Analytics of On-Shelf Availability (Term Project: December 12, 2024

Mason Phillips, Kansas State University
December 12, 2024

Abstract

This study analyzes when action needs to be taken to ensure stores have ample inventory to meet the demand for specific items. This study identifies 25,118 instances out of 40,591 data points where store managers should have ordered inventory but did not.

1. Introduction

Despite modern technology, retail stores still often face the issues of running out of stock. Even with supply chain fixes, an 8% stock out rate still plagues retailers worldwide[1]. 81% of retail associates believe their company needs better inventory tools [2]. These stocking issues have both a direct cost of lost sales, and the indirect cost of losing consumer trust.

When a retailer runs out of stock, customers often abandon that purchase. When an item a consumer wants is out of stock, about nine percent of consumers will continue shopping but will not buy a substitute item. Another 21% to 43% will go to another store to complete their purchase. In total only about 60% of customers will either return to the same store later, or substitute their item. Lost sales due to out of stock issues can account for a four percent sales loss at a typical retailer [1].

It might seem that simply overstocking could solve this issue, however retailers may not have the land to store this capital. Additionally, this raises the retailer’s days of working capital. A higher days of working capital increases a retailer's liquidity risk and deteriorates its free cash flow [3].

This study uses a dataset composed of the sales data for a number of items, for a number of stores between 2019 and 2021. The data analytics in this study aim to reduce the 8% out of stock rate that is plaguing retailers by analyzing the daily sales of an item combined with vendor’s lead time for that item, to predict any stocking issues for a retailers normal purchasing activity.

This study does not try to solve the irregular issues of out of stock, such as the toilet paper shortage that resulted from the Covid-19 pandemic, nor does it try to solve issues due to increased demand during the holiday season.

2.Methodology

The inventory dataset utilized in this research is composed of 40,591 records representing the daily inventory information of 96 unique items across 22 unique stores. Since each store does not have each product, there are 100 unique item and store combinations that comprise this data set. The inventory data was collected daily between January 1st, 2019 and May 3rd, 2021.

The inventory data contains 13 fields. Each record contains a date, store id, product/item number (SKU), starting inventory, ending inventory, items sold, units under promotion, and a few more columns representing the number of items that have been ordered, but not yet been delivered to the store.

date	store_id	sku	product_category	total_sales_units	on_hand_inventory_units	replenishmer
2019-01-02	63	57	Category 04	0	8	
2019-01-04	63	57	Category 04	0	7	
2019-01-05	63	57	Category 04	3	5	
2019-01-06	63	57	Category 04	2	11	
2019-01-09	63	57	Category 04	3	7	
2019-01-10	63	57	Category 04	2	5	
2019-01-13	63	57	Category 04	0	13	
2019-01-15	63	57	Category 04	0	13	
2019-01-16	63	57	Category 04	0	14	
2019-01-20	63	57	Category 04	2	6	
2019-01-24	63	57	Category 04	0	11	
2019-01-25	63	57	Category 04	3	9	
2019-01-30	63	57	Category 04	4	8	
2019-01-31	63	57	Category 04	0	9	
2019-02-01	63	57	Category 04	2	7	
2019-02-02	63	57	Category 04	0	8	
2019-02-03	63	57	Category 04	3	5	
2019-02-08	63	57	Category 04	0	19	
2019-02-09	63	57	Category 04	5	15	
2019-02-11	63	57	Category 04	0	13	

Figure 1. Features the first 20 records in the inventory dataset.

The inventory data is paired with a dataset containing the lead time for vendors to get items delivered to a store. This dataset contains 100 records (one for each unique store and item combination.) Each record contains 8 fields, consisting of information identifying the store, item SKU, and vendor the time that the item spends on order, in transit, and in the distribution center.

key	vendor_id	sub_vendor_id	store_id	sku	lead_time_in_dc	lead_time_in_transit	lead_time_on_order
1	1	1001	1763	1	4	3	7
2	1	1001	1763	2	2	1	7
3	1	1002	1843	2	2	1	7
4	1	1001	1763	3	4	3	7
5	2	2016	486	6	2	1	8
6	2	2073	1587	7	3	2	9
7	2	2087	1556	8	4	3	7
8	8	8010	1283	39	3	2	6
9	10	10001	1763	46	2	1	5
10	11	11001	334	52	4	3	8
11	12	12006	63	57	4	3	10
12	13	13091	98	64	3	2	9
13	13	130153	171	64	3	2	7
14	13	130160	178	64	2	1	6
15	13	130185	206	64	4	3	8
16	13	130188	209	64	2	1	5
17	13	130208	232	64	2	1	5
18	13	130223	249	64	4	3	9
19	13	130244	274	64	4	3	10
20	13	130297	337	64	3	2	8

Figure 2. Features the first 20 records in the vendor lead time dataset.

The inventory dataset contains some gaps for the days when a specific store has no sales and no changes in inventory for a specific item. A complete dataset must be extrapolated from the

inventory_pipeline	units_in_transit	units_in_dc	units_on_order	units_under_promotion	shelf_capacity
16	0	0	8	0	32
14	0	8	0	0	32
12	8	0	0	0	32
11	0	0	0	0	32
14	0	0	8	0	32
13	0	0	8	0	32
21	0	8	0	0	32
13	0	0	0	0	32
13	0	0	0	0	32
14	0	0	8	0	32
10	0	0	0	0	32
8	0	0	0	0	32
16	0	0	8	0	32
16	0	0	8	0	32
15	0	0	8	0	32
15	0	0	8	0	32
13	8	0	0	0	32
19	0	0	0	0	32
15	0	0	0	0	32
13	0	0	0	0	32

given dataset to complete the analysis of the data.

To extrapolate a completed dataset, a PySpark SQL DataFrame is created containing one column, “date” with a row for each day between the first and last date in the dataset (for this dataset 1/1/19 to 5/3/21). This dataframe is left outer joined on with a dataframe of the unique store and SKU combinations, resulting in a dataset with a record for each date for each store and SKU. This join grows the number of records in the dataset from 40,591 to 85,400. The records created as a result of this join are either forward-filled or set to 0 as appropriate. With a complete dataset, flags can now be raised for promotional and replenishment events. These items are unpredictable and thus should not be treated as regular events. Two new columns are added to the dataset, “promotion_flag” and “replenishment_flag” which is set to 0 unless there are any items under promotion or replenished, in which case, the respective field is set to 1.

Flags are then raised for any times where a change in inventory is untracked in sales, as the item is lost, stolen, damaged, etc. These inventory changes are referred to as phantom inventory. A new column, “phantom_inventory_ind” is added to the dataset, similar to before this indicator/flag is set to 0, unless sales subtracted from units at the beginning of the day do not equal the number of units at the end of the day, in which case the indicator is set to one. A column called “phantom_inventory” is also added to the dataset containing the difference between the expected items at the end of the day and the actual items.

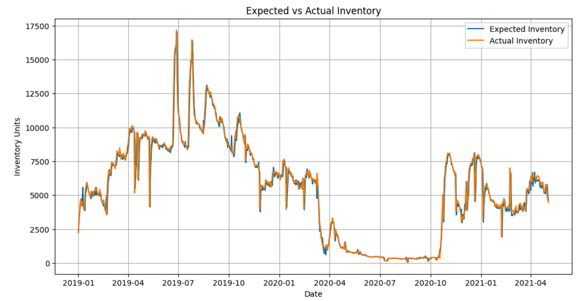

Figure 3. Represents the total phantom inventory on each day in the dataset.

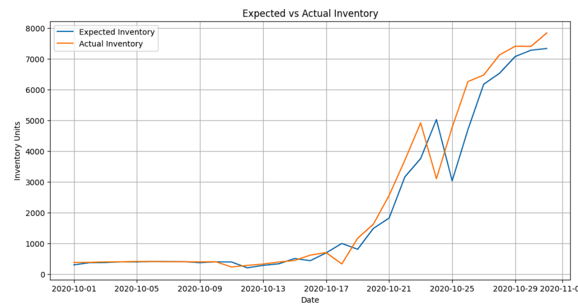

Figure 4. Represents the total phantom inventory across all items during October 2020. Anywhere the blue line is above the orange line, items are lost as phantom inventory.

On hand inventory is estimated by using the previous day’s inventory + replenishment units - (units sold + phantom inventory units). Average daily sales over the last 90 days is calculated using the sql.functions.expr() from PySpark with the parameter “'AVG(total_sales_units) OVER(PARTITION BY store_id, sku ORDER BY date ROWS BETWEEN 90 PRECEDING AND CURRENT ROW)'”.

The shortest lead time for each store and SKU combination is calculated using the Spark SQL function “Least” over each of the different lead times in the vendor dataset, and stored in a new “min_lead_time” column.

A left outer join is performed on the inventory dataset and vendor dataset, such that each record now has the minimum lead time to replenish that SKU at that store. The minimum lead time is then multiplied by the average daily sales over the past 90 days to calculate the minimum stock needed to avoid an out-of-stock problem (safety stock).

Two more columns are added to the combined dataset. One is raised to warn when the inventory on-hand is less than the safety stock. The other is raised to warn when there are not enough units in the pipeline to prevent the out of stock issue. A third column is added to raise a flag whenever there is insufficient lead time in the pipeline to meet the expected demand. All three columns are then combined to raise an alert when inventory on hand is less than safety stock, and there are either not enough items in the pipeline, or not enough lead time in the pipeline to prevent the out-of-stock issue. 25,118 unique stores, item, and date combinations are found to be at risk of an out-of-stock issue.

More flags are raised for when an item does not have any sales. A flag is raised for this reason when .05 is greater than the probability of zero sales is raised to the power of the consecutive days the item is out of stock.

All the flags are consolidated into a data frame to alert when an employee needs to take action to resolve a stocking related issue.

Using expression smoothing with α=.8 to generate a forecast.

4. Results

Action needs to be taken on 34,836 of the records in this dataset. 25,118 of these flags are from anticipated events running out of stock, and 9702 are from times when there are zero sales for a suspicious amount of time.

Within the completed datasets there were 16912 unique combinations of days, stores, and SKUs where there was 0 stock of the item. (Approximately 20% of the time). Of these, 14,700 of the days were preventable using this model. Roughly 87% of the out-of-stock issues within this dataset could be solved using this model.

5. Future Work

In the future, this model could be improved on by adding information related to substitutable and complementary products.

To reduce the risk of overstocking, if an out-of-stock alert is raised on an item that only is purchased with a complimentary item, and the complimentary item is sold out, it may be worthwhile to match the timing of the alert with another alert. For example, if a hardware store runs out of paint, the store should time to restock paint rollers at the same time as the store restocks paint, less the store will overstock on paint rollers. This would especially be an issue if paint is backordered, as the unsold paint rollers would vastly increase the store's days of working capital.

6. Conclusion

This model can predict out-of-stocks problems with x% accuracy. Considering a two percent improvement in on-shelf availability is worth one percent in increased sales [4], utilizing this model can vastly improve many retailers total revenue.

7. Citations and References

References

Corsten, D. & Gruen, T. "Stock-Outs Cause Walkouts." Harvard Business Review https://hbr.org/2004/05/stock-outs-cause-walkouts

Tapscott, K. "Out-Of-Stocks: A Hidden Crisis in Retail. Zebra (2024). https://www.zebra.com/us/en/blog/posts/2024/out-of-stocks-hidden-crisis-retail.html

"Days Working Capital (DWC)" Wall Street Prep (2024). https://www.wallstreetprep.com/knowledge/days-working-capital-dwc/

Williams, R., Seybert, M., Saker, R. & Smith, B. "Improving On-Shelf Availability for Items With AI Out of Stock Modeling." DataBricks. https://www.databricks.com/blog/2021/08/24/improving-on-shelf-availability-for-items-with-ai-out-of-stock-modeling.html