McGill University - Desautels Faculty of Management

RETL 408 - Omni-Channel Retailing

Demand Forecasting and Pricing Optimization for JD.com

Elena Dal Cin	260888938
Jonathan Steinberg	260924571

Table of Contents

Introduction - ***Industry & Project Overview***

Industry Overview

As brick-and-mortar retailers shuttered their doors at the onset of the Covid-19 pandemic, e-commerce became critical to the retail industry's survival. Chinese consumers entered strict lockdowns at the pandemic's beginning due to the government's zero-Covid policy. Online retail giants Alibaba and JD.com stepped in to bring consumers innovative, rapid solutions to meet their everyday needs. For instance, in Wuhan, both firms scaled their automated delivery technologies, and in the Hubei province, JD launched drone delivery to make deliveries in just twenty minutes rather than six hours (Lin, 2020).

As of 2022, the e-commerce market in China is nearly twice the size of the U.S. market, with a value of approximately 1.54 trillion USD compared to the U.S.'s 857 billion USD market value (Statista…Revenue, 2021). JD.com and Tmall dominate the Chinese e-commerce market, who together comprise over 90% of the market share (Sohu, 2022). Within the subset of the retail market, however, Taobao and Pinduoduo are the most frequented sites, and in Q2 2021, Taobao reached nearly half a billion monthly visitors (Research and Markets, 2022).

Founded in 2004, JD.com is one of China's largest retailers, with a total value of 697.81 billion Yuan as of December 2021 (100ec.cn, 2022). Historically, e-commerce sales from these sites have been largely restricted to the country, but with increasing globalization, international market penetration is growing (Research and Markets, 2022). To follow this trend, JD.com recently partnered with Shopify to facilitate foreign sellers listing their products on the marketplace (Shopify, 2022). JD.com's primary business model is a direct-to-consumer (D2C) online store with a secondary function to provide a marketplace to third-party sellers. As product authentication is critical for online marketplaces, JD.com allows only a limited number of third-party sellers to sell on its website, which allows the firm to maintain its strict quality standards (JD.com, 2017). A critical factor in JD's D2C success is the vertical integration of its supply chain. JD's profitability relies heavily on its vertical integration as it can realize critical economies of scale in selling traditionally low-margin goods. This assortment places it with the likes of Walmart and Amazon in consistently delivering high-quality, low-cost goods to consumers (Schiefelbein, 2018). The firm recognized this strength and, in 2007, launched the recently IPO'd JD Logistics, which serves as a logistics company providing other retailers with a vertically integrated logistics model (Biondi, 2021). The model is gaining popularity amongst luxury retailers, as ensuring high-quality service for high-end purchases is critical (Biondi, 2021). While the firm's success is evident, it continually seeks to improve its operations to boost profitability.

Project Overview

The growth of omnichannel and online shopping leaves retailers with dwindling margins, making demand forecasting and pricing optimization critical tools for growing profitability. While difficult to conduct accurately, demand forecasting is critical for retailers to make numerous decisions, from inventory to pricing. This paper will outline a demand forecast model that JD.com can use to optimize prices and maximize revenue. In evaluating these methods, the study highlights the sensitivity of pricing to demand forecasts while identifying notable characteristics of JD.com's SKUs and pricing model.

Data Description

To conduct the demand forecasting and pricing optimization, we used two datasets with information from March 1, 2018, to March 31, 2018, from JD.com. These two datasets, one outlining SKU data and the other order information, were merged to create a larger set to determine the best demand predictors for JD.com's inventory.

SKU Data

There were a total of 31,868 SKUs in the dataset, with variables sku_ID, type, brand_ID, attribute1, attribute2, activate_date, and deactivate_date. Sku_ID is a string providing an identifier for each product, type is an integer variable that specifies whether a first or third-party seller sells the SKU, and brand_ID is a string variable to identify the brand associated with the SKU. Attribute1 and attribute2 are integer variables describing each product's category's first and second most important attributes. Activate_date and deactivate_date are strings representing the dates at which the SKU is added to JD.com's inventory and removed from inventory. The full list of attributes is found in Appendix A.

Order Data

To forecast demand, we needed to understand the order distribution for SKUs' sales throughout the month. The JD_order_data table had string variables order_ID, user_ID, sku_ID, order_date, and order_time; integer variables quantity, type, promise, gift_item, dc_ori, and dc_des; and float variables original_unit_price, final_unit_price, direct_discount_per_unit, quantity_discount_per_unit, bundle_discount_per_unit, and coupon_discount_per_unit. In the table, there were a total of 549,989 orders sold to 457,298 customers (see Appendix B for the full list of features).

Merged Data

To develop a dataset for demand forecasting, we combined the SKU and order data into a single dataset entitled 'df_merged'. We then added four variables to this set: day_of_week, weekday, total_discount, and sales. day_of_week is a string variable describing the day of the week the order was placed. Weekday is a binary variable which assigns the value one if the order was placed on a business day and zero otherwise. Total_discount and sales are float variables calculated to determine the total realized discount of an SKU sold, and sales calculates the revenue for an order (Appendix C). This merged dataset contains 9,159 out of the total 31,868 SKUs, as only 28.74% of the SKUs were sold throughout the month of March 2018. The full list of this dataset's features can be found in Appendix D.

Methodology

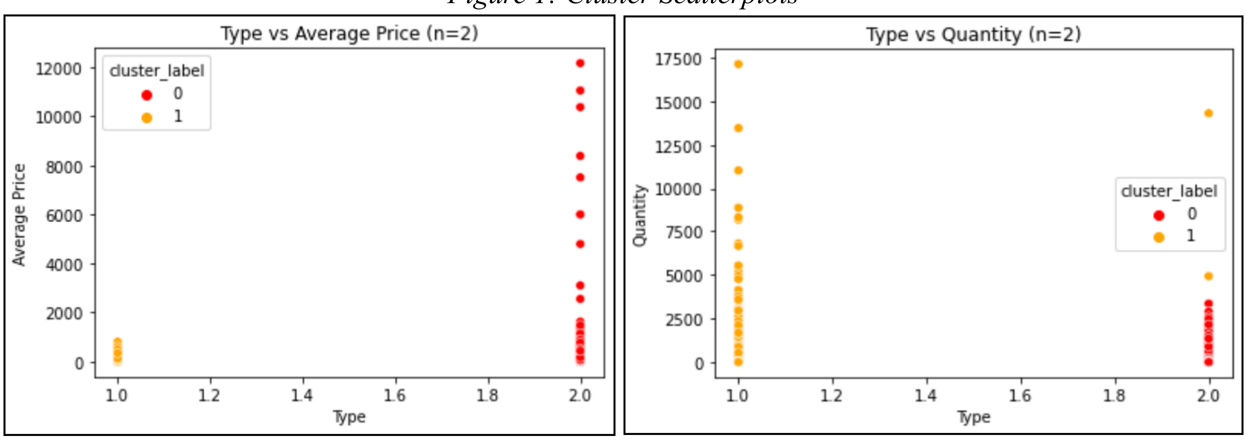
To develop the demand forecast and pricing model for JD, we first divided the df_merged dataset between a training and a test set. The training data ran from March 1 to 23 inclusive of 388,090 orders (70.56%), and the test data from March 24 to 31 inclusive of 161,899 orders (29.44%). This resulted in an approximately 70% training to 30% test data split, which is standard for a train-test split. Our analysis consisted of three primary processes (1) clustering, (2) demand forecasting, and (3) pricing optimization.

Clustering

JD carries a wide variety of products listed by the firm and third-party sellers, so we thought it critical to conduct a clustering analysis to group SKUs with a similar demand (Cohen, 2022). This analysis allowed us to identify actionable clusters for which to complete the demand forecast. Using the training set, we created variables quantity (total quantity sold over the first three weeks), average_price (average price of each SKU over the first three weeks), and type (first or third-party seller), and generated a new dataset entitled 'df_cluster'. Other features outlined in the data description were left out, as many of the SKUs had null values in these categories. Should these features have been left in, we might have removed SKUs with null values, but the validity of a clustering analysis relies on all SKUs being present.

To find the optimal number of clusters, we used the silhouette method, which aims to minimize the distance between observations in the same cluster (i.e., cohesion) while maximizing the distance between observations in different clusters (i.e., separation) (Khern-am-nuai, 2021). Iterations from two to ten clusters grant each observation a silhouette score and obtain an overall average for each number of clusters. Two through five clusters demonstrated largely similar average silhouette scores, which indicated that as the number of clusters increased, there was little difference in their optimality (Appendix E). The maximum score was for n=3 clusters; however, as there was a large drop-off between n=5 and n=6, we preliminarily selected five clusters. Upon further evaluation of the clusters, we realized that for all solutions with n greater than 2, at least one of the clusters had an insufficient amount of order data to train a robust demand prediction model (Appendix F). Consequently, we decided that two clusters were optimal. Figure 1 displays plots of the clusters with the three attributes, and Appendix G displays the number of SKUs and orders in each cluster.

Figure 1: Cluster Scatterplots



In Cluster 0, we observed much higher average prices with lower sales, while in Cluster 1, we observed lower average prices with higher sales. Notably, the clusters clearly distinguish between first and third-party sellers, as Cluster 0 is primarily third-party sellers' SKUs and Cluster 1 first-party seller SKUs. JD focuses primarily on commoditized goods, leaving apparel and longtail items to third-party sellers (Schiefelbein, 2018). As JD takes a different approach than competitors, it is logical that there is a clear distinction between these seller groups.

Demand Forecasting

After determining the two optimal clusters, we proceeded with feature engineering and selection for demand forecasting. For each SKU, by day, we created the features displayed in Appendix H for both training and test data. Our features encompass time components (order_day, day_of_week, weekday), SKU characteristics (type, attribute1, attribute2), and pricing and demand information (avg_final_price, sum_quantity_lag, and lagged discount variables). In addition, we created a feature (avg_final_price_other_lag) that measures the average price of all other SKUs in the same clusters since the demand for SKUs typically depends on the price of others that are similar. If the SKU's attribute1 or attribute2 were null, we used the respective averages of other SKUs in the same cluster. We chose this form of transformation for demand prediction instead of clustering analysis, since our aim was for pure predictive power rather than interpretability. Furthermore, we removed gift items from training and test data as we did not want to confuse the model with SKUs that had a positive demand with a price of zero. Instead, we created a gift_item_in_orders_lag feature, which refers to the number of orders that included gift items for each SKU. As the purpose of the model is prediction, we needed to lag several features to make them suitable for this purpose. For instance, since JD.com leverages dynamic pricing, it would have been inappropriate to predict demand using price information from the same day. However, the average final price was not lagged because we needed to use it to attempt to optimize JD's prices. The predictor for this model is sum_quantity.

We calculated feature importance to determine the most consequential variables in predicting sum_quantity (Appendices I & J). Data scientist Stacey Ronaghan defines feature importance as “the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. *The higher the value the more important the feature*” (Ronaghan, 2018). Sum_quantity_lag was the most important variable as it is highly correlated to sum_quantity compared to all other features included in the model. Gift_item_in_orders_lag was also highly correlated with sum_quantity and was therefore also included. Other variables highly correlated, in magnitude, to the predictor were type, avg_final_price_other_lag, and avg_final_price, in that order. We also included variables order_day, weekday, avg_total_discount_lag, attribute1, and attribute2 in the prediction. These features were added on the basis of logic to account for time components and SKU characteristics, which we hypothesized to be influential in demand prediction.

Using these inputs, we ran a Random Forest model on both clusters. Random Forest is a supervised machine learning algorithm in which a large number of decision trees are merged together for more accurate and reliable predictions (Khern-am-nuai, 2021). A random bootstrapped sample of features are considered when building each tree, which creates more diversity and thus higher predictive power (Khern-am-nuai, 2021). However, Random Forest is a black box algorithm, meaning it can only produce metrics of evaluation, and it is seldomly used for interpretation. Since we are seeking the most accurate demand forecast possible for JD.com, we are ok with sacrificing interpretability for better predictions.

Pricing Optimization

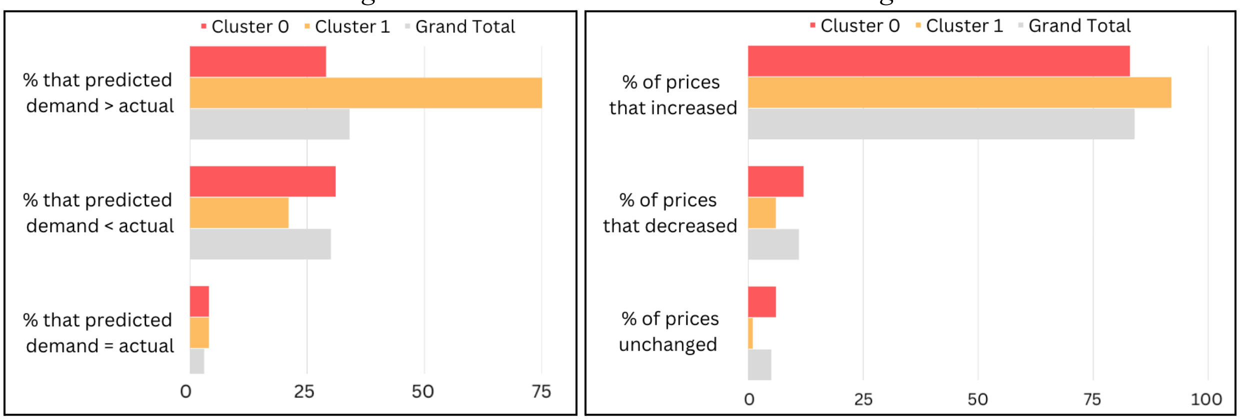
Using the random forest model’s results, we moved into the third part of our analysis - pricing optimization. This analysis aimed to find the optimal daily average prices for each SKU that maximize JD's revenue. In doing so, we assumed an inversely proportional relationship between price and demand, meaning that higher prices lead to lower demand levels and vice-versa. Given the demand forecast, we could determine the optimal price holding all other features of the model constant. We did this by using the *predict()* functionality of Random Forest, which allows us to predict demand with different levels of avg_final_price. This demand was subsequently multiplied by the price associated with it to determine the revenue at this price*demand combination. For each SKU on each date, we calculated a range of prices as follows: 0.25x, 0.5x, 0.75x, 1x, 1.25x, 1.5x, 1.75x, and 2x. We used this price range as finding the exact optimal price would be too time consuming. We thought this range sufficient to cover what JD.com would consider changing their prices to. After conducting this analysis, for each SKU, we selected the price level which optimized revenue. In doing so, we developed a re-optimized pricing model for JD that is based on the previously described demand forecast.

Results of Pricing Optimization

Upon use of our pricing model, we discovered a 150% increase in revenues for JD.com from approximately 13.6 million USD to 34.1 million USD. The behaviour between clusters differed, we suspect largely due to the distinction between first- and third-party sellers. Appendix K displays the results of our study.

Cluster 0, primarily third-party listings, had much less drastic adjustments to JD's current pricing and demand than Cluster 1 did. This model's out-of-sample (OOS) MAD 3.33 indicated that, on average, our demand prediction is +/- 3.33 units off the actual demand. Potentially due to this small MAD, Cluster 0 had a relatively even distribution of demand predictions that were greater, less than, or equal to observed demand. On the other hand, Cluster 1's demand model experienced a much larger OOS MAD of 46.02 MAD, which we suspect causes 75% of SKUs to have overestimated demand. This high MAD and demand overestimation is somewhat expected in Cluster 1 since its SKUs are those with much higher sales, as we discovered earlier from the cluster scatterplots. The result of this overestimation is highlighted in the percentage revenue increase. For Cluster 0, JD realized a 92% increase in revenue; in Cluster 1, this jump was 191%. Using our demand forecast and pricing model, over 80% of SKUs in both clusters experienced a price increase, but the quantity of the price jumps was much greater in Cluster 1.

Figure 2: Demand Estimation and Price Change



These results are significant for JD.com because, with retailers' low margins, it is critical to find increased revenues and profitability wherever possible. Our model suggests that JD chronically under-prices products and that this under-pricing is most significant for third-party sellers. This analysis indicated that JD has room to increase prices and boost revenue to over two times its current level, a critical competitive advantage in the retail industry. However, this drastic revenue jump warranted further evaluation, as it was significantly higher than hypothesized.

Limitations

Upon evaluating our demand forecast and pricing optimization model, we uncovered a handful of limitations, primarily associated with data availability, forecast accuracy, and results implementation.

Data Availability

Although we had access to extensive data on JD's March 2018 sales and inventory, many of the fields had missing values, which made it difficult to leverage them in our clustering analysis, as it requires all SKUs to be present. We used the most comprehensive SKU data to develop more accurate clusters for our decentralized demand forecasting and pricing optimization. Further, in our analysis, we could not access data informing of complementary or substitute products and of the product's category. All of these features are critical components of understanding a retailer's demand expectations and price levels, as the way in which products interact may greatly alter the outcome of our study. JD's dynamic pricing strategy also made the pricing optimization less interpretable than if they reset prices once daily, as we were forced to take an average of all the prices for a single SKU each day.

Demand Forecast Accuracy

The 150% total revenue increase prompted us to evaluate the results of our study, as it was beyond the increase we expected at the study's onset. The pricing optimization analysis relied heavily on the demand forecast, and any errors in this demand forecast were amplified in price changes. This, we believe, is what drove much of the revenue jump in the pricing optimization results.

Over 70% of observations in the test set overestimated demand, and of this subset, over 50% also had higher re-optimized prices than actual prices. For instance, on March 31, SKU #3f787efcbc realized a demand of 22 units with an average price of $197.68. However, our demand model predicted that the demand for this SKU would be 109 units for the same price. As the demand forecast consistently committed this overestimation, the newly suggested prices also became higher to account for the higher expected demand. The final result was, therefore, drastically increased revenues compared to JD's current revenue.

Measurability of Results

As we only had access to data from March 2018, we had no way to test the actual performance of the reoptimized prices. Ideally, JD would implement an A/B test, which makes price changes to a handful of SKUs while holding all other prices constant and tracking the performance of these products compared to others. We instead checked the results of our model by evaluating its performance on the test data set aside at the beginning of our analysis, but as this is historical data, we could not determine the model's true performance.

Conclusion

Should JD implement our demand forecast and pricing optimization technique, they may realize as much as a 150% increase in revenues. However, to carry this out, they should conduct A/B testing to confirm or disprove the validity of the price increases. Although the demand was consistently over-forecasted, the clustering analysis uncovered interesting differences in pricing and volume behaviours between the first and third-party sellers. Further analysis may include a deeper study of these different selling groups, but the currently proposed pricing optimization and demand forecast provide JD with a strong revenue boost in the increasingly competitive Chinese e-commerce industry.

Appendix

Appendix A: SKU Table Description

Field	Data type	Description	Sample value
sku_ID	string	Unique identifier of a product	b4822497a5
type	int	1P or 3P SKU	1
brand_ID	string	Brand unique identification code	c840ce7809
attribute1	int	First key attribute of the category	3
attribute2	int	Second key attribute of the category	60
activate_date	string	The date at which the SKU is first introduced	2018-03-01
deactivate_date	string	The date at which the SKU is terminated	2018-03-01

Appendix B: Order Table Description

Field	Data type	Description	Sample value
order_ID	string	Order unique identification code	3b76bfcd3b
user_ID	string	User unique identification code	3cde601074
sku_ID	string	SKU unique identification code	443fd601f0
order_date	string	Order date (format: yyyy-mm-dd)	2018-03-01
order_time	string	Specific time at which the order gets placed (format: yyyy-mm-dd HH:MM:SS)	2018-03-01 11:10:40.0
quantity	int	Number of units ordered	1
type	int	1P or 3P orders	1
promise	int	Expected delivery time (in days)	2
original_unit_price	float	Original list price	99.9
final_unit_price	float	Final purchase price	53.9
direct_discount_per_unit	float	Discount due to SKU direct discount	5.0
quantity_discount_per_unit	float	Discount due to purchase quantity	41.0

bundle_discount_per_unit	float	Discount due to "bundle promotion"	0.0
coupon_discount_per_unit	float	Discount due to customer coupon	0.0
gift_item	int	If the SKU is with gift promotion	0
dc_ori	int	Distribution center ID where the order is shipped from	29
dc_des	int	Destination address where the order is shipped to (represented by the closest distribution center ID)	29

Appendix C: Variable Total Discount and Sales Calculation

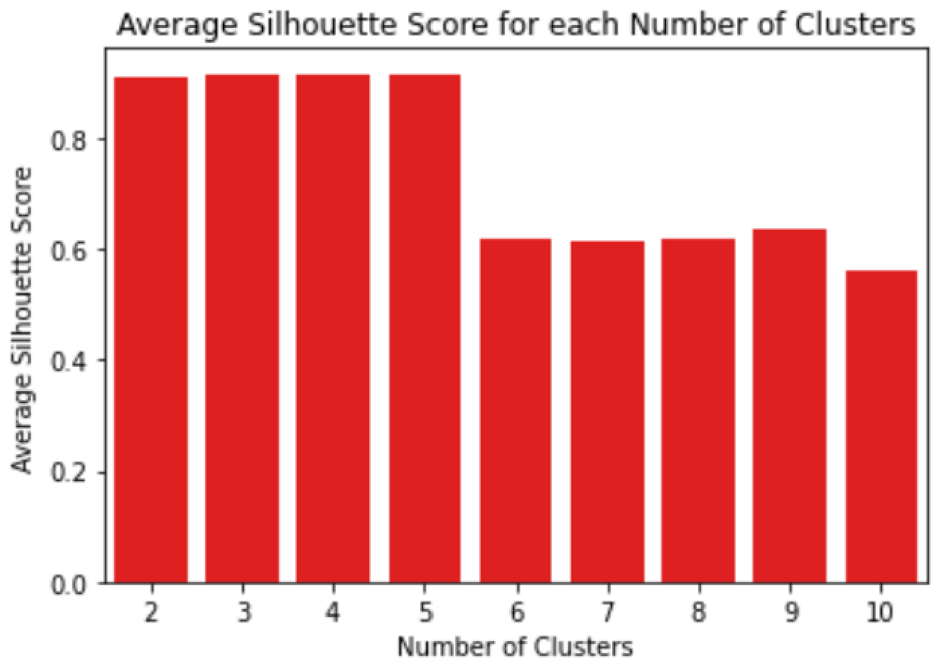
$$total\_discout = 1 - \frac{final\_unit\_price}{original\_unit\_price}$$

$$sales = 1 - final\_unit\_price * quantity$$

Appendix D: Variables added to df_merged

Field	Data type	Description	Sample value
order_day	int	Day of order	1
day_of_week	string	Name of order day	Monday
weekday	binary	Business day or not	0 - weekend 1 - weekday
total_discount	int	1-(final_unit_price/original_unit_price)	99.9

Appendix E: Silhouette Score Graph

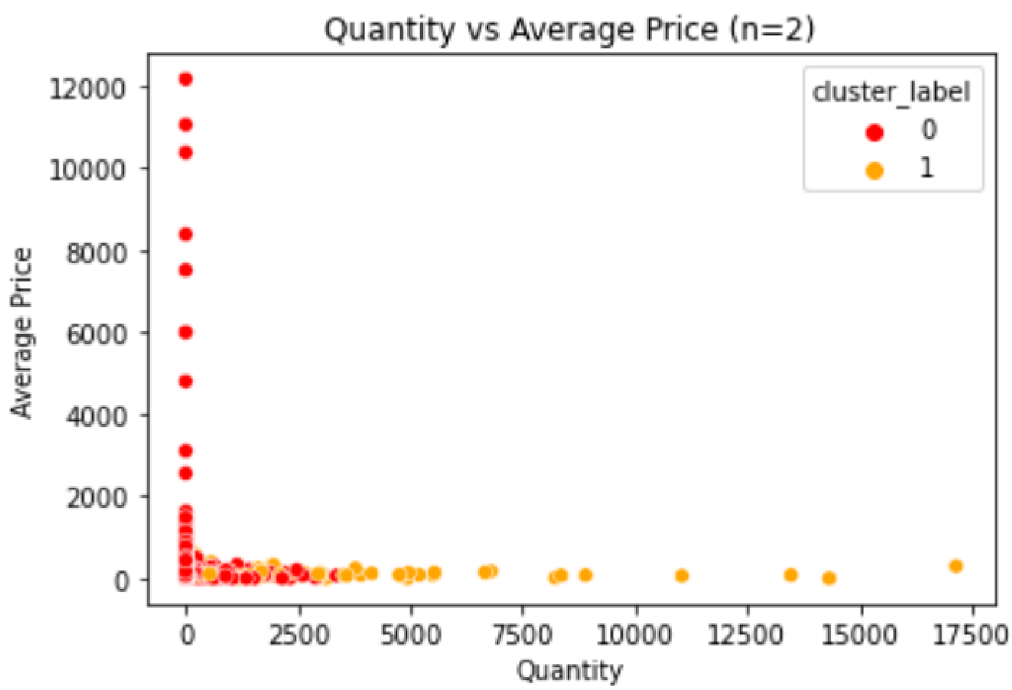


Appendix F: Clustering Results Table

		Cluster Group				
		0	1	2	3	4
n=2	Number of SKUs	7463	315	N/A	N/A	N/A
	Number of Orders	174882	213208	N/A	N/A	N/A
n=3	Number of SKUs	7455	315	8	N/A	N/A
	Number of Orders	174874	213208	8	N/A	N/A
n=4	Number of SKUs	7455	299	16	8	N/A
	Number of Orders	174874	115433	97775	8	N/A
n=5	Number of SKUs	7451	283	29	8	7
	Number of Orders	167837	76159	62982	81104	8

Appendix G: Clustering Outcomes and Additional Cluster Scatterplot

	Cluster 0	Cluster 1
Number of SKUs	7,463	315
Number of Orders	174,882	213,208



Appendix H: Feature Engineering and Selection Table

Feature	Description
sku_ID	Individual identifier by SKU
order_day	Day of order
day_of_week	Name of order day
weekday	Business day or not
type	1P or 3P
attribute1	If null, take the average of all other SKUs in the cluster.
attribute2	If null, take the average of all other SKUs in the cluster.

Feature	**Description**
sku_ID	Individual identifier by SKU
order_day	Day of order
avg_final_price	Average final price of SKU i during day j
avg_total_discount_lag	Average total discount of SKU i during day j-x
avg_direct_discount_lag	Average direct discount of SKU i during day j-x
avg_quantity_discount_lag	Average quantity discount of SKU i during day j-x
avg_bundle_discount_lag	Average bundle discount of SKU i during day j-x
avg_coupon_discount_lag	Average coupon discount of SKU i during day j-x
avg_final_price_other_lag	Average final price of all other SKUs not equal to i during day j-x
sum_quantity_lag	Sum of units of SKU i sold during day j-x
gift_item_in_orders_lag	Sum of orders that contained gift items and SKU i during day j-x
sum_quantity	**Sum of unit of SKU i sold during day j**

Note: Selected features highlighted in green, predictor highlighted in yellow
Note: Day j = current day, Day j-1 = previous day, Day j-x = most recent day SKU i was sold

Appendix I: Correlation Table

	sum_quantity
sum_quantity	1.000
avg_final_price	-0.0462
avg_total_discount_lag	0.0931
avg_final_price_other_lag	0.0855
attribute1	-0.0235
attribute2	0.0146
type	0.2928
day_of_week_Monday	-0.0014
day_of _week_Tuesday	-0.0065

day_of_week_Wednesday	0.0017
day_of_week_Thursday	0.0108
day_of_week_Friday	0.0120
sum_quantity_lag	0.7344
avg_coupon_discount_lag	0.0056
avg_bundle_discount_lag	0.0613
avg_quantity_discount_lag	0.0769
avg_direct_discount_lag	0.0165
weekday	0.0145
order_day	-0.0525
gift_item_in_orders_lag	0.1478

Note: Selected features highlighted in green, predictor highlighted in yellow

Appendix J: Feature Importance Table

Feature	**Importance**
sum_quantity_lag	0.6811
avg_quantity_discount_lag	0.0491
avg_final_price	0.0474
avg_coupon_discount_lag	0.0400
avg_final_price_other_lag	0.0398
avg_total_discount_lag	0.0375
avg_direct_discount_lag	0.0256
order_day	0.0248
attribute1	0.0114
attribute2	0.0111
avg_bundle_discount_lag	0.0092

day_of_week_Friday	0.0064
day_of_week_Thursday	0.0049
type	0.0037
weekday	0.0034
day_of_week_Monday	0.0019
day_of_week_Wednesday	0.0017
day_of_week_Tuesday	0.0013
gift_item_in_orders_lag	0.0064

Note: Selected features highlighted in green

Appendix K: Pricing Optimization Results

	Cluster 0	**Cluster 1**	**Grand Total**
Prediction Model MSE	395.29	14,933.64	N/A
Prediction Model MAD	3.33	46.02	N/A
% that predicted demand > actual	29%	75%	34%
% that predicted demand < actual	31%	21%	30%
% that predicted demand = actual	40%	4%	36%
Revenue with current prices	$5,558,444	$8,048,193	$13,606,637
Revenue with optimal prices	$10,653,258	$23,413,896	$34,067,154
Revenue % increase	92%	191%	150%
Proportion of prices that increased	83%	92%	84%
Proportion of prices that decreased	12%	6%	11%
Proportion of prices that stayed same	6%	1%	5%

Works Cited

100ec.cn. (2022, February 14). Company value of leading e-commerce platforms in China as of December 2021 (in billion yuan) [Graph]. In Statista. Retrieved December 02, 2022, from https://www.statista.com/statistics/1134335/china-most-valuable-ecommerce-companies/

Biondi, A. (2021, May 28). *What the JD Logistics IPO means for luxury.* Vogue Business. https://www.voguebusiness.com/consumers/jd-logistics-ipo-luxury-china

Cohen, M. et al. (2022). Common Demand Prediction Methods. Tangm C. S., *Demand Prediction in Retail* (pp. 29-67). Springer Nature Switzerland. https://doi.org/10.1007/978-3-030-85855-1

JD.com. (2017). *About Us*. https://corporate.jd.com/aboutUs

Khern-am-nuai, W. (2021, November 2021). *INSY 446 – Fall 2021 Data Mining for Business Analytics Session 10 – Unsupervised Learning Part 2* [PowerPoint slides]. Desautels Faculty of Management.

Lin, C. (2020, April 1). *In the Face of Lockdown China's E-Commerce Giants Deliver*. Harvard Business Review. https://hbr.org/2020/04/in-the-face-of-lockdown-chinas-e-commerce-giants-deliver

Research and Markets. (2022, March 01). *Alibaba and JD.com Remain the Largest B2C E-Commerce Market Players in China for 2021* [Press release]. https://www.globenewswire.com/en/news-release/2022/03/01/2394011/28124/en/Alibaba-and-JD-com-Remain-the-Largest-B2C-E-Commerce-Market-Players-in-China-for-2021.html

Ronaghan, S. (2018, May 11). *The Mathematics of Decision Trees, Random Forest, and Feature*

Importance in Scikit-learn and Spark. Towards Data Science. https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3#:~:text=Feature%20importance%20is%20calculated%20as,the%20more%20important%20the%20feature

Schiefelbein, L. (2018, July 25). *Is JD.com the Future of Chinese E-Commerce?*. Forbes. https://www.forbes.com/sites/lukeschiefelbein/2018/07/25/is-jd-com-the-future-of-chinese-e-commerce/?sh=53af8c50737d

Shopify. (2022, January 18). *Shopify & JD.com Unlock World's Largest Ecommerce Market for Merchants*. Shopify. https://news.shopify.com/shopify-and-jdcom-unlock-worlds-largest-e-commerce-market-for-merchants

Sohu. (2022, November 17). B2C online retail market platforms' transaction share in China in 3rd quarter of 2022 [Graph]. In Statista. Retrieved December 02, 2022, from https://www.statista.com/statistics/866847/china-b2c-online-retail-market-platforms-by-transaction-share/

Statista. (2021, November 2). Revenue of e-commerce worldwide in 2022, by country (in billion U.S. dollars) [Graph]. In Statista. Retrieved December 02, 2022, from https://www.statista.com/forecasts/1283912/global-revenue-of-the-e-commerce-market-country