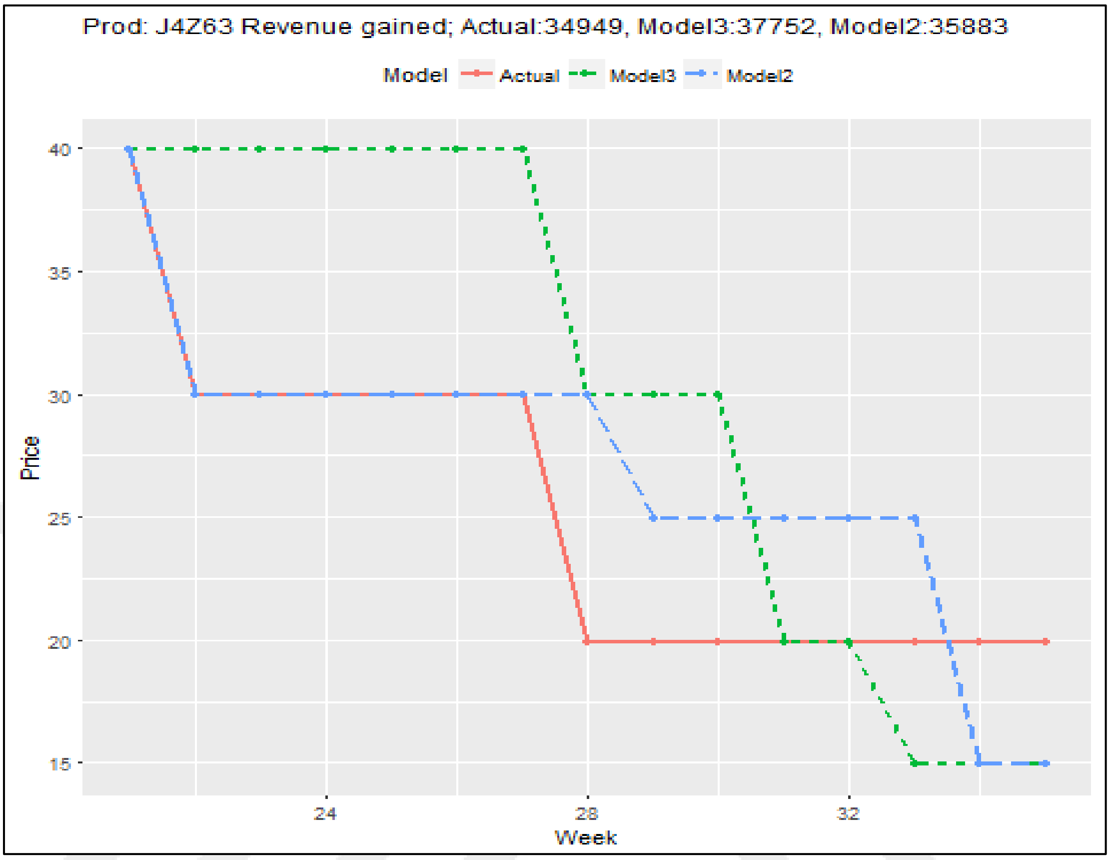


Table 3.10: Comparison of Model 3 results

	Actual		Model 1		Model 3	
Product	% of Inv. Left	Net Revenue	% of Inv. Left	Net Revenue	% of Inv. Left	Net Revenue
E1L13	3.0%	32,301	4.1%	36,092	2.7%	35,870
E1095	0.8%	4,704	0.0%	4,884	0.0%	4,884
E1090	0.2%	8,160	8.9%	7,794	7.2%	8,153
J1Z50	1.7%	25,579	4.7%	27,512	3.6%	27,491
J4Z63	1.5%	34,949	5.1%	37,720	3.3%	37,752
J1L02	2.9%	4,326	1.0%	4,200	1.0%	4,200
J4398	6.1%	5,600	0.0%	6,029	0.0%	5,979
E1069	3.7%	7,548	6.6%	8,498	5.0%	8,478
J7182	7.7%	5,753	0.0%	5,727	0.0%	5,727
E1081	0.6%	6,269	0.0%	6,479	0.0%	6,184
E8009	6.8%	3,352	14.1%	2,963	14.1%	2,963
E1L34	1.5%	14,400	0.0%	15,250	0.0%	15,050
J1041	12.2%	10,272	3.6%	11,739	3.6%	11,739

J1059	2.3%	13,931	7.6%	14,909	4.3%	15,037
E1067	13.8%	4,690	8.2%	5,173	7.9%	5,138
E1085	0.9%	5,293	2.7%	6,124	1.3%	6,174
E1102	0.5%	8,146	2.2%	8,012	1.6%	8,051
E1110	3.0%	8,035	9.8%	8,727	9.6%	8,751
J1047	3.4%	8,155	3.0%	8,700	3.0%	8,700
J1048	7.8%	2,787	0.0%	2,690	0.0%	2,690
J1L49	2.2%	6,407	0.0%	6,537	0.0%	6,537
J1Z29	27.4%	5,185	9.1%	6,143	9.1%	6,143
E1063	0.4%	15,865	8.6%	16,595	6.4%	16,901
E1078	1.1%	9,803	1.3%	10,131	0.7%	10,063
E1099	0.3%	8,121	0.9%	9,246	0.3%	9,216
E1106	0.3%	23,046	4.9%	22,541	4.8%	22,121
E1089	1.6%	8,412	0.0%	9,011	0.0%	8,791
J1Z90	2.9%	30,796	0.0%	29,713	0.0%	29,378
J4Z23	2.8%	43,346	0.0%	41,762	0.0%	41,492
J7Z80	3.2%	19,816	0.0%	20,483	0.0%	20,028
E1112	0.8%	1,917	5.8%	1,758	4.6%	1,748
E1L22	0.0%	13,235	0.0%	13,455	0.0%	12,900
J3Z56	1.5%	90,689	1.8%	95,779	1.1%	94,673
TOTAL	**2.8%**	**490,887**	**2.9%**	**512,375**	**2.2%**	**509,001**
				4.4%		**3.7%**

Table 3.10 shows that the new constraint and business rule contribute positively to inventory performance. In terms of end of season inventory performance, Model 3 performed better than all other models. And revenue loss due to new business rules is ignorable.

4. FINDINGS

In this study, we found that applying optimal pricing policies has potential to increase revenues and reduce the end of season inventories for apparel retailers. Based on the results at Table 3.10, if the optimal prices are applied there exists potentially additional 3.7% revenue gain for the company. It also shows that optimal policy can reduce the inventory left over.

When examining the all results at Appendix 1 section, we also found that staying as long as possible at full price affects the revenues positively. However, staying at full price when the high season has been passed affects revenues negatively. For this reason, instead of making aggressive discounts, the revenue can be maximized by applying smaller but more frequent discounts.

In apparel retail, since the nature of the demand is complex, the state space of the problem becomes very large. We found that solving the problem directly with the stochastic dynamic programming is not practically implementable. Instead, we showed that the problem can be solved by the approximate dynamic programming approach.

In the end, we generated a model which considers the company's business rules and found optimal pricing policies for each product-model. Computational time of the model is around one minute or less per product-model. Even though the model gives the optimal pricing policies from beginning to the end of the season, it is offered to review the policy for every week to become the results more responsive in the real life scenario.

5. DISCUSSION AND CONCLUSION

Apparel retail business has its own unique dynamics. In this business where the uncertainty is high, merchandising planners put their efforts to achieve the highest revenues during the short selling season. Product portfolio that has the large number of products makes it difficult for planners to make decisions because the majority of products do not have a sales history, customer tastes are variable over time, and competition is high. In this study, a revenue optimization model has been proposed in the apparel retail business to enable planners to make decisions under uncertainties.

Most of the models studied in the literature are modeled by theoretical distributions. In this study, a data driven approach was adopted and the demand dynamics were modeled using the historical data. A log-linear demand model which considers the initial purchase, age of the product, time, season, special days, price and inventory level was proposed. For updating the parameter estimates, an estimation methodology has been offered. In optimization stage, in addition to considering the demand as deterministic, stochastic methods that would respond to uncertainties and assess possible risks were also offered. Due to the fact that the state space required for the solution was very large, approximate dynamic programming methods has been implemented. Some approximations were applied and the problem that was impossible to solve practically has been solved. In addition to that, a heuristic solution taking into account the additional business rules has been developed and the results were compared with the optimal policies. In the end a simulation approach which mimics the reality of the solution has been applied and the results were compared against the merchandising planner's pricing policies. It was shown that the proposed models could contribute to the revenues.

We also found that staying as long as possible at full price affects the revenues positively. However, staying at full price when the high season has been passed affects revenues negatively. For this reason, instead of making aggressive discounts, the revenue can be maximized by applying smaller but more frequent discounts. This results are similar to the study of Gupta and et al. (2004).

Proposed model achieved the best results when inventory left over was considered at the end of the season. On the other hand, in terms of revenues gained, proposed model achieved additional 3.7% more revenue.

To our knowledge, this study is the only one that explains the demand with time, price, and inventory level, and attempts to solve the problem stochastically. Compared to the actual pricing policies of the merchandising planners, it was seen that stochastic model could increase the revenues. However, when considering the computational time of the solution, the methods used in the approximate dynamic programming solution need to be improved. Considering the many assumptions of the deterministic solution, risk averse stochastic models can produce more suitable solutions for real life.

Experiments in this study have shown that forecasting model can be improved by applying new methodologies. For example, we saw that even if they were in the same category, some product's behavioral pattern was not in harmony with the forecasting model. The products that show different demand behaviors can be clustered by using data mining techniques. Thus, output of the demand model can be improved. Deep learning algorithms which are popular in the recent times can help better estimating the parameters and explaining their effects.

REFERENCES

Books

Powell, W. B., 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality* (Vol. 703). John Wiley & Sons.

Talluri T., Van Ryzin G., 2004. *The Theory and Practice Revenue Management*. Springer Science+Business Media, New York, pp. 175-235.

Winston, W. L., & Goldberg, J. B., 2004. *Operations research: applications and algorithms* (Vol. 3). Boston: Duxbury press, pp. 1029

Periodicals

Akçay, Y., Natarajan, H. P., & Xu, S. H., 2010. Joint dynamic pricing of multiple perishable products under consumer choice. *Management Science*, vol 56, 1345-1361.

Bitran, G.R., Mondschein, S.V., 1997. Periodic Pricing of Seasonal Products in Retailing. *Management Science*, pp. 64–79.

Bitran, G., Caldentey R. & Mondschein S., 1998. Coordinating Clearance Markdown Sales of Seasonal Products in Retail. *Operations Research*, pp. 609–623.

Caro, F., Gallien, J., 2010. Inventory management of a fast-fashion retail network. *Operations Research*, vol 58(2), 257-273.

Caro, F., Gallien, J., 2012. Clearance Pricing Optimization for a Fast-Fashion Retailer. *Operations Research*, vol 60, no.6, pp. 1404–1422.

Chen, M., & Chen, Z. L., 2015. Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. *Production and Operations Management*, 24(5), 704-731.

Chew, E. P., Lee, C., Liu, R., Hong, K. S., & Zhang, A., 2014. Optimal dynamic pricing and ordering decisions for perishable products. *International Journal of Production Economics*, 157, 39-48.

Dasu, S., & Tong, C., 2010. Dynamic pricing when consumers are strategic: Analysis of posted and contingent pricing schemes. *European Journal of Operational Research*, vol 204(3), 662-671.

Elmaghraby, W., Keskinocak, P., 2003. Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices and Future Directions. *Management Science*, vol 49, no.10, pp. 1287–1309.

Feng, Y., Gallego, G., 2000. Perishable Asset Revenue Management with Markovian Time Dependent Demand. *Management Science*, pp. 942–955.

Friend, S. C., Walker, P. H. 2001. Welcome to the new world of merchandising. *Harvard Business Review*, 79(10), 133-141.

Gallego, G., Van Ryzin G., 1994. Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons. *Management Science*, pp. 1000–1020.

Gallego, G.; Van Ryzin, G., 1997. A multiproduct dynamic pricing problem and its applications to network yield management, *Operations Research*, vol 45, no 1, pp. 24-41.

Gupta, D., Hill, A.V., Chameeva, T., 2004. A Pricing Model for Clearing End-of-Season Retail Inventory. *European Journal of Operations Research*, pp. 518–540.

Heching, A., Gallego, G., Van Ryzin, G., 2002. Mark-down pricing: An empirical analysis of policies and revenue potential at one apparel retailer. *Journal of Revenue and Pricing Management*, pp. 139–160.

Koenig, M., & Meissner, J., 2010. List pricing versus dynamic pricing: Impact on the revenue risk. *European Journal of Operational Research*, vol 204(3), 505-512.

Lazear, E.P., 1986. Retail Pricing and Clearance Pricing. *The American Economic Review*, pp. 14–32.

Levin, Y., McGill, J., & Nediak, M., 2008. Risk in revenue management and dynamic pricing. *Operations Research*, vol 56, no2, 326-343.

Levin, Y., McGill, J., & Nediak, M., 2010. Optimal dynamic pricing of perishable items by a monopolist facing strategic consumers. *Production and Operations Management*, vol 19, 40-60.

Lim, A.E.B., Shanthikumar, J.G., 2007. Relative Entropy, Exponential Utility, and Robust Dyynamic Pricing. *Operations Research*, vol 55, no.2, pp. 198–214.

Maglaras, C., Meissner, J., 2006. Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing & Service Operations Management*, 8(2), 136-148.

Maglaras, C., & Eren, S., 2015. A maximum entropy joint demand estimation and capacity control policy. *Production and Operations Management*, vol 24, 438-450.

Mantrala, K.M., Rao, S., 2001. A Decision-Support System that Helps Retailers Decide Order Quantities and Mark-downs for Fashion Goods. *Interfaces*, vol 31, no.3, pp. 146–165.

Ofek, E., & Avery, J., 2012. "J.C. Penney's 'Fair and Square' Pricing Strategy.". *Harvard Business School Case*, 513-036.

Pashigian, B. P., 1988. Demand uncertainty and sales: A study of fashion and markdown pricing. *The American Economic Review*, 936-953.

Perakis, G., & Sood, A. (2006). Competitive multi-period pricing for perishable products: A robust optimization approach. *Mathematical Programming*, vol 107(1-2), 295-335.

Smith, S. A., McIntyre, S. H., & Achabal, D. D. (1994). A two-stage sales forecasting procedure using discounted least squares. *Journal of Marketing Research*, 44-56.

Smith, S., Achabal, D., 1998. Clearance and Inventory Policies for Retail Chains. *Management Science*, pp. 285–299.

Soysal, G. P., & Krishnamurthi, L., 2012. Demand dynamics in the seasonal goods industry: An empirical analysis. *Marketing Science*, 31(2), 293-316.

Şen, A., 2013. A comparison of fixed and dynamic pricing policies in revenue management. *Omega*, vol 41, 586-597.

Warner, E.J., Barsky, R.B., 1995. The Timing and Magnitude of Retail Store Markdowns Evidence from Weekends and Holidays. *The Quarterly Journal of Economics*, pp. 321–352.

Xu, X. I. A. O. W. E. I., & Hopp, W. J., 2005. Dynamic pricing and inventory control: The value of demand learning. *Department of Industrial Engineering*, Northwestern University.

Yao, X., Sanli, T., & Gulcu, A., 2015. Markdown optimization for multiple products with a common markdown budget constraint. J*ournal of Revenue and Pricing Management*, vol 14(6), 442-450.

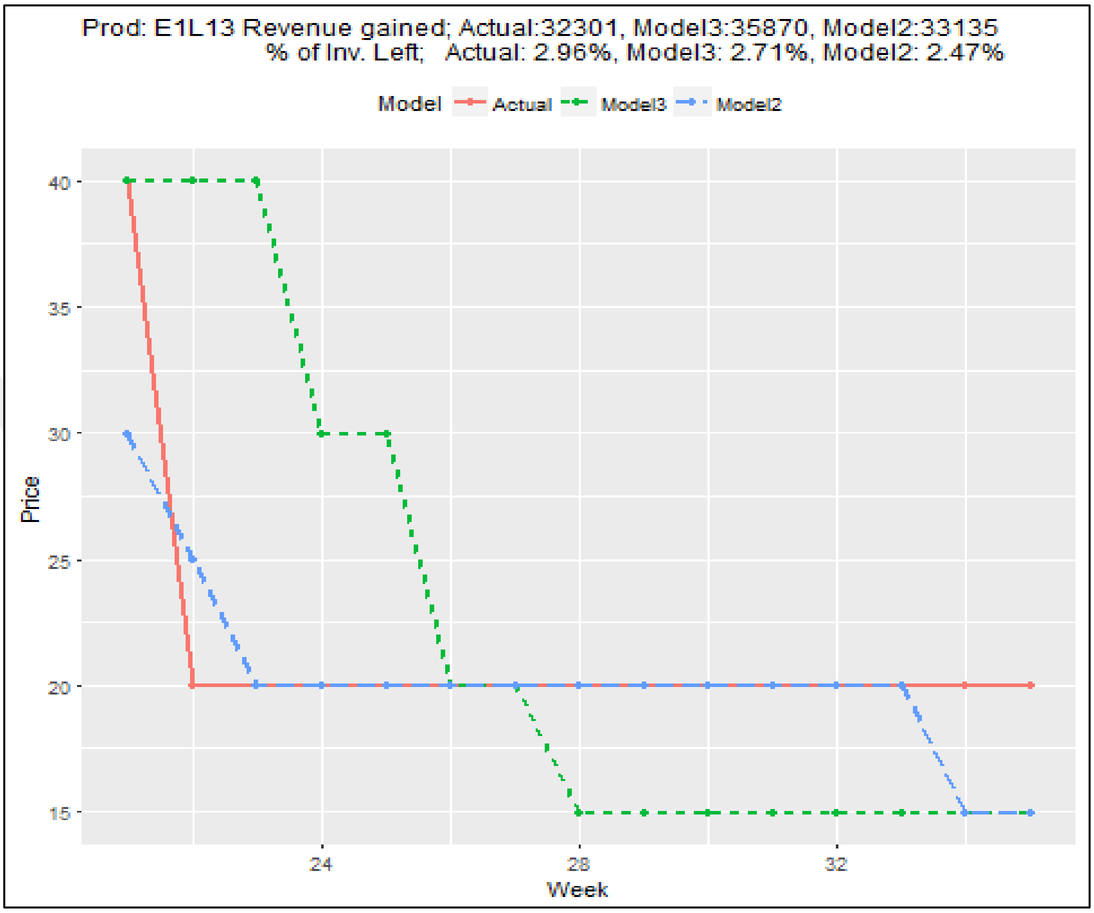
Zhang, S., G. J. Fitzsimons., 1999. Choice-process satisfaction: The influence of attribute alignability and option limitation. *Organ. Behav. Human* Dec. 77(3) 192–214

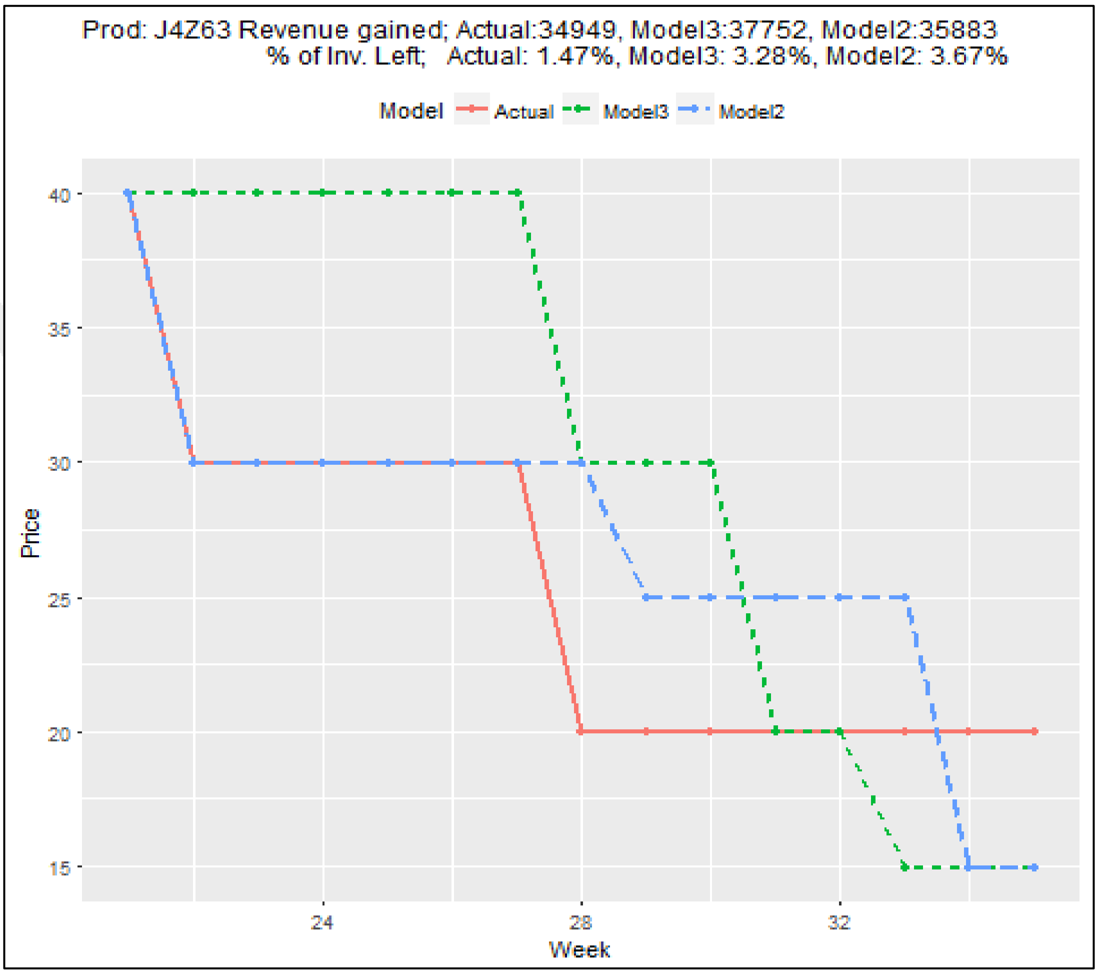
Zhao, W., & Zheng, Y. S., 2000). Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management science*, 46(3), 375-388.

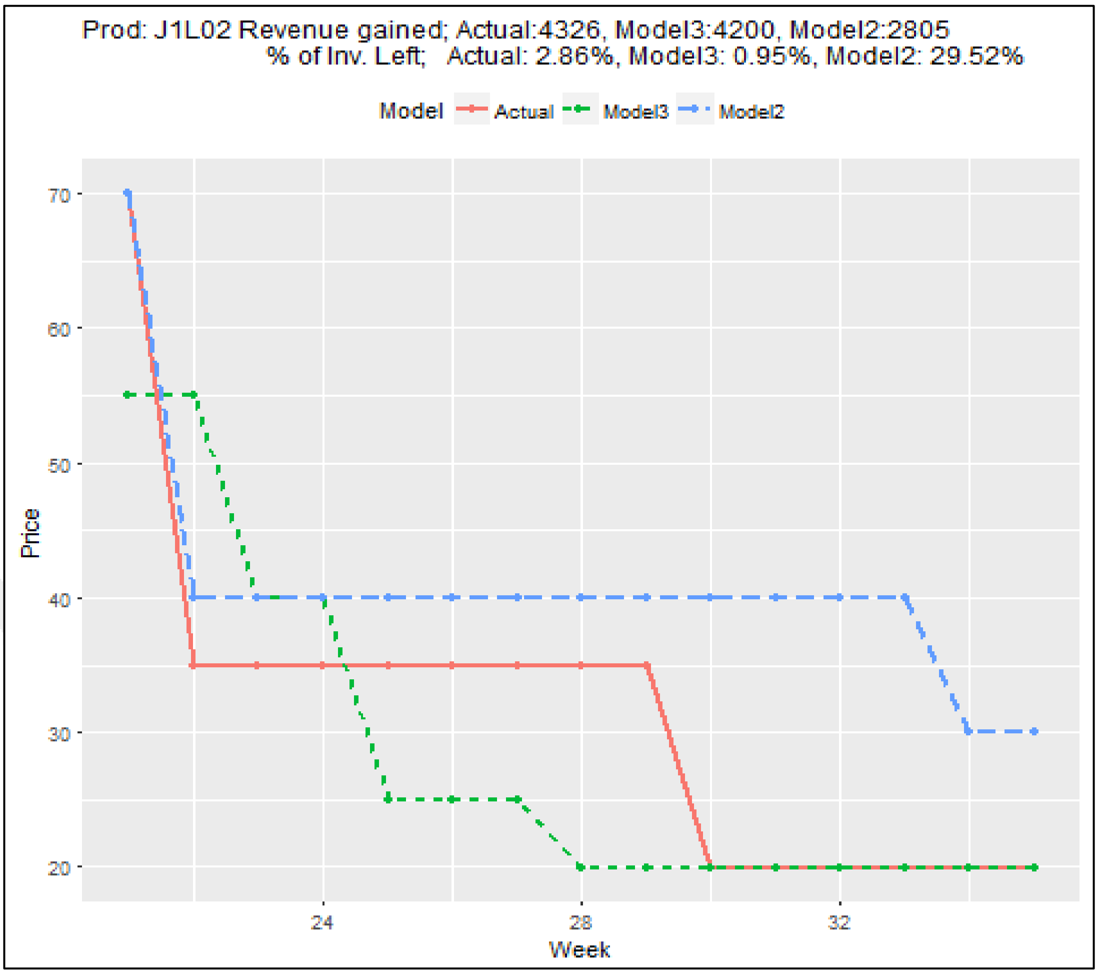
Zhao, L., Tian, P., & Li, X., 2012. Dynamic pricing in the presence of consumer inertia. *Omega*, 40(2), 137-148.

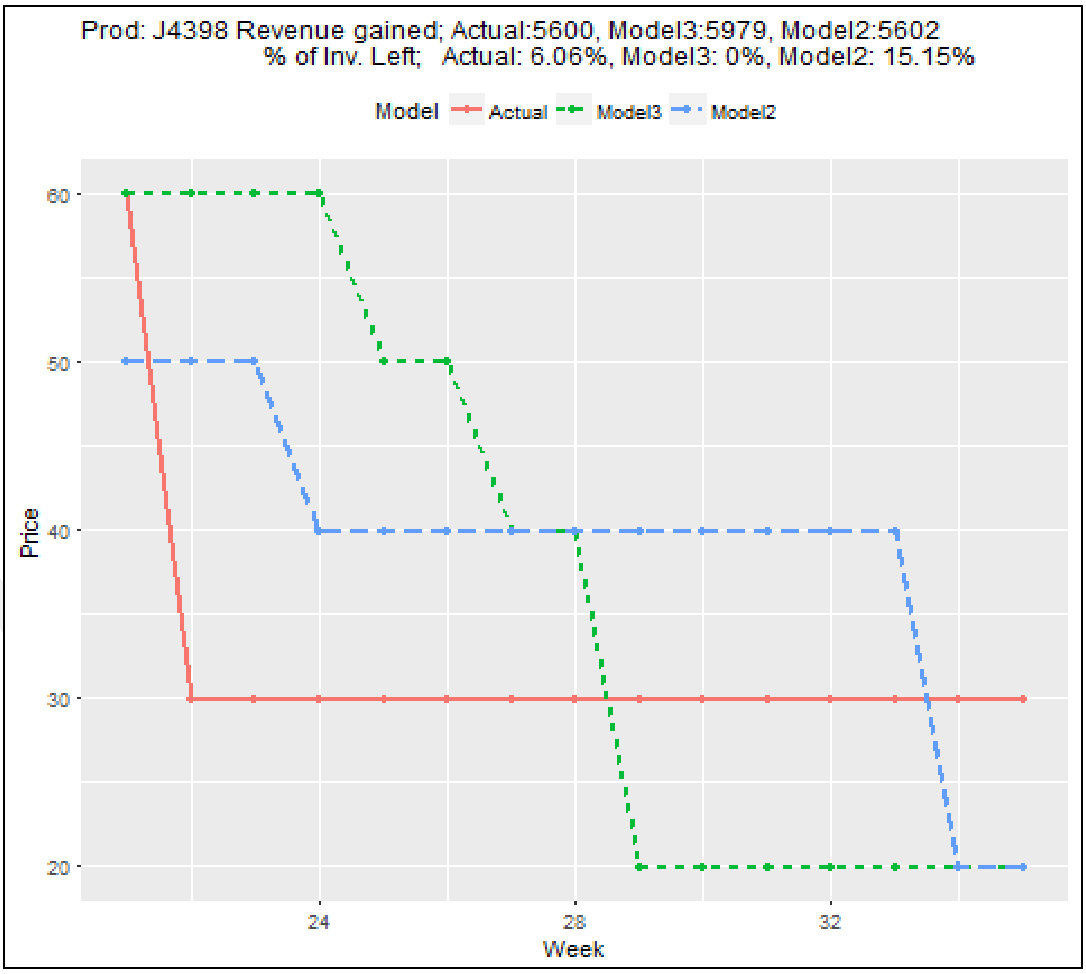
APPENDICES

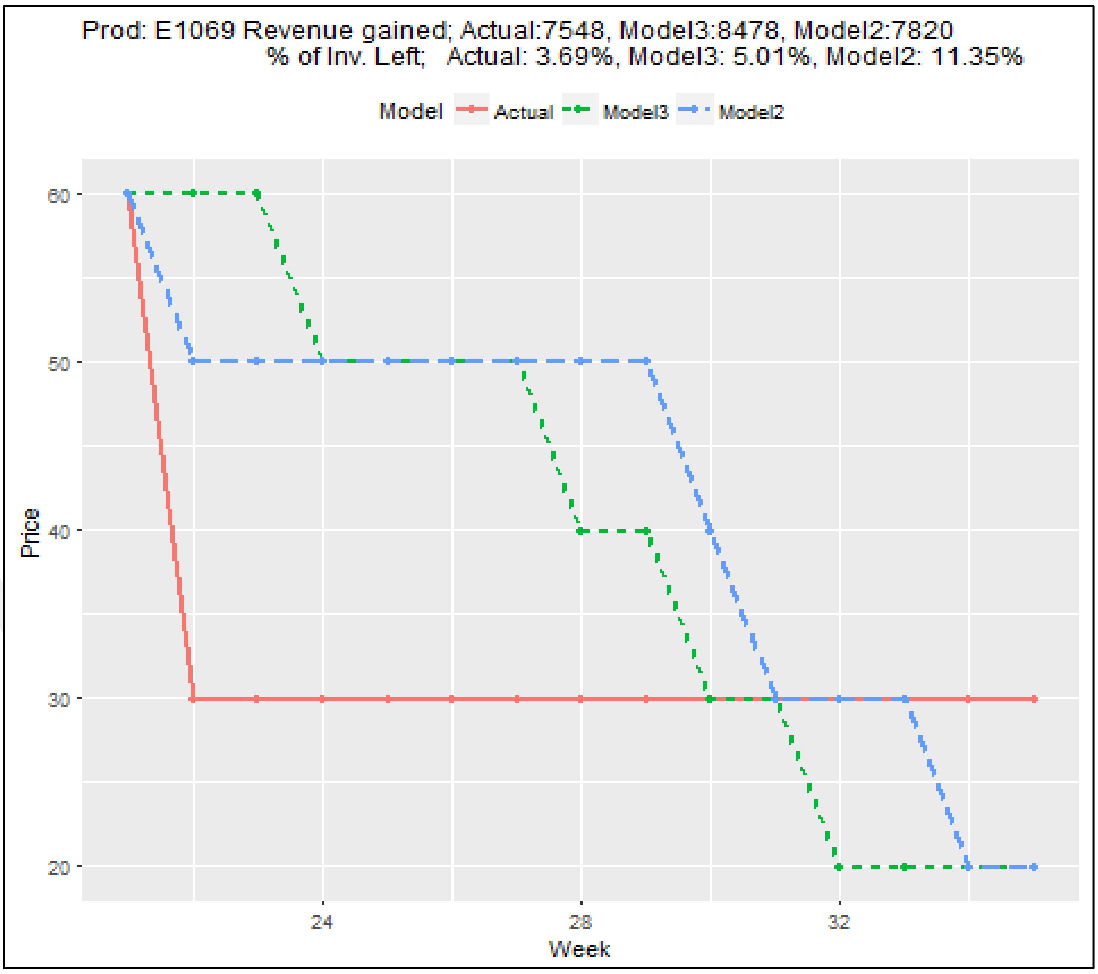
APPENDIX 1: Optimal pricing policies by Model vs. Actual

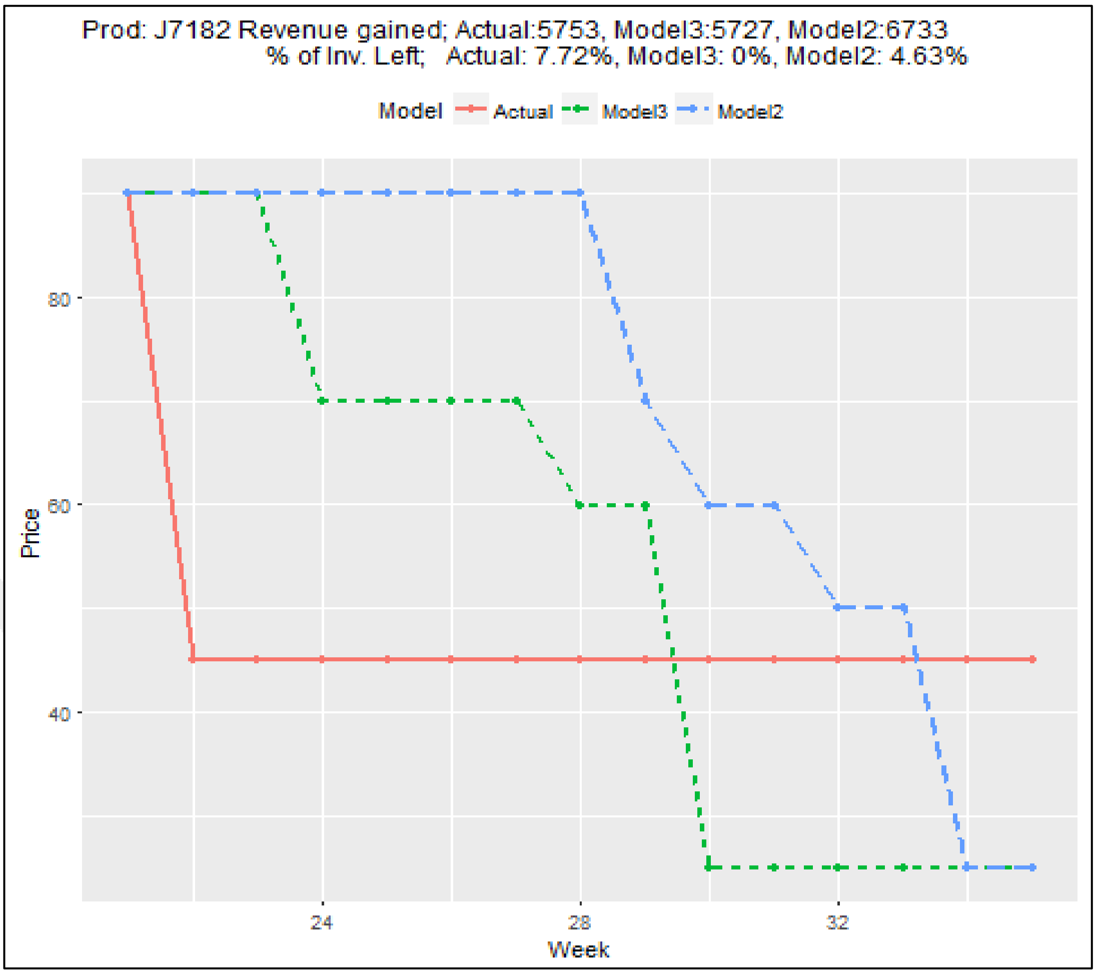


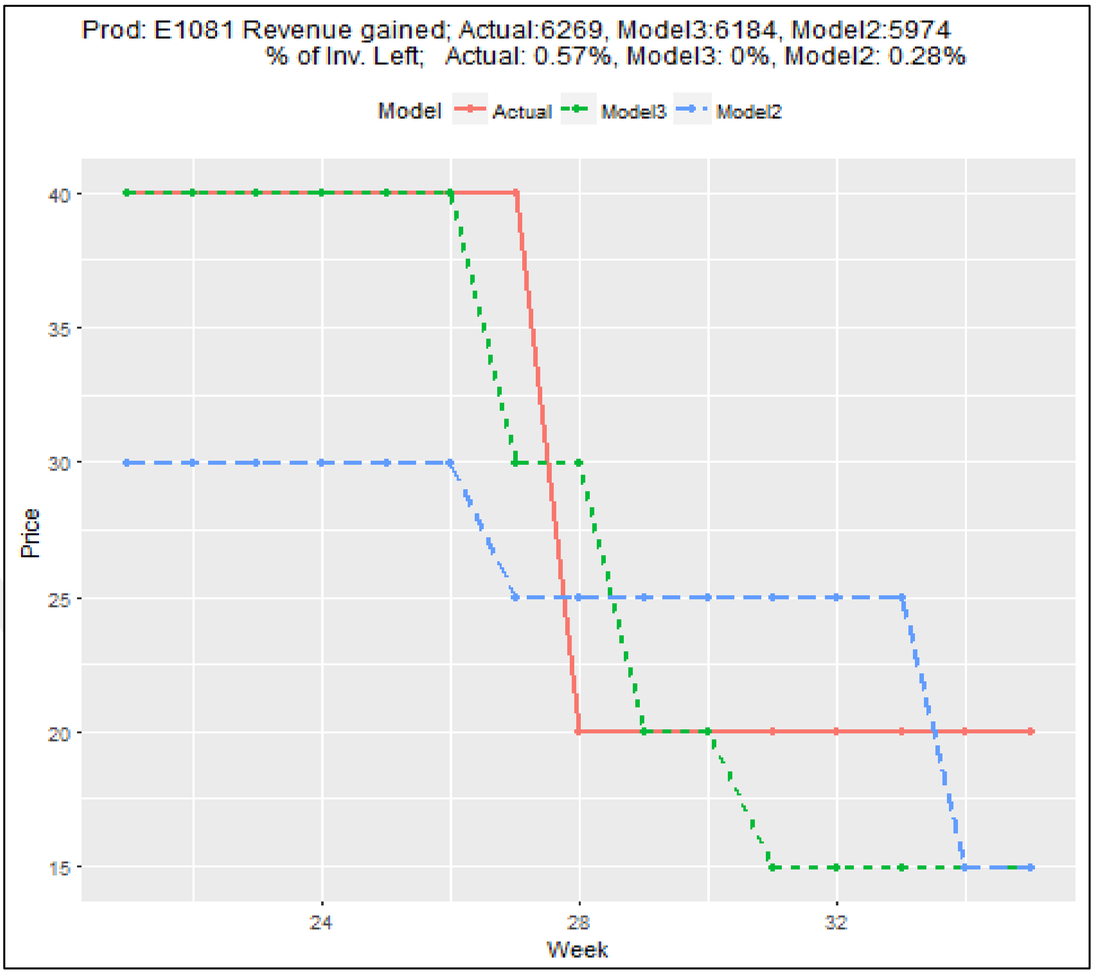

Prod: J4Z63 Revenue gained; Actual:34949, Model3:37752, Model2:35883
% of Inv. Left; Actual: 1.47%, Model3: 3.28%, Model2: 3.67%
Model
Actual
Model3
Model2
Price
40
35
30
25
20
15
24
28
32
Week

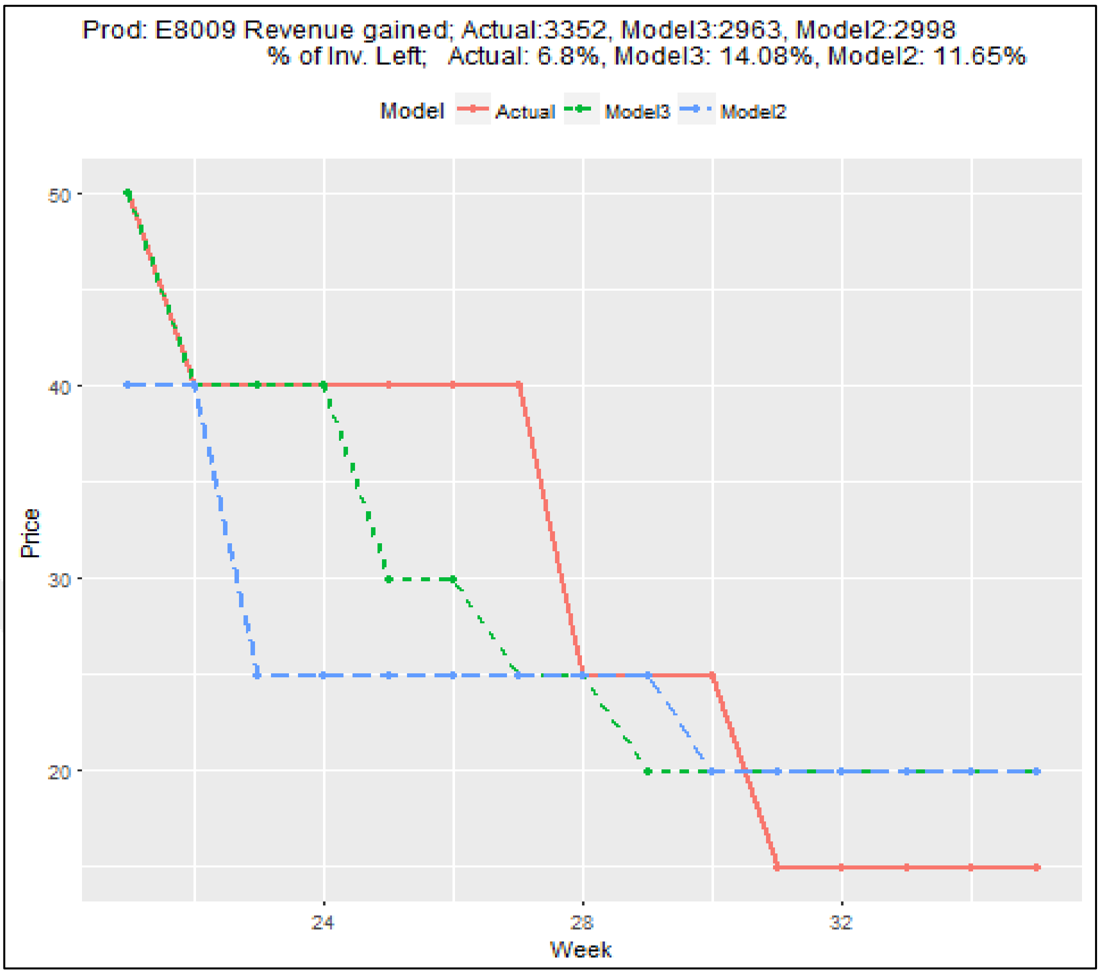

Prod: J1L02 Revenue gained; Actual:4326, Model3:4200, Model2:2805
% of Inv. Left; Actual: 2.86%, Model3: 0.95%, Model2: 29.52%
Model
Actual
Model3
Model2
Price
70
60
50
40
30
20
24
28
32
Week

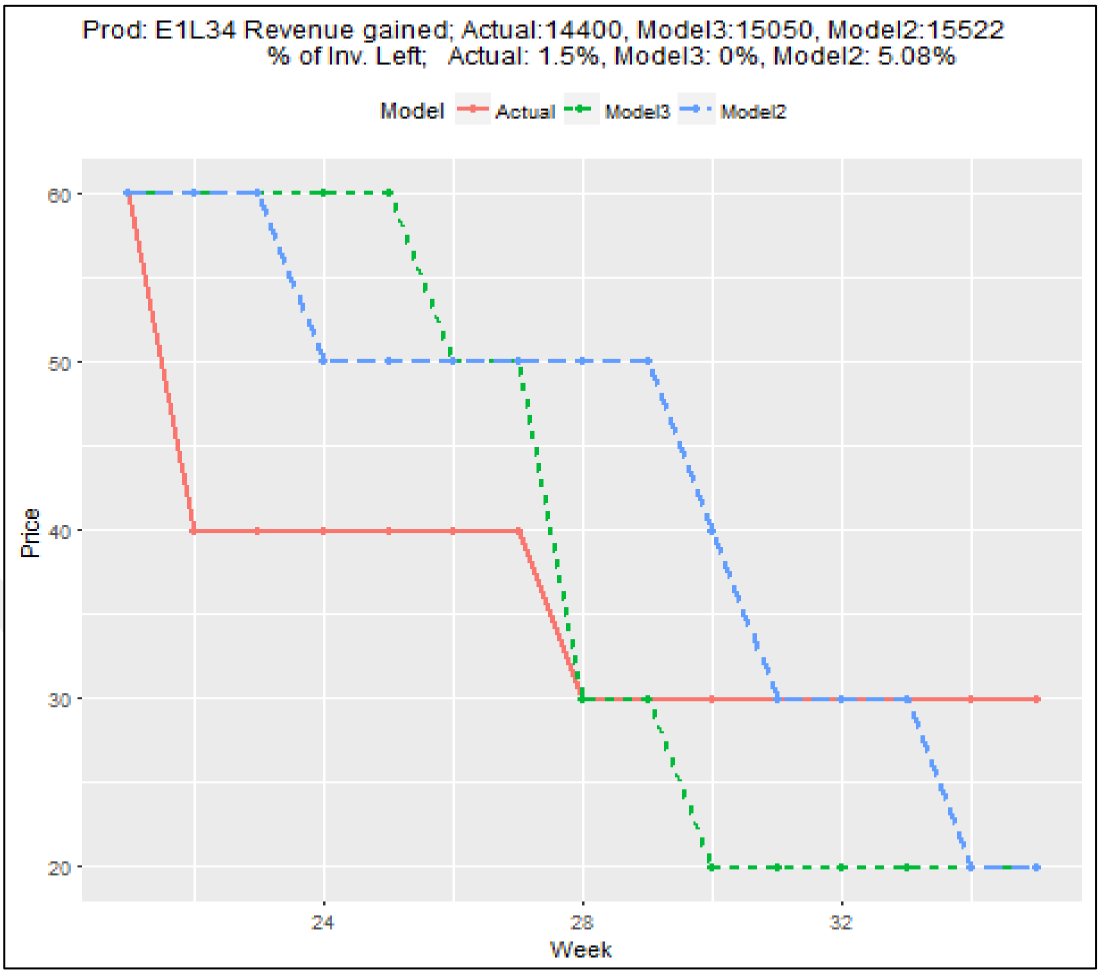

Prod: J4398 Revenue gained; Actual:5600, Model3:5979, Model2:5602
% of Inv. Left; Actual: 6.06%, Model3: 0%, Model2: 15.15%
Model
Actual
Model3
Model2
Price
60
50
40
30
20
24
28
32
Week

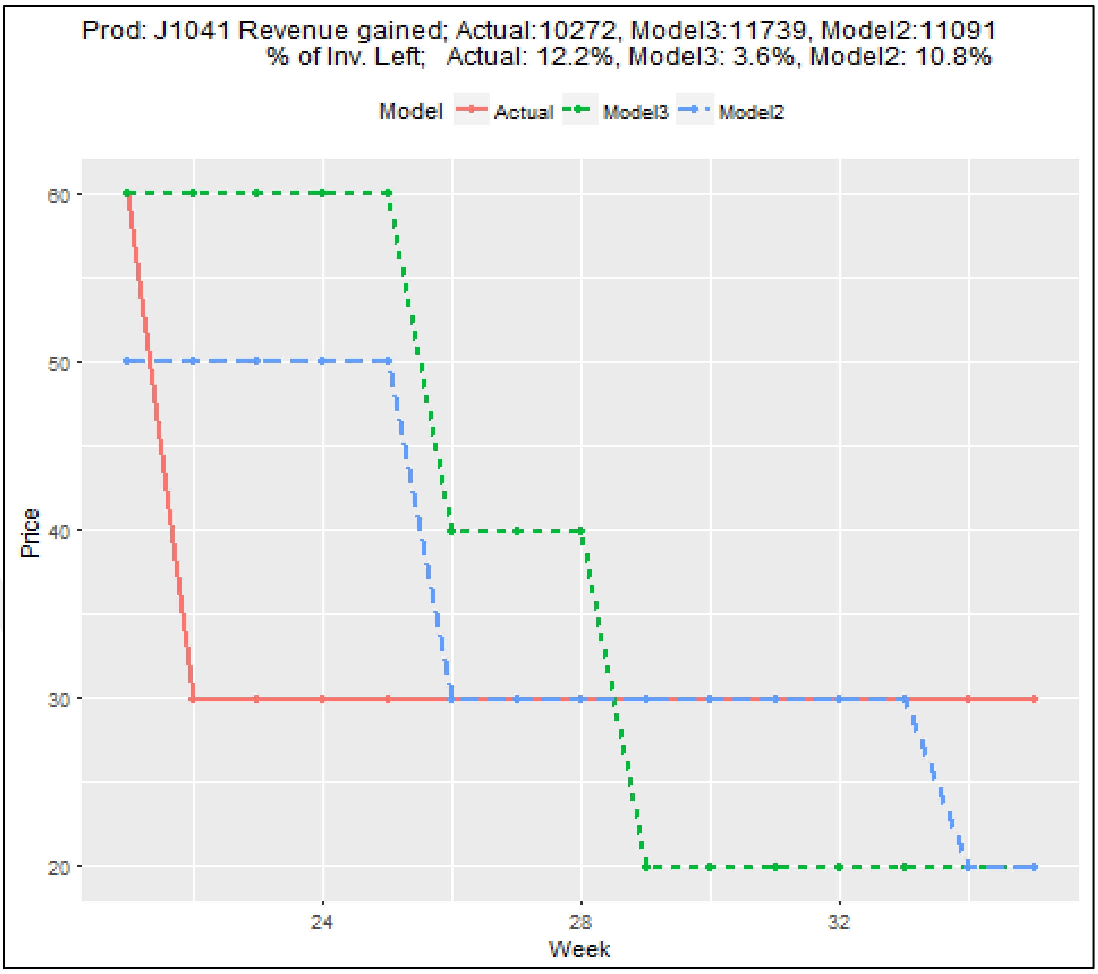

Prod: E1069 Revenue gained; Actual:7548, Model3:8478, Model2:7820
% of Inv. Left; Actual: 3.69%, Model3: 5.01%, Model2: 11.35%
Model
Actual
Model3
Model2
Price
60
50
40
30
20
24
28
32
Week

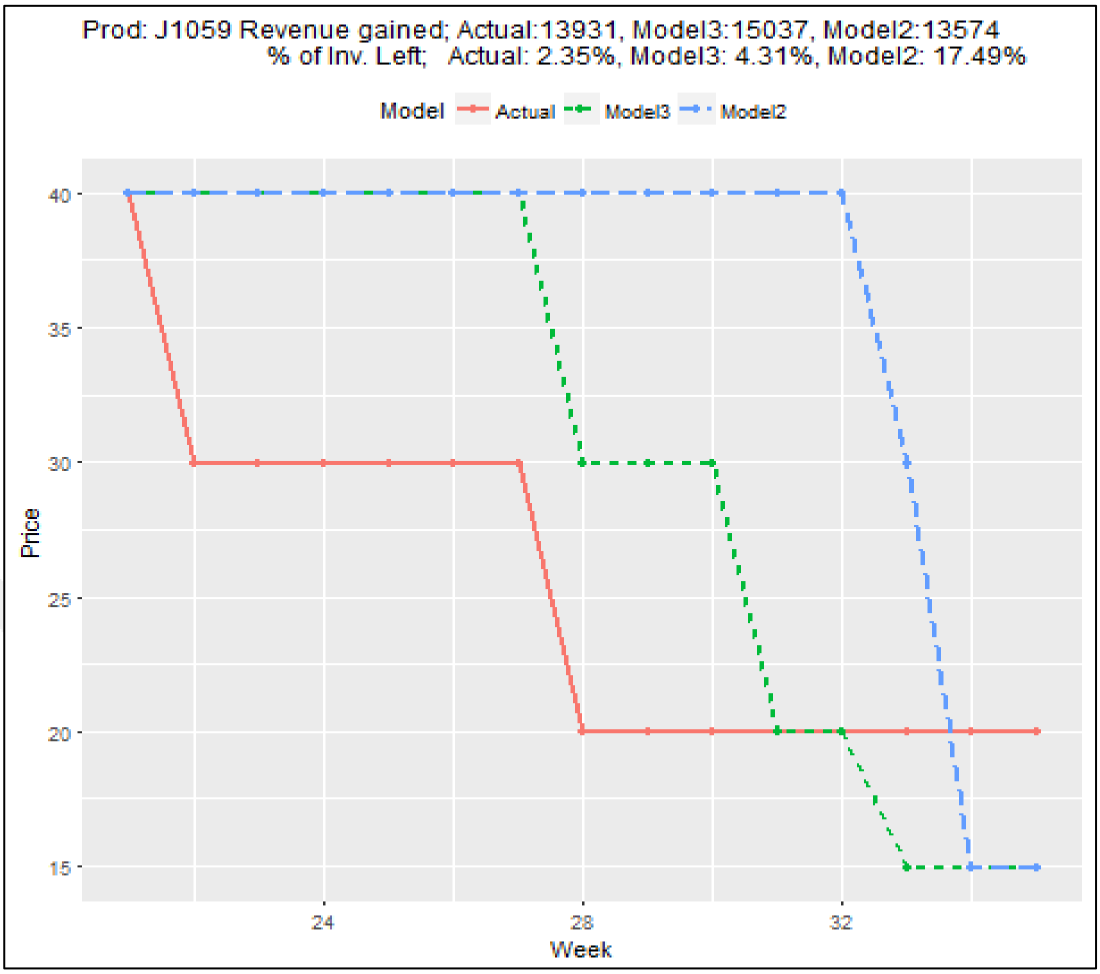

Prod: J7182 Revenue gained; Actual:5753, Model3:5727, Model2:6733
% of Inv. Left; Actual: 7.72%, Model3: 0%, Model2: 4.63%
Model
Actual
Model3
Model2
Price
80
60
40
24
28
32
Week

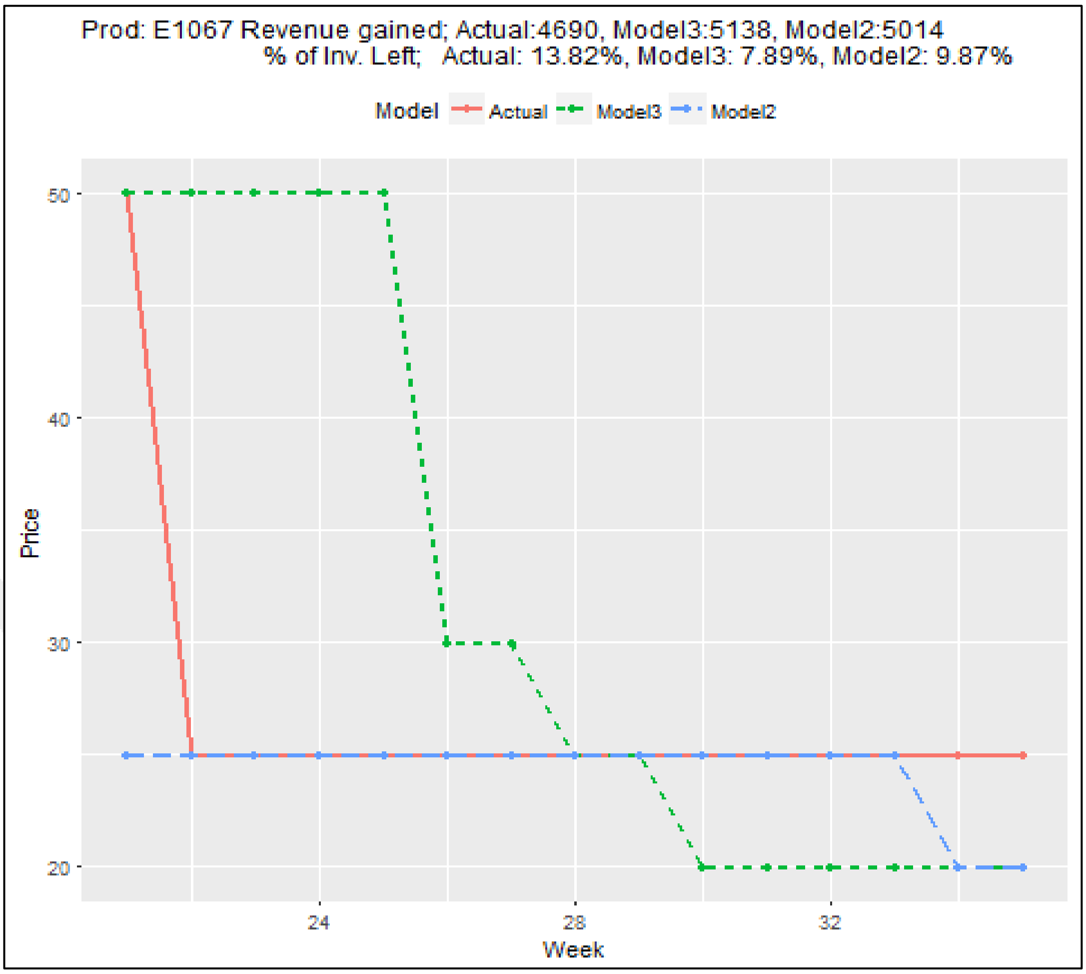

Prod: E1081 Revenue gained; Actual:6269, Model3:6184, Model2:5974
% of Inv. Left; Actual: 0.57%, Model3: 0%, Model2: 0.28%
Model
Actual
Model3
Model2
Price
40
35
30
25
20
15
24
28
32
Week

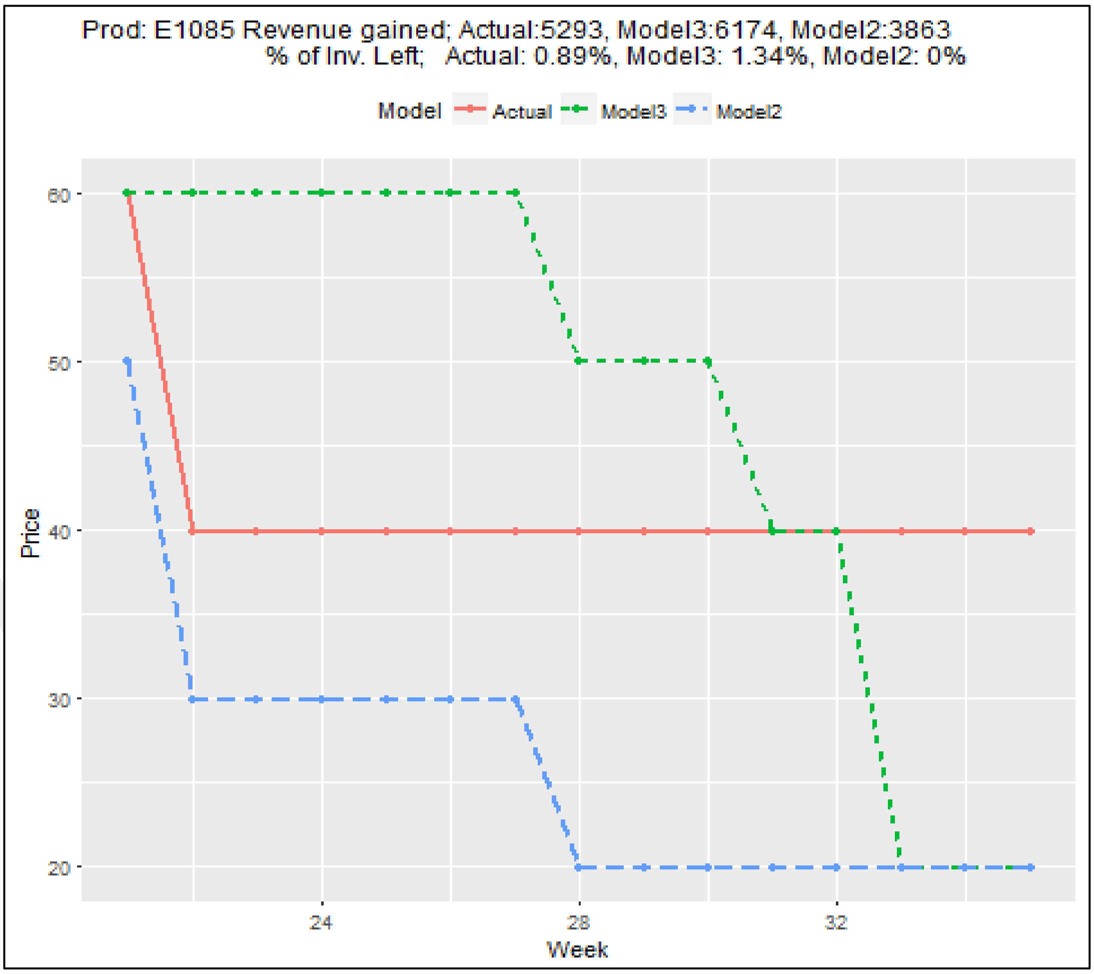

Prod: E8009 Revenue gained; Actual:3352, Model3:2963, Model2:2998
% of Inv. Left; Actual: 6.8%, Model3: 14.08%, Model2: 11.65%
Model
Actual
Model3
Model2
50
40
30
20
Price
24
28
32
Week

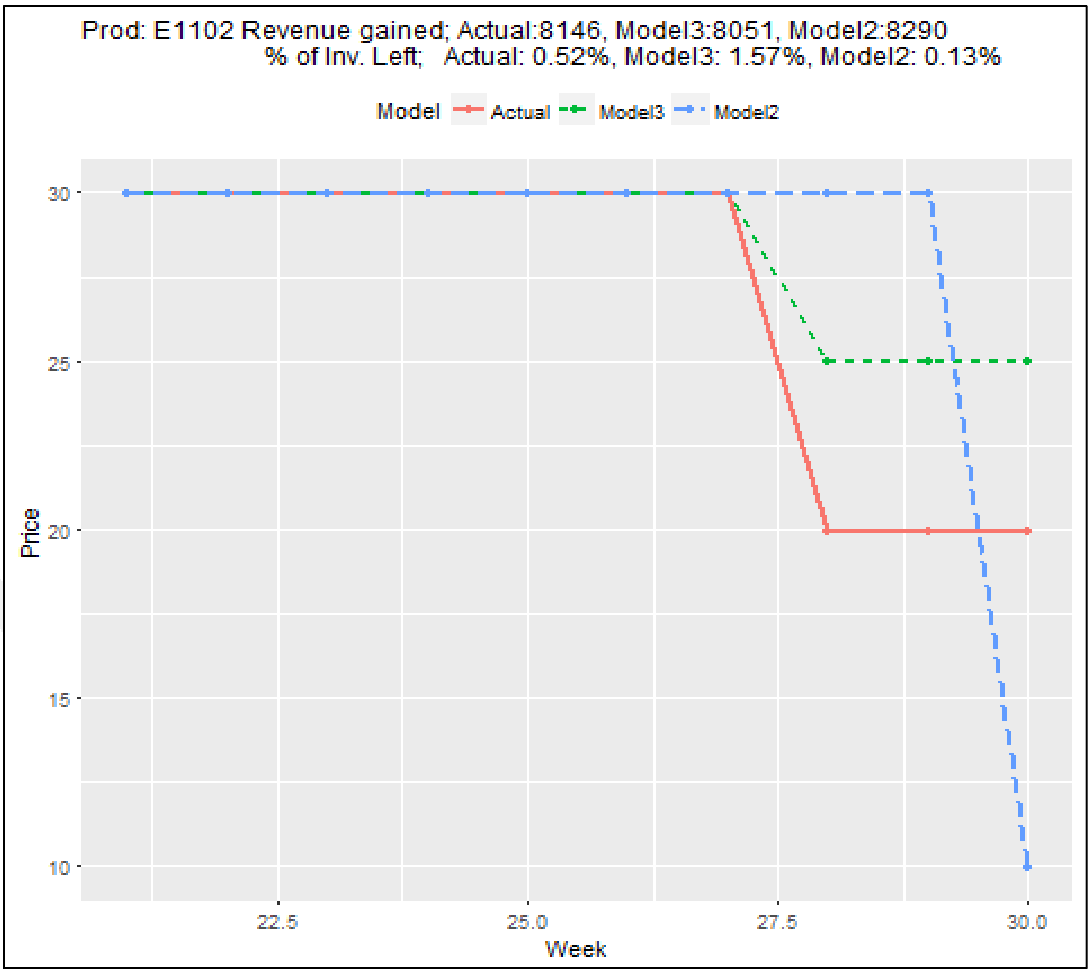

Prod: E1L34 Revenue gained; Actual:14400, Model3:15050, Model2:15522
% of Inv. Left; Actual: 1.5%, Model3: 0%, Model2: 5.08%
Model
Actual
Model3
Model2
Price
60
50
40
30
20
24
28
32
Week


Prod: J1041 Revenue gained; Actual:10272, Model3:11739, Model2:11091
% of Inv. Left; Actual: 12.2%, Model3: 3.6%, Model2: 10.8%
Model
Actual
Model3
Model2
Price
60
50
40
30
20
24
28
32
Week

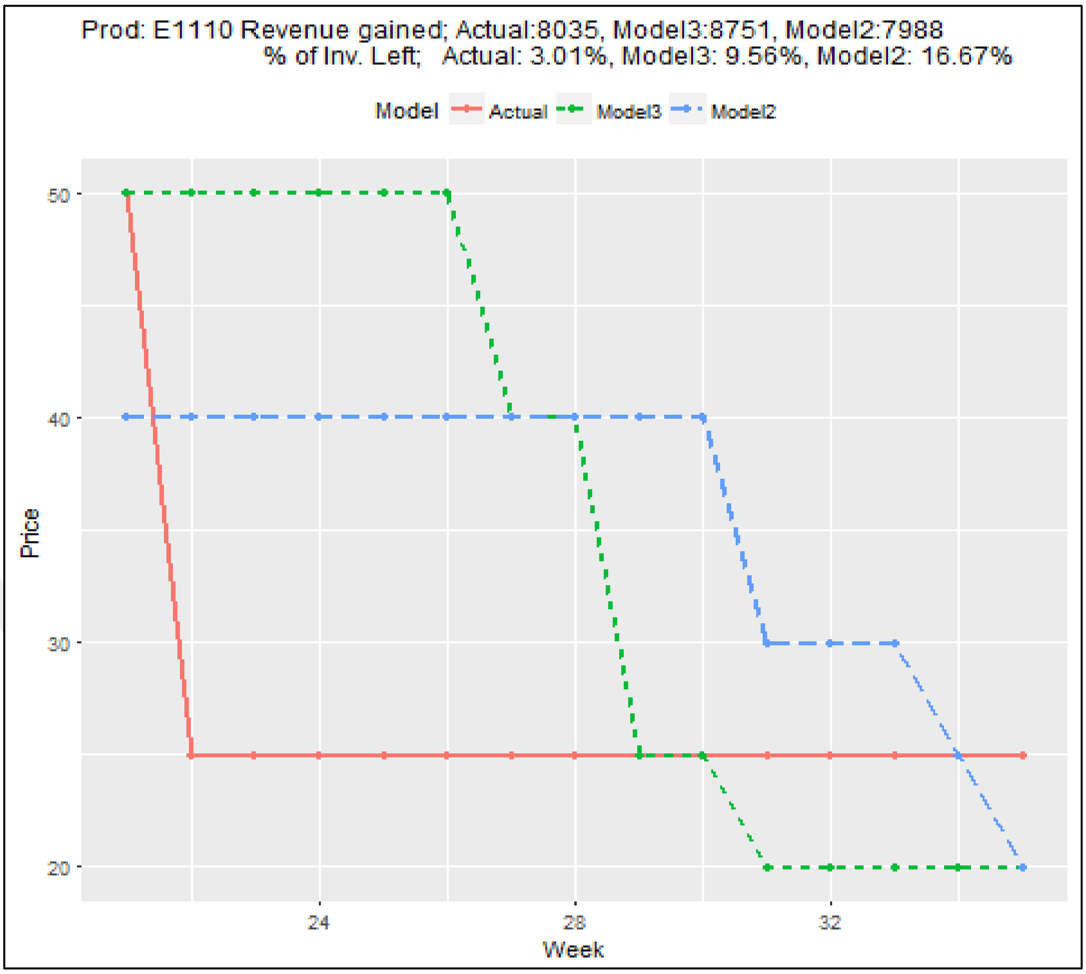

Prod: J1059 Revenue gained; Actual:13931, Model3:15037, Model2:13574
% of Inv. Left; Actual: 2.35%, Model3: 4.31%, Model2: 17.49%
Model
Actual
Model3
Model2
40
35
30
25
20
15
Price
24
28
32
Week


Prod: E1067 Revenue gained; Actual:4690, Model3:5138, Model2:5014
% of Inv. Left; Actual: 13.82%, Model3: 7.89%, Model2: 9.87%
Model
Actual
Model3
Model2
50
40
30
20
Price
24
28
32
Week

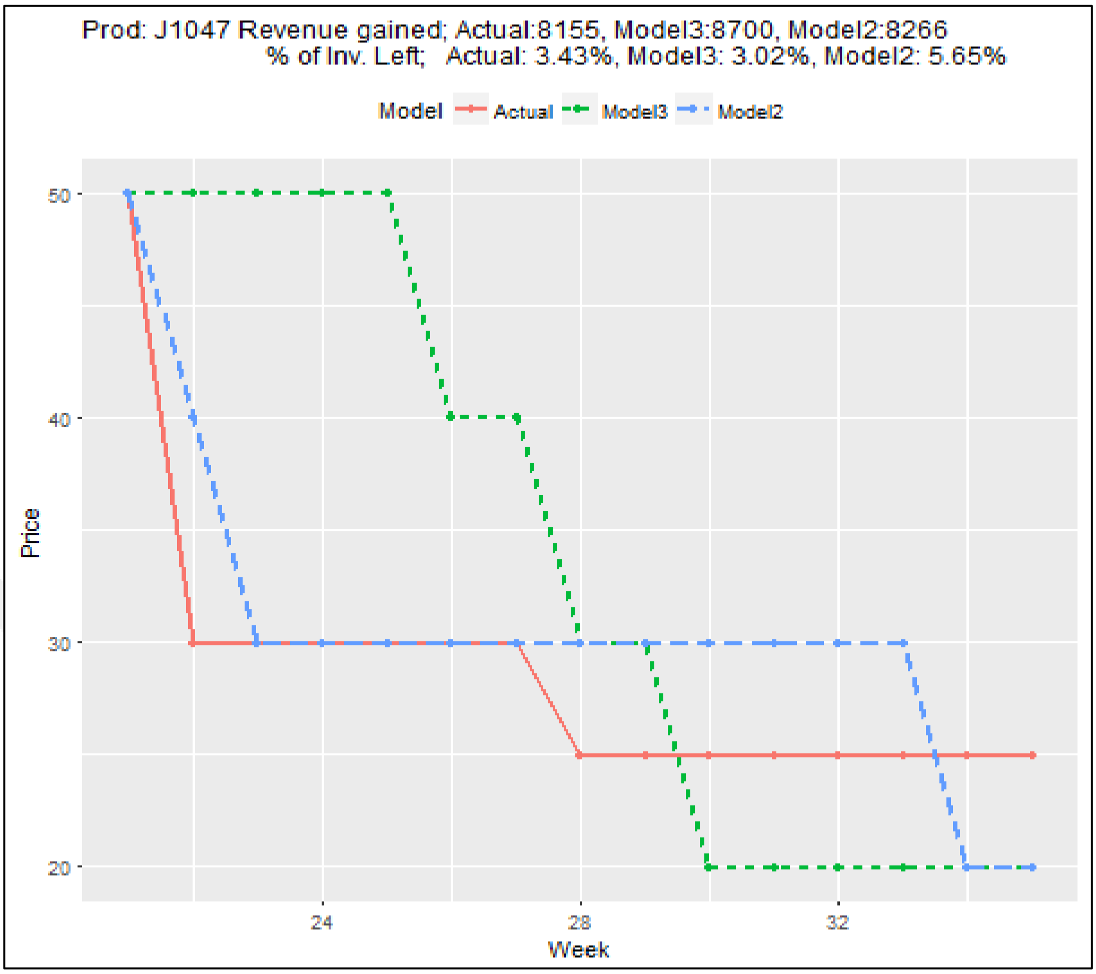

Prod: E1085 Revenue gained; Actual:5293, Model3:6174, Model2:3863
% of Inv. Left; Actual: 0.89%, Model3: 1.34%, Model2: 0%
Model
Actual
Model3
Model2
60
50
40
30
20
Price
24
28
32
Week



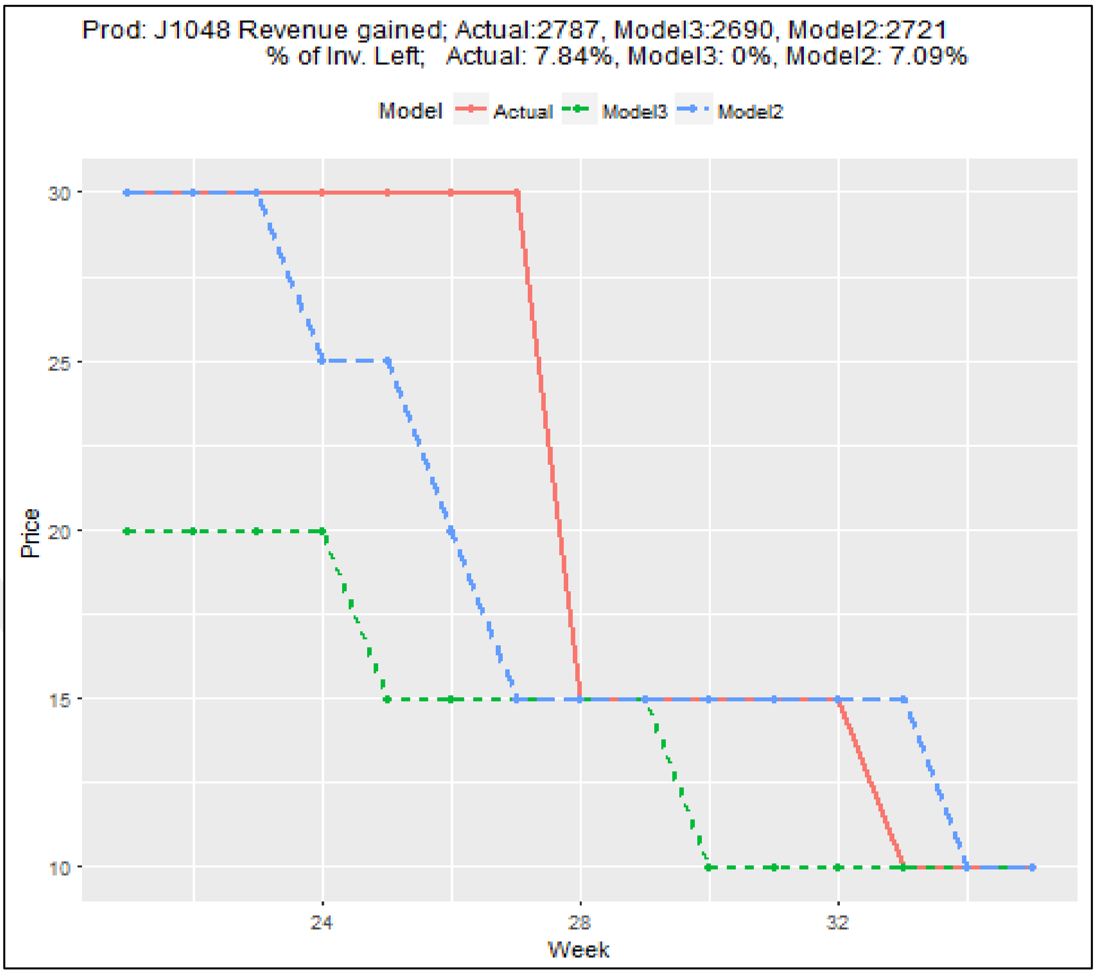
Prod: E1102 Revenue gained; Actual:8146, Model3:8051, Model2:8290
% of Inv. Left; Actual: 0.52%, Model3: 1.57%, Model2: 0.13%
Model
Actual
Model3
Model2
Price
30
25
20
15
10
22.5
25.0
27.5
30.0
Week

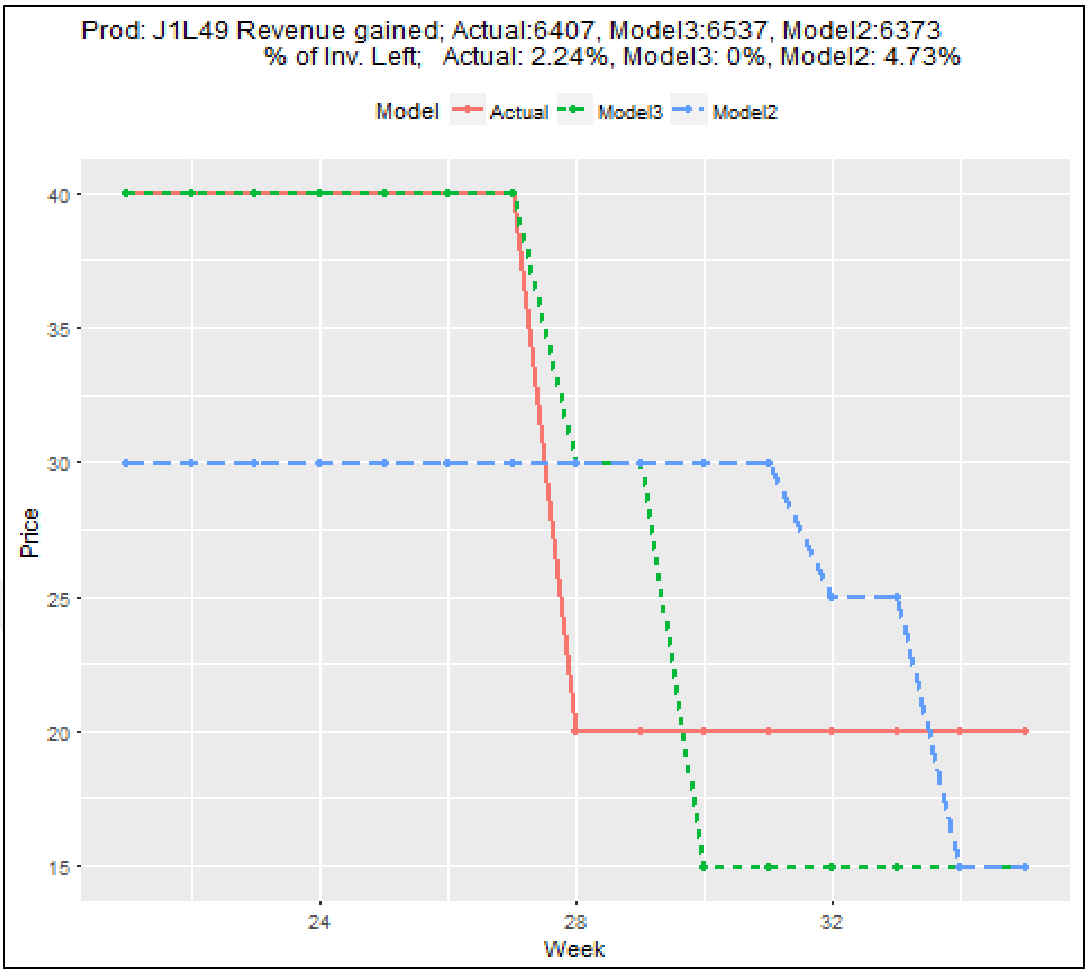


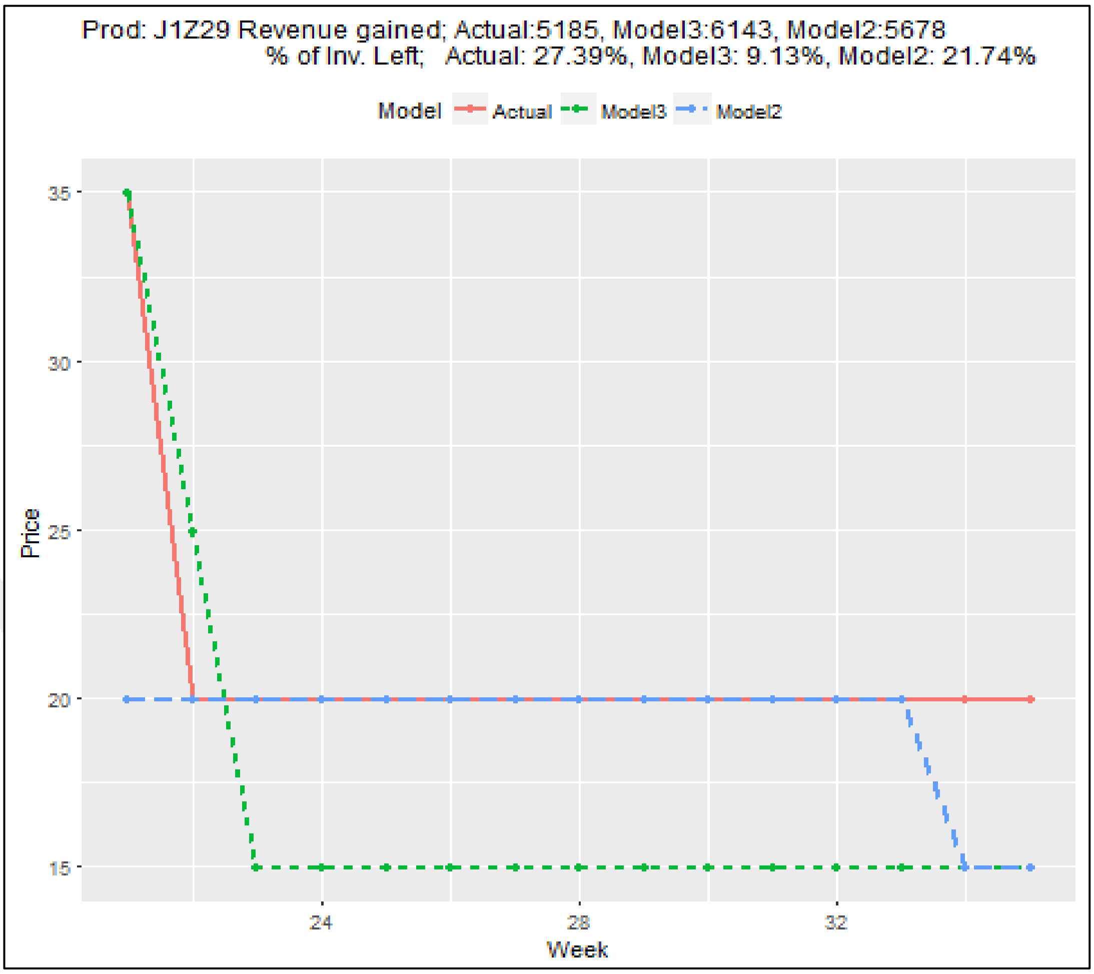
Prod: E1110 Revenue gained; Actual:8035, Model3:8751, Model2:7988
% of Inv. Left; Actual: 3.01%, Model3: 9.56%, Model2: 16.67%
Model
Actual
Model3
Model2
50
40
30
20
Price
24
28
32
Week

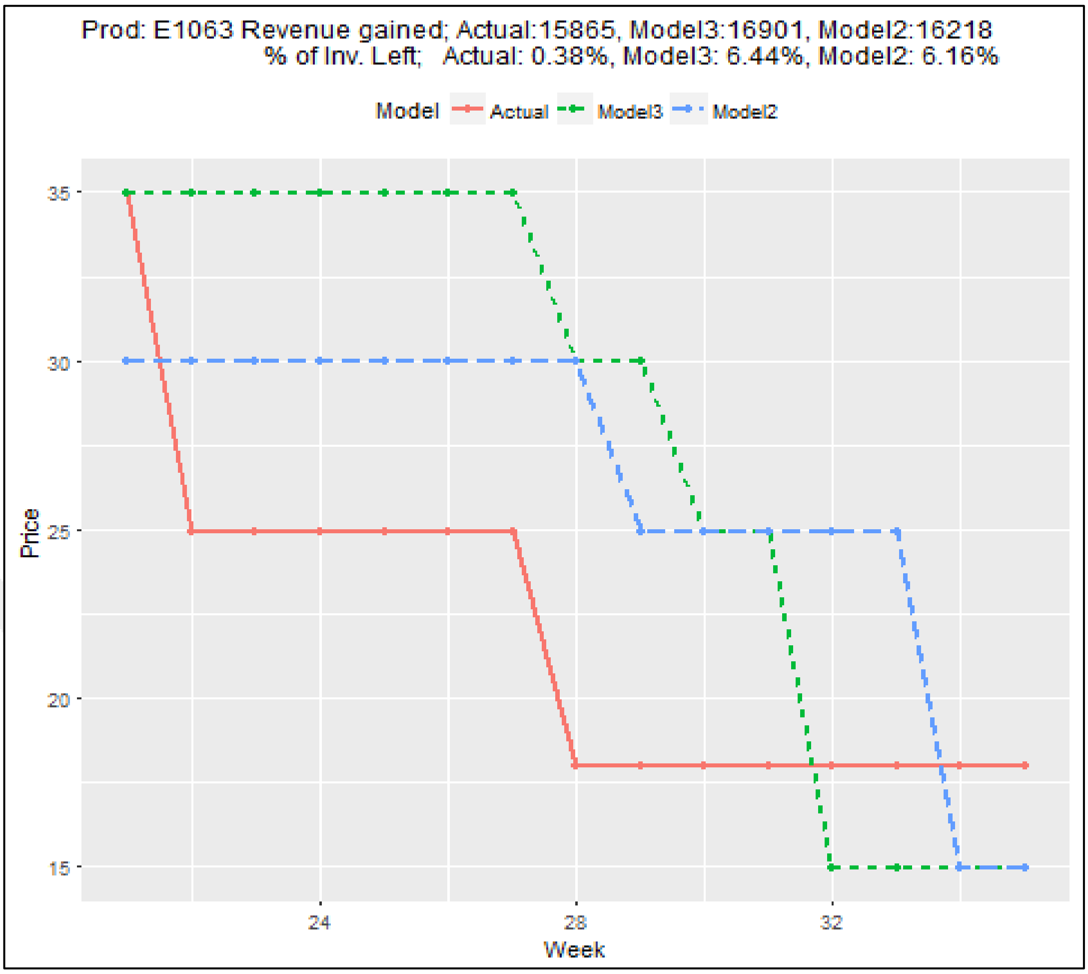


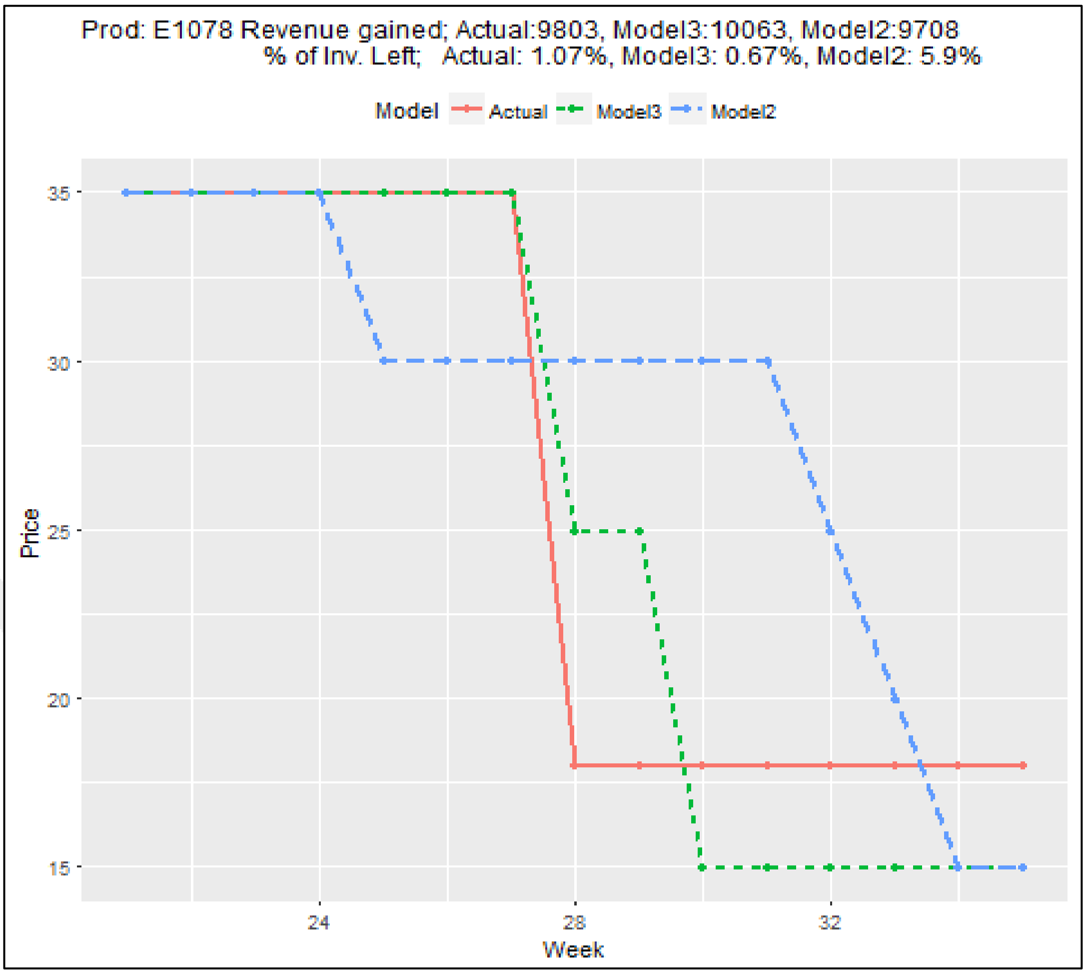
Prod: J1047 Revenue gained; Actual:8155, Model3:8700, Model2:8266
% of Inv. Left; Actual: 3.43%, Model3: 3.02%, Model2: 5.65%
Model
Actual
Model3
Model2
Price
50
40
30
20
24
28
32
Week

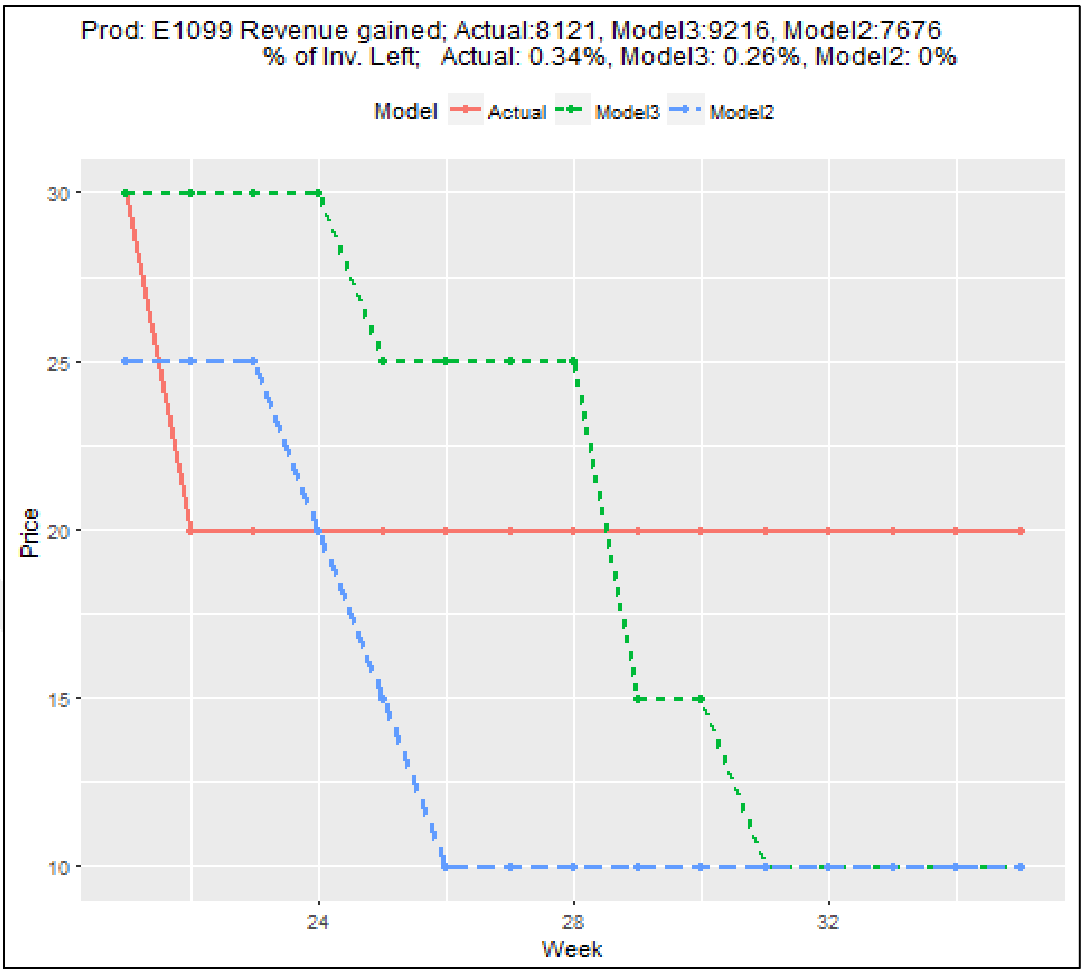

Prod: J1048 Revenue gained; Actual:2787, Model3:2690, Model2:2721
% of Inv. Left; Actual: 7.84%, Model3: 0%, Model2: 7.09%
Model
Actual
Model3
Model2
Price
30
25
20
15
10
24
28
32
Week


Prod: J1L49 Revenue gained; Actual:6407, Model3:6537, Model2:6373
% of Inv. Left; Actual: 2.24%, Model3: 0%, Model2: 4.73%
Model
Actual
Model3
Model2
Price
40
35
30
25
20
15
24
28
32
Week

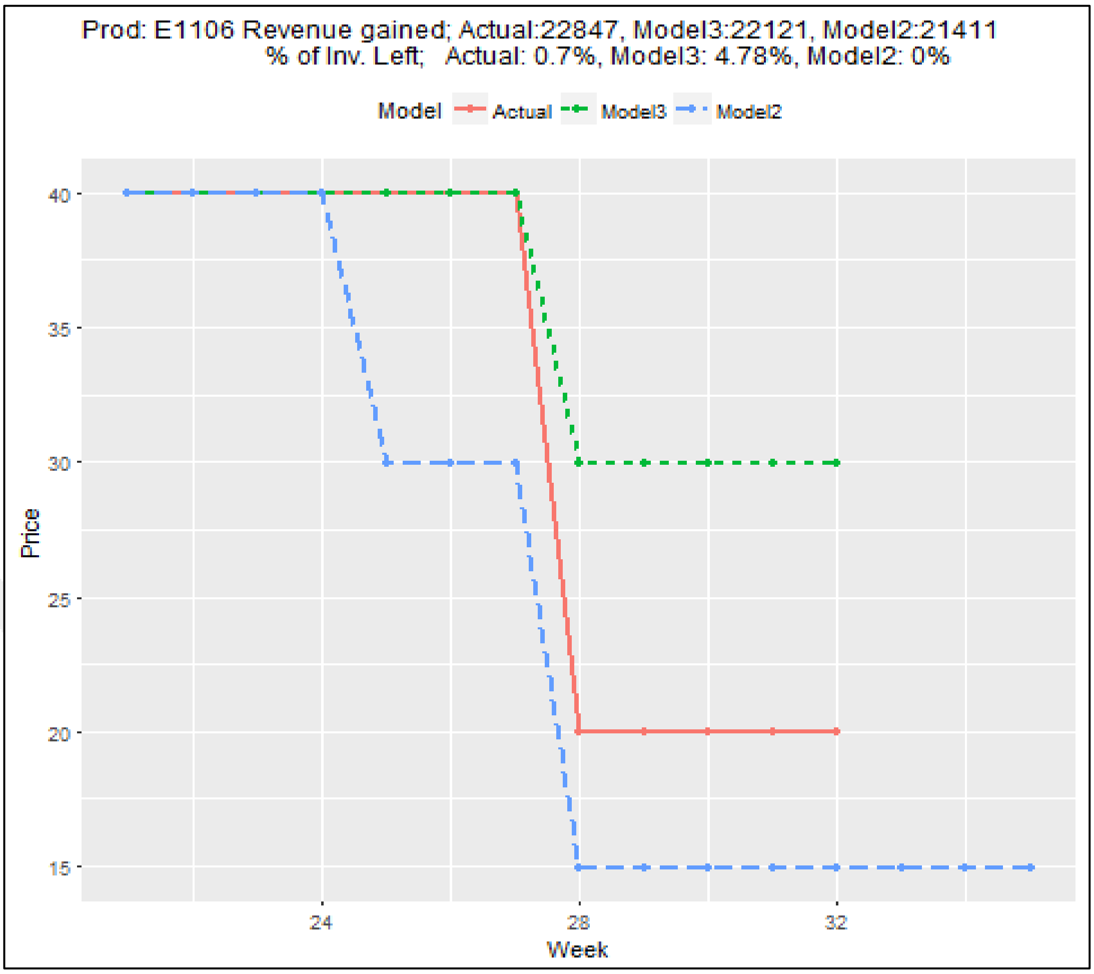

Prod: J1Z29 Revenue gained; Actual:5185, Model3:6143, Model2:5678
% of Inv. Left; Actual: 27.39%, Model3: 9.13%, Model2: 21.74%
Model
Actual
Model3
Model2
35
30
25
20
15
Price
24
28
32
Week

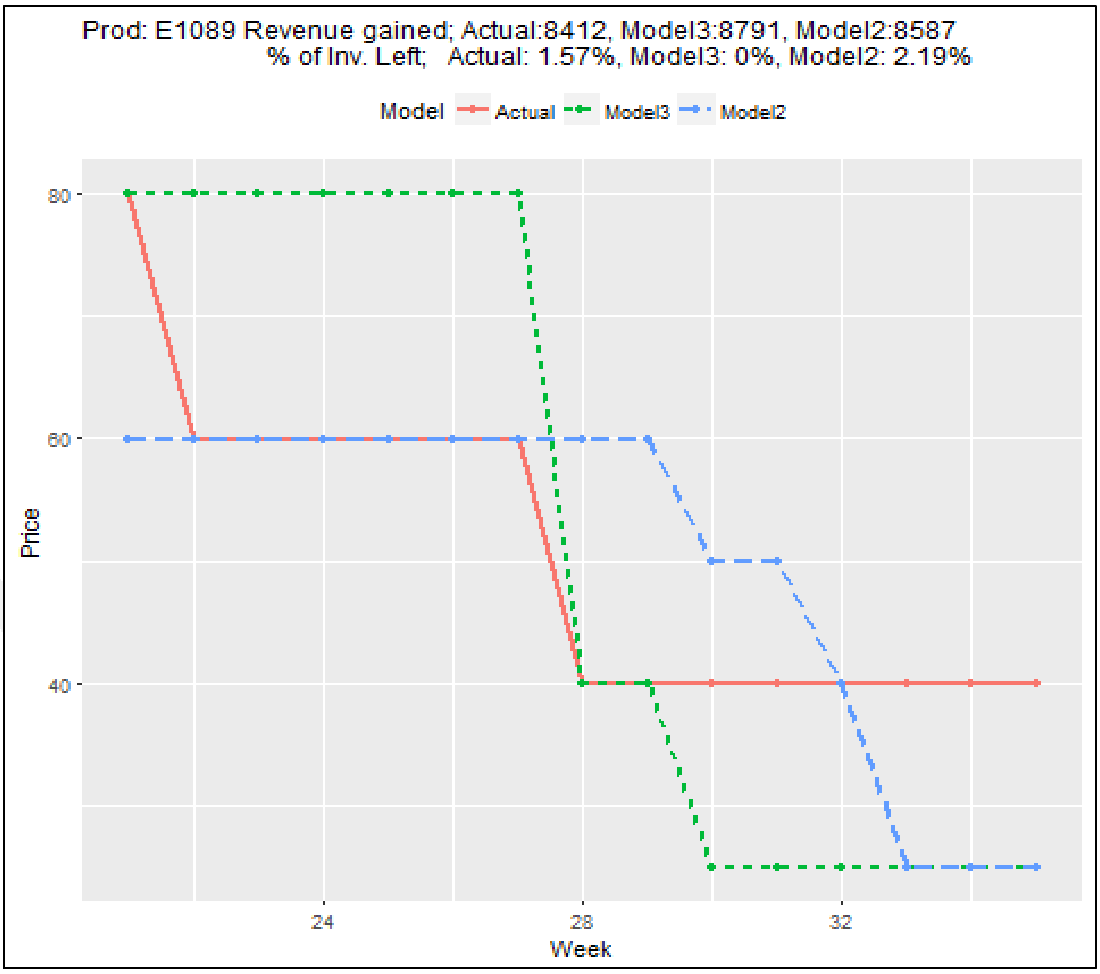

Prod: E1063 Revenue gained; Actual:15865, Model3:16901, Model2:16218
% of Inv. Left; Actual: 0.38%, Model3: 6.44%, Model2: 6.16%
Model
Actual
Model3
Model2
Price
35
30
25
20
15
24
28
32
Week

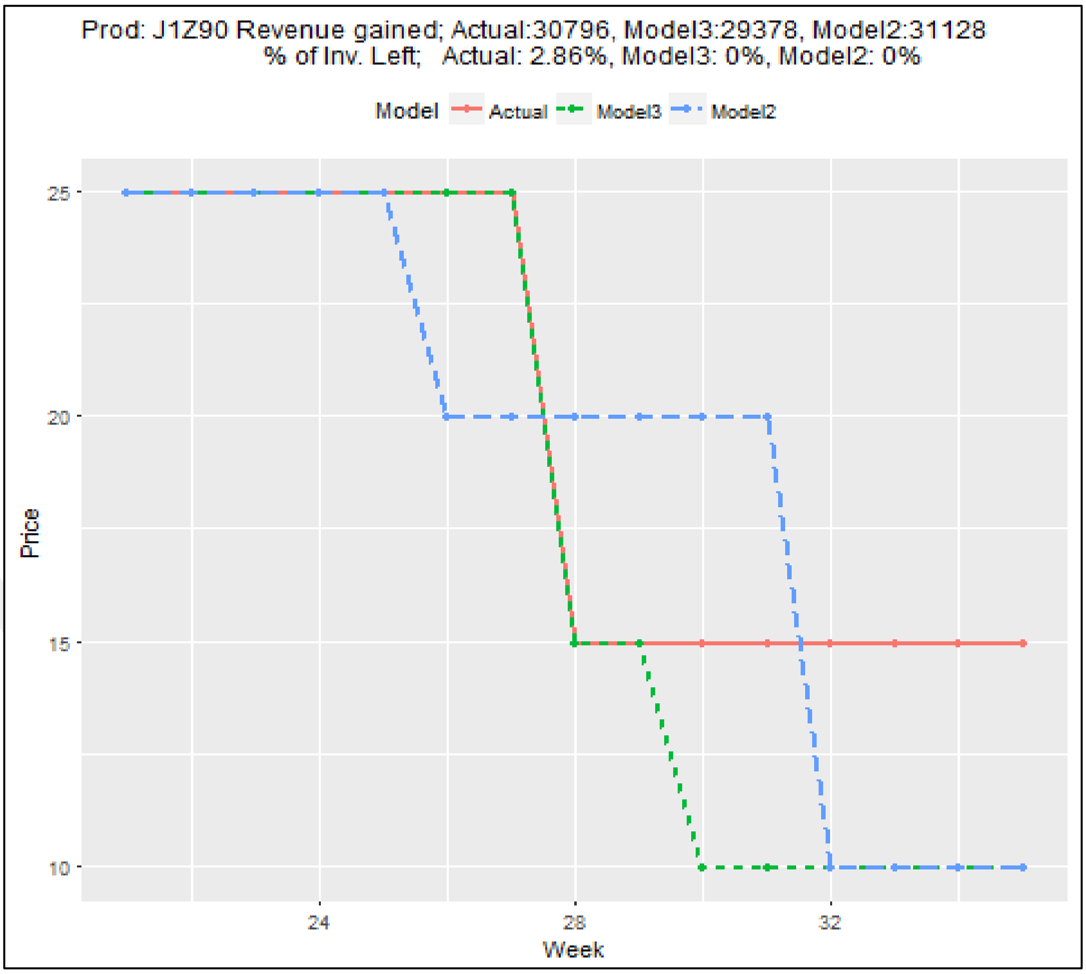

Prod: E1078 Revenue gained; Actual:9803, Model3:10063, Model2:9708
% of Inv. Left; Actual: 1.07%, Model3: 0.67%, Model2: 5.9%
Model
Actual
Model3
Model2
35
30
25
20
15
Price
24
28
32
Week

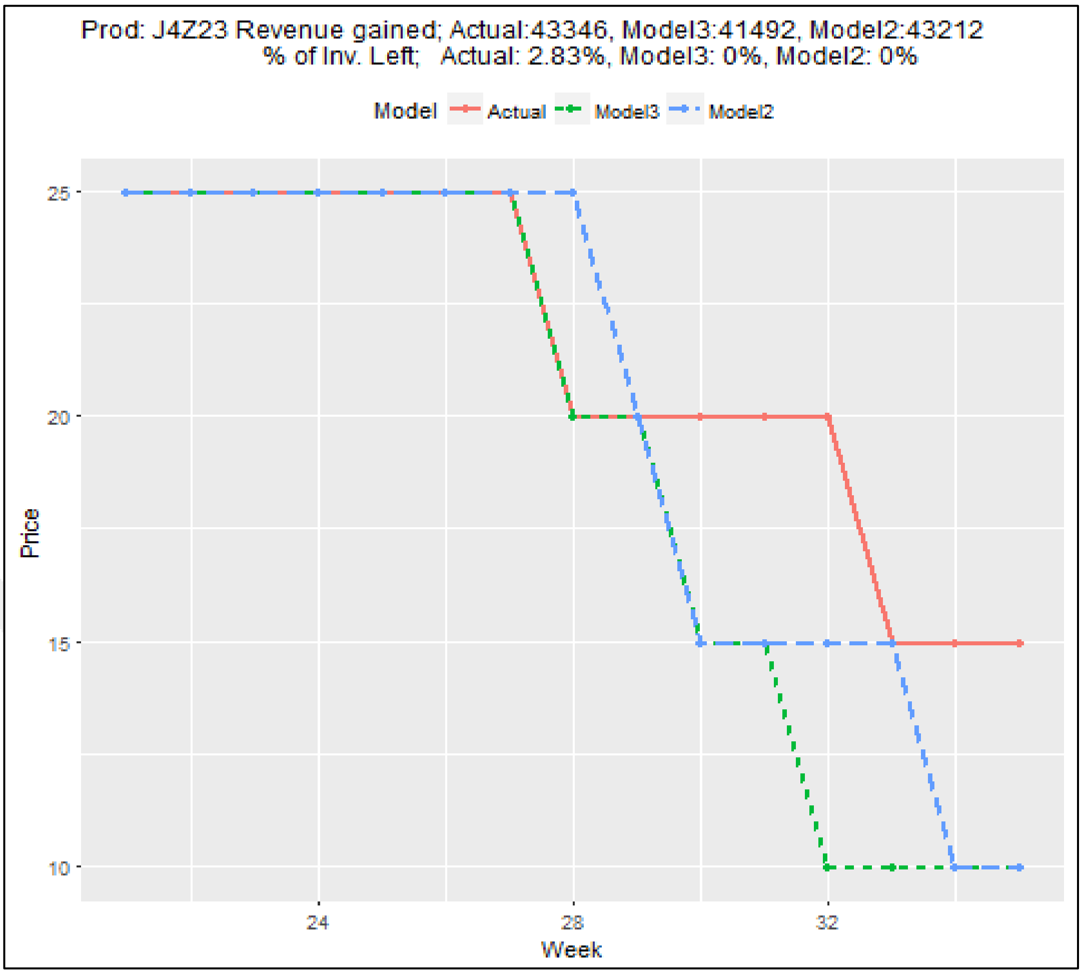


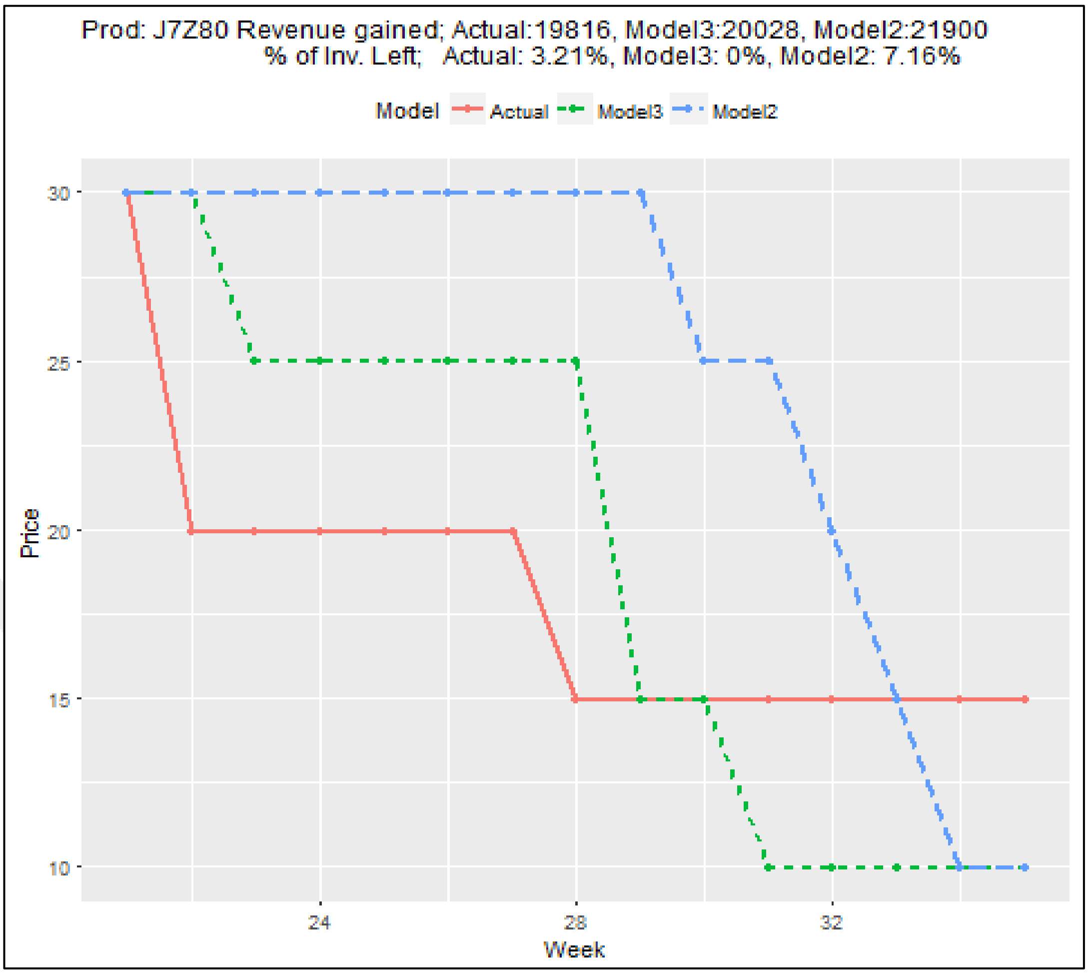
Prod: E1099 Revenue gained; Actual:8121, Model3:9216, Model2:7676
% of Inv. Left; Actual: 0.34%, Model3: 0.26%, Model2: 0%
Model
Actual
Model3
Model2
30
25
20
15
10
Price
24
28
32
Week

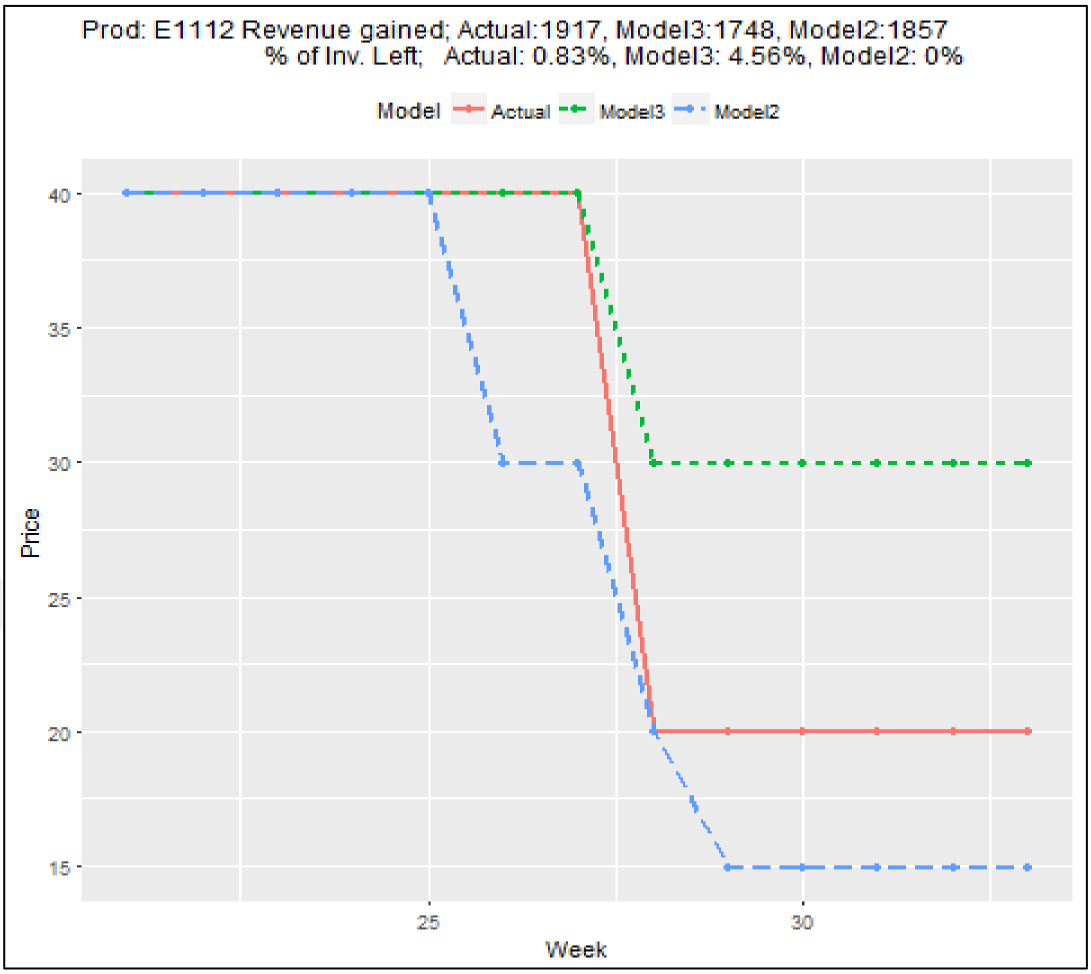

Prod: E1106 Revenue gained; Actual:22847, Model3:22121, Model2:21411
% of Inv. Left; Actual: 0.7%, Model3: 4.78%, Model2: 0%
Model
Actual
Model3
Model2
Price
40
35
30
25
20
15
24
28
32
Week

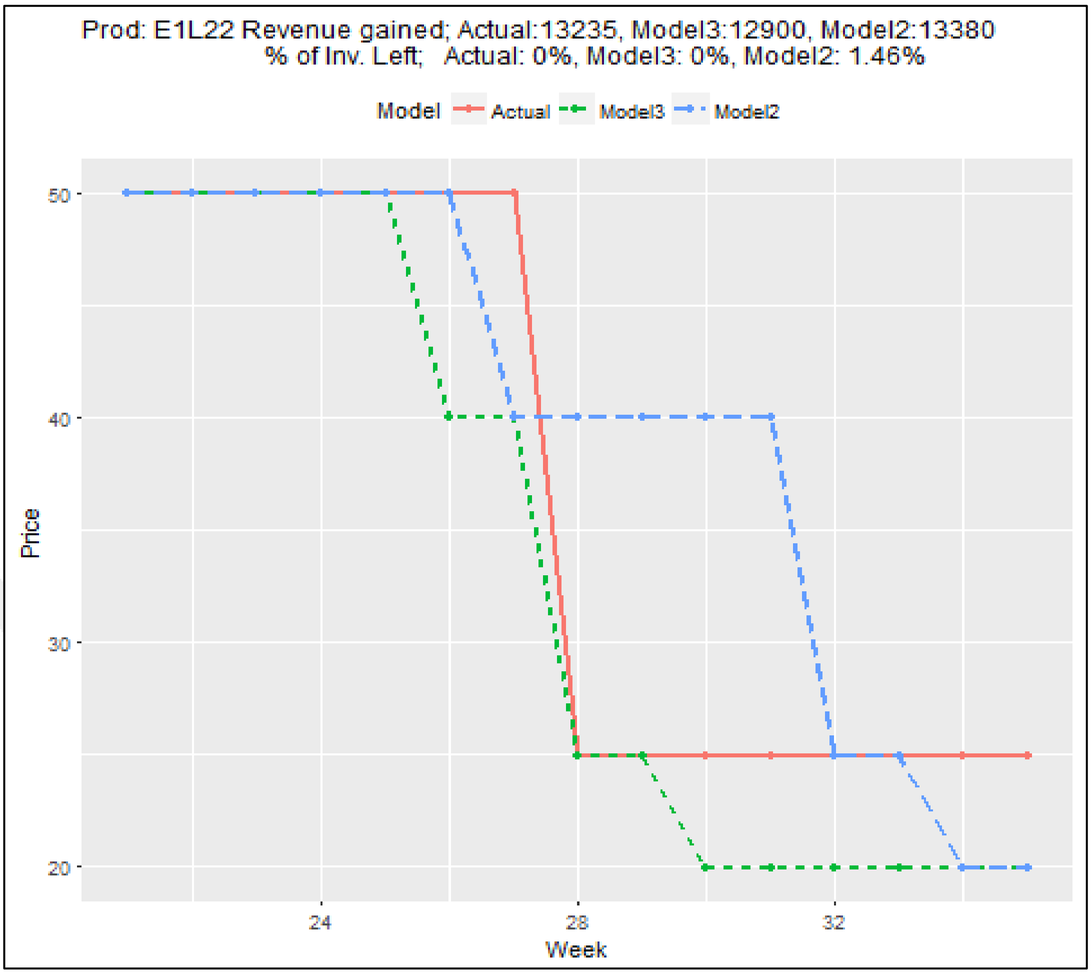

Prod: E1089 Revenue gained; Actual:8412, Model3:8791, Model2:8587
% of Inv. Left; Actual: 1.57%, Model3: 0%, Model2: 2.19%
Model
Actual
Model3
Model2
Price
80
60
40
24
28
32
Week

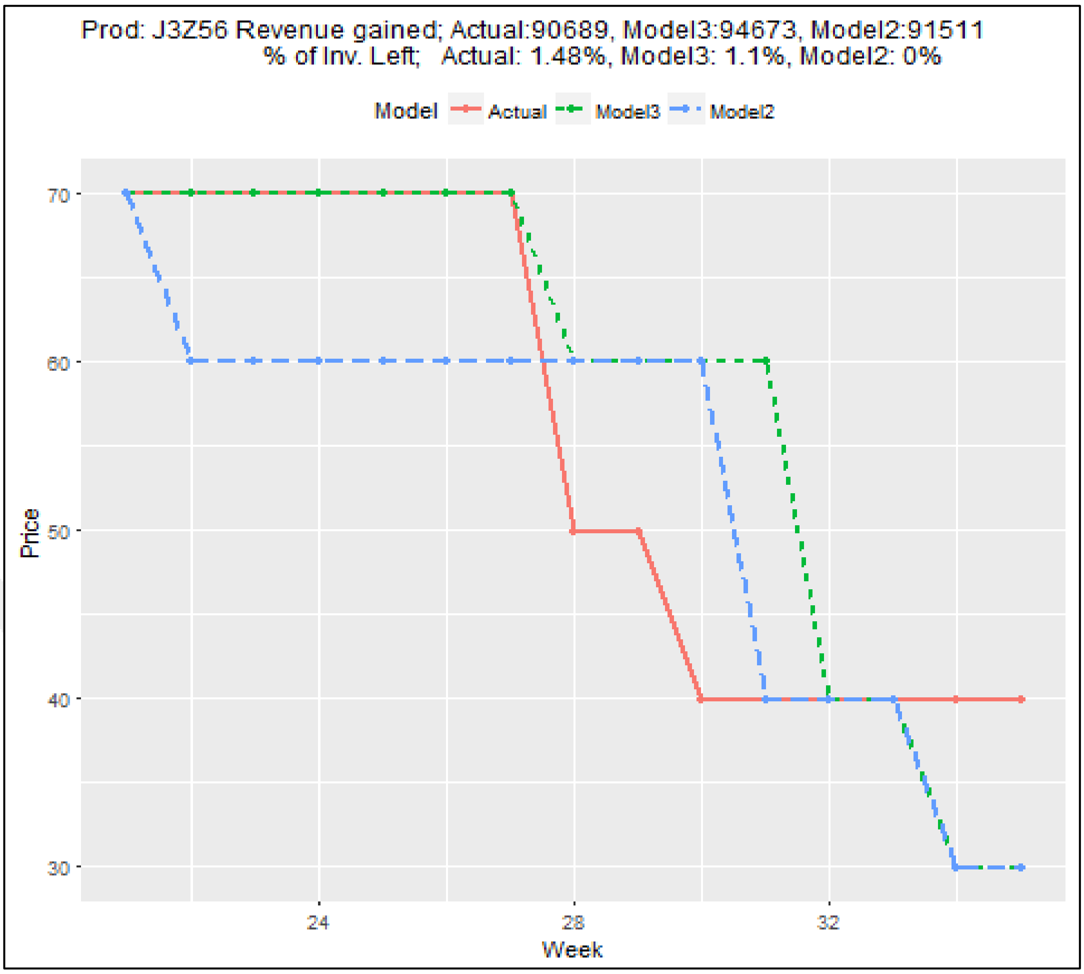

Prod: J1Z90 Revenue gained; Actual:30796, Model3:29378, Model2:31128
% of Inv. Left; Actual: 2.86%, Model3: 0%, Model2: 0%
Model
Actual
Model3
Model2
25
20
15
10
Price
24
28
32
Week


Prod: J4Z23 Revenue gained; Actual:43346, Model3:41492, Model2:43212
% of Inv. Left; Actual: 2.83%, Model3: 0%, Model2: 0%
Model
Actual
Model3
Model2
25
20
15
10
Price
24
28
32
Week


Prod: J7Z80 Revenue gained; Actual:19816, Model3:20028, Model2:21900
% of Inv. Left; Actual: 3.21%, Model3: 0%, Model2: 7.16%
Model
Actual
Model3
Model2
Price
30
25
20
15
10
24
28
32
Week


Prod: E1112 Revenue gained; Actual:1917, Model3:1748, Model2:1857
% of Inv. Left; Actual: 0.83%, Model3: 4.56%, Model2: 0%
Model
Actual
Model3
Model2
Price
40
35
30
25
20
15
25
30
Week


Prod: E1L22 Revenue gained; Actual:13235, Model3:12900, Model2:13380
% of Inv. Left; Actual: 0%, Model3: 0%, Model2: 1.46%
Model
Actual
Model3
Model2
Price
50
40
30
20
24
28
32
Week


Prod: J3Z56 Revenue gained; Actual:90689, Model3:94673, Model2:91511
% of Inv. Left; Actual: 1.48%, Model3: 1.1%, Model2: 0%
Model
Actual
Model3
Model2
Price
70
60
50
40
30
24
28
32
Week

APPENDIX 2: Codes

```
######### FORECASTING MODEL ###########
load("../YAZ_ HAFTALIK.rdata")
t[which(is.na(t$INDIRIM_TUTAR)),"INDIRIM_TUTAR"]=0

prepareDataSet <- function(dataSet){

  dataSet = dataSet[!is.na(dataSet$ILK_FIYAT),]
  dataSet$KIRIKLIK = dataSet$STOK_ADET /
(dataSet$RENK_SAYISI*dataSet$BEDEN_SAYISI*dataSet$MAGAZA_SAYISI)
  dataSet[which(dataSet$KIRIKLIK <0),"KIRIKLIK"] =0
  dataSet[which(dataSet$KIRIKLIK >1),"KIRIKLIK"] =1
  dataSet$LOG_KIRIKLIK = ifelse(dataSet$YAS>26, log(dataSet$KIRIKLIK),
log(1))
  dataSet$FIYAT_RATE = (dataSet$FIYAT/dataSet$ILK_FIYAT)
  #dataSet$FIYAT_RATE =
(dataSet$FIYAT/dataSet$ILK_FIYAT)^dataSet$KACINCI_INDIRIM
  dataSet[which(dataSet$SATIS_MIKTAR==0),"SATIS_MIKTAR"] =1

dataSet[which(dataSet$ONCEKI_SATIS_MIKTAR==0),"ONCEKI_SATIS_MIKTA
R"] =1
  dataSet$LOG_SATIS_MIKTAR = log(dataSet$SATIS_MIKTAR)
  dataSet$LOG_ONCEKI_SATIS_MIKTAR =
log(dataSet$ONCEKI_SATIS_MIKTAR)
  dataSet$LOG_STOK_ADET = log(dataSet$STOK_ADET)
  dataSet$LOG_ILK_ALIM = log(dataSet$TOPLAM_ALIM)
  dataSet$LOG_FIYAT_RATE = log(dataSet$FIYAT_RATE)
  dataSet$TARIH = as.Date(as.character(dataSet$TARIH), format = "%d.%m.%Y")
  dataSet$AY = as.factor(month(dataSet$TARIH))
  dataSet$AY1 = ifelse(as.factor(month(dataSet$TARIH))==1,1,0)
  dataSet$AY2 = ifelse(as.factor(month(dataSet$TARIH))==2,1,0)
  dataSet$AY3 = ifelse(as.factor(month(dataSet$TARIH))==3,1,0)
  dataSet$AY4 = ifelse(as.factor(month(dataSet$TARIH))==4,1,0)
  dataSet$AY5 = ifelse(as.factor(month(dataSet$TARIH))==5,1,0)
  dataSet$AY6 = ifelse(as.factor(month(dataSet$TARIH))==6,1,0)
  dataSet$AY7 = ifelse(as.factor(month(dataSet$TARIH))==7,1,0)
  dataSet$AY8 = ifelse(as.factor(month(dataSet$TARIH))==8,1,0)
  dataSet$AY9 = ifelse(as.factor(month(dataSet$TARIH))==9,1,0)
  dataSet$AY10 = ifelse(as.factor(month(dataSet$TARIH))==10,1,0)
  dataSet$AY11 = ifelse(as.factor(month(dataSet$TARIH))==11,1,0)
  dataSet$AY12 = ifelse(as.factor(month(dataSet$TARIH))==12,1,0)
  dataSet$KALAN_STOK_YUZDE = dataSet$KUMULATIF_SATIS_MIKTAR /
dataSet$TOPLAM_ALIM
  dataSet$LOG_KALAN_STOK_YUZDE = log(dataSet$KALAN_STOK_YUZDE)
  dataSet$KAMPANYA = ifelse(dataSet$INDIRIM_TUTAR>0,1,0)

  return(dataSet)
```

```
}

  regularSet = t[which(t$SEZON_YAS > 14 & t$FIYAT>0 & t$STOK_ADET>0 &
t$KUMULATIF_SATIS_MIKTAR>0 & t$TOPLAM_ALIM>200 &
t$RAMAZAN_BAYRAMI == 0 & t$KURBAN_BAYRAMI == 0 &
t$ANNELER_GUNU == 0 & t$BABALAR_GUNU == 0 & t$SEVGILILER_GUNU
== 0 & t$YILBASI == 0),]
  regularSet = prepareDataSet(regularSet)

  clearanceSet = t[which(t$SEZON_YAS > 14 & t$SEZON_YAS <40 & t$FIYAT>0
& t$STOK_ADET>0 & t$KUMULATIF_SATIS_MIKTAR>0 &
t$TOPLAM_ALIM>200 & t$RAMAZAN_BAYRAMI == 0 &
t$KURBAN_BAYRAMI == 0 & t$ANNELER_GUNU == 0 &
t$BABALAR_GUNU == 0 & t$SEVGILILER_GUNU == 0 & t$YILBASI == 0) ,]
  clearanceSet = prepareDataSet(clearanceSet)

  fullClearanceSet = t[which(t$SEZON_YAS > 14 & t$SEZON_YAS <40 &
t$FIYAT>0 & t$STOK_ADET>0 & t$KUMULATIF_SATIS_MIKTAR>0 &
t$TOPLAM_ALIM>200 ) ,]
  fullClearanceSet = prepareDataSet(fullClearanceSet)

  nMarka="1"
  train = clearanceSet[which(clearanceSet$YIL==3  &
substr(clearanceSet$URUN_KOD,1,1) == nMarka),]
  test = clearanceSet[which(clearanceSet$YIL==4  &
substr(clearanceSet$URUN_KOD,1,1) == nMarka),]

  fulltrain = fullClearanceSet[which(fullClearanceSet$YIL==3  &
substr(fullClearanceSet$URUN_KOD,1,1) == nMarka),]
  fulltest = fullClearanceSet[which(fullClearanceSet$YIL==4  &
substr(fullClearanceSet$URUN_KOD,1,1) == nMarka),]

l = step(lm(LOG_SATIS_MIKTAR ~
            LOG_ILK_ALIM  +
            LOG_KIRIKLIK +
            LOG_FIYAT_RATE +
            SEZON_YAS +
            RAMAZAN_BAYRAMI +
            ANNELER_GUNU +
            BABALAR_GUNU +
            SEVGILILER_GUNU +
            AY1 +
            AY2 +
            AY3 +
            AY4 +
            AY5 +
            AY6 +
            AY7 +
```

```
               AY8 +
               AY9 +
               AY10 +
               AY11 +
               AY12
             , data = fulltrain))

  summary(l)

#her urun icin beta hesapla
beta = data.frame(URUN_KOD = unique(fulltrain$URUN_KOD), INTERCEPT =
as.numeric(l$coefficients[1]))
 beta
 fulltrain$LOG_TAHMIN = predict(l, newdata = fulltrain) -
as.numeric(l$coefficients[1])

   for (i in 1:nrow(beta)){
     newdata = fulltrain[which(fulltrain$URUN_KOD==beta[i,"URUN_KOD"]),]
     lx = lm((LOG_SATIS_MIKTAR-LOG_TAHMIN)~1, data= newdata)
     beta[i,"INTERCEPT"] = as.double(lx$coefficients[1])

fulltrain[which(fulltrain$URUN_KOD==beta[i,"URUN_KOD"]),"LOG_TAHMIN"]
=
fulltrain[which(fulltrain$URUN_KOD==beta[i,"URUN_KOD"]),"LOG_TAHMIN"]
+ (lx$coefficients[1])
   }

  regularSeason = t[which(t$SEZON_YAS <= 14 & t$FIYAT>0 &
t$STOK_ADET>0 & t$KUMULATIF_SATIS_MIKTAR>0 &
t$TOPLAM_ALIM>200 ),]
  regularSeason = prepareDataSet(regularSeason)

  nMarka="1"
  RegularPrev = regularSeason[which(regularSeason$YIL==3  &
substr(regularSeason$URUN_KOD,1,1) == nMarka),]
  Regulartrain = regularSeason[which(regularSeason$YIL==4  &
substr(regularSeason$URUN_KOD,1,1) == nMarka),]

  #gecen senenin modeli ile betasiz tahmin yapıp
  #yeni senenin gercklesen verileri ile betaları bulacagiz

  #once gecen senin regular period ile AY katsayilarini bulalım
  l_Regular_Prev = step(lm(LOG_SATIS_MIKTAR ~
               LOG_ILK_ALIM  +
               SEZON_YAS +
               AY2 +
               AY3 +
               AY4
```

```
            , data = RegularPrev))

  summary(l_Regular_Prev)


View(l_Regular_Prev$coefficients[c("(Intercept)","LOG_ILK_ALIM","SEZON_YAS","AY2","AY3","AY4")])

  #gecen senenin katsayilari ile bu seneyi tahminedip, hataları beta0 olarak kabul edeceğiz.

  Regulartrain$LOG_TAHMIN =
l_Regular_Prev$coefficients["LOG_ILK_ALIM"]*Regulartrain$LOG_ILK_ALIM +

l_Regular_Prev$coefficients["SEZON_YAS"]*Regulartrain$SEZON_YAS +

ifelse(is.na(l_Regular_Prev$coefficients["AY2"]),0,(l_Regular_Prev$coefficients["AY2"]))*Regulartrain$AY2 +
ifelse(is.na(l_Regular_Prev$coefficients["AY3"]),0,(l_Regular_Prev$coefficients["AY3"]))*Regulartrain$AY3 +
ifelse(is.na(l_Regular_Prev$coefficients["AY4"]),0,(l_Regular_Prev$coefficients["AY4"]))*Regulartrain$AY4

  plot(Regulartrain$LOG_TAHMIN,Regulartrain$LOG_SATIS_MIKTAR)

    #her urun icin beta hesapla
 betaRegular = data.frame(URUN_KOD = unique(Regulartrain$URUN_KOD),
INTERCEPT = 0)
 betaRegular

   for (i in 1:nrow(betaRegular)){
     newdata =
Regulartrain[which(Regulartrain$URUN_KOD==betaRegular[i,"URUN_KOD"]),]
     lx = lm((LOG_SATIS_MIKTAR-LOG_TAHMIN)~1, data= newdata)
     betaRegular[i,"INTERCEPT"] = as.double(lx$coefficients[1])

Regulartrain[which(Regulartrain$URUN_KOD==betaRegular[i,"URUN_KOD"]),"LOG_TAHMIN"] =
Regulartrain[which(Regulartrain$URUN_KOD==betaRegular[i,"URUN_KOD"]),"LOG_TAHMIN"] + (lx$coefficients[1])
   }

   MarkdownSeason = t[which(t$FIYAT>0 & t$STOK_ADET>0 &
t$KUMULATIF_SATIS_MIKTAR>0 & t$TOPLAM_ALIM>200 ),]
  MarkdownSeason = prepareDataSet(MarkdownSeason)

  nMarka="1"
```

```
  MarkdownSeason = MarkdownSeason[which(MarkdownSeason$YIL==4  &
substr(MarkdownSeason$URUN_KOD,1,1) == nMarka),]

  #2013 senenin katsayilari ve 2014 betalari ile tahmin yap
  MarkdownSeason = merge(MarkdownSeason, betaRegular, by = "URUN_KOD")

  c_LOG_ILK_ALIM =
ifelse(is.na(l$coefficients["LOG_ILK_ALIM"]),0,l$coefficients["LOG_ILK_ALIM"]
)
  c_LOG_KIRIKLIK =
ifelse(is.na(l$coefficients["LOG_KIRIKLIK"]),0,l$coefficients["LOG_KIRIKLIK"])
  c_SEZON_YAS =
ifelse(is.na(l$coefficients["SEZON_YAS"]),0,l$coefficients["SEZON_YAS"])
  c_LOG_FIYAT_RATE =
ifelse(is.na(l$coefficients["LOG_FIYAT_RATE"]),0,l$coefficients["LOG_FIYAT_R
ATE"])
  c_RAMAZAN_BAYRAMI =
ifelse(is.na(l$coefficients["RAMAZAN_BAYRAMI"]),0,l$coefficients["RAMAZAN
_BAYRAMI"])
  c_AY2 = ifelse(is.na(l$coefficients["AY2"]),0,l$coefficients["AY2"])
  c_AY3 = ifelse(is.na(l$coefficients["AY3"]),0,l$coefficients["AY3"])
  c_AY4 = ifelse(is.na(l$coefficients["AY4"]),0,l$coefficients["AY4"])
  c_AY5 = ifelse(is.na(l$coefficients["AY5"]),0,l$coefficients["AY5"])
  c_AY6 = ifelse(is.na(l$coefficients["AY6"]),0,l$coefficients["AY6"])
  c_AY7 = ifelse(is.na(l$coefficients["AY7"]),0,l$coefficients["AY7"])
  c_AY8 = ifelse(is.na(l$coefficients["AY8"]),0,l$coefficients["AY8"])
  c_AY9 = ifelse(is.na(l$coefficients["AY9"]),0,l$coefficients["AY9"])

  summary(l)
  #gecen senenin katsayilari ile bu seneyi tahminedip, hataları bulalim
   MarkdownSeason$GERCEK_INTERCEPT =
MarkdownSeason$LOG_SATIS_MIKTAR - (
                     c_SEZON_YAS*MarkdownSeason$SEZON_YAS +
                     c_LOG_ILK_ALIM*MarkdownSeason$LOG_ILK_ALIM +
                     c_LOG_KIRIKLIK*MarkdownSeason$LOG_KIRIKLIK +
                     c_LOG_FIYAT_RATE*MarkdownSeason$LOG_FIYAT_RATE +

c_RAMAZAN_BAYRAMI*MarkdownSeason$RAMAZAN_BAYRAMI +
                     c_AY2*MarkdownSeason$AY2 +
                     c_AY3*MarkdownSeason$AY3 +
                     c_AY4*MarkdownSeason$AY4 +
                     c_AY5*MarkdownSeason$AY5 +
                     c_AY6*MarkdownSeason$AY6 +
                     c_AY7*MarkdownSeason$AY7 +
                     c_AY8*MarkdownSeason$AY8 +
                     c_AY9*MarkdownSeason$AY9)

   MarkdownSeason$CORRECTED_INTERCEPT = 0
```

```
  MarkdownSeason$REGRESSED_INTERCEPT = 0

 for (i in 1:nrow(betaRegular)){
   for (j in
min(MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"UR
UN_KOD"]),"SEZON_YAS"]):max(MarkdownSeason[which(MarkdownSeason$UR
UN_KOD==betaRegular[i,"URUN_KOD"]),"SEZON_YAS"])){


     livedata =
MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS <=j),]

     if (nrow(livedata)>0){


       result = tryCatch({

         # ortalama ile beta tahmini
          corrected_intercept = mean(livedata[which(livedata$SEZON_YAS>j-3 &
livedata$SEZON_YAS<j),"GERCEK_INTERCEPT"])

         if (is.na(corrected_intercept)){

MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"CORRECTED_INTERCEPT"] =
MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"GERCEK_INTERCEPT"]
         } else {

MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"CORRECTED_INTERCEPT"] =
corrected_intercept
         }

         #regressyon ile beta tahmini
          l_corr = lm(GERCEK_INTERCEPT~SEZON_YAS ,
data=livedata[which(livedata$SEZON_YAS<j),c("GERCEK_INTERCEPT","SEZO
N_YAS")])
          pred_intercept = l_corr$coefficients["(Intercept)"] +
(ifelse(is.na(l_corr$coefficients["SEZON_YAS"]),0,l_corr$coefficients["SEZON_Y
AS"])*j)
         if (is.na(pred_intercept)){

MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"REGRESSED_INTERCEPT"] =
MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"GERCEK_INTERCEPT"]
```

```
          } else {

MarkdownSeason[which(MarkdownSeason$URUN_KOD==betaRegular[i,"URUN_
KOD"] & MarkdownSeason$SEZON_YAS ==j),"REGRESSED_INTERCEPT"] =
(pred_intercept*0.5 + corrected_intercept*0.5)
          }

        }, warning = function(w) {

        }, error = function(e) {

        }, finally = {

        }
        )
      }

    }

  }


# Deterministic Optimum
MaxRevenue_d <- function (u, h, f, s) {
  #mean(mpe[which(mpe<2)]), sd(mpe[which(mpe<2)])
  #hata = rnorm(1,0.0956355,0.5750935)
  hata = 0
  min_f = 9
  ind = .75

  # initial condition
  if (h > 35 | f<min_f | s<=0) {

    if (h > 35){
      return(-1 * max(0,s) * (DynSeason[which(DynSeason$URUN_KOD == u &
DynSeason$SEZON_YAS==15),"ILK_FIYAT"]) * 0.5)
    } else {
      return(0)
    }
  }
  else {
    return(max(
      (ifelse((f<min_f ),0,((f_talep_d(u, h, f, s, hata) * f) + MaxRevenue_d(u, (h + 1), f,
((s - f_talep_d(u, h, f, s,hata) ))))))
      ,(ifelse(((f_round(f*ind,5))<min_f ),0,(( f_talep_d(u, h, (f_round(f*ind,5)), s,
hata) * (f_round(f*ind,5))) + MaxRevenue_d(u, (h + 1), (f_round(f*ind,5)), (s -
f_talep_d(u, h, (f_round(f*ind,5)), s, hata))))))
    )
```

```
      )
    }
  }


#Heuristic
heuristicResults <- function(u){

    require(ggplot2)
    det = compareResults(u,21,T)
    stoch = compareResults(u,21,F)
    maxWeek = max(det$SEZON_YAS)
    actRevenue=round(det[which(det$SEZON_YAS==maxWeek),"NET_REVENUE"])

Model1Revenue=round(det[which(det$SEZON_YAS==maxWeek),"OPT_NET_REVE
NUE"])

Model2Revenue=round(stoch[which(stoch$SEZON_YAS==maxWeek),"OPT_NET_R
EVENUE"])

    df2 = data.frame(Model = "Actual", Week = det$SEZON_YAS, Price = det$FIYAT)
    df2 = rbind(df2, data.frame(Model = "Model3", Week = det$SEZON_YAS, Price =
det$OPT_FIYAT))
    df2 = rbind(df2, data.frame(Model = "Model2", Week = stoch$SEZON_YAS, Price
= stoch$OPT_FIYAT))


    x = sort(df2[which(df2$Model=="Model3"),"Price"],decreasing = F)
    x1= x

    # don=T
    # while(don){
      counter=1
      fiyat = x[1]
      for (i in 1:length(x)){
         if (fiyat != x[i]){
           if (counter >= 2 ){
             #sorun yok
             counter = 1
           } else {
              if (i!=length(x)){
                x[i] = fiyat
              }
             counter = counter+1
           }

         } else {
           counter = counter +1
```

```
        }
       fiyat = x[i]

     }


   #print(u)
   return (data.frame(urun=u, hafta=df2[which(df2$Model=="Model3"),"Week"],
stok=0, origfiyat=sort(x1,decreasing = T), kararfiyat=sort(x,decreasing = T), talep=0))
```