Markdown Pricing For a Large Scale Retailer

Vibhati Burman
TCS Research
Tata Consultancy Services
Chennai, India
vibhati.b@tcs.com

Rajesh Kumar Vashishtha
TCS Research
Tata Consultancy Services
Chennai, India
r.vashishtha@tcs.com

Srividhya Sethuraman
TCS Research
Tata Consultancy Services
Chennai, India
srividhya.sethuraman1@tcs.com

Ganesh Radhakrishnan*
TCS Retail
Tata Consultancy Services
Chennai, India

Prashanth Ganesan*
TCS Retail
Tata Consultancy Services
Chennai, India

Suresh Kumar V
TCS Retail
Tata Consultancy Services
Chennai, India
sureshkumar.v2@tcs.com

Sharadha Ramanan
TCS Research
Tata Consultancy Services
Chennai, India
sharadha.ramanan@tcs.com

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671641

***Abstract*—A large European-based international discount store chain retailer, operating in more than 1000 stores across countries, wanted to make a transition from traditional cost-based markdown pricing to an optimal markdown pricing that maximizes the overall revenue as well as clears the inventory. Their regular pricing strategy was to price lowest among all their competitors since they are a discount store chain. Their markdown pricing problem is both critical and complex because there are many requirements to be satisfied, albeit simultaneously. The overall revenue had to be maximized as well as the given percentage of inventory was to be cleared by the end of the markdown period. Also, inventory had to remain in the stores till the end of the season so that the footfalls do not decrease and the markdown pricing was to be at the item group level for interrelated items. We developed a novel markdown solution by utilizing the retailers' historical markdown performance data to come up with the markdown price estimates. Our approach was to obtain initial markdown price estimates for each item, apply ML/DL algorithms to forecast sales, compute price elasticities and build a nonlinear markdown price optimization system to recommend optimal prices. In this paper, we give the details and results for one category - the Clothing category. We did back testing for multiple periods to compare our sales forecasting model outputs using initial price, with their actuals. We have employed a distributed computing and parallel execution framework in cloud to obtain optimal markdown prices for products in the clearance season. Our final recommended price was greater by around 20% than the actual markdown prices for most items and this led to around 6% decrease in sales units while satisfying their inventory constraints. Our optimal pricing solution resulted in 10% increase in revenue for Clothing as compared to their actuals. Our markdown solution was later scaled to other categories as well as to stores in other countries.**

***Index Terms*—Demand Forecast, Clothing, Big Data, Deep learning, Large scale retailer, Discount store chain, Markdown pricing, Non- linear optimization, Machine Learning.**

I. Introduction

A markdown is a permanent reduction, usually over a relatively short time interval, in the price of a retail merchandise. Unlike promotions, markdown prices are permanent in nature. Markdown could be introduced for several reasons including reduced demand due to change in customer perception induced by social media platforms, planned assortment changes, or due to incorrectly estimated inventory levels of the retail merchandise. For retailers who sell short-life-cycle products such as seasonal and fashion items, markdown pricing is a crucial consideration. Further, industries such as Retail, Hospitality, Aviation, Manufacturing and others are fast moving from cost-based pricing to dynamic and revenue-based markdown pricing to avoid underpricing and under selling of items. Although markdown prices promote sales, they also lead to reduced margins. Therefore, the challenge is to optimally price products for markdown that maximize revenue as well as the sales units.

There are several challenges a retailer faces in coming up with optimal markdown prices that would be effective for clearing the remaining inventory while also optimizing the overall revenue. First, lack of historical markdown data for products going into their first markdown is one challenge. Second, the demand and price elasticity estimation is difficult due to evolving customer behaviour, promotional events, marketing campaigns, competitor strategy, and the lack of price changes in the past. Third, retailers typically have more than one requirement to be satisfied, albeit simultaneously – that is, revenue to be maximized, given percentage of inventory to be cleared, and inventory to remain in stores till end of season so that footfalls do not do not decrease.

In this paper, we have addressed the retailers' problem of coming up with an initial markdown price for each retail merchandise in a novel way. We first estimate the initial markdown prices by utilizing the retailers' historical markdown performance data and then optimize it further to meet the retailers' requirements. This is a novel way of estimating markdown prices as the existing approaches do not utilize the retailers' past markdown performance data to come up with initial markdown prices. We do not need to use in-season markdown data for demand estimates as seen in the

*Ganesh Radhakrishnan and Prashanth Ganesan were employees of TCS when this work was done.

We thank Uma Maheswari G., TCS Research, for her inputs to this paper.

978-1-6654-3902-2/21/$31.00 ©2021 IEEE

literature. We also address the impact of remaining inventory level on the demand. This is achieved by using temporal features along with the remaining inventory as a feature for demand forecasting. To overcome the challenge in estimating price elasticity posed by the lack of price changes in the history data, we perform grouping of articles at different levels of hierarchy and based on their attributes. In this paper, we have formulated and solved markdown pricing as a non-linear optimization problem. This formulation helps the retailer to come up with an optimal price that is a balance among revenue maximization, end inventory minimization and non-zero weekly sales rate to encourage customer footfalls.

We have also addressed the problem of markdown pricing at item group level for inter-related items. Items may be inter-related in different ways such as inter-linked items, substitutes or dependent items. We observed from history that the same markdown price was given to similar items having almost same regular prices and going for markdown at the same time. For example, similar tops and t-shirts having almost same regular prices, were given the same markdown price. We came up with optimal markdown pricing at item group level for inter-related items.

The organization of this paper is given as follows: Section II, a review of related work is presented. In Section III, characteristics of the retailer data is summarized. Section IV, the methodology followed is discussed. In Section V, experiments done and the results obtained are presented. Finally, in Section VI, the conclusions of the work and the direction of future research work are discussed.

II. RELATED WORK

Markdown pricing has been extensively studied along different perspectives in literature. In [1], the authors study the inventory dependence on demand and its impact on pricing and distribution across multiple retail locations. Their solution considers pricing, inventory allocation and store consolidation. Our pricing is with respect to inventory dependent demand, as inventory allocation as per our scenario, takes place at Distribution Center (DC) level. Their solution is at a single store or a group of stores level, whereas our solution is flexible and can be applied at any level of granularity, namely, store-article or store group article or article DC level.

In [2], the authors formulate a dynamic program corresponding to the multi-period and multi-product inventory and pricing coordination problem for a product group within a given country using revenue maximization as the objective for a fast fashion retailer. They do not consider substitution effects in their model. In our work, we propose an optimization model that operates at article-national level wherein price elasticity is calculated in the presence of similar articles that act as substitutes to each other at one of the levels.

In [3], the authors have developed and deployed a markdown pricing optimization system for food, furniture and consumable products across Walmart stores. They use deep reinforcement learning, simulation and optimization to arrive at optimal markdown prices. The deep learning models as part of our solution do not use any simulated data. Our solution addresses the challenges associated with the rapidly evolving clothing/apparel industry. Further, our solution also offers optimal markdown prices at item group level consisting of inter-related items.

In [4], the authors present an optimization approach for a limited but potentially large amount of inventory that is stored at multiple fulfillment centers and must be sold by a certain exit date. Their objective is to maximize gross profit that is total revenue minus the shipping cost. Their scope of optimization is limited to optimizing markdown price for a single product, taking into account demand cannibalization of other products in e-commerce. In our work, we perform price optimization for all articles scheduled for markdown for a discount store chain comprising of brick-and mortar stores wherein the calculation of unsold inventory across different stores is a rather complex scenario.

In [5], the authors propose a markdown pricing strategy for holiday baskets for their corresponding remaining inventory after the holiday. Their objective is to maximize profits and they study effect of pricing under two classes of purchases- one, where repeated purchases are allowed and two, where the repeated purchases are not allowed. They do not compute price elasticity of related products in a basket explicitly.

In [6], the authors study single period ordering and markdown pricing policies for short lifecycle products. They propose profit maximizing models to determine optimal order size, initial price, markdown time and price. Their work deals with perishables or highly seasonal products where the markdown horizon is small and fixed. In our work, we provide markdown pricing solutions for both seasonal and non-seasonal products where the demand estimation at each step is inventory contingent.

In [7], the authors demonstrate their markdown optimization solution for a fast-fashion retailer. In their approach, the coefficients for some regressors and the coefficient for price elasticity are estimated and updated periodically and demand is computed based on them. Our solution involves a non-linear markdown price optimization model and does not rely on in-season markdown period sales for accurate demand estimates.

In [8], the authors perceive the markdown optimization problem as one that involves joint decisions on inventory allocation and pricing subject to business rules. They incorporate a rolling-horizon approach wherein they perform demand estimation using the latest demand information. In their work, an inventory allocation logic is scheduled from a central warehouse, as soon as the markdown commences. In [9], we have shown the implementation of a novel age based prediction model to accurately forecast demand for fashion items for the next season. In our current scenario, inventory allocation does not happen from a central warehouse during markdown. Our solution works very well even in the absence of in-season markdown data because of the features considered in our models and the inventory estimation strategy employed in the demand prediction models.

In [10], the author has discussed the challenges in markdown pricing specifically in the apparel retail sector. An empirical demand model is employed based on weighted least squares estimation approach. Their empirical model, involving regression trees, is used for estimating dynamic demand. In [11], the authors propose a demand forecasting model based on regression trees. They combine this model with an algorithm to optimize pricing and built a decision support tool for an online retailer. In contrast, we employ models based on boosting algorithms to arrive at initial sales estimates for an offline retailer.

In [12], the authors analyze results based on deterministic and stochastic demand for short lifecycle items like perishables that have a finite selling horizon. In [13], the authors study markdown problem in apparel sector and highlight the challenges involved in pricing substitute products. These two approaches do not take into account the retailer's conventional markdown pricing methodology to arrive at initial price points. In our model, we identify the discounts offered by the retailer in the past and incorporate significant insights from the same.

In [14], the author discusses results pertaining to markdown optimization model implemented for different retailers. Their examples are based on a very small subset of data and do not contain steps to calculate the impact metrics clearly specified. In contrast, our markdown optimization model was tested for articles set for markdown across different time periods and can be implemented at any level of granularity.

III. DATA CHARACTERISTICS

In this section, we discuss the significant characteristics of retailer's data used for this work. The data of a large discount store chain is anonymized and used for this study. The data is structured and consists of performance, attribute and inventory related information for more than 300 brick-and-mortar stores for around 20 categories within a country. In this paper, we give details of our markdown optimization solution for the category – Clothing, which has around 7000 articles. The product hierarchy followed for each category is as follows: Category, Product group, Material group and Article. Ladies wear, Men's wear, Baby wear are some of the product groups inside Clothing category. Sweaters, Socks, Jumpsuits are examples of some material groups inside the product group Ladies wear. Articles within a material group share the same attributes though the attribute values may differ. Article is also referred as SKU (Stock Keeping Unit)/ item/ product/ merchandise in this work.

The markdown duration for articles belonging to Clothing category was 6 weeks. From the historical performance data, which included 4 years of markdown sales data, we could observe that every year around 15% of all the articles in Clothing went into markdown. For most of the material groups, on an average, discount offered was between 20- 30 percent. However, the minimum and maximum discount percentage observed in the Clothing category were 5% and 80% respectively. Few product groups like Ladies wear and Kidswear had larger percentage of articles going into markdown.

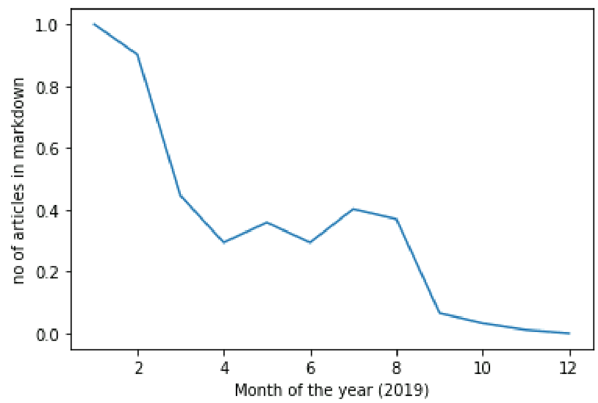

Fig. 1. Number of articles in markdown vs month.

The number of articles set for markdown are spread across the year. A graph showing the variation of number of articles that are subject to markdowns in a year for different months is illustrated in Figure 1. For demand prediction we make use of temporal features for addressing season related demand fluctuations.

The inventory at any given time and the number of stores at which this inventory was distributed, were among the most important features for the estimation of price elasticity. Most of the articles set for markdown in Clothing at any given time had only 2 price changes that could be captured from historical sales data. The regular price of articles in Clothing varied from 99 cents to 19.95 Euro, whereas the markdown price varied from 99 cents to 11.95 Euro.

Figure 2 shows a plot of normalized sales units vs. markdown week number for an article. We observe that as the markdown period progresses, the demand for the article falls. This type of pattern is observed typically for other articles, where the sales peaks within one or two weeks from the time markdown period starts. This decline in sales with time could be attributed to the decrease in inventory levels at the stores as time progresses. Owing to this observation, we have markdown week number and inventory for each week as features in the demand prediction module.

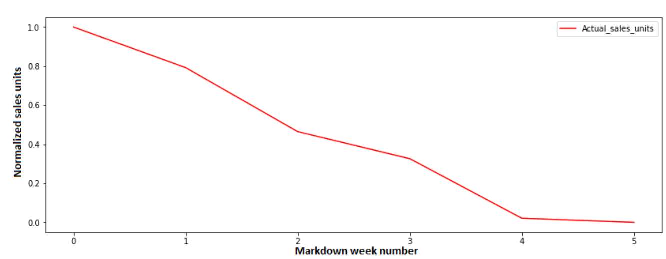

Fig. 2. Normalized sales units vs Markdown week number

In the Figure 3, the green plot represents the sales change rate and the yellow plot represents the markdown price for one SKU. We observe that the markdown price offered and the sales change rate for this SKU are highly corelated. The Pearson correlation coefficient between sales change rate and markdown price for all SKUs is also found to be high implying that markdown price is an important feature impacting sales units.

Fig. 3. Markdown price and sales change rate with respect to time.

IV. METHODOLOGY

Figure 4 shows the architecture of the markdown optimization solution proposed in this paper. This diagram exhibits the flow of data through different stages to generate the markdown price recommendation as output. The performance data, attribute data, and inventory data are taken as inputs and are pre-processed before going to the price estimation and demand estimation modules. All modules developed for the end to end functioning of this pricing solution are presented in details along with suitable justification of tools/ techniques employed, in the subsequent sections.

A. Markdown Price and sales estimation

In the proposed solution, we have used the retailer's historical markdown data to come up with initial markdown price estimates for each retail merchandise. These initial price estimates are used to forecast markdown period demand. The obtained initial price estimates and the corresponding demand forecasts are later optimized in the optimization module.

The initial estimates of the markdown price are obtained by utilizing the retailers' historical markdown data. These markdown price estimates serve as inputs to the demand prediction models. Estimating the initial prices based on the retailers' historical strategy also helps us in validating the performance of the sales prediction models. Since many articles go into markdown for the first time, we consider the similarity of the new retail merchandise, for which the markdown price needs to be prescribed, with the merchandise that have undergone markdown in the past. From the history data, we consider only those articles and their respective performance, attributes which have performed well, in terms of inventory clearance, in markdown duration. Material group of the item, regular retail price, temporal features, and sales rate are considered for calculating Euclidean distance based similarity. The hypothesis here is that the articles in a material group that are priced similarly and sell similarly during the regular sales period, will have similar markdown prices, provided the markdown period inventory and time of the year are also similar. Here, we have used k-nearest neighbours (KNN) regressor for finding similarity and estimating initial markdown prices. KNN works well here since the data dimensionality is small and the number of data samples is large. Finally, the calculated initial price estimates are used as one of the inputs for demand forecasting.

For estimating markdown period demand for the articles, we consider the historical markdown data of the retailer. We perform data pre-processing and feature building before training Machine Learning (ML) and Deep Learning (DL) based models for estimating demand.

The markdown demand of an item i, at time t is formulated as:

$$\begin{aligned} d_{i,t} = f(&MPE_{i,t}, RSR_i, MWN_i, IL_{i,t}, \\ &SI_{i,t}, attribute_i, RP_i, nholiday_t) \end{aligned} \tag{1}$$

where,

- $MPE_{i,t}$: is the markdown price estimate for the article i at time t. This is obtained from the initial markdown price estimation module.
- RSR_i: regular period sales rate of the article i.
- MWN_i: Markdown week number of the article i. This is the number of weeks since markdown started.
- $SI_{i,t}$: Seasonality indices of the article i at time t
- $attribute_i$: Attributes corresponding to article i
- $IL_{i,t}$: Inventory left for the article i at time t.
- RP_i: Regular period price for article i.
- $nholidays_t$: Represents the number of holidays from time step $t-1$ to time step t.

In this formulation, we have considered the impact of attribute values, regular period sales and price, along with temporal features. We also use markdown week number and remaining inventory as independent variables. Using markdown week number i.e., the no. of weeks since the markdown started, along with remaining inventory at each week allows the model to learn the impact of remaining inventory on demand as the markdown progresses. The results obtained from the sales prediction model are passed on to markdown price optimization module as initial price and sales pair for each article to be further optimized.

Here, we have selected eXtreme gradient boosting (Xgboost) and Artificial neural network (ANN) models for estimating demand. Both these models are capable of capturing non- linear dependencies and scale well for large datasets and are thus suitable for our use case.

B. Markdown Price elasticity of Demand

Understanding price elasticity of demand for different articles is key to pricing as it facilitates in anticipating the nature of demand for achieving different financial objectives. The elasticity estimator module considers the features that significantly impact the sales of an article as inputs. The output of the elasticity estimator module is the respective elasticity coefficients of articles assigned at various levels.

For estimating price elasticity coefficients, an important criterion we have considered is that the article must have undergone a minimum of three price changes in the past in order

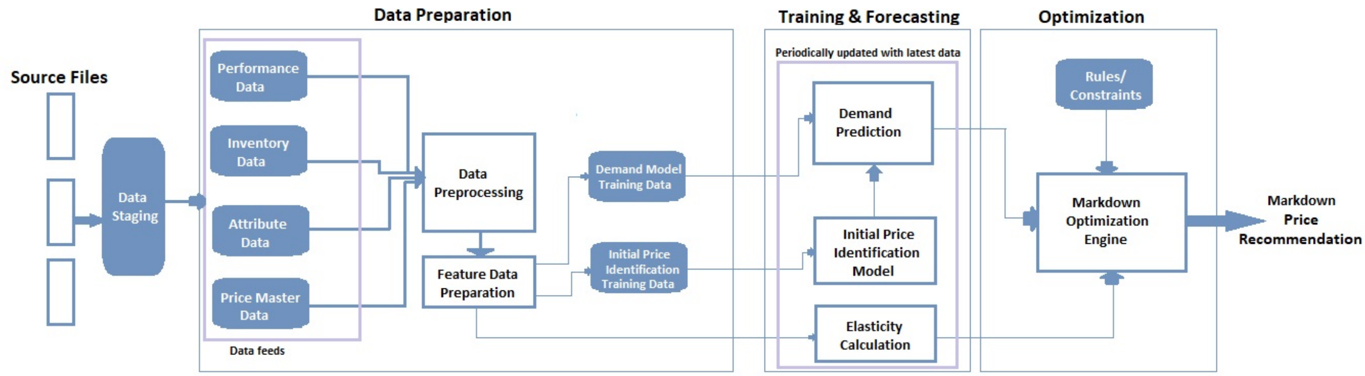

Fig. 4. Markdown system architecture.

to arrive at accurate elasticity estimates. For articles that satisfy this criteria, Elastic Net regression is employed to arrive at elasticity coefficients using the log-log model. We wanted the price elasticity coefficients to fall within a certain range so as to prevent the optimization engine from prescribing unfeasible prices for certain articles. As Elastic Net regression handles regularization along with removal of insignificant features, we employed it to arrive at price elasticity coefficients. An article is assigned with an elasticity coefficient only if it satisfies criterions based on coefficient of determination and probability value (where p- value should be less than or equal to 0.05 and r2 score should be greater than or equal to 0.75). In real-world data, however, many articles do not satisfy this criterion. For such articles, we identify similar articles to assign elasticity estimates. For calculating article similarity, we consider article brand, material group, available inventory, retail price, number of stores where the inventory is distributed, temporal features such as the week of year and weekly seasonality indices, and binary flags indicating promotion and markdown. Furthermore, articles which fail the statistical criteria based on p-values and r2 score, are subject to further grouping at the next higher merchandise hierarchy level. Grouping of articles at various levels allows us to obtain accurate elasticity estimates for articles that had undergone minimal or no price changes in the past.

The elasticity estimates computed from this module flows into the price optimization module that takes into account the prediction results along with elasticity estimates to obtain optimal markdown prices for articles in the given markdown duration.

C. Markdown Price Optimization

The final module of the solution is the markdown price optimization module. We use a nonlinear markdown price optimization system to recommend optimal markdown prices. The optimal markdown prices maximize the overall yield and satisfy all the business constraints.

The inputs for this module are markdown price estimates, predicted sales units and price elasticity values. The markdown price estimates and their corresponding predicted sales units are obtained from the prediction model. The price elasticity values are obtained from the elasticity estimator module. Yield is the weighted sum of sales units, revenue and margin, and the weights are business inputs that are based on the objectives of the retailer. A retailer can have varied objectives for introducing markdown, for example, it can be to clear most of the inventory or to maximize revenue with some percentage of inventory to be cleared. According to the objective, the weights are arrived at and are given as business inputs. The minimum and maximum markdown prices and the percentage inventory to be cleared are given as business constraints. The remaining inventory at the beginning of markdown and cost prices for the SKUs are the other inputs. The module outputs the optimal markdown prices for all the items.

The objective function of the optimization system is formulated as a non-linear weighted function of the normalized yield. The business constraints included given percentage of inventory to be cleared by end of markdown and that the inventory to remain in stores till end of season so that store footfalls do not decrease. Another important aspect is that the sku-store level combinations are large and therefore the computing resources and time will increase almost exponentially. Our price optimization system scales well with the given infrastructure to give results in a reasonable time. Our markdown price optimization solution generates price recommendations for 100 SKUs and 100 stores combinations in about 20 minutes.

The business constraints of the retailer are taken as minimum and maximum markdown prices and the percentage inventory to be cleared for each retail merchandise. In our model, percentage inventory to be cleared is taken as a hard constraint. This implies that the optimal price should satisfy the condition that percentage of inventory to be sold before end of season is equal to or close to the value of percentage inventory to be cleared. The nonlinear demand curve for an article is obtained using its initial markdown price, predicted sales units and price elasticity. We use a SLSQP solver, a nonlinear solver in SciPy, for solving the optimization problem to recommend optimal markdown prices as output.

Markdown Price Optimization Formulation:

$$\max \sum_{i=1}^{n} (\alpha_1 * (S_{md,i}/S_{curr,i}) + \alpha_2 * (R_{md,i}/R_{curr,i}) + \alpha_3 * (M_{md,i}/M_{curr,i})) \tag{2}$$

subject to:

$$\begin{aligned} &a. && P_{min,i} \leq P_{md,i} \leq P_{max,i}. \\ &b. && I_{cl,i} \leq S_{md,i} \leq I_{total,i}. \\ &c. && S_{md,i} = f(P_{md,i}). \\ &d. && R_{md,i} = S_{md,i} * P_{md,i}. \\ &e. && M_{md,i} = S_{md,i} * (P_{md,i} - C_{md,i}). \\ &f. && \alpha_1 + \alpha_2 + \alpha_3 = 1. \end{aligned} \tag{3}$$

Where,

- $$S_{md,i} = S_{curr,i} * (P_{md,i}/P_{curr,i})^{\beta} \tag{4}$$
- $i = \{1, 2, 3, ...n\}$ where n is the number of items for which markdown price is given.
- α_1, α_2, α_3 are scaling factors for sales units, revenue, margin respectively.
- $P_{min,i}, P_{max,i}$ are minimum and maximum price for SKU i during markdown period.
- $P_{md,i}$ is markdown price for SKU i (decision variable).
- $C_{md,i}$ is the total cost of SKU i during the markdown period.
- $S_{md,i}$ is the number of units of SKU i sold during the markdown at markdown price.
- $R_{md,i}$ is revenue during markdown period from SKU i.
- $M_{md,i}$ is margin achieved during markdown period for SKU i.
- $S_{curr,i}$ is the predicted number of markdown sales units for item i.
- $R_{curr,i}$ is the revenue for item i, computed based on the predicted sales units, during the markdown period.
- $M_{curr,i}$ is the margin for item i, computed based on the predicted sales units, during markdown period.
- $\beta_{md,i}$ is the markdown price elasticity for item i.
- $I_{cl,i}$ is amount of inventory to be cleared for SKU i during markdown period.
- $I_{total,i}$ is total inventory of the SKU i available during markdown period.

The objective function as shown in eq. 2 is to maximize the overall yield, where yield is a weighted function of sales units, revenue and margin. The business constraints and the sales, revenue, margin calculations are covered in eq. 3. The weights α_1, α_2 and α_3 satisfy (f) and are given as business inputs to the model. The constraint (a) refers to the minimum and maximum price constraints and (b) implies that the units sold during markdown period is greater than or equal to the percentage of inventory to be cleared and is less than or equal to total inventory available. The markdown sales units, revenue and margin calculations are given by (c, d, e). The markdown sales units in (c), is a function of predicted sales units, markdown price and price elasticity.

Eq. 4 specifies the demand as an iso-elastic function of price. Here $P_{curr,i}$ is the predicted initial markdown price, $S_{curr,i}$ is the predicted demand during markdown period and $P_{md,i}$ is the markdown price (decision variable).

Our optimization solution recommends optimal markdown prices that maximize the overall yield. Our solution helps solve the problem of dynamically coordinating inventory and pricing decisions for unsold DC inventory during the regular sales period itself.

V. EXPERIMENTS AND RESULTS

In this section, we discuss the performance of our markdown optimization solution. For our experiments, we have considered the data of a large discount chain retailer. The data is anonymized, structured and is composed of performance, attribute and store characteristics of around 300 brick and mortar stores. The performance data is aggregated at week-national level owing to the fact that there is only one DC per country, and the pricing is done at country level. For demonstrating the results, we have selected the category-Clothing. This is because the quantity and variety of products available in this department, during markdown, is large. The markdown period considered here for all the articles is 6 weeks and we perform our experiments with respect to a single optimal markdown price for the entire markdown duration. The inventory at the beginning of the markdown, for each article is available as an input from the retailer. Here, we have assumed that there is no replenishment during the markdown period. For training our price and demand estimation models, we consider the performance and attribute data of the retailer from start of year 2016 to end of year 2018. For testing, we have considered the data of retailers' 2019 markdown performance. We have considered around 1300 articles that undergo markdown from 2016 to 2019.

For initial price estimations, we have used a KNN regressor with $k = 3$. Figure 5 shows the results obtained for our initial price estimates. In this graph, the x-axis represents randomly selected test articles and y-axis represents the corresponding markdown prices. The plot also shows the markdown price estimated predicted by KNN. We observe that the initial price

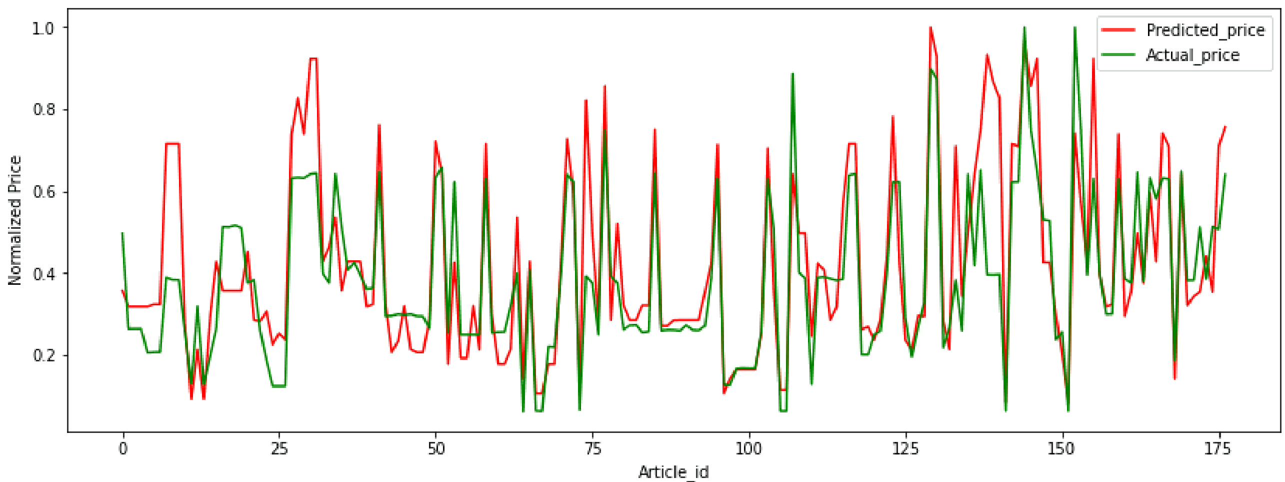

Fig. 5. Actual markdown prices and predicted markdown prices.

estimates obtained from the KNN based model is quite similar to the actual prices used by the retailer for markdown.

The demand prediction models are developed using ML and DL based algorithms. The architecture of the ANN model is shown in Figure 6. From Figure 6, we see that the inputs for the ANN based prediction models are markdown price estimates, attribute values, regular period sales and price, temporal features, markdown week number and remaining inventory . These inputs are scaled using a standard scaler before getting into the model . The ANN based demand prediction model is built using tensorflow deep learning framework [15]. We have implemented ANN with three dense layers along with batch-normalization [16] and dropout [17]. We have trained our ANN using the Adam optimizer with a learning rate of 0.001. For XGBoost model, we use the scikit-learn package [18], and its hyperparameters are tuned using Bayesian-Hyperparameter Optimization technique [19]. For the purpose of deployment and testing in real-world environment, we fix the random seed value.

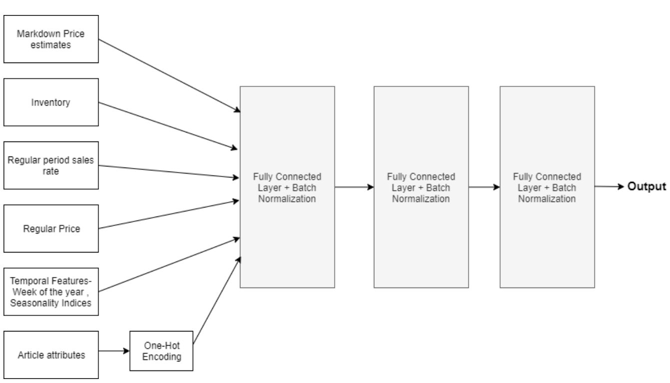

Fig. 6. Architecture of the ANN based prediction model.

To train our ML/DL based prediction models, we have used three different loss functions – 1) Mean Squared Error (MSE)

TABLE I
PREDICTION MODEL RESULTS.

	MAE (Normalized)	RMSE (Normalized)	1-MAPE
XG- MSE	0.06357	0.08880	0.798
XG - Huber	0.10087	0.12801	0.821
XG -Logcosh	0.0396	0.1258	0.819
DL -MSE	0.062973	0.096719	0.790
DL - Huber	0.064982	0.097758	0.796
DL -Logcosh	0.065935	0.1020760	0.785

2) Logcosh 3) Huber loss. We have evaluated the performance of our ML/DL based demand prediction models by using the following metrics- a) Mean Absolute Error(MAE), b) Root Mean Squared Error(RMSE), and c) Mean absolute percentage error(MAPE). In Table V, we have summarized the results obtained. From Table V we observe that for Xgboost we get the best estimates in terms of MAPE using huber loss function. For DL model, we get best results in terms of MAPE using MSE loss function. For any category, both XGBoost and ANN models are trained and the best model is selected based on test error.

From Figure 7 we observe the plots of actuals vs predicted sales units obtained from the demand prediction model. In this graph, the x-axis represents randomly selected test period articles that undergo markdown. The y-axis represents the normalized sales units for the 6 week markdown period. Since our initial price esimates are very close to the actual markdown prices given by the retailer, we are comparing the predictions of our demand forecast models with the actuals. From the plots we observe that there are a few instances of under and over predictions which can be attributed to some external factors influencing customer behaviours. For a majority of articles, the predicted sales units is very close to the actual sales units.

Next, we will discuss the results obtained from our markdown price optimization module. For Clothing category, approximately 15% of all the articles sold in a particular year

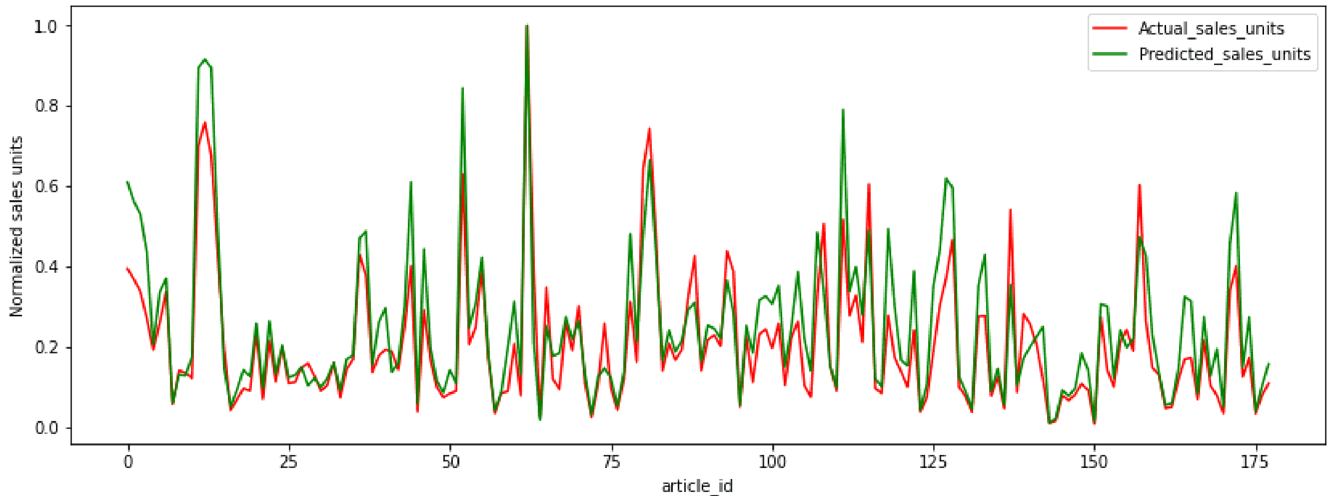

Fig. 7. Actual and predicted sales units.

went for markdown. For these articles, revenue from markdown sales contributed to 10% of their total revenue for the year. Our recommended prices were greater by around 20% than the actual markdown prices for most items. This led to around 6% decrease in sales units while satisfying their percentage of inventory cleared value, as compared to actuals. Our solution resulted in 10% increase in revenue as compared to the actuals.

The optimization was run with constraints on the percentage of inventory to be cleared by the end of markdown. When the constraint was set to 50% of inventory to be cleared, the results showed that there was considerable increase in price compared to the actual markdown given. This increased the markdown contribution to the year's revenue by 1% but appeared at the cost of higher percentage of inventory remaining at the end of the markdown. For successive increase in the percentage of inventory cleared, the recommended prices reduced while increasing the sales and hence clearing the inventory much more. The results are summarized in Table II.

Our markdown pricing is made flexible by using what-if scenarios. The what-if scenarios helps the retailer to recommend optimal prices when business rules change over a period of time. These changes include change in percentage of inventory to be cleared value and change in minimum and maximum values of price for optimization.

Change in the minimum and maximum prices:
The retailer can change the minimum and maximum price ranges and obtain optimum markdown prices for them. The markdown price range that gives higher revenue can be chosen. For instance, price range 1 and price range 2 correspond to [25%, 75%] and [15%, 30%] price ranges. For an item, this translates to price ranges [1.24, 3.71] and [3.46, 4.21]. The recommended prices for this item are 3.71 and 4.07 for price range 1 and price range 2 respectively. The price range 2 leads to a higher revenue and is thus chosen as shown in Table III.

Change in value of percentage of inventory cleared:
When the given percentage of inventory cleared constraint is not achievable the optimization runs with reduced percentage of inventory cleared and recommends optimal price. For example in Table IV, given 90% of inventory cleared is not achievable for an item hence, a lower achievable value of 75% of inventory cleared is recommended. Here, 9000, 8100 and 6750 corresponds to total, 90%, 75% percentage of inventory cleared respectively as shown in Table IV.

Markdown pricing for item groups:
We identified similar item groups within which the same markdown price needs to be applied. We ran optimization so that a single optimal price is obtained for all the similar items having same regular prices that maximizes their overall revenue. Specifically, the items, Ladies Basic cotton s-xxl Bordeaux and Ladies Basic. cotton s-xxl Black, belong to the same product group, i.e., Ladies t-shirt basic. We recommended the same markdown price of 2.54 Euro for them while running the optimization at an item group level.

VI. CONCLUSION

We developed a nonlinear markdown price optimization system for a large discount store chain retailer. Our system recommends optimal prices that maximizes the overall revenue as well as satisfies all their business constraints. Their business constraints included given percentage of inventory to be cleared by end of markdown, inventory to remain in stores till end of season so that store footfalls do not decrease and markdown pricing to be at item group level for inter-related items. One important aspect of this problem that can turn into a potential bottleneck is the increase in requirement of computing resources and running time as the SKU-store combinations increase. So, in order to address the volume of SKU store combinations, we leveraged open source distributed computing and parallel execution frameworks on cloud with elastic computing capability to meet workload changes. Our markdown pricing system consists of an initial price model

TABLE II
CHANGE IN % OF INVENTORY CLEARED VALUE VS. CHANGE IN PRICE, SALES AND REVENUE.

% of Inventory cleared	Avg price change with respect to markdown Price	Sales change with respect to Actuals	Revenue change with respect to Actuals	Change in markdown contribution to overall revenue of actuals
50	+22 %	-6.5%	+8.8%	+1%
75	+9.7%	-0.4%	+1.6%	+0.16%
90	-4.4%	+8.3%	-6.2%	-0.6%

TABLE III
WHAT IF SCENARIO 1- CHANGE IN MIN AND MAX CONSTRAINTS.

Price range	Min and max markdown price range %	Min and max price range	Recommended Price
price range 1	25- 75%	1.24- 3.71	3.71
price range 2	15- 30%	3.46- 4.21	4.07

TABLE IV
WHAT IF SCENARIO 2 – RECOMMEND ACHIEVABLE % OF INVENTORY CLEARED VALUE.

Inventory	90% of inventory cleared (given constraint)	75% of inventory cleared
9000	8100 (not achievable)	6750 (achievable, recommended)

based on their historical markdowns, ML/DL based sales forecast model, price elasticity computation model and a nonlinear markdown price optimization model. Our pricing solution is flexible due to the what-if scenarios that help the retailer to recommend optimal prices when business rules change over a period of time.

We have given in this paper, details of application of our markdown pricing system to the Clothing category for the retailer. Clothing category consists of around 7000 items, of which about 15% items goes for markdown every year. We did backtesting for multiple periods to compare our sales forecasting model outputs using initial price, with their actuals. Our recommended price was greater by around 20% than their actual markdown price for most items and this led to around 6% decrease in sales units while satisfying their inventory constraints. Our solution resulted in 10% increase in revenue for Clothing.

We also obtained markdown pricing at a SKU group level for the inter-related SKUs. Our markdown optimization system is generic and can be applied for different categories and at different levels - SKU-store, SKU store cluster or SKU DC level. Our nonlinear optimization model scales much better than traditional methods. Our markdown price optimization solution generates price recommendations for 100 SKUs and 100 stores combinations in about 20 minutes. This is when optimization is run on distributed cluster of 3 nodes with 16 cores and 64GB RAM.

As future work, we will be adding functionalities to our system to handle multiple markdowns. Additionally, our system would be designed to accommodate both short-term and long-term markdown planning components which could greatly facilitate the retailer in efficient inventory planning and allocation.

REFERENCES

[1] S. A. Smith and N. Agrawal, "Optimal markdown pricing and inventory allocation for retail chains with inventory dependent demand," *Manufacturing & Service Operations Management*, vol. 19, no. 2, pp. 290–304, 2017.

[2] F. Caro, F. Babio, and F. Peña, "Coordination of inventory distribution and price markdowns for clearance sales at zara," in *Operations in an Omnichannel World*. Springer, 2019, pp. 311–339.

[3] Y. Chen, P. Mehrotra, N. K. S. Samala, K. Ahmadi, V. Jivane, L. Pang, M. Shrivastav, N. Lyman, and S. Pleiman, "A multiobjective optimization for clearance in walmart brick-and-mortar stores," *INFORMS Journal on Applied Analytics*, vol. 51, no. 1, pp. 76–89, 2021.

[4] A. Vakhutinsky, K. Mihic, and S.-M. Wu, "A prescriptive analytics approach to markdown pricing for an e-commerce retailer," *Journal of Pattern Recognition Research*, vol. 14, no. 1, pp. 1–20, 2019.

[5] H. Wang, S. Li, and J. Luo, "Optimal markdown pricing for holiday basket with customer valuation," *International Journal of Production Research*, vol. 56, no. 18, pp. 5982–5996, 2018.

[6] M. Nagare and P. Dutta, "Single-period ordering and pricing policies with markdown, multivariate demand and customer price sensitivity," *Computers & Industrial Engineering*, vol. 125, pp. 451–466, 2018.

[7] F. Caro and J. Gallien, "Clearance pricing optimization for a fast-fashion retailer," *Operations research*, vol. 60, no. 6, pp. 1404–1422, 2012.

[8] M. Chen, Z.-L. Chen, G. Pundoor, S. Acharya, and J. Yi, "Markdown optimization at multiple stores," *IIE Transactions*, vol. 47, no. 1, pp. 84–108, 2015.

[9] R. K. Vashishtha, V. Burman, R. Kumar, S. Sethuraman, A. R. Sekar, and S. Ramanan, "Product age based demand forecast model for fashion retail," *arXiv preprint arXiv:2007.05278*, 2020.

[10] S. C. Yıldız and M. Hekimoğlu, "Markdown optimization in apparel retail sector," in *International Conference on Advances in National Brand and Private Label Marketing*. Springer, 2020, pp. 50–57.

[11] K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi, "Analytics for an online retailer: Demand forecasting and price optimization," *Manufacturing & Service Operations Management*, vol. 18, no. 1, pp. 69–88, 2016.

[12] G. Bitran and R. Caldentey, "An overview of pricing models for revenue management," *Manufacturing & Service Operations Management*, vol. 5, no. 3, pp. 203–229, 2003.

[13] Ö. Coşgun, U. Kula, and C. Kahraman, "Analysis of cross-price effects on markdown policies by using function approximation techniques," *Knowledge-Based Systems*, vol. 53, pp. 173–184, 2013.

[14] S. A. Smith, "Clearance pricing in retail chains," in *Retail Supply Chain Management*. Springer, 2008, pp. 271–291.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[16] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *arXiv preprint arXiv:1805.11604*, 2018.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[19] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*. PMLR, 2013, pp. 115–123.