Markdown Optimization Strategy

MGSC 670 – Revenue Management

Desautels Faculty of Management, McGill University

Prepared by

Addison Ji

Jennifer Liu

Chiara Lu

Rebecca Zhang

Zhiming Zhang

Prepared for

Dr. Maxime Cohen

May 21st, 2024

# Introduction

In this simulation, the task is to manage the sale of 2,000 units of a brand-new fashion apparel item. The primary goal is to devise and implement an optimal pricing strategy to maximize total revenue over a 15-week sales period. Initially, the apparel is priced at $60, and this price must be maintained for at least the first week. After this initial period, there is flexibility to apply one of several discount strategies, reducing the price to either $54 (a 10% discount), $48 (a 20% discount), or $36 (a 40% discount). It is important to note that once the price is reduced, it cannot be increased again. Additionally, due to long production lead times, no extra inventory can be procured during the sales season. The challenge is to balance timing and discount levels to sell all units and maximize revenue.

# Generate Data

The data can be generated by playing games multiple times and collecting them in the same format as the historical data provided. Instead of creating a scraper using “selenium”, the team took a different approach by using the source code in the HTML, which contains all the formulas. The game first generates an array of 15 random demands to represent real-world unpredictable demand for the 15 weeks. As the user clicks through the price levels, different multipliers are applied to generate the quantity sold based on the current price level. The inventory balance and revenue are calculated as the user progresses through the 15 weeks (see Appendix 1). The perfect foresight strategy calculates the maximum possible revenue using all possible combinations of prices based on the pre-generated random demand. When testing the strategies, executing functions to mimic user clicks on the screen and recording the differences in performance across many repetitive attempts is faster and more efficient to deploy.

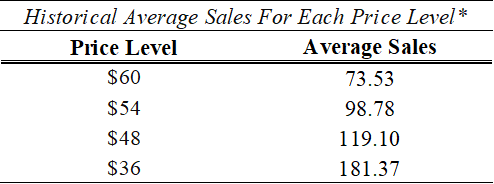
**Strategy 1: Heuristic Approach**

The team developed 3 sub-strategies using heuristic approach (code can be found in the attached zip: `Heuristic\_approach.ipynb`). The first strategy simulates a player's pricing strategy by maintaining a starting price of $60 for three weeks, then lowering the price to the next discounted level if the sum of the last three weeks' sales falls below a threshold. The second strategy also begins with an initial price and only lowers the price if sales for two consecutive weeks fall below a given threshold. The third strategy maintains the initial price $60 for a specified initial period, only moving to the next price level if the cumulative sales up to that week are less than the product of the threshold and the number of weeks passed. The simulation keeps track of player’s revenue, perfect foresight revenue, difference between the two, and the ratio of difference and perfect revenue (difference percentage), terminating when the inventory is exhausted, or the sales season concludes.

To find the optimal threshold for strategy 1, thresholds ranging from 270 to 399 in increments of 5 were tested. For strategy 2, thresholds from 100 to 150 were tested in the same increments. Strategy 3 required testing both thresholds (100 to 150) and initial periods (1 to 4 weeks). The optimal thresholds and initial periods were identified based on the median difference percentage across 1,000 simulations. For strategy 1, the optimal threshold was 300, achieving a median difference percentage of 5%. For strategy 2, the optimal threshold was 115, with a median difference percentage of 3.8%. For strategy 3, the optimal threshold was 105 with an initial period of 1 week, achieving a median difference percentage of 5.9%. Once these optimal values were identified, each strategy was run 10,000 times, and the results were saved to an Excel file. This approach ensures that the chosen thresholds and initial periods maximize the alignment of the player's revenue with the perfect foresight revenue, optimizing the pricing strategy.

# Strategy 2: Linear Optimization

In this section, we employ a linear optimization approach based on average historical sales for each price level. Specifically, the average sales for each price point ($60, $54, $48, $36) are used as benchmarks, as shown in Table 1. The discounts are applied only when real-time sales fall below these historical averages. This ensures that markdowns are data-driven and strategically timed to optimize revenue. The code file for this approach can be found in the attached zip: `Linear\_optimization.ipynb`.



*Table 1: Mean Sales based on Historical Demand Data*

The linear optimization approach was implemented 10,000 times to verify its performance. The results indicate that this markdown strategy, grounded in historical sales data, performs robustly, consistently achieving high revenue and maintaining a relatively low difference percentage compared to the perfect foresight strategy, as presented in the Result section.

# Strategy 3: Inventory Monitoring

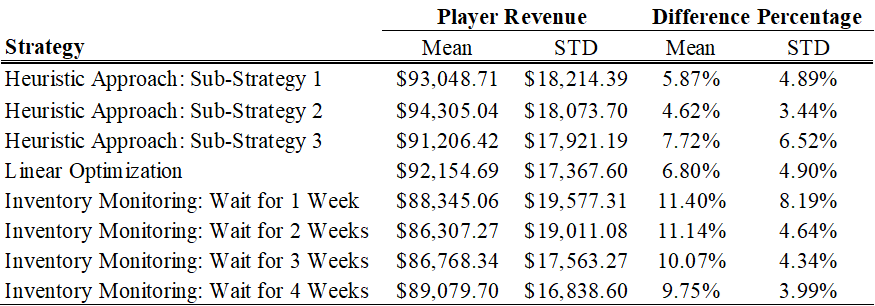
Monitoring inventory levels allows the player to apply promotions when the inventory decrease is slow. When approaching the end of the simulation weeks, it is more opportune to markdown rather than earning nothing. In the notebook `Inventory\_monitoring\_strategy.ipynb`, the player tracks weekly sales and calculates the slope of recent sales data. This slope indicates whether sales are increasing, decreasing, or stable. By analyzing the slope, the player can detect trends and make timely adjustments to pricing strategies.

The core strategy for adjusting prices is based on the observed slope of sales trends. If the slope indicates a declining trend, and a certain waiting period has passed since the last price change, the player considers lowering the price to boost sales. This waiting period, of either 1, 2, 3, and 4 weeks ensures that price adjustments are not made too frequently, simulating a more realistic market response. The player continues to monitor sales and inventory levels week by week, making further price adjustments as needed based on the observed trends. The threshold changes every time the code is run, as demand is randomized.

Additionally, the code includes an analysis of the range of observed slopes over 1,000 simulations. This analysis helps in understanding the variability in sales trends and identifying appropriate thresholds for triggering price changes. By experimenting with different slope thresholds and waiting periods, the player aims to find the optimal combination that maximizes revenue. The best inventory monitoring strategy is to wait for 4 weeks to observe the inventory decrease slope, this would yield a 9.75% difference percentage between actual and optimal revenue (See Appendix 3 for other wait weeks’ results).

# Results

The performance of the three strategies is compared based on the mean difference from the actual revenue and optimal revenue in percentage (See Appendix 3). The best performing strategy is the Heuristic Approach sub-strategy 2. This Heuristic Approach outperformed the others due to its lower mean and standard deviation in the differences between game results and the perfect foresight strategy, also yielding higher average revenue (See Table 2).



*Table 2: Performance Metrics Comparison Table for Different Markdown Strategies*

# Limitations and Conclusion

Noteworthy, the limitations of the simulation include the **static demand elasticity assumption**- the demand model assumes a static elasticity that doesn't account for varying consumer sensitivity to price changes over time. Consumer behavior can be influenced by factors like trends, marketing efforts, and competitor pricing. The simulation operates in a **closed market** and there is no oversight of competitors’ pricing situation. Another limitation of the simulation is the **$0 End-of-Season Stock Value**. In practice, there might be alternative ways to liquidate remaining stock, such as selling to discount outlets or repurposing items.

In retrospect, this simulation captures demand uncertainty and creates a proxy for the marginal increase in quantity demanded should a price change incur. With the Heuristic Approach strategy and the current simulation settings, the fashion retail store is going to achieve a median difference of 4%, an average difference of 4.6% and a standard deviation of 3.4% running 10,000 simulations, when setting the sales threshold at 115 to decide when to apply the next markdown. This nuanced strategy allows for more informed decision-making, optimizing pricing tactics to better align with consumer purchasing behavior and ultimately enhancing overall revenue performance.

# Appendices

*Appendix 1: html source code*

A screen shot of a computer code

Description automatically generated

*Appendix 2: Markdown Optimization Approach Summary*

Linear Optimization

Simulate Data

Heuristic Approach

Inventory Monitoring

Criteria:

-Mean difference in actual revenue and optimal revenue

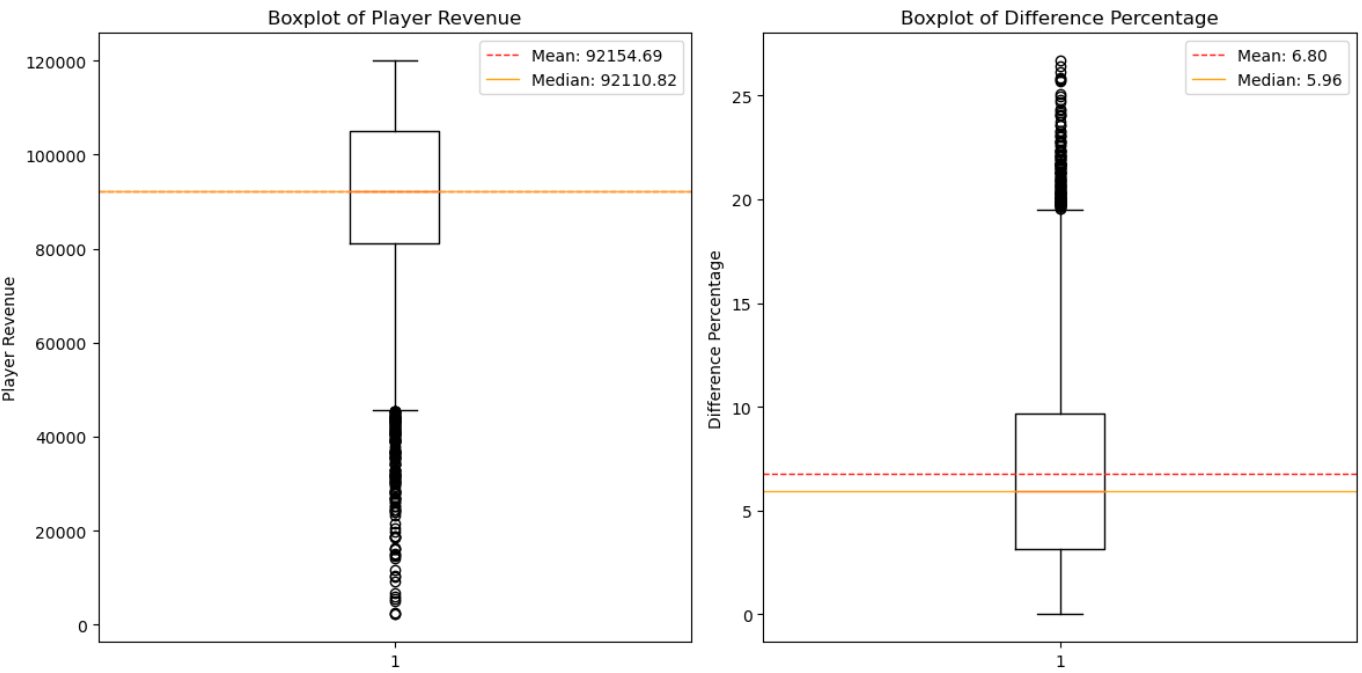
- Standard Deviation

*Appendix 3: Boxplots for All Markdown Strategies*

**Strategy 1 – Heuristic Approach**

|  |  |  |
| --- | --- | --- |
| *Sub-Strategy 1* | *Sub-Strategy 2* | *Sub-Strategy 3* |
|  |  |  |

**Strategy 2 – Linear Optimization**



**Strategy 3 – Inventory Monitoring**

