INTERNSHIP REPORT

on

PREDICTION OF BULLWHIP EFFECT IN A SUPPLY CHAIN MODEL

INDIAN INSTITUTE OF TECHNOLOGY (B.H.U) VARANASI

Under The Supervision Of

Dr. Lakshya

Associate Professor, Department of Mechanical Engineering

Submitted By:

Aryan Kumar-21135029

ACKNOWLEDGEMENT

I would take this opportunity to express my profound gratitude and deep regards to Dr. Lakshya for giving me an opportunity to do my internship project under him, for his guidance and for giving the utmost freedom and flexibility throughout the progress of this project. His dedication to our intellectual growth and development has been truly exemplary, serving as a beacon of inspiration for me to strive for excellence in my academic pursuits.

ABSTRACT

For organisations to keep up with the increasing demand, the supply chain requires efficiency. However, due to the unpredictable nature of demand, the bullwhip effect is a highly prominent in all industries. The role of AI and machine learning has been investigated in managing the risk which is accompanied by the bullwhip. The purpose of project is to study the effectiveness of various machine learning models in the prediction of the bullwhip effect. The bullwhip effect was studied using the linear regression, KNN and XGBoost, Random Forest based on variables - Demand, Receive, Forecast, Net stock, Lead-Time-Delivery, Safety Stock, Order-up-to-Level, Order and cost.

Keywords: Bullwhip Effect, KNN algorithm, XGBoost, Supply Chain, Risk Management

CONTENTS

1. **CHAPTER 1 INTRODUCTION**
    1.1 What is Bullwhip Effect?
    1.2 Motivation
    1.3 Organisation of Report
2. **CHAPTER 2 LITERATURE REVIEW**
    2.1 Bullwhip Effect
    2.2 Causes of the bullwhip effect
    2.3 AI in bullwhip effect
    2.4 Research gaps
    2.5 Research objectives
3. **CHAPTER 2 MODELLING BULLWHIP**
    3.1 Ways of measuring bullwhip
        3.1.1 Operations research theory
        3.1.2 Filler theory
        3.1.3 Adhocracy
        3.1.4 Control theory
    3.2 Four stage supply chain simulation
        3.2.1 Simple 4-stage supply chain
        3.2.2 Calculations for the dataset generation
4. **CHAPTER 4 AI ON VIRTUAL DATA SET**
    4.1 Data preparation
    4.2 Data exploration
    4.3 Data cleaning
    4.4 Data processing
    4.5 Machine learning algorithms employed
        4.5.1 KNN
        4.5.2 Random Forest
        4.5.3 XGBoost
5. **CHAPTER 5 RESULTS**

5.1 Results

6. CHAPTER 6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

6.2 Future Scope

CHAPTER 1: INTRODUCTION

This chapter gives a brief introduction to the bullwhip effect and its origins. It also explains why it is important to study and predict this phenomenon. Also, the proposed work has been briefly described, along with the motivation of the authors behind this work. Lastly, the structure of this dissertation is briefly mentioned.

1.1 WHAT IS BULLWHIP EFFECT?

Bullwhip effect is a phenomenon observed in the supply chains whenever there is fluctuation in the demand of a commodity. As the fluctuations in the demand increase, the demand forecasting methods give inaccurate results to the upstream members of the supply chain. This erroneous forecasting causes upstream stages of supply chain to either underestimate or overestimate the demand and subsequently maintain inventory levels that are very high and very low. The fluctuations in the inventories, due to very high or very low demands, are progressively amplified as we move upstream in the supply chain. This progressive amplification of the "fluctuations", or more accurately variance, in the inventory level of upstream members of the supply chain due to fluctuations in the customer demand is called the bullwhip effect. This effect is certainly undesirable due to its negative effect on the inventory costs and extra load that it puts on a supply chain.

This effect was first observed by Jay Forrester in 1961. Later in the 90s, this effect became apparent in the supply chains of Proctor and Gamble (P&G) and Hewlett-Packard. The term bullwhip was also coined at this moment only.

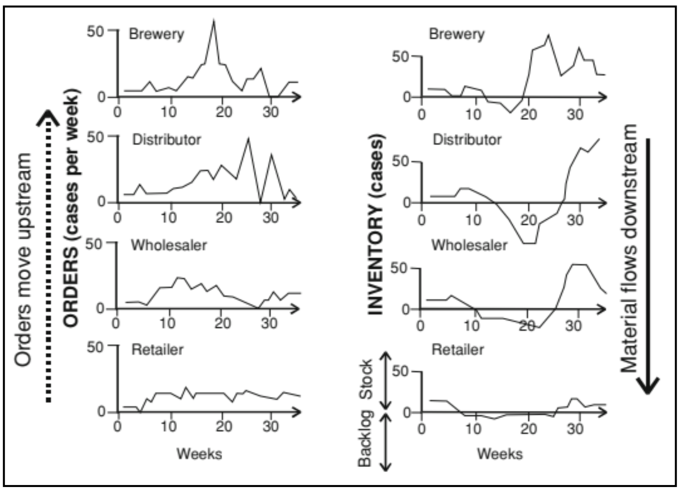


Figure 1.1: The bullwhip effect in a supply chain: Sterman (1989)

1.2 MOTIVATION

With the arrival of the coronavirus pandemic, organisations around the world suffered a major decrease in demand. The world moved into a recession (World Bank, 2021). Organisations realised the importance of having a lean system in order to prevent losses. The supply chains of various industries are moving towards a more resilient and lean structure. However, the threat posed by the bullwhip effect to a lean supply chain still looms large as information sharing is not viable in the highly competitive economies, moreover, poor demand forecasting methods contribute substantially to the bullwhip effect. (Prakash et al 2014) Implementation of artificial intelligence techniques in the supply chains of various industries can help forecast the demand with a higher accuracy and prevent the demand amplification phenomenon. (Amin et al 2008) These works are the basis of current research as use of AI algorithms in predicting the bullwhip effect has been explored in the current project.

This project aims to model a 4 stage supply chain using the principles laid down by Forrester (1961) and Sternman's beer game (1983). Using this model, the demand amplification is modelled using the variables such as Demand, Receive, Forecast, Net stock, Lead Time Delivery Safety Stock, Order up to Level, Order and cost.

CHAPTER 2: LITERATURE REVIEW

2.1 BULLWHIP EFFECT

Supply Chain management is defined as the strategic, systematic and coordination of business functions and the tactics followed between these functional groups, both within and across companies (Mentzer et al., 2012). The long term performance of individual companies and the entire supply chain is improved. One of the major problems faced in a supply chain is the undulating demand. The demand causes irregular movement of goods making it harder to accurately forecast sales, manage inventory and reduce service levels (Zotteri G, 2013). These fluctuations are known as the Bullwhip Effect. It was first studied by Forrester in 1958. It is observed when the variations in demand orders amplify and move across the supply from a local retailer to a large scale manufacturer (Forrester, 1958). The impact of the bullwhip effect is extremely detrimental to any organisation. Extensive research has been conducted to reduce the Bullwhip effect using methods such as Forecasting and Replenishment, Vendor Managed Inventory, Collaborative Planning etc. (Li et al 2013, Jaipuria et al 2014, Hussain et al 2012, Bhattacharya et al. 2011, Ciancimino et al 2012). Even though there have been major advancements in Information and Communication Technology. The Bullwhip effect still haunts the supply chain.

2.2 CAUSES OF THE BULLWHIP EFFECT

Significant effort has been put in research to find the underlying causes of this effect. Based on the literature, the main causes behind the bullwhip effect can be divided into two major categories namely behavioural and operational causes (Udenio et al., 2017). Behavioural (or psychological) causes include all the causes that arise due to improper decision made by humans responsible for running the supply chain. This can be due to various reasons such as lack of cognitive capability of the individual or stressful work atmosphere (Yang et al., 2021) Mostly, over or underestimation of the supply or demand capabilities is the underlying cause under this category. (Udenio et al., 2017)

The operational reasons for the bullwhip are mentioned below,

- Amplification of demand order variation (Lee et al 1997)
- Lack of coordination between stakeholders in the supply chain (Bhattacharya et al 2011)
- Lack of transparent information sharing (Bhattacharya et al 2011)
- Excess inventories throughout the supply chain to avoid stockout (Suki 2009)
- Inventory Stock-outs (Sun et al 2005)

The operational bullwhip effect can be addressed directly using various techniques. One of the techniques is Improved information sharing between the various stages of the supply chain. The amplification of demand order-variation occurs across the supply chain because of distortion, concealment and minimal sharing of information. The information regarding end user demand is used by the downstream members of the supply chain to estimate the inventory information. When inventory is accumulated, the inventory holding costs increase and chances of inventory obsolescence increase. (Li et al 2013)

Improved demand forecasting is another technique used to reduce the operational bullwhip effect. All the stakeholders of the supply chain can accurately align their production process with customer demand. Moreover, the accurate customer demand predictions can smoothen the fluctuations over time. (Disney et al 2003)

The operational bullwhip effect can be reduced when all organisation in the supply chain implement a robust replenishment policy. The BWE can be controlled with a policy which is devised with the collaboration of all stakeholders and free and open information sharing. (Kim et al 2010)

Lastly, operational bullwhip effects can be reduced by reducing the lead time between the stakeholders of a supply chain, and reduction of lead time. The product availability increases, excess stock decreases. The decision making process is optimised and the overall bullwhip effect decreases. (Li et al 2013)

2.3 ARTIFICIAL INTELLIGENCE AND THE BULLWHIP EFFECT

Prakash et al, 2014 suggested the integration of artificial intelligence in the supply chain to mitigate the bullwhip effect by directly monitoring the demand forecasting and order batching with the aid of machine learning algorithms. Prakash et al, 2014. Implemented artificial neural network in mitigating the bullwhip and compared the results with conventional forecasting methods, wherein artificial neural networks was found to significantly outperform the conventional forecasting methods, which further suggests the idea of exploring other machine learning algorithms for mitigating the bullwhip effect.

Donnel et al, 2006, proved in his study that the bullwhip effect can be significantly mitigated by integrating genetic algorithm by applying it to a beer distribution game, suggesting the capabilities of genetic algorithm to optimally determine order policy when facing irregular demand and lead time. Since genetic algorithm has the capability to find the optimal ordering policy for each member of the supply chain, it solves the problem of sales promotions which poses as one of the major problems associated with the bullwhip effect.

2.4 RESEARCH GAPS

After extensive literature review, we concluded that there aren't many papers which address the behavioural causes of the bullwhip effect. Mitigating the behavioural causes of the bullwhip effect could help us reduce the effect which plagues the supply chain. Even with a magnitude of papers used to reduce the operational and quantitative factors of the bullwhip effect, the problem still persists. This indicates the need for tools mitigate the behavioural causes of BWE. Moreover, there has been a lack of artificial intelligence tools because of the lack of research on the causes. (De Almedia et al 2015) As mentioned earlier, artificial intelligence can prove to be fruitful in order to reduce the bullwhip effect. To address both the gaps, A 4 stage supply chain model was used to simulate the bullwhip effect and its behavioural causes.

2.5 RESEARCH OBJECTIVES

The objective of the paper is to investigate the behavioural cause of the bullwhip effect. We aim to model a 4 stage supply chain model using the principle of industrial dynamics laid down by Forrester in 1961 and include the behavioural aspects of the bullwhip effect by modelling the same on Sternman's beer game (1983). The model would take customer demand as an input. The behaviour of the subsequent stages of the supply chain are dependent on the customer input. Consequently we model the bullwhip effect caused by the behaviour of the stakeholders in the various supply chains. The characteristics of every stage of the supply chain are monitored. Each stage of the supply chain is characterised variables such as Demand, Receive, Forecast, Net stock, Lead Time Delivery, Safety Stock, Order up to Level, Order and cost.

CHAPTER 3: MODELLING BULLWHIP

This chapter skims through various theories than can qualitatively or quantitatively gauge the bullwhip effect. Subsequently, this chapter explains the working of the 4 stage supply chain model developed to generate the dataset and the pre-processing operations performed on the generated dataset.

3.1 Ways of measuring bullwhip

Geary et al., in 2003, have reviewed a few theories that can help in gaining the knowledge about the current state of bullwhip in a supply chain. The ways in which these theories quantify, or analyse qualitatively, bullwhip effect are written in this section:

3.1.1 Operation research theory:

According to this theory, the bullwhip effect in a simple supply chain can be given by the formula in the equation 3.1 (Chen at al., 2000). Chen's simple supply chain, identical to this project, used simple moving average as the forecasting method to gauge future demand.

$$Bullwhip = \frac{\sigma_O^2}{\sigma_D^2} \qquad (3.1)$$

3.1.2 Filter theory:

In this theory, concept of size of "messages" and "disturbances" are used to determine the value of bullwhip while expressing the problem in the frequency domain. (Towill et al. 1994)

3.1.3 Ad-Hocacy

Sometimes the pragmatic approach of the experienced personnel managing the supply chain can be fruitful. Thus, this theory relies on the hunch of the experienced managers to re-engineering the supply chain, as and when the amplification arises.

3.1.4 Control theory

The quantification of the bullwhip in this theory is quite similar to the operation research theory. It uses an analogy between noise bandwidth and bullwhip, the ratio of variances of output and input (order and demand) is equated to the noise bandwidth (bullwhip) (Y.Z. Tsypkin, 1964).

$$Bullwhip = \frac{\sigma_O^2}{\sigma_D^2} = \frac{1}{\pi}\int_0^{\pi} |F^*(j\bar{w})|^2 d\bar{w} = \frac{W_n}{\pi} \qquad (3.2)$$

3.2 Four stage supply chain simulation

3.2.1 Simple 4 stage supply chain

The 4 stages are namely retailer, wholesaler, distributor and factory.

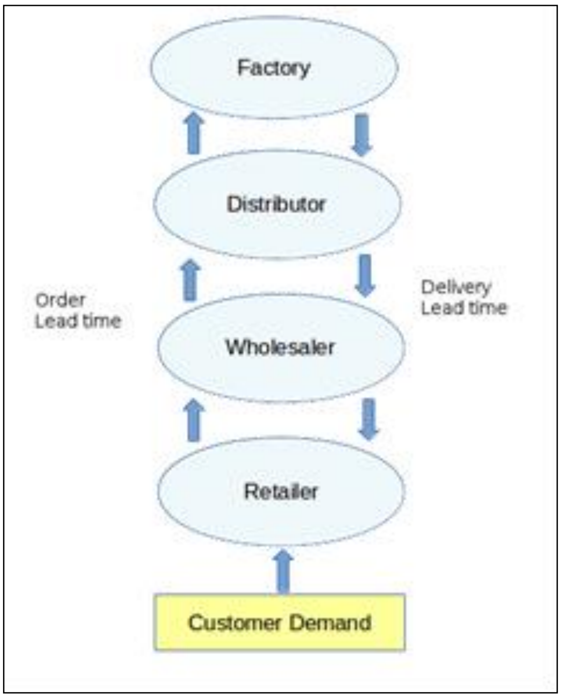


Figure 3.1: Basic 4 stage supply chain model

Each stage in the system is the customer of upstream and a supplier of downstream stage in the supply chain. For instance, the retailer observes customer demand and based on its current inventory situation places orders to its supplier, in this case the wholesaler. It receives orders made after a delivery lead time. The system is non-coordinated that is there is no information being shared between the stages of the supply chain. The decisions made by each stage are based on the simple moving average forecasting method. Each of the stages tries to predict the demand by averaging the last five demands of the stage downstream.

Time	Demand	Receive	Forecast	NS	LTD	SS	OUT	Order	Cost	Bullwhip
1	100	100	100	0	100	0	100	100	0	
2	100	100	100	0	100	0	100	100	0	
3	105	105	105	0	105	0	105	105	0	
4	105	105	105	0	105	0	105	105	0	
5	110	110	110	0	110	0	110	110	0	

Showing 1 to 5 of 5 entries

Previous 1 Ne

Table 3.1: Variables in the Supply chain Model

In the subsequent section it can be seen that no external factors affect the supply chain. Only the decisions made by each stage based on the demand forecasting contribute to bullwhip. Furthermore, the various variables that are mentioned in the table 1 except demand is calculated using the formulae mentioned contribute to bullwhip.

3.3.2 Calculations for the dataset generation

The table of data generated by model for each stage is shown in the table 1. As mentioned earlier, the demand of each stage is the order of the stage downstream. The other variables in the table are explained briefly in this section.

Net stock (NS) is defined here as a current inventory level. This is given by,

$$NS_t = NS_{t-1} + Received - Demand \qquad (3.3)$$

Safety stock (SS) is the amount of inventory that companies keep in order to protect themselves against stock out situations during lead time. The safety stock is given by the following formula:

$$SS = Z \times \sigma \times \sqrt{L} \qquad (3.4)$$

The σ is standard deviation of the demand. The symbol L is the Lead time, which is the number of days between the time an order is placed and the time it is received. The SS was rounded off to 2 decimal places. Z is the safety factor, which is a constant associated with the service level. Service level is the probability of not running out of stock during the next replenishment cycle.

The service level was taken to be 0.95 in the simulation. Let us see how service level and safety factor are related. Since service level is a probability, it can be expressed as the area under the curve of a standard normal distribution curve. So, safety factor can be defined as the z-score at which the probability of not running out of stock during the next replenishment cycle is equal to the service level. As an example, if Z=1.645 (approx.) then stock will not run out of stock 95 times out of 100.

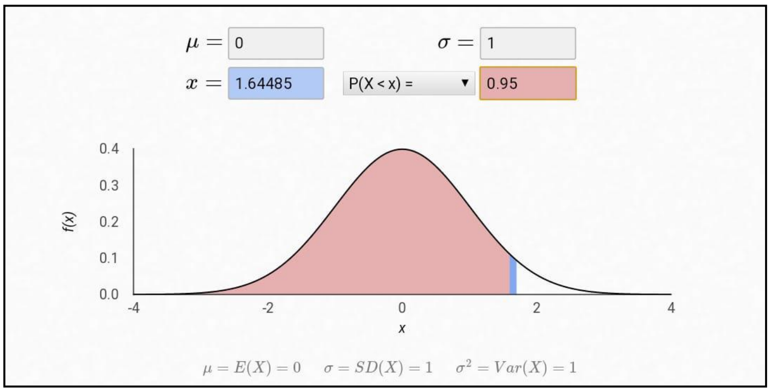


Figure 3.2: Service level in normal distribution

Lead time demand (LTD) is the average demand during lead time. Here, value of lead time was taken to be 1. Lead time demand is given by,

$$LTD = Forecast \times L \qquad (3.5)$$

Order Up to Level (OUT) is a replenishment policy. Each period companies review stock levels and place an order to bring its stock levels up to a target level.

$$OUT = LTD + SS \qquad (3.6)$$

Order (O), stocks ordered but not yet arrived, is given by,

$$O = OUT - NS \qquad (3.7)$$

To quantify bullwhip, the operation research model is used. In this model, the ratio of variances is commonly used in dispersion statistics to compare variability from various distributions. If the ratio is greater than 1, it is called as variance amplification. In other case when it is less than one, it is called as the isolation effect. (Campuzano et al. 2011)

Cost, it is the expense incurred in storing the product in inventory and back ordering. For calculating cost, there are two possible cases:

- In the case when the stock is not sufficient to fulfil the demand. It is calculated as:

$$Cost = Holding\ cost\ per\ piece \times Net\ stock, \quad if\ Net\ Stock > 0 \qquad (3.9)$$

- In all other cases,

Cost = Backordering cost per piece x Quantity of unfulfilled order *(3.10)*

For the purpose of the simulation, the back ordering and the holding cost are taken to be 0.5 and 2 units respectively. Also, no external factors other than service level, demand, lead time, number of time periods used for forecasting in simple moving average, affect the bullwhip in the simulation.

CHAPTER 4 AI ON VIRTUAL DATA SET

The data is generated using the bullwhip model above. To implement variability in the demand, the customer demand is simulated using a sinusoidal curve as shown in the figure 4.2.

4.1 Data preparation

In order for the data to be used by the algorithm, data pre-processing is required. This is done to ensure that the raw data is converted into a format that is manageable by the algorithm. Also, in this step all the instances of inconsistent data are removed to avoid errors in prediction and improve efficacy. For the project, the process of data pre-processing had three steps, namely Data Exploration, Data Cleaning, and Data Processing. These steps are described briefly in this section.

4.1 Data Exploration:

In this process, the initial dataset is first visualized by human before further processing it. This is done to check for any inaccuracies such as missing values, typographical errors etc. Apart from this, various visualization tools are also used to see identify patterns in the data and decided which variables must be utilized for the regression.

4.2 Data Cleaning:

In this process, all the unnecessary and incorrect data points are removed in order to increase the accuracy of the analysis. This is done to make the data more consistent with other data points. The data is also standardized in this part of the process only to maintain uniformity in the dataset.

4.3 Data Processing:

In this process, the pre-processed data is converted into a more interpretable form by the application of various algorithms. All the labelled data is encoded with a numerical value. The data table is scaled for the machine learning algorithms in this step only. The dataset is further split into training, testing and validation data sets.

The training data set is used to "train" the neural network, which is essentially the process of deciding the various weights to be used for the analysis. Successively, to check if these weights are selected properly, validation data set is utilized. The outputs of the algorithms are compared with the actual values of the validation dataset to gauge the accuracy of the dataset. Lastly, the test dataset is utilized to evaluate the efficacy of the model.

	Demand	Receive	Forecast	NS	LTD	SS	OUT	Order	Cost	Bullwhip
count	828.000000	828.000000	828.000000	828.000000	828.000000	828.000000	828.000000	828.000000	828.000000	828.000000
mean	124.528080	124.463104	123.346727	19.750000	123.346727	21.335761	144.682488	124.932488	25.747446	1.178901
std	74.783188	82.811118	71.455368	54.129313	71.455368	31.702924	87.044084	83.003476	49.577952	0.104387
min	-16.280000	-26.510000	-14.740000	-362.120000	-14.740000	0.900000	-11.470000	-26.510000	0.000000	1.000000
25%	65.467500	65.752500	65.422500	-8.312500	65.422500	8.320000	85.342500	65.752500	8.251250	1.120000
50%	125.675000	123.165000	124.950000	10.010000	124.950000	16.220000	144.255000	124.020000	19.340000	1.140000
75%	170.527500	171.540000	166.822500	42.920000	166.822500	23.492500	180.420000	172.192500	29.403750	1.200000
max	573.080000	832.820000	388.850000	534.760000	388.850000	346.470000	665.120000	832.820000	724.240000	1.760000

Table 4.1: Basic statistical details like percentile, mean, standard deviation etc. of the dataset

```
Demand       0
Receive      0
Forecast     0
NS           0
LTD          0
SS           0
OUT          0
Order        0
Cost         0
Bullwhip     0
dtype: int64
```

```
Demand       float64
Receive      float64
Forecast     float64
NS           float64
LTD          float64
SS           float64
OUT          float64
Order        float64
Cost         float64
Bullwhip     float64
dtype: object
```

Figure4.1 : A) Checking for missing values in each data point B) Exploring the data of each data point

	Demand	Receive	Forecast	NS	LTD	SS	OUT	Order	Cost	Bullwhip
0	111.0	110.00	104.0	-1.00	104.0	6.88	110.88	111.88	2.00	1.000
1	117.0	111.88	106.2	-6.12	106.2	7.30	113.50	119.62	12.24	1.098
2	123.0	119.62	109.6	-9.50	109.6	8.19	117.79	127.29	19.00	1.287
3	128.0	127.29	113.2	-10.21	113.2	11.42	124.62	134.83	20.42	1.405
4	133.0	134.83	117.8	-8.38	117.8	12.71	130.51	138.89	16.76	1.530

Table 4.2: First few rows of the dataset

	Demand	Receive	Forecast	NS	LTD	SS	OUT	Order	Cost	Bullwhip
0	1.079815	0.794282	1.471047	2.013201	1.471047	0.086524	0.904166	0.805220	0.013808	1.000
1	1.130718	0.805220	1.498303	1.984658	1.498303	0.092601	0.923528	0.850255	0.084502	1.098
2	1.181621	0.850255	1.540425	1.965815	1.540425	0.105478	0.955231	0.894883	0.131172	1.287
3	1.224040	0.894883	1.585024	1.961857	1.585024	0.152212	1.005705	0.938755	0.140975	1.405
4	1.266459	0.938755	1.642013	1.972059	1.642013	0.170877	1.049232	0.962378	0.115708	1.530

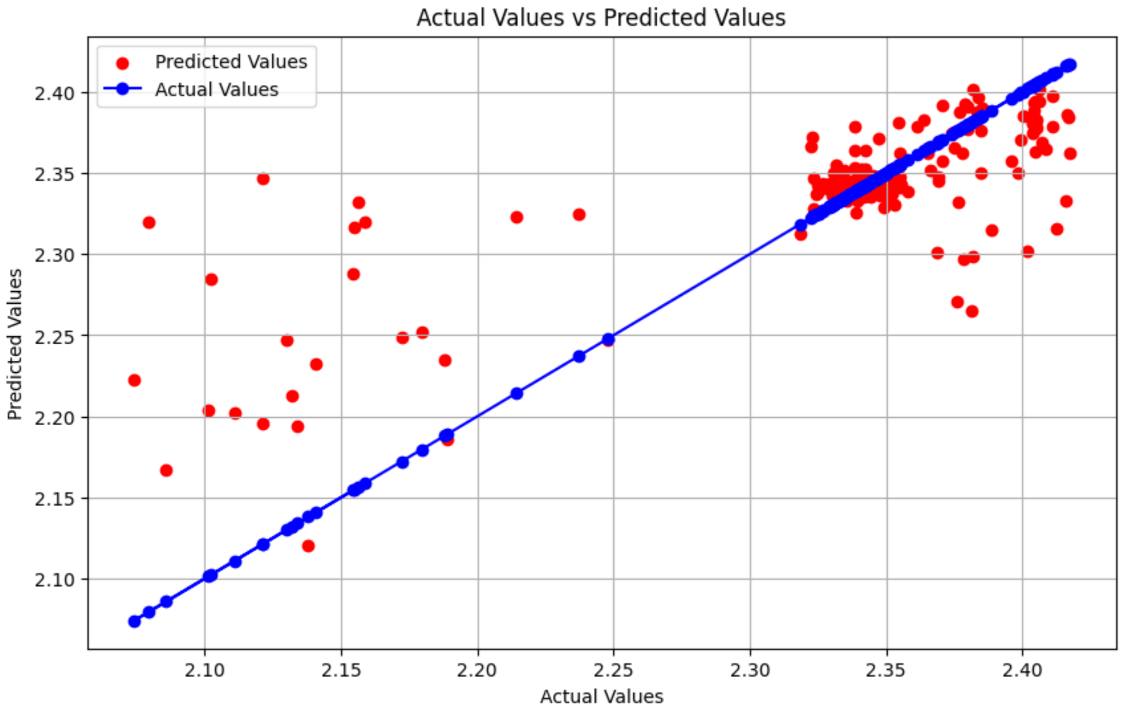
Table 4.3: Dataset after processing and scaling the data point

Figure 4.2: *4.2a represents Customer Demand Signal in the Bullwhip Game; 4.2b represents Customer Demand vs Retailer Demand; 4.2c represents Customer Demand vs Wholesaler Demand; 4.2d represents Customer Demand vs Distributor Demand; 4.2e represents Customer Demand vs Wholesaler Demand; 4.2f represents Customer Demand vs aggregate Demand*

4.5 Machine Learning algorithms employed

4.5.1 K-Nearest Neighbours

K-Nearest Neighbours is an instance-based machine learning algorithm which stores history of all available cases and uses the information to classify new cases based on the criteria of similarity measure or distance function, wherein the K value represents the number of neighbours, Javier et al, 2009. The intuitive and simple-to-implement nature of the algorithm, makes it a plausible candidate for integration in the supply chain. Non-parametric nature of KNN allows the algorithm to work for demand fluctuations of any type since there are no preliminary assumptions to be met. KNN, being a lazy learner, responds quickly to changes in the input during real time use. However, KNN suffers from the curse of dimensionality and thus would fail to perform well where input variables for bullwhip effect are too many. Moreover, KNN is a slow algorithm wherein an imbalanced data set can cause problems, making it highly sensitive to outliers.



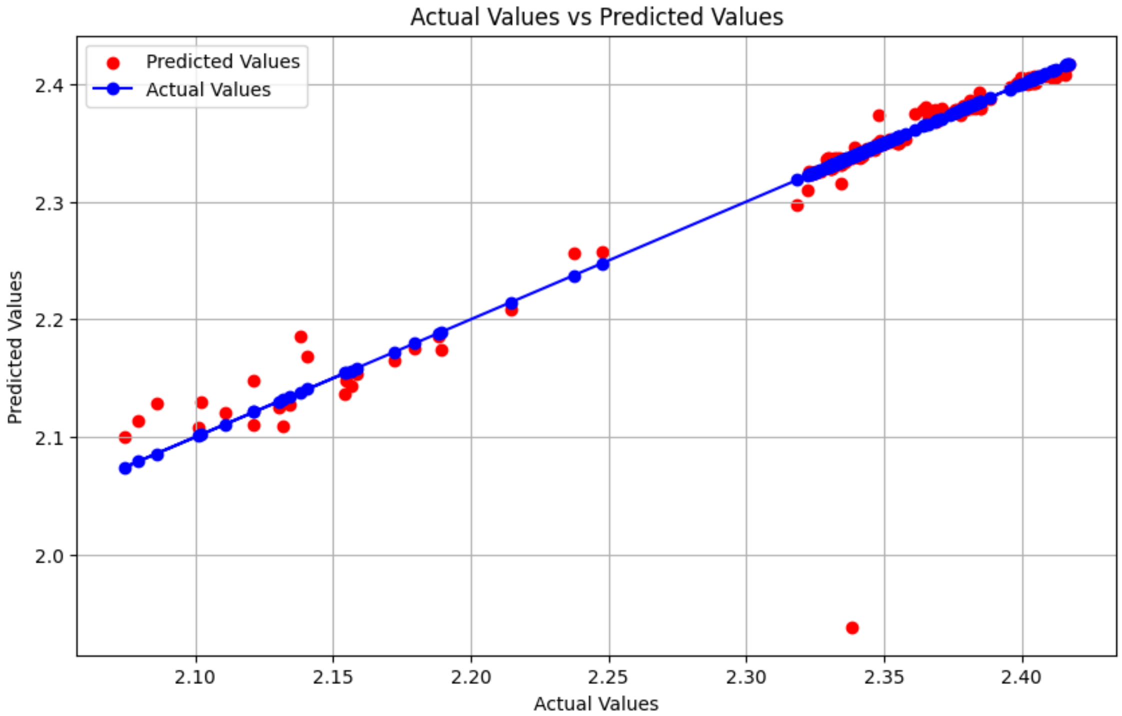
The low MSE and MAE indicate that the KNN model's predictions are quite close to the actual values.

The R^2 value of 0.5603 suggests that the model explains a moderate portion of the variance in the data, but there is still room for improvement.

The KNN model appears to perform reasonably well, with small errors in predictions and a moderate level of explained variance. However, there might be potential to enhance the model's accuracy, possibly by fine-tuning hyperparameters, using additional features, or exploring different modeling techniques.

4.5.2 Random Forest Regressor

The Random Forest Regressor is a powerful ensemble learning method used for regression tasks. It operates by constructing multiple decision trees during training and outputting the average of their predictions for regression tasks. This technique helps to improve the model's accuracy and robustness compared to individual decision trees.. Random Forest, unlike K-Nearest Neighbours, performs well even with missing values. Low computation of decision trees make it suitable for oceans of data as seen in several industries.

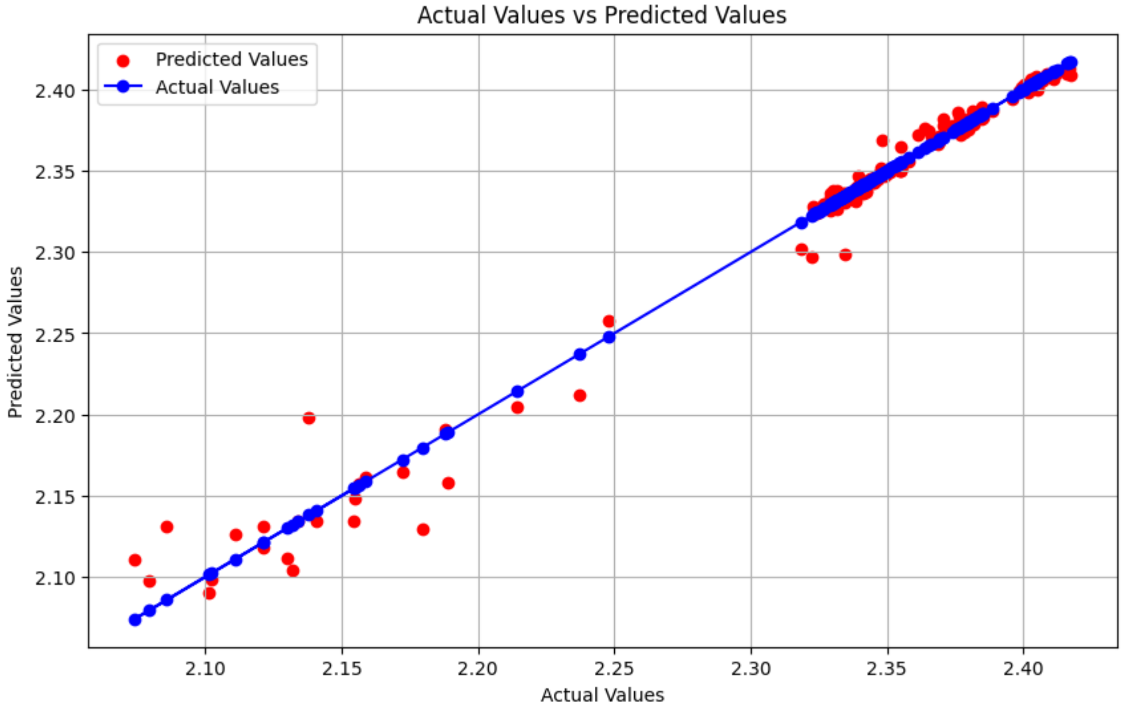


4.5.3 XG BOOST

XGBoost (Extreme Gradient Boosting) is a powerful and efficient open-source implementation of the gradient boosting framework. It has become popular due to its performance and speed in various machine learning competitions and real-world applications. Here are some key details about XGBoost:

Key Features:

1. Gradient Boosting Algorithm:
   - XGBoost is based on the gradient boosting framework, which builds an ensemble of decision trees sequentially. Each tree tries to correct the errors of the previous one by minimizing a specified loss function.
2. Regularization:
   - XGBoost includes L1 (Lasso) and L2 (Ridge) regularization to prevent overfitting, making the model more robust and generalizable.
3. Parallelization:
   - XGBoost supports parallel processing, which speeds up the training process. It can leverage multiple CPU cores and distributed computing for faster computation.
4. Tree Pruning:
   - Instead of using a greedy algorithm to grow the trees, XGBoost employs a technique called "max depth" pruning. It starts by growing the tree to a maximum depth and then prunes back to a predefined depth, which helps in reducing overfitting and improving performance.
5. Handling Missing Values:
   - XGBoost can handle missing values internally by learning the best direction to go when a value is missing, rather than requiring pre-imputation.
6. Custom Loss Functions:
   - Users can define custom loss functions to tailor the boosting process to specific problem requirements.
7. Scalability:
   - XGBoost is designed to be highly scalable. It can handle large datasets and can be deployed on distributed systems such as Hadoop and Spark.



CHAPTER 5 RESULTS

Here's a detailed comparison of the three algorithms based on the provided performance metrics:

Performance Metrics:

1. **Mean Squared Error (MSE)**:
   - **XGBoost**: .00009268254997124444
   - **Random Forest Regressor**: 0.00086793308841459 24
   - **KNN**: 0.002420691655278982
2. **R-squared (R^2)**:
   - **XGBoost**: 0.9831641633223367
   - **Random Forest Regressor**: 0.8423394724441486
   - **KNN**: 0.5602800163795265

3. **Mean Absolute Error (MAE)**:
   - **XGBoost**: 0.004831535978783734
   - **Random Forest Regressor**: 0.006282895962217262
   - **KNN**: 0.02751889365002037

   Analysis:

1. **Mean Squared Error (MSE)**:
   - **XGBoost** has the lowest MSE, indicating that it has the smallest average squared difference between predicted and actual values. This suggests that XGBoost's predictions are the closest to the actual values among the three models.
   - **Random Forest Regressor** has a higher MSE than XGBoost but is still significantly lower than KNN.
   - **KNN** has the highest MSE, suggesting its predictions are less accurate compared to the other two models.
2. **R-squared (R^2)**:
   - **XGBoost** has the highest R^2 score of 0.9831641633223367, indicating that it explains approximately 98.32% of the variance in the data. This suggests a very high level of accuracy.
   - **Random Forest Regressor** has a decent R^2 score of 0.8423394724441486, explaining about 84.23% of the variance, which is good but not as high as XGBoost.
   - **KNN** has the lowest R^2 score of 0.5602800163795265, explaining only 56.03% of the variance. This indicates a moderate level of accuracy, but much lower than the other two models.
3. **Mean Absolute Error (MAE)**:

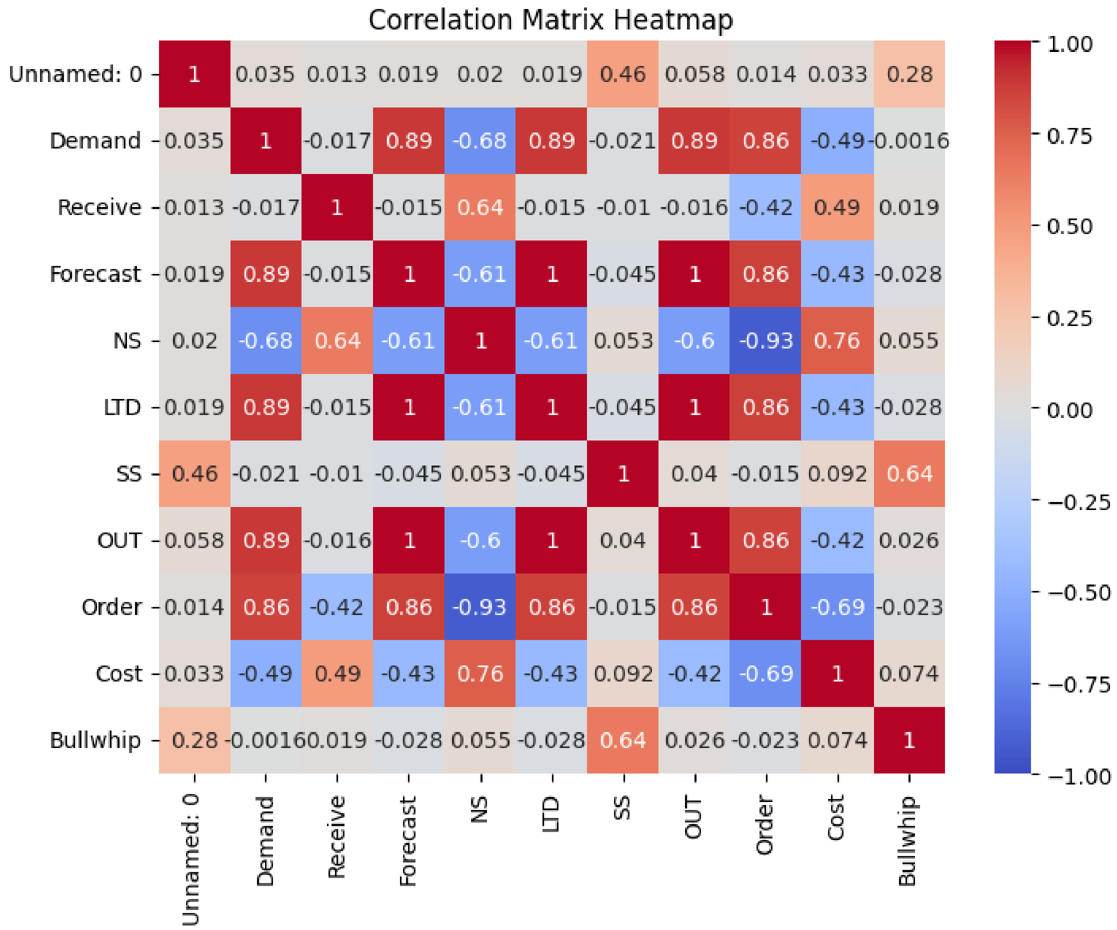
- **XGBoost** has the lowest MAE, indicating the smallest average absolute errors between predicted and actual values. This again suggests that XGBoost's predictions are the most accurate.
- **Random Forest Regressor** has a higher MAE than XGBoost but lower than KNN, indicating moderate prediction errors.
- **KNN** has the highest MAE, indicating the largest average errors, suggesting less accurate predictions compared to the other two models.

Conclusion:

- **XGBoost** clearly outperforms the other two algorithms in all three metrics (MSE, R^2, MAE), indicating that it provides the most accurate and reliable predictions. Its high R^2 score and low error metrics suggest that it has the best fit to the data and makes the most precise predictions.
- **Random Forest Regressor** performs well but is not as accurate as XGBoost. It has a reasonably high R^2 score and lower error metrics than KNN, making it a good choice but not the best in this comparison.
- **KNN** performs the worst among the three algorithms, with the highest error metrics and the lowest R^2 score. It indicates that KNN's predictions are less accurate and it explains less variance in the data compared to XGBoost and Random Forest.

Overall, **XGBoost** is the best performing model in this comparison, followed by **Random Forest Regressor**, and **KNN** is the least accurate.

A correlation Matrix was also constructed to understand the correlation between bullwhip and the various inputs. A multi-correlation matrix was obtained.


Correlation Matrix Heatmap
Unnamed: 0
Demand
Receive
Forecast
NS
LTD
SS
OUT
Order
Cost
Bullwhip
1 0.035 0.013 0.019 0.02 0.019 0.46 0.058 0.014 0.033 0.28
0.035 1 -0.017 0.89 -0.68 0.89 -0.021 0.89 0.86 -0.49 -0.0016
0.013 -0.017 1 -0.015 0.64 -0.015 -0.01 -0.016 -0.42 0.49 0.019
0.019 0.89 -0.015 1 -0.61 1 -0.045 1 0.86 -0.43 -0.028
0.02 -0.68 0.64 -0.61 1 -0.61 0.053 -0.6 -0.93 0.76 0.055
0.019 0.89 -0.015 1 -0.61 1 -0.045 1 0.86 -0.43 -0.028
0.46 -0.021 -0.01 -0.045 0.053 -0.045 1 0.04 -0.015 0.092 0.64
0.058 0.89 -0.016 1 -0.6 1 0.04 1 0.86 -0.42 0.026
0.014 0.86 -0.42 0.86 -0.93 0.86 -0.015 0.86 1 -0.69 -0.023
0.033 -0.49 0.49 -0.43 0.76 -0.43 0.092 -0.42 -0.69 1 0.074
0.28 -0.0016 0.019 -0.028 0.055 -0.028 0.64 0.026 -0.023 0.074 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

CHAPTER 6 CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

The paper's objective was to model a 4-Stage supply chain, measure the bullwhip effect, understand its underlying causes and possible solutions and model a prediction artificial intelligence algorithm. The Bullwhip effect in supply chains is majorly caused due to demand forecasting, price fluctuation, gaming during shortage and order batching. One proposed solution to the problem can be integration of the entire supply chain with time constraint information sharing. However this is not favorable as the various stages of the supply chain have diverse organizational goals. More over in competitive markets, data theft is a serious concern. New demand forecasting techniques such as ANN, Fuzzy Logic, and Genetic algorithms can be used to forecast the demand with higher accuracy in the absence of supply chain integration. The artificial intelligence technique, artificial neural network forecasts accuracy is better than the traditional approaches (Prakash et al 2014). We concluded that the advanced demand forecasting techniques which use regressors and perform poorly with multi collinearity must not be used to predict the bullwhip effect. Algorithms which use the projection of the data points such as KNN algorithms are effective for its prediction (Rinkaj Goyal et al 2014).

6.2 FUTURE SCOPE

Future research could be aimed at exploring algorithms which handle high degrees of collinearity. Moreover, research with a larger dataset can also be conducted to verify the results of this paper. The advancement in e-commerce and e-business through internet has made the supply chain system more dynamic and requires JIT approach. Further work can be carried out to investigate how e-commerce affects the demand amplification and create satisfied customers.

Appendix 1

Code for implemented Machine Learning Algorithms

aryan-rohit/BullWhip-Effect-prediction-using-ML (github.com)

REFERENCES

1. Amin-Naseri MR, Tabur BR. Neural network approach to lumpy demand forecasting for spare parts in process industries. Proceedings of the IEEE International Conference on Computer and Communication Engineering. May 13–15, 2008, Kuala Lumpur, Malaysia. 1378–1382p. 978-1-4244-1692-9/08,
2. Bhattacharya R, Bandyopadhyay S (2011) A review of the causes of bullwhip effect in a supply chain. Int J Adv Manuf Technol 54(9–12): 1245–1261
3. Campuzano F., Mula J. (2011) Bullwhip Effect in Supply Chains. In: Supply Chain Simulation. Springer, London. https://doi.org/10.1007/978-0-85729-719-8_3.
4. Chen, F., Drezner, Z., Ryan, J., & Simchi-Levi, D. (2000). Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information. *Management Science, 46*(3), 436-443.
5. Ciancimino E, Cannella S, Bruccoleri M, Framinan JM (2012) On the bullwhip avoidance phase: the synchronized supply chain. Eur J Oper Res 221(1):49–63
6. D.R. Towill, A. Del Vecchio, Application of filter theory to the study of supply chain dynamics. Prod. Plan. Control. **5**, 82–96 (1994)
7. Disney SM, Towill DR (2003) Vendor-managed inventory and bull- whip reduction in a two-level supply chain. Int J Oper Prod Manag 23(6):625–651
8. Forrester JW (1958) Industrial dynamics—a major breakthrough for decision makers. Harv Bus Rev 36(4):37–66
9. Forrester JW. Industrial Dynamics. The M.I.T. Press. 1961.
10. Hau L. Lee, V. Padmanabhan, Seungjin Whang, Information Distortion in a Supply Chain: The Bullwhip Effect. Management Science 43 (4) 546-558 [doi: https://doi.org/10.1287/mnsc.43.4.546 }
11. Hussain M, Drake PR, Lee DM (2012) Quantifying the impact of a supply chain's design parameters on the bullwhip effect using simu- lation and Taguchi design of experiments. Int J Phys Distrib Logist Manag 42(10):947–968
12. Il-Gyo Chong, Chi-Hyuck Jun, Performance of some variable selection methods when multicollinearity is present, Chemometrics and Intelligent Laboratory