Intelligent Reasoning Systems Project

ReplenishNow

NUS Institute of System Science, MTech Intelligent Systems

Benjamin Lim
Bianca Cao
Toh Zhi Yuan

5-21-2023

Contents

Table of Figures

1.0 Executive Summary

ReplenishNow is a recommendation engine designed for small-to-medium enterprises seeking to improve the accuracy and efficiency of their inventory restocking strategies. The system implements existing industry methods to forecast inbound stock to balance between safeguard against stock-out (no products to fulfil customer orders) whilst minimizing inventory to avoid over stocking (unnecessary overhead and depreciation costs).

Under the hood, ReplenishNow uses a combination of machine reasoning and learning to forecast and monitor the supply and demand levels to make daily adjustments to inventory levels in real time.

The application's front-end is deployed using ReTool – A web-based internal tool creator with in-built authentication and user interface functions that enables rapid development across a wide spectrum of experience. ReTool has expanded the user demographic beyond just development and planning teams to include sales, procurement, and warehouse teams for end-to-end decision making and closed-loop communication on a single platform.

The end result is a resilient supply chain which will mitigate the risk of overstocking or stockouts and reduce the inventory holding costs, leading to improved efficiency and profitability.

2.0 Market Research

2.1 Domain Knowledge

Industry experts have established best practices for forecasting inbound inventory to prevent both stock outs and overstocking scenarios that are commonly utilized by companies across various sectors and sizes. Many of the afore-mentioned approaches utilize a consistent and repeatable business logic but are implemented manually through human analysis.

Demand forecasting: Accurate demand forecasting is crucial for maintaining optimal inventory levels. Analyse historical sales data, market trends, and factors influencing demand to anticipate future demand patterns. This allows you to adjust your inventory levels accordingly and minimize stockouts.

Safety Stock: Safety stock acts as a buffer to protect against unexpected fluctuations in demand or supply chain disruptions. It represents additional inventory held above the average expected demand to ensure availability during peak periods or delays. Calculate safety stock levels based on historical data and lead time variability.

Inventory Management Systems: Utilize advanced inventory management software or systems to automate and optimize inventory control. These systems can track real-time inventory levels, monitor sales, generate purchase orders, and provide valuable insights for decision-making.

Supplier Relationship Management: Cultivate strong relationships with suppliers to enhance communication and collaboration. Develop mutually beneficial agreements, negotiate favourable terms, and establish clear lines of communication for timely updates on stock availability and lead times.

Just-In-Time (JIT) Inventory: JIT inventory management focuses on receiving goods only when needed, minimizing the need for excess stock. By streamlining the supply chain and reducing carrying costs, businesses can improve efficiency and minimize the risk of stockouts.

ABC Analysis: Implement the ABC analysis technique to classify items based on their value and prioritize inventory management efforts accordingly. Categorize items as A (high-value, low-quantity), B (moderate-value, moderate-quantity), and C (low-value, high-quantity) to allocate resources effectively.

Continuous Monitoring and Data Analysis: Regularly monitor inventory levels, sales patterns, and market trends. Leverage data analysis techniques to identify demand patterns, lead time variations, and potential stockout risks. By staying vigilant and proactive, you can make data-driven decisions to optimize inventory levels and prevent stockouts.

Collaboration Across Departments: Foster collaboration between different departments, such as sales, operations, and procurement. Encourage cross-functional communication to align inventory management strategies with sales forecasts and customer demand.

Regular Inventory Audits: Conduct routine physical inventory counts and audits to identify discrepancies, minimize inaccuracies, and uncover any issues with inventory management. This helps maintain accurate stock levels and reduces the risk of stockouts due to inventory discrepancies.

Continuous Improvement: Implement a culture of continuous improvement by regularly reviewing and optimizing inventory management processes. Seek feedback from employees and stakeholders, analyse performance metrics, and adapt strategies to evolving market conditions.

Data used for forecasting strategy is almost always confidential between companies and its suppliers, such as lead time and pricing. Regardless of the industry and company size, it is predictable that the data required to fuel the recommendation engine is available for input by the user on assumption they are utilizing the above methods without any novel changes to the thought process. From data analysis and demand forecasting, we can learn new knowledge on recalibrating the safety stock level and re-ordering point to achieve dynamic replenishment with rule-based decision making.

2.2 Problem Statement

Companies adopt robust inventory management and forecasting processes to safeguard against stock-out scenarios in their supply chain downstream to customers or internal stakeholders. The same processes also help to prevent over-stocking, whereby costs for non-moving goods are accumulated, such as opportunity cost for warehouse space for product with higher value or liquidity, depreciation costs, insurance and utility overhead.

The problem is that forecasting methods are often inaccurate due to fixed algorithms and underlying, tribal knowledge behind the decision making. This potentially disturbs the balance between overstocking and stockout and leading to either event happening.

For Small-To-Medium enterprises, these methods are also not often executed efficiently as they do not have established management systems and they often have inherent bias for overstocking to avoid bad customer relations.

Bottom line, inaccuracy and inefficiency in current forecasting methodology leads to **increased costs** in both operational expenses and manhours to manage the issues (whether over or under supply) as they arrive.

3.0 Project Definition

3.1 Objective

To design and deploy a bespoke recommendation system for SMEs that utilizes artificial intelligence to suggest time-critical replenishment decisions to achieve an optimal inventory volume that prevents both stock out and overstock situations.

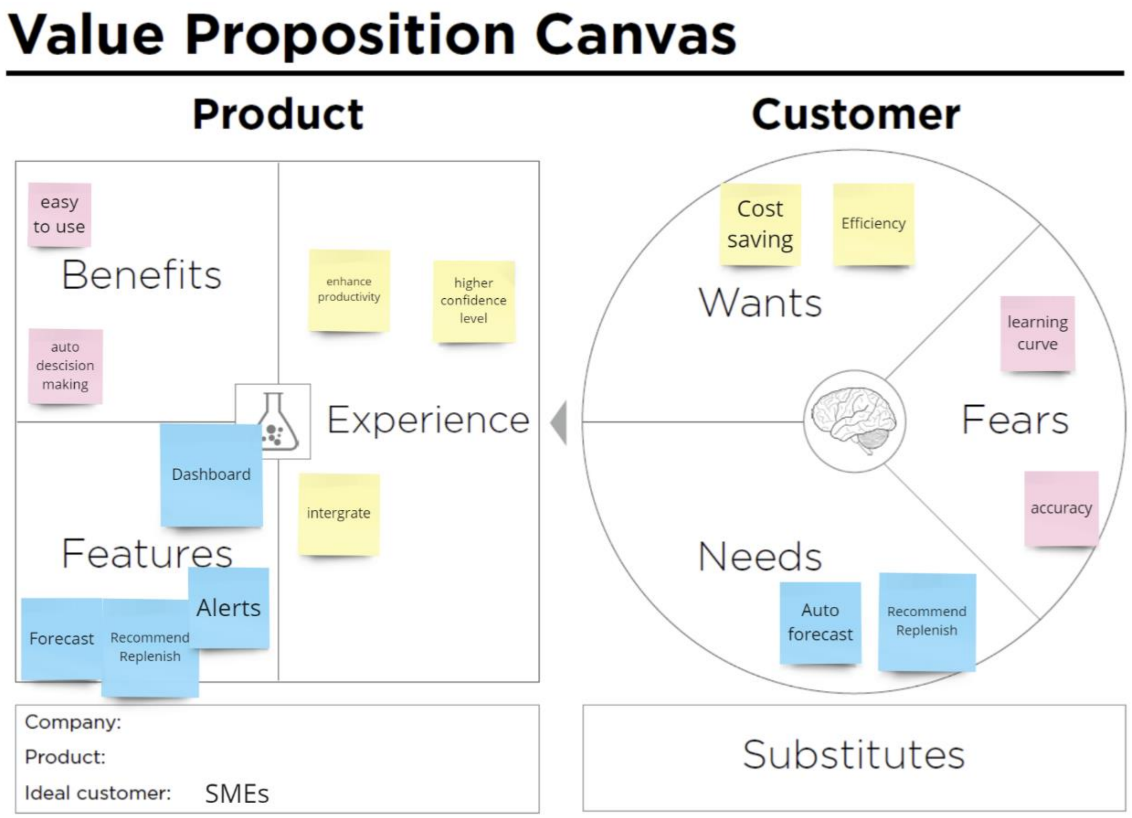


Figure 1 – Value Proposition Canvas

3.2 Requirements

The application must achieve the following:

1. **Decision Automation:** Business & process rule-based module. Understanding current business rules used by SME to enable decisive supply chain management based on real time data.
    - Automatic inventory replenishment based on reorder points or demand forecasting.
    - Intelligent exception handling and alerting for out-of-stock situations, unexpected demand spikes or lead time delays.
    - Automate decision-making and continuously improve inventory planning accuracy

2. **Knowledge Discovery & (Big) Data Mining Techniques:** Analytics and discovery from different SKU (Stock Keeping Unit) relations and find out if certain events may impact. (promotions, volumes
   - Data collection and integration from multiple sources, including sales history, demand forecasts, supplier performance, and production schedules.
   - Predictive analytics to forecast future demand and identify potential inventory risks or opportunities
3. **Business Resource Optimization:** Minimize excess inventory and spoilage, overhead for warehouse space and upkeep, Just-In-Time approach to procurement.
   - Optimization of inventory levels to trade-off between stockouts and excess inventory
   - Real-time monitoring of inventory turnover, lead times, and other key performance indicators (KPIs) to support continuous improvement
4. **Interactive UI:** Proactively queries and alerts users on changes in supply chain and actions to take. Dashboard that displays critical business metrics (sales forecasts, executive-related data)
   - UI to support intuitive and user-friendly inventory planning
   - Intelligent alerts and recommendations based on customer preferences or historical trends
   - Cognitive search and discovery to quickly find relevant inventory and demand data
   - Cognitive process automation to optimize inventory planning workflows and reduce manual effort.
   - Data analytics and visualization to identify trends, patterns, and anomalies in inventory and demand data

3.3 Assumptions

The application was designed with the following assumptions/constraints

1. Costs of inventory transportation (inbound and outbound) is not a decision factor as to when restock orders should be made.
2. Users have existing database for tracking their existing and forecasted inventory levels, and also supply chain information such as lead time they can upload into the application in bulk.

4.0 System Design

4.1 Overview

An effective inventory replenishment system ensures that an organization has the right amount of inventory at the right time to meet customer demands, while minimizing inventory holding costs. The design of an inventory replenishment system involves several key factors, such as demand forecasting, lead time analysis, safety stock calculation, order quantity determination, and order frequency.

Demand forecasting involves analysing past sales data to determine future demand. Accurate demand forecasting helps to prevent stockouts and overstocking, which can lead to lost sales and increased holding costs, respectively. Lead time analysis is the process of determining how long it takes for an order to be fulfilled from the time it is placed. This information is essential in determining the appropriate order point and order frequency.

Safety stock calculation involves determining the minimum amount of inventory that should be held to prevent stockouts due to unexpected demand or lead time variations. Order quantity determination is the process of determining the optimal amount of inventory to order each time an order is placed. This helps to minimize ordering costs and holding costs. Finally, order frequency is determined based on lead time, order quantity, and demand patterns.

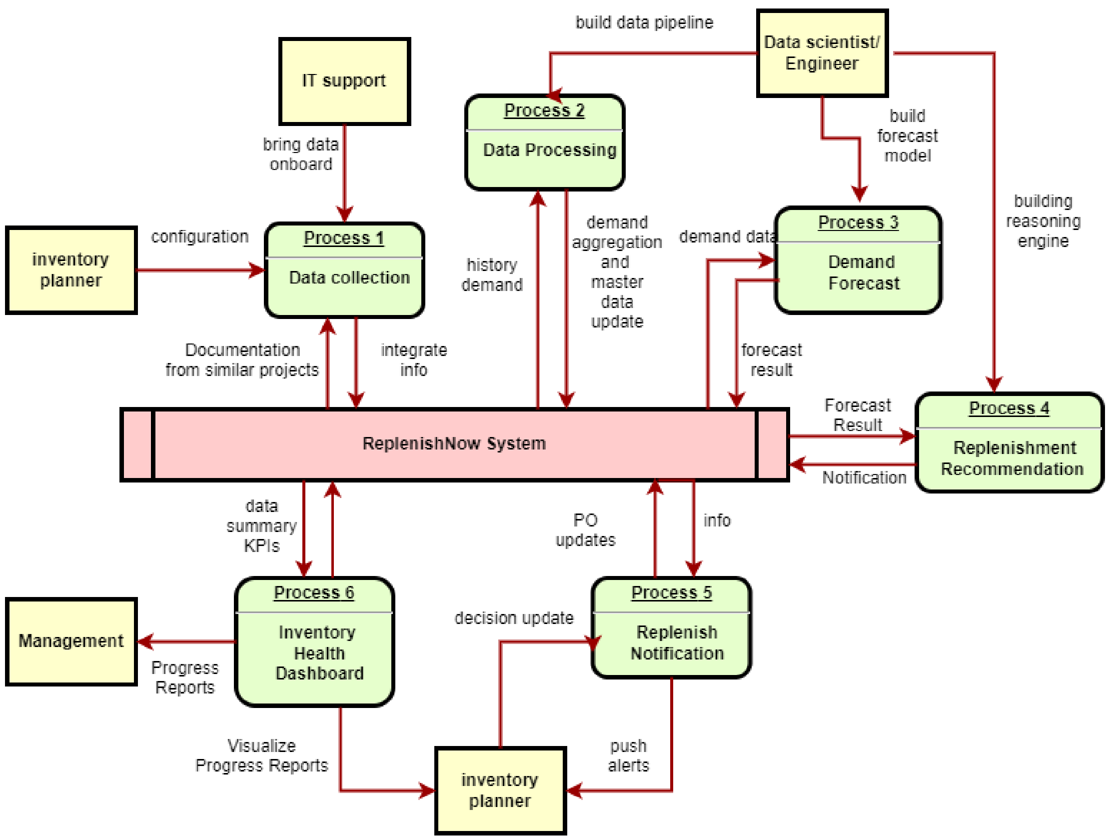


Figure 2 - Overall Data Flow Chart

Overall, the design of an inventory replenishment system is a complex process that involves various factors, but it is critical for the success of any organization that manages inventory. A well-designed system ensures that inventory levels are optimized to meet customer demands while minimizing holding and ordering costs. In our solution, we will have 4 key features:

1. SKU level Demand forecasting on historical sales order
2. Safety Stock Calculation using classical inventory theory
3. Rule-based inventory replenishment decision making
4. Inventory optimization using integer programming solver

4.2 Application Architecture

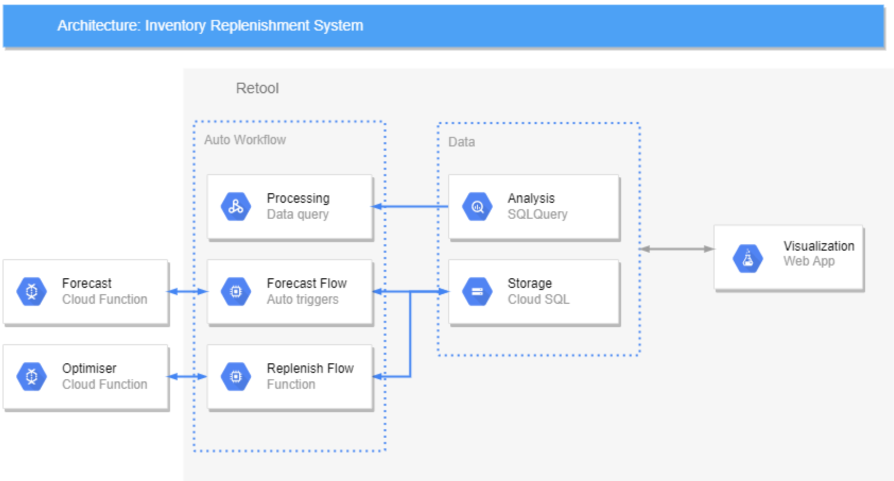


Figure 3 - System Architecture

4.2.1 Database Design

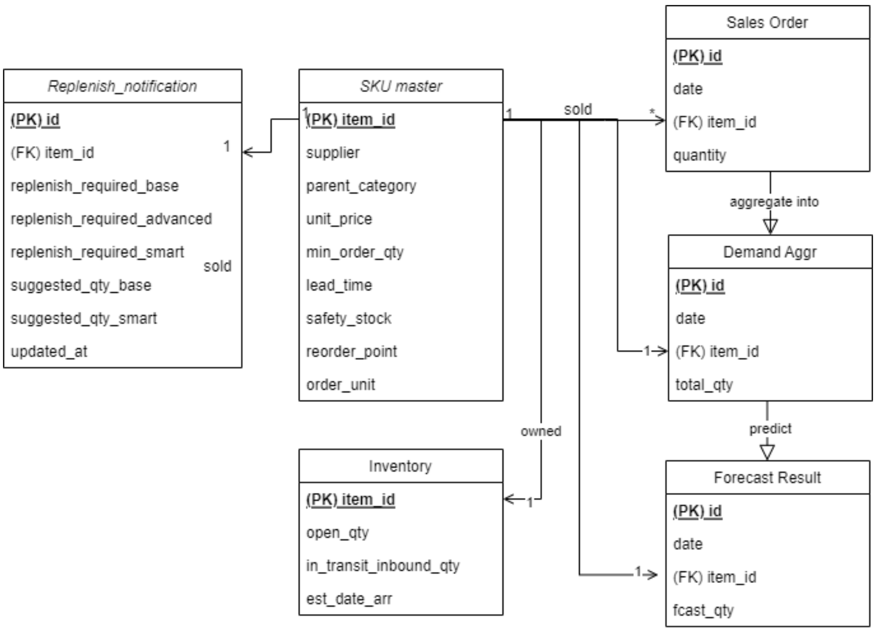


Figure 4 - Relational Database Schema

Database schema (ERD)

This is a minimum database entity relationship diagram.

SKU master: to store item master data (data source: system or input by planner)

Inventory: to store inventory open quantity and in-transit quantity for each item

(data source: warehouse management system WMS)

Sales order: to store all the outbound records from customer (data source: Sales system)

Demand Aggregation: to aggregate sales data for forecast, automatically process when sales order update We can build data pipeline (ETL) or SQL stored procedure to aggregate sales order on SKU level (group by) and sum up numbers to daily/weekly basis (resample). Alternatively, we can integrate with Retool Workflow: embed the processing here to update “demand_aggr” table

Forecast Result: being updated with latest forecast values for each active SKU items.

Replenish Notification: to store all the replenishment suggestion from 3 different rule/algos.

4.2.2 UI with Retool

Retool offers four key advantages in the development and deployment of ReplenishNow.

1. Firstly it provides a high level of customization, allowing us to tailor the interface to meet our specific needs. This can help us to create a unique and engaging user experience that is optimized for our particular use case.
2. Secondly, developing our own web UI and backend service from scratch can be a time-consuming and costly process. By using Retool, we can significantly reduce development costs and accelerate the time-to-market for our application.
3. Thirdly, Retool provides a user-friendly drag-and-drop interface that requires no coding experience, which means that our team can quickly create and modify applications without the need for specialized technical skills.
4. Finally, Retool offers robust support and maintenance, which helps us to avoid the potential costs and headaches associated with managing our own infrastructure. By leveraging Retool's existing platform and expertise, we can focus our resources on improving and expanding the functionality of our application, rather than on maintaining the underlying infrastructure.

The system required highly customization for each individual company client, including data source configuration, system integration, data characteristics.

4.2.3 Backend with Retool

- Retool workflow: scheduled and automated pipeline
    - data processing routines
    - Forecasting API triggers
    - Replenish Decision Maker
    - Optimiser API triggers
- Retool query functions: to calculate and retrieve data from tables be used in UI
- 2 cloud function API:
    - Forecasting API deployed via Azure function
    - Replenish Optimiser API via Google Cloud function

4.3 Knowledge-Based (KB)

The forecasting module, we trained and saved the model in a file. The re-order decision maker uses domain knowledge. Both will be explained in the 'our methodology' part in the below 2 chapters.

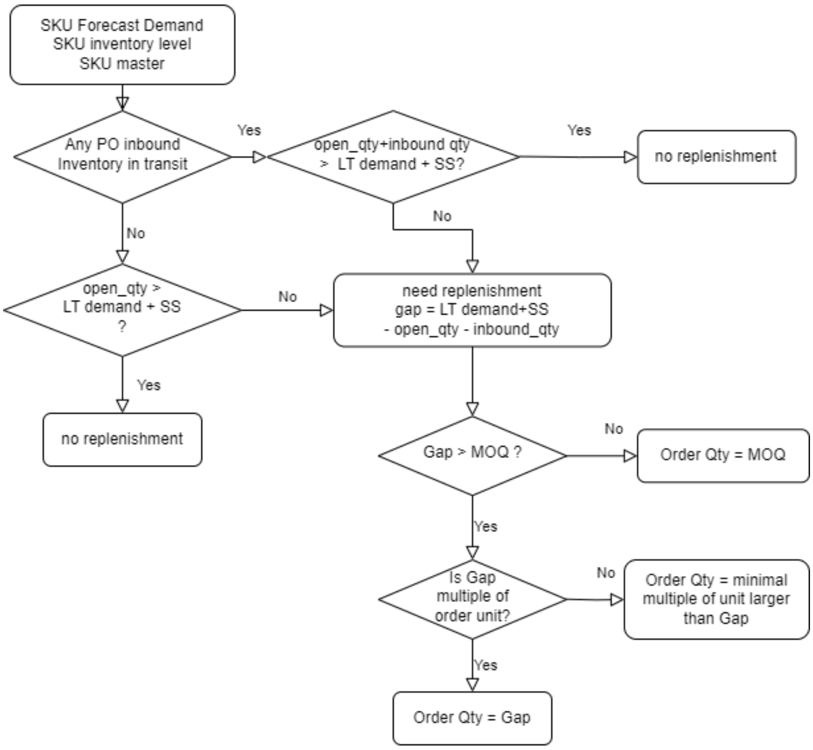


Figure 5 - Replenish Strategy Flow Chart A

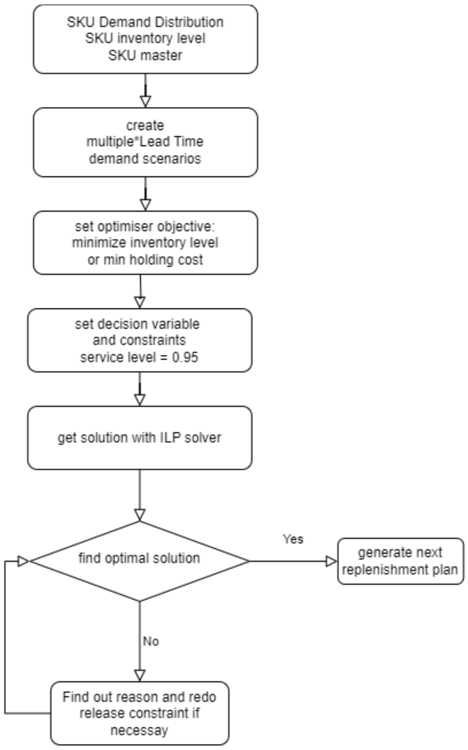


Figure 6 - Replenish Strategy Flow Chart B

5.0 Demand Forecasting

Goal of this module is to:

1. build auto forecasting model pipelines for each SKU
2. propose a hybrid model to outperform the baseline model (moving average)
3. analysis and evaluate the performance of the hybrid model
4. Utilize the forecasting values as replenishment decision input

5.1 Theory

5.1.1 Demand Characteristics

Smooth represents a relatively stable pattern and it's easy to forecast. It shows a regular demand and regular time. i.e. This type of products can be sold every day or every week. (Essential household items: e.g., bread, milk and eggs) Erratic pattern means regularity in time, but the quantity varied dramatically. It's unpredictable & challenging to forecast highly variable nature. i.e. This type of products can be sold every day or every week however, for example, one day it may sell 2 in quantity whereas, another day it could sell 200 in quantity. (e.g. luxury items). Intermittent illustrate sporadic demand patterns. It can be difficult to forecast. shows irregularity in time and regularity in quantity pattern. i.e., seasonal products

Finally, lumpy type is unpredictable no matter what type of time series models is used It shows irregularity in time and irregularity in the quantity pattern. A solution for this type of product is to have a safety stock. e.g., expensive capital goods

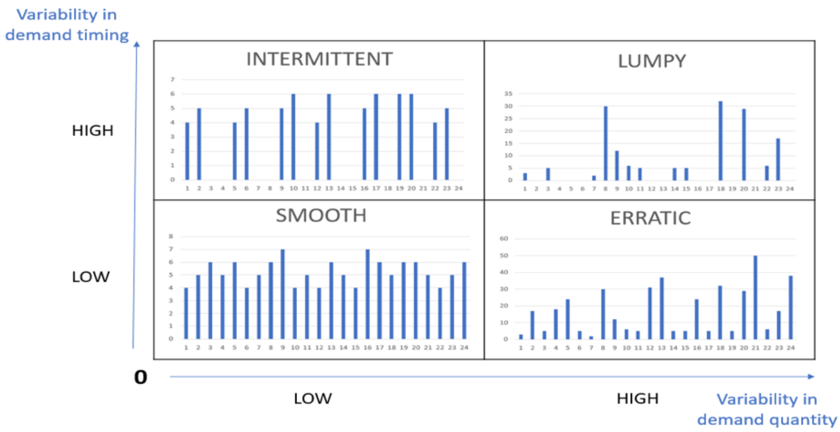


Figure 7 - Illustration of 4 Demand Characteristics (Source: Frepple APS)

Here we picked high-running items from the dataset to make sure the time series do not have a lot zero interval, namely only smooth pattern, so we can focus on the items are more predictable. Metrics for smooth demand: mean inter-demand interval < 1.32 square of the Coefficient of Variation < 0.49.

5.1.2 Forecasting Models

Although our implementation makes use of PyCaret, an open-source, low-code, end-to-end machine learning library in Python that automates the workflows, we still conducted study on a variety of forecasting models to understand what the promising candidates for our use case are. We must know what is behind the scenes before trusting it has the capacity and help us to speed up all the model selection productively.

	Model
ets	ETS
exp_smooth	Exponential Smoothing
arima	ARIMA
auto_arima	Auto ARIMA
par_cds_dt	Passive Aggressive w/ Cond. Deseasonalize & Detrending
lar_cds_dt	Least Angular Regressor w/ Cond. Deseasonalize & Detrending
huber_cds_dt	Huber w/ Cond. Deseasonalize & Detrending
lr_cds_dt	Linear w/ Cond. Deseasonalize & Detrending
ridge_cds_dt	Ridge w/ Cond. Deseasonalize & Detrending
en_cds_dt	Elastic Net w/ Cond. Deseasonalize & Detrending
lasso_cds_dt	Lasso w/ Cond. Deseasonalize & Detrending
br_cds_dt	Bayesian Ridge w/ Cond. Deseasonalize & Detrending

knn_cds_dt	K Neighbors w/ Cond. Deseasonalize & Detrending
theta	Theta Forecaster
et_cds_dt	Extra Trees w/ Cond. Deseasonalize & Detrending
dt_cds_dt	Decision Tree w/ Cond. Deseasonalize & Detrending
lightgbm_cds_dt	Light Gradient Boosting w/ Cond. Deseasonalize & Detrending
omp_cds_dt	Orthogonal Matching Pursuit w/ Cond. Deseasonalize & Detrending
gbr_cds_dt	Gradient Boosting w/ Cond. Deseasonalize & Detrending
rf_cds_dt	Random Forest w/ Cond. Deseasonalize & Detrending
catboost_cds_dt	CatBoost Regressor w/ Cond. Deseasonalize & Detrending
ada_cds_dt	AdaBoost w/ Cond. Deseasonalize & Detrending
xgboost_cds_dt	Extreme Gradient Boosting w/ Cond. Deseasonalize & Detrending
llar_cds_dt	Lasso Least Angular Regressor w/ Cond. Deseasonalize & Detrending
naive	Naive Forecaster
snaive	Seasonal Naive Forecaster
polytrend	Polynomial Trend Forecaster
croston	Croston
grand_means	Grand Means Forecaster

Classical Models

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)},$$

Seasonal Naïve Model:

Decomposition

A good practice is to split a time series into several components that represents pattern categories

Single Decomposition: additive y_t= s_t + T_t + R_t ;multiplicative y_t= s_t * T_t * R_t

STL Decomposition: Seasonal and Trend decomposition using Loess method it can handle any type of seasonality and is robust to outliers

Regression

includes different linear regression models and non-linear ones. A simple weighted moving average model can perform well under certain conditions. For example,

Demand_t = 0.4*Demand_{t-7} + 0.3 *Demand_{t-14} + 0.2 *Demand_{t-21} +0.1*Demand_{t-28}.

Though it looks trivial at first glance, the model considers weekly seasonality, long-term trend and decayed history. This is the real case observed by one of our team members in the work. An industry planner who has 10-year domain experience created it, but she knows little on data science.

Exponential Smoothing (star representative: Holt-Winters)

The Idea is to produce weighted averages of historical observations as forecasting results the weights keep decaying exponentially for the older observations. The model is widely used and reliable for a wide range of time series efficiently. Holt-Winters additive model is one of the variants that capture seasonality in the additive way. In the other hand, **ARIMA models** should be used on stationary data only while Exponential smoothing methods are appropriate for non-stationary data (i.e. data with a trend and seasonal data).

Prophet is invented by Facebook team. It's very easy to use and able to take in multi seasonality as well as event or holiday factor. In addition, Prophet uses a Bayesian framework that allows for uncertainty estimation. Users can alter the key parameter via built-in functionalities like trend changepoint detection and holidays effects modelling, which can improve the accuracy of the forecast with minimum effort.

Machine Learning Models

(We investigate them but decide to put it out-of-scope as it requires more data to train)

The most popular ones are Recurrent Neural Network and Long Short-Term Memory model. RNN is designed to process the data in the sequential form. input and output layer as the conventional neural network models hidden layers are composed of recurrently connects cells (memory states) Although RNN has the capability of capturing the temporal dependence in **sequential** data, it has a **disadvantage when facing a long-term past observation**. The gradient decays exponentially when learning through time, resulting in little information retained by the RNNs.

Long Short-Term Memory (LSTM) targeting to tackle the *vanishing gradient problem of RNNs* when processing long-term temporal dependencies. It combines RNN with a standalone memory cell and a series of gates across the network.

Hybrid Model

Certainly, all the models mentioned above can be fused together to achieve higher accuracy.

E.g. TBATS concludes key features of the models: Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components. Belows shows the model from the winner of the M4 forecasting competition. It integrates multiple techniques (both statistics and machine learning ones)

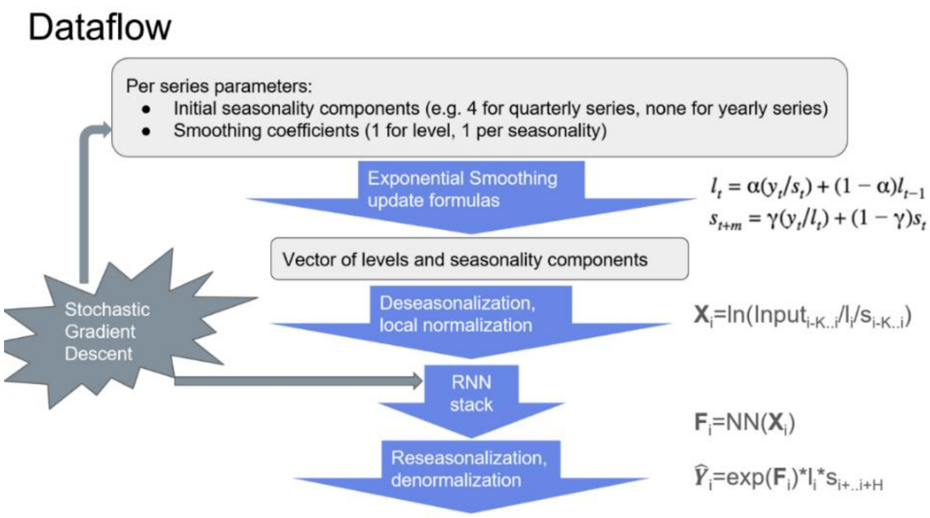


Figure 8 - Dataflow Diagram

5.2 Our Methodology

5.2.1 Our Use Case - Supply Chain Sales/Demand Forecasting

The entire predictive model system for demand forecasting evolved into four parts:

1. The classic periodic models are simple to implement and perform good enough especially on certain categories.
2. The Machine Learning models can be introduced, when we aim for higher accuracy and bring complex factors into consideration e.g. promotion, marketing change, etc.
3. New product models are still challenging in the industry. However, a more practical approach is to utilize similarity-based logic.
4. Finally, a routing model can be proposed to automatically match each commodity within a specific period and determine the suitability of meta model adaptation based on algorithmic identification

Again, due to the limited timeline and data availability, we wrapped the seasonality analysis, model building, model selection, fine tuning, model blender, evaluation all together with **PyCaret**.
We are focusing on the **smooth** demand type given the nature of our dataset.

5.2.2 Model Evaluation and Cross Validation

Metrics

Evaluating the effectiveness of the forecasting model Metrics are MSE, RMSE, MAPE. Furthermore, the symmetric Mean Absolute Percentage Error (sMAPE) & Mean Absolute Scaled Error MASE are more critical.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2. \qquad RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}} \qquad \mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n}.$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \qquad \mathrm{SMAPE} = \frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

We can use MAPE to compare different models and MASE to validate if the performance is better than the Naive forecast.

Cross Validation Issue

Running CV on time series forecasting is very different from other classification/regression models, as it works on a rolling basis instead of discrete picked data points. We cannot split train and test data set randomly as the data points are closely correlated in time sequence. Thus, we need to separate the series based on time.

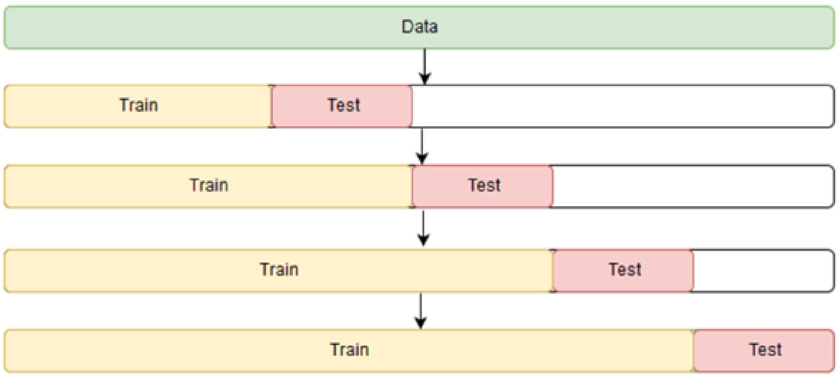


Figure 9 - Illustration of Cross Validation on Time Series (Source: Analytics India Magazine)

5.3 Data Gathering and Processing

From open-source dataset to extract 7month sales data and SKU master information.
Mock the Lead time and supplier as data is not available.
Need to aggregate the data from sales order to SKU level daily demand.

Processing on collected data *sequence interchangeable*

1. Data cleaning: drop duplicates and missing lines
2. column name mapping and data type casting – sales order must at least have Datetime Index, SKU code and quantity column
3. Group by SKU level demand and join with calendar date (fill in zero if no sold on the date)
4. Time frequency resampling by day, week, month, year, etc. (here we chose day)

Further improvement: introduce Multivariate, Feature Engineering and SKU Correlation as well.

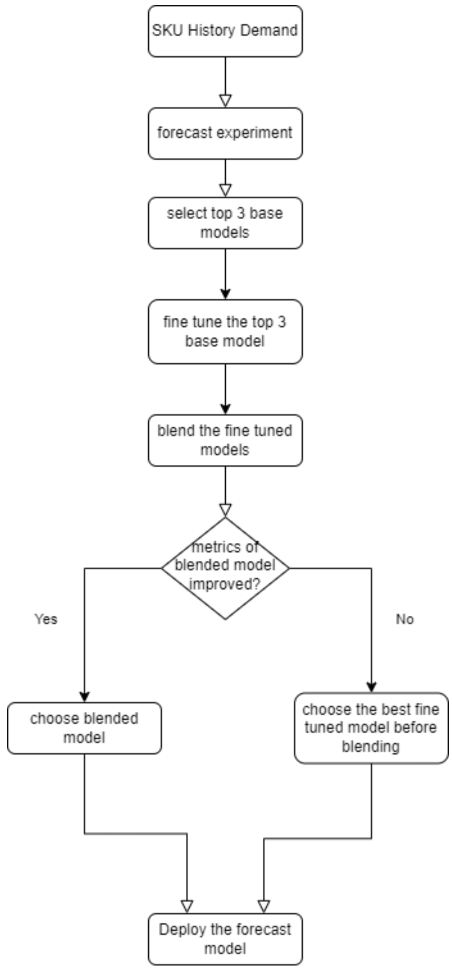


Figure 10 - Forecasting Training Procedure Flowchart

6.0 Replenishment

6.1 Theory: Inventory Policy

To balance the supply and demand, there are mainly 2 types of strategy: Cycle stock inventory& Safety Stock inventory (policy based) e.g. ABC inventory analysis.
By balancing the levels of safety stock and cycle stock, organizations can ensure that they are meeting customer demand while minimizing the costs associated with excess inventory.

6.1.1 Concept

- Vendor lead time (VLT): the length of time elapsed between the issuance of the replenishment request and the replenishment of goods;
- Inventory-on-hand: the actual inventory on hand;
- Inventory level: the difference between the amount of existing inventory and the amount of out-of-stock;
- Inventory position: the sum of inventory level and inventory in transit;
- Safety Stock meant to prepare enough inventory for a demand strike and avoid over stock.
- Service level: the ability to meet customer needs immediately, according to different operating modes, service level usually has two ways to define it:
  Type I: the probability that there is no stock-out during the replenishment lead time.
  Type II: the ratio of demand that is met immediately during the replenishment lead time to the total demand.

Total Cost = replenishment (setup + variable cost) + holding (capital + storage) + stockout loss

calculate economic order quantity (EOQ): to balance the costs associated with holding inventory (carrying costs) against the costs associated with ordering inventory (ordering costs).

5. set a customer service goal first e.g. 95%
6. inventory holding cost base holding on current interest rate
7. shrinkage due to shelf-life or other deprecation

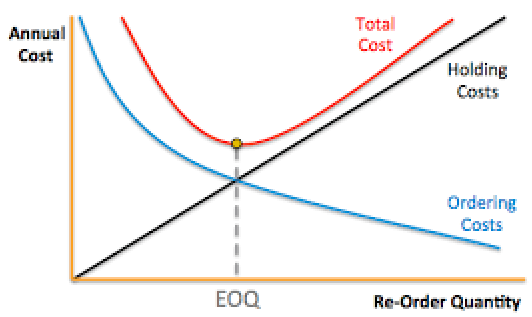
6.1.2 Classic Inventory Replenishment Models

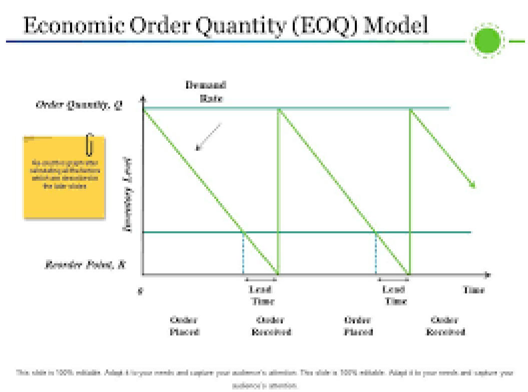
1. EOQ – economic order quantity model

$$Q^* = \sqrt{\frac{2DK}{h}}$$

D= annual demand quantity; K= fixed cost per order; h = annual holding cost per unit

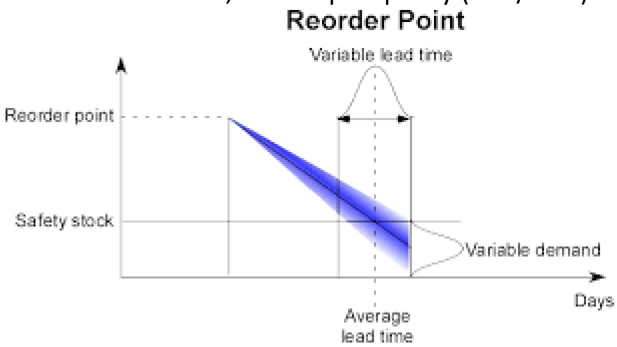
illustration





2. Newsvendor (Demand is random)

3. Periodic review, order-up-to policy (T, S)
   where T is the fixed time between orders and S is the order-up-to-level.
   Orders are placed according to a fixed schedule like every T days.
4. Continuous review, order-up-to policy (Min/Max) or (s,S)



   [Safety Stocks: Beware of Formulas – Michel Baudin's Blog]
5. Continuous review, fixed order quantity policy (Reorder Point, Order Quantity) (R, Q)
   where R is the reorder point and Q is the fixed order quantity.
   order-up-to level - current inventory level.
   Ref: [Top 3 Most Common Inventory Control Policies - Smart Software (smartcorp.com)]

6.1.3 Safety Stock and its calculation

Safety Stock meant to prepare enough inventory for a demand strike and avoid over stock.
The calculation can be done on *either a daily or weekly basis*, depending on the level of granularity required for your inventory management system. Here we mainly working on high-volume, fast-moving inventory with frequent stockouts, so it's calculated on daily level. However, weekly level is acceptable for any SKU has a more stable and relatively predictable demands.

Assuming that demand during successive unit time periods are independent and identically distributed random variables drawn from a normal distribution, the safety stock can be calculated as:[9]

$$SS = z_\alpha \times \sqrt{E(L)\sigma_D^2 + (E(D))^2\sigma_L^2}$$

where,

- α is the service level, and z_α is the inverse distribution function of a standard normal distribution with cumulative probability α; for example, z_α=1.65 for 95% service level. The service level can be easily calculated in Excel by typing in the formula =normsinv(probability%). For eg entering =normsinv(95%) will return 1.65 as the answer.[10]
- $E(L)$ and σ_L are the mean and standard deviation of lead time.
- $E(D)$ and σ_D are the mean and standard deviation of demand in each unit time period.[11]

The reorder point can then be calculated as:

$$ROP = E(L) \cdot E(D) + SS$$

The first term in the ROP formula $E(L)E(D)$ is the average demand during the lead time. The second term ss is the safety stock. If the lead time is deterministic, i.e. $\sigma_L = 0$, then the ROP formula is simplified as $ROP = L \cdot E(D) + z_\alpha \sigma_D \sqrt{L}$.

Image source: Wikipedia

One thing to note here is that the above are assumptions based on **historical data** for future replenishment. In production, usually we will have forecasting, the estimate of the future, so E(D) can be estimated directly using the predicted values and σ_D should be the deviation from the predicted value, not the deviation from the mean.

6.2 Our Methodology: Hybrid Strategy Based on Demand Forecast and Optimiser

6.2.1 Introducing Demand Forecast for Decision

We implement this as our basic replenishment strategy which is called: continuous review, Dynamic Reorder point based on demand forecast.

Logic rules look like this:

(When to order)

IF Day ends AND weekday falls in workday THEN do stock check

Loop through each SKU:

IF inventory level <= Re-ordering Point, THEN place orders now.

(How much to order)

IF place order required, THEN calculate order quantity (desired) = Lead Time demand forecast

IF place order required AND SKU has min order quantity
AND order quantity (desired) < min order quantity, THEN order quantity = MOQ

IF place order required AND SKU has min order quantity
AND order quantity (desired) >= min order quantity,
THEN order quantity = order quantity (desired)

Re-ordering Point = Safety Stock + Lead Time total Demand Qty

Lead Time total demand Qty is counted as expected lead time * expected average demand, but we can replace with the forecast result if it's more accurate.

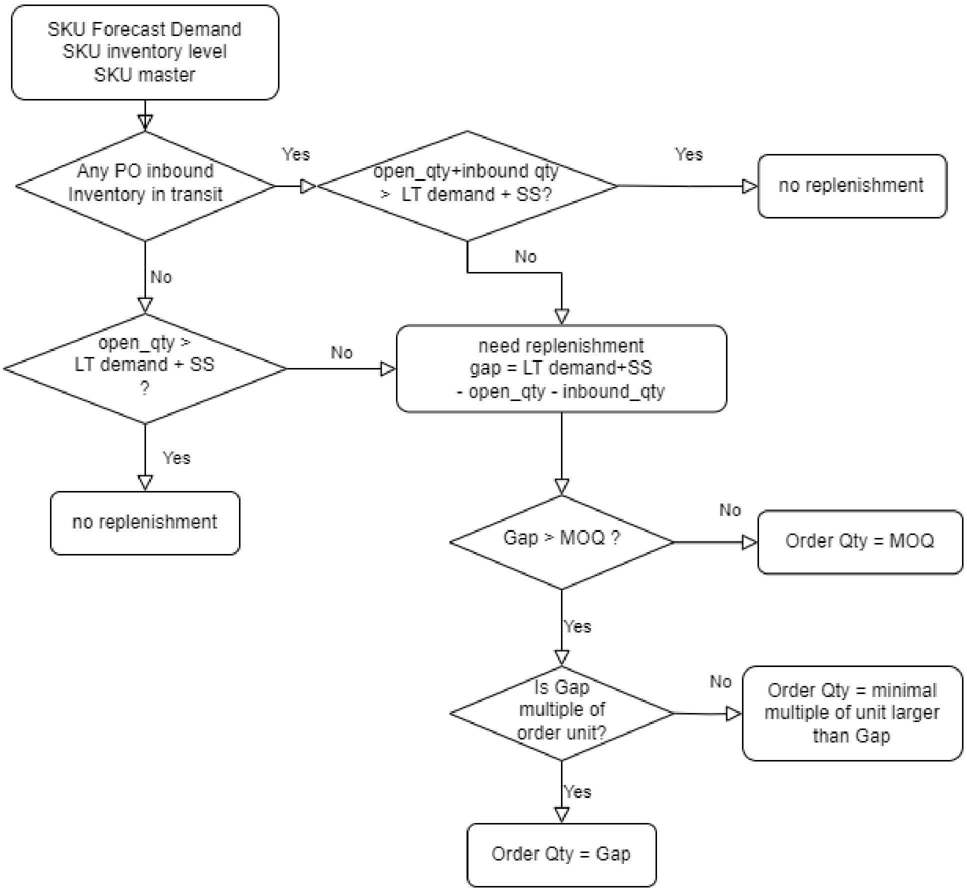


Figure 11 - Replenish Strategy Flowchart (Basic)

6.2.2 Inventory Optimization- Our simple ILP solver

The design of an ILP (Integer Linear Programming) optimizer that takes in demand scenarios and other master data involves a complex process that utilizes mathematical programming to develop an optimal replenishment schedule. The optimizer is designed to consider a wide range of factors, including inventory levels, demand forecasts, supplier lead times, safety stock, and order quantities, to suggest the most effective replenishment schedule possible. The optimizer can generate solutions that minimize costs and optimize inventory levels, while also trade-off specific business requirements and constraints.

Not only it can suggest further replenishment time point and quantity, but it also offer helpful insights like reasonable order frequency when examining the historical data. Ultimately, we must balance the accuracy with model runtime to put in real-time production system.

Input: multiple demand scenarios, initial inventory level, SKU lead time, service rate, SKU MOQ

$D_{s,t}$:= demand quantity at day t, scenario s

I_{init} := initial inventory level at day t, scenario s

service level := success rate of fulfillment

LT := lead time of replenishment

MOQ := minimum order quantity, order quantity must be multiple of it.

S := number of demand scenarios input

T := timespan of the demand input

Decision variables: future item inbound quantity

O_t := order quantity at day t

$I_{s,t}$:= inventory level at day t, scenario s

$F_{s,t}$:= failure (whether demand fulfilled) at day t, scenario s (stockout⇒0, else⇒1)

Constraint: meet service level, replenishment interval, quantity follow multiple of MOQ

initialize: $I_{s,0} = I_{init}$

order interval: O_t = 0 when t is not a workday

update inventory level:

$$I_{s,t+1} = I_{s,t} - D_{s,t} \;\; if\ t < LT$$

$$I_{s,t+1} = I_{s,t} - D_{s,t} + O_{t-LT+1} * order\_unit \;\; else$$

linearization: stock state = 0 if stock out else = 1

$$I_{s,t} <= stock\_state_{s,t} * big\_M$$

$$I_{s,t} >= 1/big\_M - (1 - stock\_state_{s,t}) * big\_M$$

big_M defined based on inventory magnitude

service rate:

$$\sum_{s=1}^{S} \sum_{t=1}^{T} stock\_state_{s,t}/(S * T) \geq service\_level$$

Objective: minimize inventory level (holding cost)

$$min \sum_{s=1}^{S} \sum_{t=1}^{T} I_{s,t}/(S * T)$$

Some default setting:
service level = 95 %
order_interval = 3
big_M = 2000 (given the dataset inventory level)

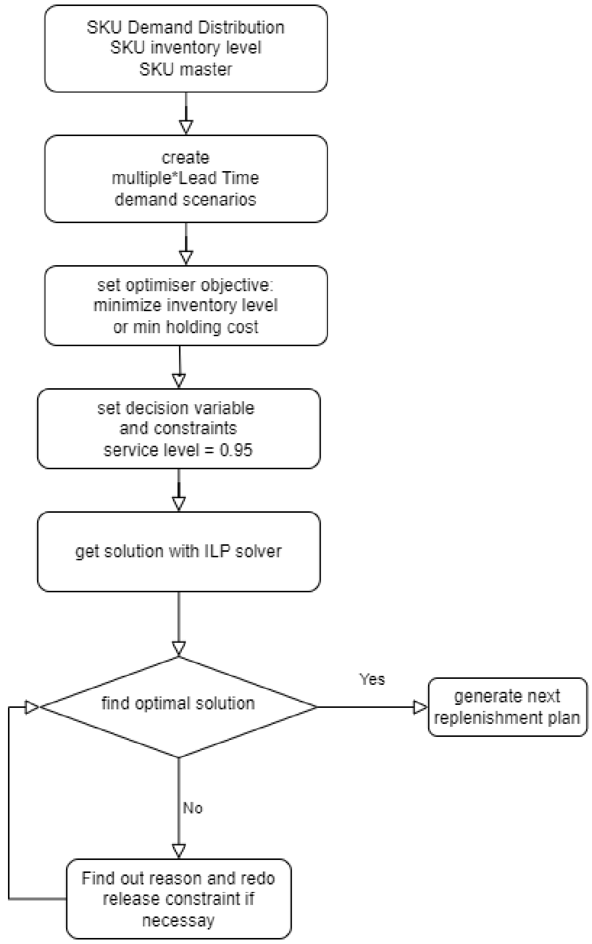


Figure 12 - Replenish Strategy Flowchart (Optimiser)

7.0 Solution Implementation and Testing

7.1 Result

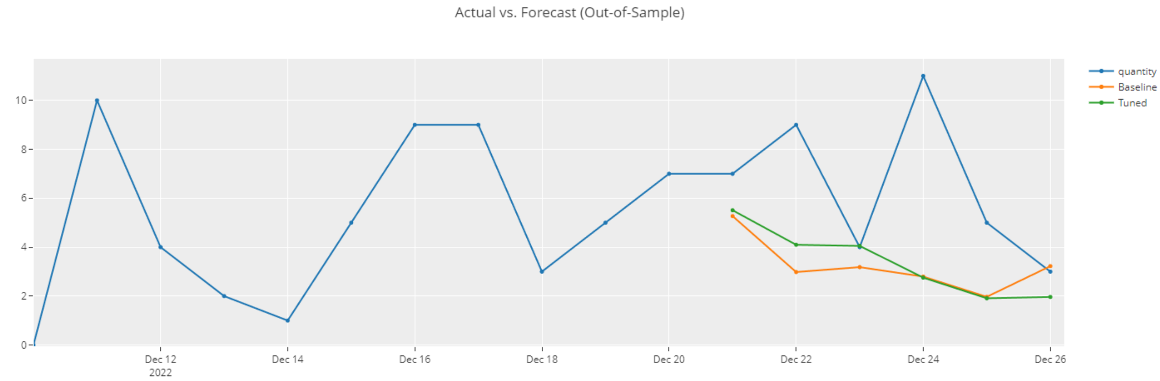
7.1.1. Forecasting

To evaluate the performance of the tuned intelligent forecast model, a concurrent comparison against the actual order and baseline (algorithmic) approach was conducted over a 6 day period from 21 December 2022 to 26 December 2022.

The results show an improvement in the tuned model, which reduced the moving average delta against the actual by **4.39%,** when compared against the baseline.

Date	Actual	Baseline	Baseline vs Actual	Tuned	Tuned vs Actual
21/12/2022	7	5.2697	1.7303	5.5094	1.4906
22/12/2022	9	2.9833	6.0167	4.0975	4.9025
23/12/2022	4	3.1831	0.8169	4.0508	-0.0508
24/12/2022	11	2.7978	8.2022	2.7538	8.2462
25/12/2022	5	1.9697	3.0303	1.9108	3.0892
26/12/2022	3	3.2246	-0.2246	1.9642	1.0358

Avg Baseline Δ	3.2620	Avg Tuned Δ	**3.1189**

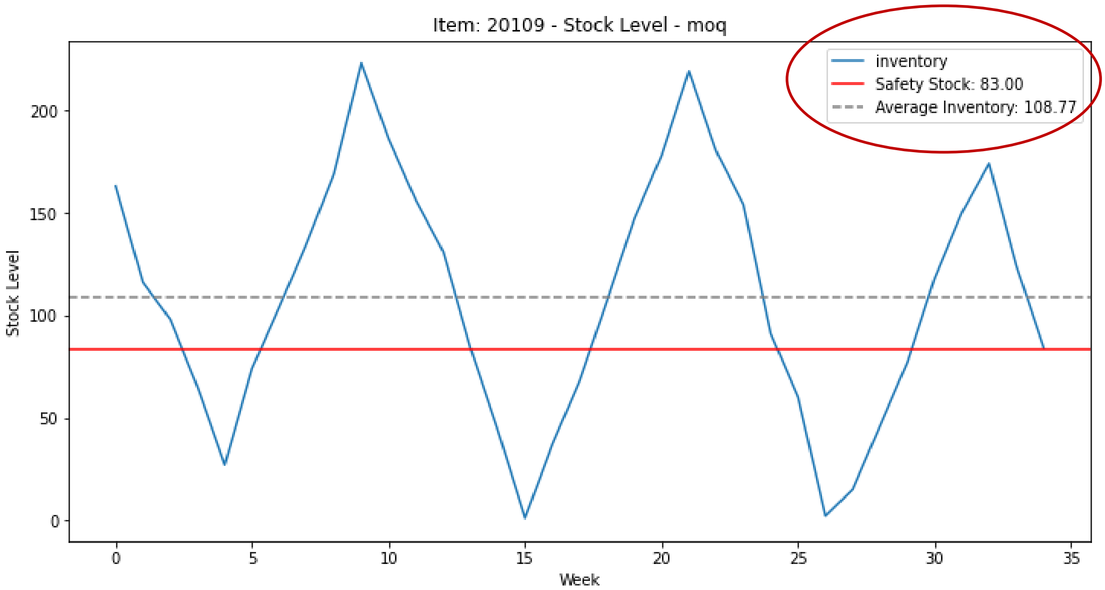


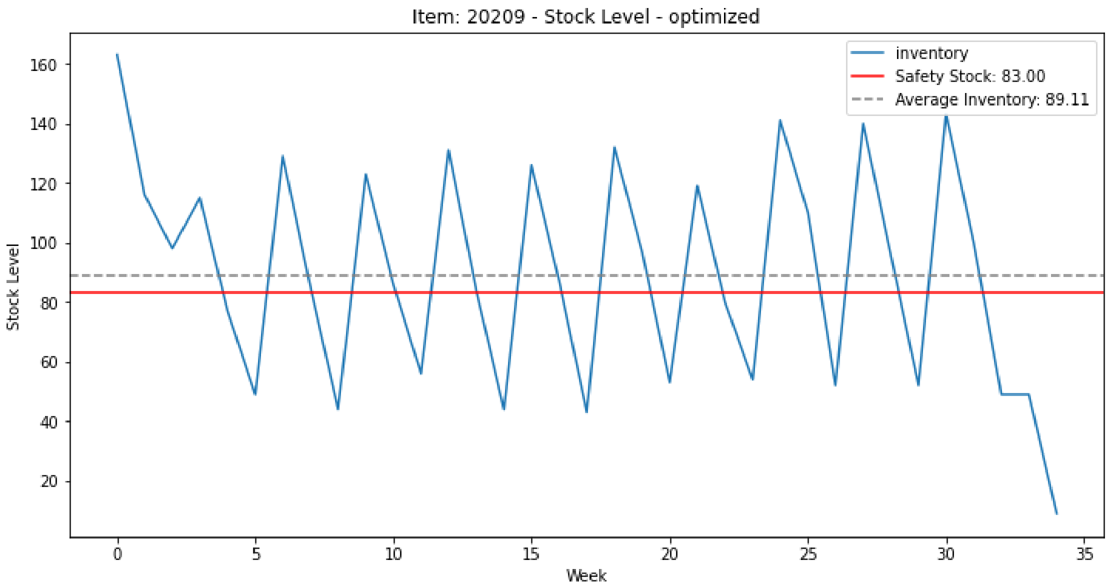
Tuned model results

	cutoff	MASE	RMSSE	MAE	RMSE	MAPE	SMAPE	R2
0	2022-12-02	0.6396	0.2960	0.7133	0.7756	3212632687887857.5000	2.0000	0.0000
1	2022-12-08	2.6277	1.6370	2.8722	4.2460	472569265497806.9375	1.3188	-0.4852
2	2022-12-14	2.6400	1.3012	3.0012	3.4475	0.5099	0.5310	-1.4311
Mean	NaT	1.9691	1.0781	2.1956	2.8230	1228400651128555.0000	1.2833	-0.6387
SD	NaT	0.9401	0.5697	1.0494	1.4840	1416265749415168.0000	0.6002	0.5942

7.1.1. Replenishment Strategy

Taking item_id 20109 as example, compare the MOQ strategy with our ILP optimiser result on history demand. The average inventory level is lower (89 V.S. 109 => 18%)





7.2 What Went Well

With reference to design goals of this application, the following requirements were satisfied as detailed below.

1. **Decision Automation:** The application can generate re-order point recommendations to user based utilizing rule-based and machine reasoning techniques to enable decisive supply chain management based on real time data, such as to prevent restock and overstock situations.
2. **Knowledge Discovery & (Big) Data Mining Techniques:** Algorithms in the cognitive system can analyse historical sales data to identify patterns in product movement relative to the season or period. This new-found information in turn is utilized to forecast future demand, upon which the demand-supply comparison is drawn to generate restock proposals to end user.
3. **Business Resource Optimization:** The same algorithms that forecast the required inventory level to meet the demand also consider the minimal safety stock values as dictated by the user. The ability of the system to consider multiple variables will minimize the holding value and cost cause by overstocking.
4. **Interactive UI:** ReTool serves as a user-friendly environment that proactively alerts users on re-order points and also a dashboard display key KPIs in a visually interactive graphic to comprehend the inventory health.

In addition to meeting the project requirements, the application offers additional benefits that have business value to the end user.

1. **Inventory/Order Management:** Users may utilize ReplenishNow as a stock take for their current inventory by regularly updating the SKU quantity, and also create or modify existing metadata for individual SKU. Users on the procurement and sales steams may also track their customer and supplier orders using the same platform. This is also more efficient that maintaining a separate ERP database such as SAP, Oracle or Excel spreadsheet.

7.3 What Can Be Improved

Forecasting:

1. enhance modelling by investigate demand patterns and routing to a smaller model pool for selection, evaluation, fine tuning and blender.
2. Include external factors and apply multivariate models; check SKU relationships
3. Introduce special models, and Machine learning model to outperform the accuracy

Replenishment:

1. Provide different strategies based on SKU demand patterns
2. Fully integrate optimiser
3. Build simulator to evaluation strategy
4. Implement A/B test on different methods and different input

UI:

1. Advanced interface: like speech assistant
2. Enhance All-in-one visualization dashboard for planner
3. Generate seasonal inventory health reports

System wise:

1. End-to-end data pipeline to bring various data source onboard
2. Standardize the I/O interface to make sub-modules interchangeable
3. Fully automate the backend functions with job scheduler and message queue

8.0 Conclusion

ReplenishNow is an application intended as a bespoke recommendation system for SMEs that utilizes artificial intelligence to suggest time-critical replenishment decisions to achieve an optimal inventory volume that prevents both stock out and overstock situations.

Evaluating the performance of the intelligent model against baseline forecasting systems, we see a quantitative improvement on two fronts. First is the inventory forecasting accuracy, which was closer to the actual sales by 4.4%. Secondly is the inventory optimization to reduce present stock volume, which had improved by 18.0%.

Qualitatively, the user interface and methods of data entry and extraction are intuitive and simple, with additional inventory and sales management functions beyond the original scope for replenishment forecasting. Users are also given the opportunity to modify the application as they see fit, with minimal coding experience.

ReplenishNow can fulfill objectives to recommend restock strategies to companies to reduce costs that improves upon existing baseline methodologies.

9.0 References

Research Paper

A hybrid method of Exponential Smoothing and Recurrent Neural Networks for time series forecasting Slawek Smyla,∗ aUber Technologies, 555 Market St, 94104, San Francisco, CA, USA

Boylan, J. E., Syntetos, A. A., & Karakostas, G. C. (2008). Classification for forecasting and stock control: a case study. Journal of the operational research society, 59(4), 473-481.

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, *109*, 1997-2028.

Books

Zipkin Paul H., Foundations of Inventory Management, Boston: McGraw Hill, 2000, ISBN 0-256-11379-3

Porteus, Evan L. Foundations of Stochastic Inventory Theory. Stanford, CA: Stanford University Press, 2002. ISBN 0-8047-4399-1

Website:

Safety stock - Wikipedia
Integer programming - Wikipedia

Demand classification: why forecastability matters - frepple APS

[Safety Stocks: Beware of Formulas – Michel Baudin's Blog]

[Top 3 Most Common Inventory Control Policies- Smart Software (smartcorp.com)]

[How to improve time series forecasting accuracy with cross-validation? (analyticsindiamag.com)]

10.0 Personal Reflection

10.1 Benjamin Lim

Reflecting on my journey through the university course on intelligent systems, machine reasoning, and machine learning, I am filled with a profound sense of growth and accomplishment. This educational experience has been a transformative one, broadening my understanding of the fascinating field of artificial intelligence and its potential applications in my work.

Initially, as I embarked on this course, I possessed a basic understanding of intelligent systems and their role in modern technology. However, as I delved deeper into the curriculum, I realized the vast extent of possibilities that machine reasoning and machine learning offer. I was exposed to various algorithms, techniques, and models that enable machines to learn from data, reason, and make informed decisions. The intricate world of neural networks, deep learning, and reinforcement learning unfolded before my eyes, captivating my curiosity and igniting a passion within me.

One of the most remarkable aspects of this course was its practical approach. Through hands-on projects and real-world case studies, I gained invaluable experience in applying intelligent systems and machine learning techniques to solve complex problems. I vividly recall the excitement and satisfaction of seeing algorithms I had implemented generate accurate predictions, classify data, and even recognize patterns. These moments validated my efforts and inspired me to explore further possibilities.

Moreover, the course also emphasized the ethical considerations surrounding the use of intelligent systems. I became acutely aware of the potential impact of AI on society, including issues of privacy, bias, and fairness. It compelled me to critically analyze the ethical implications of deploying AI solutions in my work. This newfound awareness has equipped me to approach AI development responsibly and to advocate for ethical practices within my professional domain.

10.2 Bianca Cao

As I contemplate the relevance of this course to my work, I am filled with anticipation. The knowledge and skills I have acquired in intelligent systems and machine learning are poised to revolutionize my professional endeavors. With this newfound expertise, I can envision countless applications in my field, where AI can streamline processes, enhance decision-making, and uncover valuable insights.

For instance, in my work as a financial analyst, intelligent systems can help me identify patterns in vast datasets, enabling more accurate predictions and risk assessments. Machine reasoning techniques can assist in uncovering hidden correlations and optimizing investment strategies. Additionally, I can leverage machine learning algorithms to automate routine tasks, freeing up time for more strategic analysis and fostering innovation within my organization.

Beyond my immediate work, I recognize the broader implications of intelligent systems and machine learning across industries. The potential to revolutionize healthcare, transportation, cybersecurity, and countless other fields is immense. Armed with the knowledge gained from this course, I feel a responsibility to contribute to the responsible and ethical advancement of AI in these areas.

In conclusion, completing the university course in intelligent systems, machine reasoning, and machine learning has been a profound and enlightening experience. It has expanded my horizons, equipped me with invaluable skills, and instilled a deep appreciation for the transformative power of AI. I am excited to embark on this new chapter of my professional journey, where I can harness these capabilities to drive innovation, make informed decisions, and shape a better future.