International Journal of Operations Management and Services.
ISSN: 2277–3193, Volume 11, Number 1 (2021), pp. 1-10

http://www.ripublication.com

Inventory Management using Demand Sales Forecasting

Tejas Belgamwar
Kalyan, MH (421301)

Ninad Sayare
Nagpur, MH (440034)

Abstract

The purpose of the exercise was to find fitting solution to the problems faced in supply chain – How to decide the Inventory? How to maintain lean operations without hampering service levels? In our experience in the industry, we have found that in the lack of a better solution, companies have adopted simplistic methods, incorrectly assessing their need and in turn taking a strategic backseat in the competitive space. Ironically the answer to this archaic problem has always been in their hands – Data. Our module leverages sophisticated statistical models like Holts winter, AR, ARMA, ARIMA & SARIMA available in Python Libraries. The module is made so that an average excel user will be able to leverage available data and decide optimum inventory.

Keywords: Inventory Management system, maximum profit, lean inventory, Time series forecasting, Trend, Seasonality

1. INTRODUCTION

Distribution centre in any part of the supply chain, be it a supplier, wholesaler or a retailer has to keep sufficient deliverable products in his inventory for a faster turnaround time. Inventories also represent the second largest asset category for manufacturing companies, next only to plant and equipment. The proportion of inventories to total asset generally varies between 15 to 30 percent. Given substantial investment in inventories, the importance of inventory management cannot be overemphasized [1]. Inventory if in excess than demand, it makes the operation less profitable as much of the cash is locked in the Inventory holding and if in shortage, company's serviceability suffer – affecting the company severely in the long term. Hence making the decision of Inventory very critical to company's functioning.

Decision regarding the inventory can be made after analysing the Historical data.

The Module in this exercise leverages various statistical models which identifies the inherent trend and seasonality in the time series data and helps us predict the future (Baseline forecast). After predicting the future sales, we might not get accurate results because of Events (Not present in Historical data) – E.g. Effect of Pandemic or Launch of a new product. Hence we configure events in the module to quantify their impacts, thus bringing our baseline forecast more closer to the truth.

2. UNDERSTANDING THE MODULE

2.1 What is a Bucket ?

For the purpose of the project let's assume the 'No. of days for which the user wants to hold the inventory for' is called Bucket. Assuming every organization wants to be as lean as possible, they would want their 'Bucket' to be as low as possible. The number of days a user chooses to hold inventory depends on the lead time between the ordering time and the delivery time [2].Lead time is however subject to change as per different organizations. Therefore, we have kept this particular metric as a input for the user.

2.2 Model Configuration

User would be required to define below metrics in the excel module which are crucial in its functioning (Refer Figure 1)

- **Raw data** – "Raw Data" sheet to be filled with sales data. The format has been kept in alignment with General Enterprise resource planning (ERP) outputs.
- **Last Historical data** – User can define which date should be considered as last historical date. The model would be then anchored around that date.
- **Horizon** – Horizon in Months to be considered for forecasting
- **Bucket** – Bucket to be defined for each SKU
- **Approach to be followed** – User to have the option to specify a forecasting approach he wants to be followed.

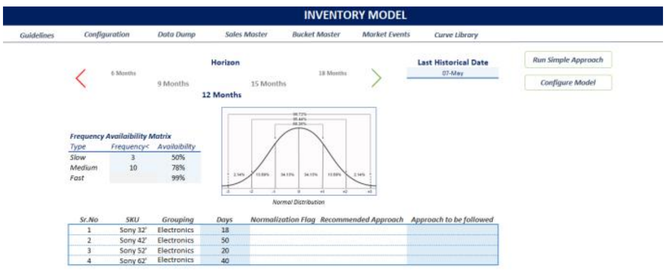

Figure 1: Configuration Dashboard for Model

3. AVAILABLE APPROACHES

Module gives the user choice between two broader approach to be followed for calculating the Inventory.

- Simple Approach
- Statistical Modelling Approach

3.1 The Simple Approach

Simple approach is a approach followed by some in the industry. Albeit the method is fairly simple to understand and use, it fails to capture inherent trend and seasonality.

3.1.1 Assumptions

- Data is stationary i.e. there are no discernible trends or seasonality in the data
- Data can be fitted on a normal distribution curve. An example of normal distribution can be referred in Figure 2.

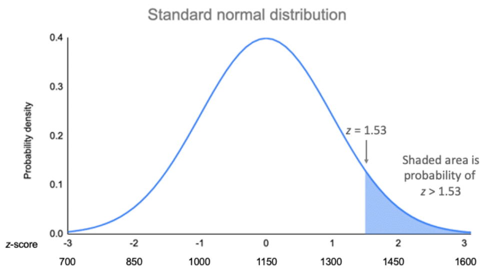

Figure 2: Normal Distribution

3.1.2 User Inputs

- Product stratification guidelines for Slow, Medium and fast moving products based on its Sale frequency. (refer Figure 3)
- % Availability to be maintained for each product groups. (refer Figure 3)

Frequency X Availaibility Matrix

Type	*Frequency<*	*Availaibility*
Slow	3	50%
Medium	10	78%
Fast		99%

Figure 3: Frequency X Availability Matrix

3.1.3 Methodology

Consecutive bucket sales (CBS)

User has already specified that he wants to hold inventory for ‘X’ no. of days i.e. Bucket. Based on this input we create a database of Consecutive bucket sales.

For e.g. – If Bucket specified is 7 Days then B1 is Sales from Day 1 to Day 7. B2 would be Day 2 to Day 8 and so on till we reach the “Last Historical date”. Database of all Bucket sales would be called “Bucket Master”.

Inventory Quantity

Final Inventory amount then is decided as per the “Frequency X Availability matrix”. Frequency of the SKU would help us categorize the product in Slow, Medium and Fast moving products. Corresponding availability for the product group is then used to calculate the “X” component in the below equation.

$$\text{Final Inventory Amount} = \text{Avg.} + X * SD$$

Avg. : Average of all the Consecutive Bucket Sales (CBS)

SD : Standard Deviation of Consecutive Bucket Sales (CBS)

3.2 The Statistical Approach

In the statistical approach, Historical timestamped data is analysed and divided into trend, seasonality and noise. Each set of data then converted to an equation which satisfies them – Process called as Fitting. This fitted equation is then used to predict the future. Fortunately, using Python libraries we can directly use the statistical models without any pre-requisite mathematics expertise. The quantitative methods adopted play a crucial role in forecasting data in various industries like pharmaceuticals[3], where scarcity of such supplies could over burden health infrastructure. Various studies discussed that models like AR and ARMA models can reduce the bullwhip effect (which means that a downstream demand fluctuation will lead to a larger fluctuation in the variance of upstream ordering) [4].

Below are the statistical models we have used for the purpose of this exercise.1

3.2.1 Holts-Winters

The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level ℓ_t, one for the trend b_t, and one for seasonal component s_t, with corresponding smoothing parameters α, β and γ. We use m to denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data m=4, and for monthly data m=12

There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series. With the additive method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation the series is seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component will add up to approximately zero. With the multiplicative method, the seasonal component is expressed in relative terms (percentages), and the series is seasonally adjusted by dividing through by the seasonal component. Within each year, the seasonal component will sum up to approximately m [7].

3.2.2 Auto Regressive (AR)

In a multiple regression model, we forecast the variable of interest using a linear combination of predictors. In an autoregression model, we forecast the variable of interest using a linear combination of *past values of the variable*. The term *auto*regression indicates that it is a regression of the variable against itself.

Thus, an autoregressive model of order p can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

where ε_t is white noise. This is like a multiple regression but with *lagged values* of y_t as predictors. We refer to this as an **AR(**p**) model**, an autoregressive model of order p [7].

3.2.3 Moving Average Model (MA)

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

where ε_t is white noise. We refer to this as an **MA(**q**) model**, a moving average model of order q. Of course, we do not *observe* the values of ε_t, so it is not really a regression in the usual sense [7].

3.2.4 Non-Seasonal ARIMA Models

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for Autoregressive

Integrated Moving Average (in this context, "integration" is the reverse of differencing). The full model can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

where y′t is the differenced series (it may have been differenced more than once). The "predictors" on the right hand side include both lagged values of yt and lagged errors [7]. We call this an **ARIMA (p,d,q) model**, where:

P: order of the autoregressive part

D: degree of first differencing involved

Q: order of the moving average part

3.2.5 Seasonal ARIMA Model

So far, we have restricted our attention to non-seasonal data and non-seasonal ARIMA models. However, ARIMA models are also capable of modelling a wide range of seasonal data. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

where m= number of observations per year. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an ARIMA(1,1,1) model (without a constant) is for quarterly data (m=4), and can be written as

$$(1 - \phi_1 B)\ (1 - \Phi_1 B^4)(1 - B)(1 - B^4) y_t = (1 + \theta_1 B)\ (1 + \Theta_1 B^4) \varepsilon_t.$$

The additional seasonal terms are simply multiplied by the non-seasonal terms [7].

4. EVALUATION

One of the aforementioned statistical methods would be better than the other based of the inherent properties of the data. We can either look at the data and make an appropriate decision or we can run all the forecasts and compare errors (RMSD) to make an decision. We have followed the latter approach in our module

4.1 Testing & Training data

To decide which approach to be followed we divide the data into two sets 'Training Data' and 'Testing Data'

e.g. If the horizon is of 24 months and bucket size specified is 1 month. Testing data would be 22 Months i.e. (Bucket (1)*2) and the remaining months would be taken as Training Data.

After having made split between Training and Testing data. We use the Training data to forecast for testing data horizon. The forecast we then get is then compared with original testing data. The difference between them helps us decide the recommended approach.

4.2 Error Quantification

To quantify the difference between Testing data and Forecasted data for the same horizon, we are using Root Mean Square Deviation (RMSD)

The RMSD represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T} (x_{1,t} - x_{2,t})^2}{T}}.$$

$X_{1,t}$: Forecasted data for the Testing data horizon

$X_{2,t}$: Testing data

Please refer Figure 4 to understand the RMSD outputs for each product X forecasting combination

Model	1	2	3	4
HW	7.2	54.1	238.6	23.9
AR	0.0	0.0	0.0	0.0
ARMA	0.0	0.0	0.0	0.0
ARIMA	25.0	52.2	97.9	131.5
SARIMA	33.2	29.4	75.2	94.4

'yhton Load | Sheet4 | Bucket Master | HW | AR | ARMA_ARIMA | SARIMA | RMSE

Figure 4: RSMD for four products using various models.

5. EVENTS

The baseline forecast we are getting through different approaches is purely mathematical in nature and based on the Historical data**.** If the user knows that there are going to be some Events in the near future which can impact the sale nos. He would need to define those in the module, so as to account its impact in the forecast.

These events can be of two types

Internal Events – Marketing campaign start, sales force augmentation, distribution channel augmentations etc

External Events – Change in Government regulations, Market entry of a competitor, COVID Lockdown etc

5.1 Modelling the Events

Market demand is induced by various external events.[5] A user would need to input the below metrics to define an Event.

Peak Impact –User would need to enter in Peak Impact of the Event he is configuring.(refer figure 5) This input can be calculated using Historical data (e.g. 80% decrease in case of a Lockdown) or based on relevant Market Research.

Start data – When would the Impact of the event start getting applied on the baseline forecast

Event Name	*On/Off*	*SKU*	*Start Date*	*Peak Share/Impact*	*Uptake Curve*	*Time to Peak (Days)*
Event 1	On	Sony 32'	16-05-2021	5%	Curve type 2	20
Event 2	On	Sony 32'	17-05-2021	10%	Curve type 5	10
Event 3	Off					
Event 4	Off					
Event 5	Off					
Event 6	Off					
Event 7	Off					
Event 8	Off					
Event 9	Off					
Event 10	Off					

Figure 5: Event Panel

Uptake curve – The Events Peak Impact can be reached in many ways. The user can choose from a Curve Library which best fits the situation. (refer Figure 6)

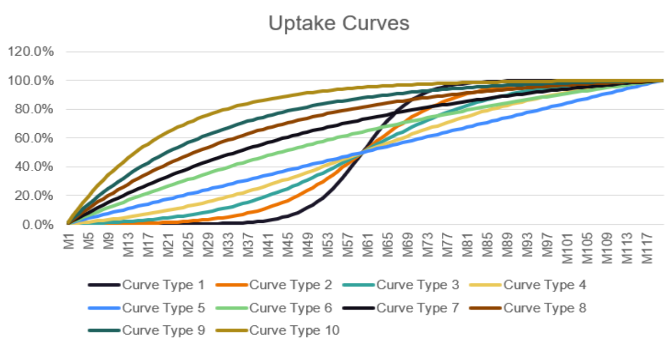

Figure 6: Curve Library

Case in Point – Finalizing Inventory of an Electronic product

STEP 1 – Adding the data

Paste the raw data (refer Figure 7) in the 'Raw Data' sheet of the module.

A	B	C
Product	Date	Quantity
Sony 32'	01-01-2018	589
Sony 32'	01-02-2018	561
Sony 32'	01-03-2018	640
Sony 32'	01-04-2018	656
Sony 32'	01-05-2018	727
Sony 32'	01-06-2018	697
Sony 32'	01-07-2018	640
Sony 32'	01-08-2018	599
Sony 32'	01-09-2018	568

Figure 7: Raw data snippet

STEP 2 – Configuring the module

In accordance with the data, user would first configure the Last Historical date and the Horizon of the data (Total Months) as depicted in Figure 8.

Specify Bucket for each product mentioned in the raw data. In our case it would be 60 Days as the product is being imported.

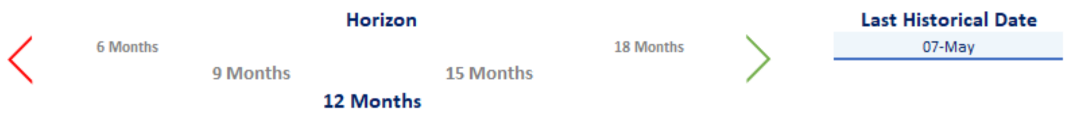

Figure 8: Time frame Module

STEP 3 – Kickstarting the Python Engine

After configuring the model the user would need to upload the file on a specified url and click on Generate forecast to start the the Python Module in the background. Post completion, download the excel module embedded with all the forecast data.

STEP 4 – Decide the Approach to be followed

Normalization flag would tell the user whether the Simple Approach can be used or not.

Apart from that, Module suggests Recommended Statistical Approach to the user by comparing RMSD values of each Forecasting methodlogy.

Step 5 – Define the Events

In our case lets assume a event of Lockdown coming up in the next 60 days. User would need to configure the Start date of the lockdown, Peak Impact and the Uptake curve.

Step 6 – Setting the Inventory

As per the Bucket Input, selected forecasting approach and the impact of defined events, the user would get a definitive Inventory quantity at an SKU level.

CONCLUSION

On several occasions, complex forecasting models that are discussed in the paper are more accurate [6] and access to such models along with the ease of their usage will help the organisation overcome the flaws in their simplistic approaches. Such a module can offer more realistic insight and would help to be better prepared for unforeseeable times. Using this insight, an organization would be able to maintain optimum inventory levels, taking substantial strides in making their operations leaner without affecting serviceability.

REFERENCES

[1] S.Angel Raphella, S.Gomathi Nathan and G.Chitra, "Inventory Management- A Case Study", International Journal of Emerging Research in Management & Technology, ISSN: 2278-9359, Vol.3 June 2014.

[2] Lishura Chen, "Statistical Forecasting Modeling to Predict Inventory Demand in Motorcycle Industry: Case Study", Industrial Engineering and Management, ISSN: 2169-0316, Vol.7, April 2019.

[3] A T Bon and T K Ng, "An Optimization of Inventory Demand Forecasting in University Healthcare Centre", IOP Conf. Series: Materials Science and Engineering, 2017.

[4] Xigang Yuan, Xiaoqing Zhang, Dalin Zhang, "Analysis of the Impact of Different Forecasting Techniques on the Inventory Bullwhip Effect in Two Parallel Supply Chains with a Competition Effect", Hindawi Journal of Engineering, 2020.

[5] Elliot Maître, Zakaria Chemli, Max Chevalier, Bernard Dousset, Jean-Philippe Gitto, Olivier Teste, "Event detection and time series alignment to improve stock market forecasting", CEUR, June 2020.

[6] Kesten C. Green, J. Scott Armstrong, "Simple versus complex forecasting: The evidence", Journal of Business Research, April 2015.

[7] Rob J Hyndman, George Athanasopoulos, "Forecasting: Principles and Practice", April 2018.