
Faculty of Social Sciences, Business and Economics, and Law (FSEJ)

November 2023

# Master's Thesis

## LEVERAGING ADVANCED ANALYTICS FOR BACKORDER PREDICTION AND OPTIMIZATION OF BUSINESS OPERATIONS IN THE SUPPLY CHAIN

Jobair Islam / Student ID: 2002099

Master's Degree Program in
Governance of Digitalization

ÅBO AKADEMI UNIVERSITY

| | |
|---|---|
| **Faculty of :**<br>Social Sciences, Business and Economics, and Law (FSEJ) | **Type of work:**<br>Master's Thesis |
| **Degree Program:**<br>MDP in Governance of Digitalization | **Language:**<br>English |
| **Authored by:**<br>Jobair Islam | **Supervised by:**<br>Prof. Jozsef Mezei |

**Title of thesis:**

Leveraging Advanced Analytics for Backorder Prediction and Optimization of Business Operations in the Supply Chain

**Abstract:**

Businesses can unlock valuable insights by leveraging advanced analytics techniques to optimize supply chain processes and address backorders. Backorders occur when a customer order cannot be fulfilled immediately due to lack of available supply. Root causes of backorders can range from supply chain complications and manufacturing miscalculations to logistical challenges. While a surge in demand might initially seem beneficial, backorders come with inherent costs, leading to supply chain disruptions, dissatisfied customers, and lost sales. This research aimed to assess the efficacy of predictive analytics in detecting early backorder signs and to understand how parameter tuning influences the performance of these predictive models. The foundation of this study was laid through an exhaustive literature review. In-depth Exploratory Data Analytics/ EDA was utilized to investigate datasets, followed by rigorous preprocessing steps, including data cleaning, feature engineering, scaling, and resampling. Machine learning models were subsequently trained, tuned, and assessed using appropriate evaluation metrics. Findings from this research underscored the value of predictive analytics in early backorder identification. They also offered a comparative analysis of machine learning algorithms and highlighted the significance of parameter tuning. Additionally, they established the necessity of multi-metric evaluations for imbalanced datasets. Thus, the study has provided a fundamental framework that can serve as a basis for future research endeavors.

**Keywords:**

Advanced analytics, AI and ML, Backorder prediction, EDA, Imblanced dataset, Linear Regression, Parameter tuning, Predictive modeling, Random Forest, RUS, SMOTENC, Supply chain, XGBoost.

| | |
|---|---|
| **Date:** 13.11.2023 | **Number of pages:** 143 + V |
| 25 figures, 14 tables, 13 appendices | |

# Table of Contents


**1 INTRODUCTION ..... 1**

**1.1 Background and motivation ..... 1**

**1.2 Problem statement ..... 3**

**1.3 Research objectives ..... 4**

**1.4 Research questions ..... 4**

**1.5 Significance and contributions of the study ..... 5**

**1.6 Limitations and delimitations of the work ..... 6**

**1.7 Structure of the paper ..... 7**

**2 EXAMINING LITERATURE AND THEORETICAL CONSTRUCTS ..... 8**

**2.1 Overview of supply chain management and its challenges ..... 8**

**2.2 Bullwhip effect and Backorder prediction in supply chain management ..... 11**

**2.3 Challenges and complexities associated with backorder prediction ..... 13**

**2.4 Understanding machine learning and its applications ..... 14**

*2.4.1* Machine learning- what it means ..... 15

2.4.1.1 Machine learning approaches: supervised versus unsupervised ..... 16

2.4.1.2 Machine learning process at a glance ..... 16

2.4.1.3 Data in machine learning ..... 19

*2.4.2* Machine learning use cases and the power of prediction ..... 21

**2.5 Existing methods and techniques for backorder prediction ..... 22**

*2.5.1* Traditional approach versus modern approach in backorder prediction ..... 23

2.5.1.1 Traditional approach ..... 23

2.5.1.2 Modern/ML approach ..... 25

*2.5.1.2.1 Enhancement in backorder prediction ..... 26*

*2.5.2* Modern/ML approach is better ..... 27

*2.5.3* Recent relevant studies at a glance ..... 29

**2.6 Rationale and uniqueness of the paper ..... 33**

**3 RESEARCH DESIGN AND METHODOLOGY ..... 36**

**3.1 Methodological approach ..... 37**

**3.2 Ethical considerations ..... 38**

**3.3 Tools/software and libraries ..... 38**

**3.4 Data collection ..... 39**

*3.4.1* Dataset overview ..... 39

**3.5 Data preprocessing techniques ..... 42**
*3.5.1* Data descriptives- key insights ..... 44
*3.5.2* Handling loaded dummy values and missing values ..... 45
*3.5.3* Outlier detection and handling ..... 48
*3.5.4* Binarization of the categorical features ..... 50
*3.5.5* Feature selection process for ML models ..... 51
3.5.5.1 Cardinality checking and dropping the "sku" column ..... 52
3.5.5.2 Bivariate analysis of categorical features with the target column ..... 52
3.5.5.3 Chi-squared test ..... 53
3.5.5.4 Bivariate analysis of numerical features with the target column ..... 55
3.5.5.5 Correlation matrix ..... 57
3.5.5.6 SelectKbest with Mutual Information / MI Score ..... 60
3.5.5.7 Summary of all observations / Finalizing the features for ML model ..... 62
*3.5.6* Handling duplicates and resetting indices ..... 62
*3.5.7* Scaling the dataset ..... 63
*3.5.8* Handling imbalanced training set / Resampling training set ..... 64

**3.6 Model building ..... 66**
*3.6.1* Model selection ..... 66
*3.6.2* Model training ..... 67
3.6.2.1 Logistic Regression ..... 70
3.6.2.2 Decision/Classification Trees ..... 73
3.6.2.3 Random Forest ..... 76
3.6.2.4 XGBOOST ..... 79

**3.7 Model evaluation metrics ..... 83**

**4 EXPERIMENTAL RESULTS AND ANALYSIS ..... 86**

**4.1 Base models evaluation ..... 87**

**4.2 Hyperparameter tuned models evaluation ..... 90**

**4.3 Comparative analysis: base Vs. tuned models ..... 93**

**4.4 Finding the best models ..... 95**
*4.4.1* With "ORIGINAL" sampling approach ..... 95
*4.4.2* With "RUS" sampling approach ..... 96
*4.4.3* With "SMOTENC" sampling approach ..... 97

**4.5 Elite model showdown: Best of current study Vs. best of state-of-the-art ..... 98**

**4.6 Model criticism, usability, and further discussion ..... 100**

**5 CONCLUSION AND FUTURE WORK ..... 102**

**5.1 Recapitulation of the study's outcomes ..... 103**

**5.2 Reflection on contributions and limitations ..... 104**
*5.2.1* Contributions revisited ..... 105
*5.2.2* Reflecting on limitations ..... 106

**5.3 Personal research challenges ..... 107**

**5.4 Addressing practical implementation challenges in SCM ..... 108**

**5.5 Future directions ..... 110**

**REFERENCES** ..... **113**

**LIST OF FIGURES** ..... **128**

**LIST OF TABLES** ..... **129**

**LIST OF APPENDICES** ..... **130**

# 1 Introduction

The Danish fairy tale, authored by the famous writer Hans Christian Andersen, suggests “The ugly duckling did not realize that he was a swan until he came in contact with swans, saw his reflection in water, and figured out that he was himself a swan, too” (Bates, 2010, Chapter 2).

The aforementioned abstract has the connotation of how data and information can help reveal hidden insights. Similarly, within the complex world of supply chain management, hidden insights and opportunities can be uncovered through the power of data and advanced analytics. Much in the same manner that the ugly duckling discovered his true nature when exposed to new information, businesses can also unlock valuable insights by leveraging advanced analytics techniques to optimize their operations in the supply chain.

## 1.1 Background and motivation

Supply chain management, SCM, encompasses the entire journey of transforming raw materials into finished goods. It ensures the availability of essential products, such as food, health items, and various commodities that facilitate daily human lives, including work, travel, and entertainment. Without the supply chain, these goods would not be accessible to the users (University of Maryland, n.d.).

Morana (2013) claims that, in today’s business landscape, the crucial role of SCM is well acknowledged in the success of companies across various sectors. With increasing external pressures, globalization, and competition, SCM has become essential for optimizing processes and streamlining production and delivery cycles. By embracing SCM practices paired with technological and organizational innovations, businesses can proactively respond to the dynamic factors and challenges (economic, environmental, or social constraints) present in the market.

In accordance with IBM (n.d.a), supply chain management has undergone a significant transformation, shifting from a focus on physical assets to the management of data, services, and comprehensive product solutions. Today's supply chains have a far-reaching impact on various aspects, including product quality, delivery efficiency, costs, customer experience,

and overall profitability. Notably, the volume of data available to supply chains has multiplied in recent years. By 2017, supply chains had already gained access to 50 times more data than they possessed a mere five years earlier, in 2012. However, a considerable portion of this data remains underutilized, resulting in missed opportunities. Thus, crucial time-sensitive data, such as weather information, labor shortages, political developments, and demand fluctuations, often goes unnoticed. Some studies (Luther, 2022; Kathuria, n.d.) find that this leads to backorders.

Notwithstanding, backorders, a very familiar term in supply chain management, occur when a customer order cannot be fulfilled immediately due to lack of available supply. Backorders can arise from various factors. From the supply perspective, a company may encounter situations where they exhaust their stock of a particular item due to challenges in the supply chain, underestimation of manufacturing capacities, or failures in delivering products to physical retail locations. On the demand side, backorders often occur when there is a high level of consumer interest in a product, particularly in the case of new releases or highly sought-after items (Kenton, 2022). Kathuria (n.d.) stated that backorders can be troublesome because, although they may initially appear as a positive situation for a manufacturer, they eventually raise additional tangible and intangible costs that can be burdensome. Backorders can lead to unhappy customers, lost sales, and disruptions in business operations.

Not only this, but backorders can also cause a bullwhip effect, another phenomenon that can have significant implications for SCM. A bullwhip effect is a chain reaction that results from the fluctuations in demand at the downstream end of a supply chain. It usually occurs when a small shift in customer demand leads to a bigger swing in what companies produce and stockpile. Several studies found that the bullwhip effect and backorders are interconnected, one can cause the other (“How backorders can”, n.d.; Burtler, 2022; Wallstreetmojo, 2023; Intellipaat, 2023, Georgiev, n.d.). Even though the bullwhip effect, along with its consequences for supply chains, has been extensively examined by scholars, it remains a significant and contemporary challenge in the field of supply chain research (Forrester, 1961; Wang and Disney, 2016, as cited in Weisz et al., 2022). Some recent examples of the bullwhip effect on inventories are:

In October 2022, Nike disclosed an excess inventory valued at $9.7 billion, attributing it to increased purchasing due to prolonged lead times. Concurrently, ASOS faced substantial

losses, writing off up to £130m of stock. In a related context, recent independent research has revealed that inventory mismanagement leads to a staggering $163 billion worth of supply chain waste annually, caused by product expiration or overproduction. (Phillips, 2022).

Addressing the bullwhip effect and effectively managing backorders require proactive measures in supply chain management. Leveraging advanced analytics techniques, such as Machine learning/ ML has emerged as a promising solution. By utilizing predictive analytics and machine learning algorithms, businesses can proactively conduct accurate demand forecasting, identify potential backorders and thus optimize inventory management, and enhance overall supply chain performance.

## 1.2 Problem statement

With the rise of e-commerce enterprises and the growing preference for online shopping, the complexity of supply chain management issues has also escalated. This poses a significant challenge for managers, who are responsible for effectively managing various aspects of the supply chain, such as production, inventory, transportation systems, and more (Dehghan-Bonari et al., 2021; Gao et al., 2022).

Although advanced analytics techniques, such as predictive analytics and Machine learning/ML, show potential for addressing backorder challenges in SCM, there is a research gap in understanding how these techniques can be effectively employed within the supply chain context. Islam & Amin (2020) and Gao et al. (2022) claimed that ML algorithms are not widely adopted in various aspects of business decision-making processes due to their lack of clarity and flexibility. However, the current literature lacks comprehensive studies that evaluate the comparative effectiveness of different machine learning algorithms in predicting backorders and explore the impact of parameter tuning on algorithm performance. Notably, most of the existing research has not employed a multi-metric evaluation technique to verify the models' effectiveness. Hence, there is a need to assess the performance of machine learning models using relevant evaluation metrics, such as Confusion Matrix, Accuracy, Precision, Recall, F1-score, AUC/Area Under the Curve (ROC) and/or any other suitable metrics, to determine the most viable strategies for optimizing business operations.

## 1.3 Research objectives

The primary objective of this research is to investigate the accuracy and performance of predictive analytics models in identifying early warning signs of potential backorders, enabling businesses to implement proactive inventory management strategies and minimize stockouts. The secondary objectives derived based on it are the followings:

- To evaluate and compare the effectiveness of different machine learning algorithms in predicting backorders within the context of supply chain operations. Further to this, the impact of parameter tuning on the performance and accuracy of these algorithms will also be explored.

- To assess the performance of machine learning models for backorder prediction, considering various evaluation metrics, such as Confusion Matrix, Accuracy, Precision, Recall, F1-score, AUC and/or any other suitable metrics. The objective is to compare the effectiveness of these models within the supply chain domain and identify the most suitable approaches for optimizing business operations.

## 1.4 Research questions

This paper aims to identify the best predictive model to achieve above-mentioned research objectives. The following research questions, RQs, will guide the investigation:

**RQ-1.** How can predictive analytics be effectively utilized to identify early warning signs of potential backorders within the supply chain?

**RQ-2.** What is the comparative effectiveness of different machine learning algorithms in predicting backorders in the context of supply chain operations? Additionally, how does parameter tuning impact the performance and accuracy of these algorithms?

These research questions aim to explore the application of predictive analytics and machine learning algorithms in identifying and managing backorders within the supply chain. By addressing these questions, this research seeks to contribute to the existing knowledge and

provide valuable insights for businesses in improving their inventory management strategies and overall supply chain performance.

## 1.5 Significance and contributions of the study

The significance of this work lies in its contribution to the supply chain management field. This study offers a comprehensive and innovative approach to forecasting inventory backorders. Some key points to highlight the contributions of this research include:

- **Practical implications:** By investigating the use of predictive analytics and machine learning in identifying potential backorders, this research provides practical insights and recommendations for businesses to enhance their inventory management strategies. Implementing proactive measures to minimize stockouts and improve supply chain efficiency can lead to cost savings, improved customer satisfaction, and increased competitiveness in the market.

- **Operational efficiency:** The findings of this study can help businesses streamline their supply chain operations by optimizing inventory levels, reducing backorder instances, and minimizing excess inventory. This can result in improved resource utilization, reduced lead times, and better overall operational efficiency.

- **Enhanced customer experience:** By effectively managing backorders and reducing stockouts, businesses can meet customer demands more effectively and enhance the overall customer experience. This can lead to higher customer satisfaction, increased loyalty, and improved brand reputation.

- **Competitive advantage:** Adopting advanced analytics techniques for backorder prediction can provide businesses with a competitive advantage in the market. The ability to anticipate and mitigate backorder situations can help businesses stay ahead of their competitors, meet customer demands more effectively, and achieve a higher level of operational excellence.

Apart from this, the significance of this study extends beyond the realm of SCM. It has the potential to be a lifesaver in critical situations, especially for emergency products, such as medicine. This research holds great importance in ensuring the timely delivery of life-saving medications to those in need, thereby contributing to the well-being and safety of individuals. Furthermore, by investigating and analyzing the efficacy of different predictive analytics techniques and their applications in backorder prediction, this research not only provides practical recommendations but also adds to the academic body of knowledge in the field of supply chain management.

## 1.6 Limitations and delimitations of the work

While conducting this research, certain limitations and delimitations were encountered that should be acknowledged to offer a thorough comprehension of the study's scope and potential constraints.

On one hand, the analysis and conclusions are based on the available data, which may have limitations in terms of completeness or accuracy. The study utilized a specific dataset, and the findings may not be fully representative of the entire population or industry. Certain assumptions were made during the research process, which may introduce biases or uncertainties in the outcomes. The findings may have limited generalizability to other industries, supply chain setups, or geographic regions. Apart from this, this paper briefly acknowledged the presence of the bullwhip effect in the supply chain but did not provide comprehensive discussions on potential solutions to address this issue.

On the other hand, the delimitations of this study lie in its focus on the application of predictive analytics and machine learning for backorder prediction within the supply chain context. Further to this, the study evaluates specific machine learning algorithms, while alternative algorithms may yield different results. The research employs certain evaluation metrics, such as Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC/AUC score, ROC/AUC curve, Precision-Recall curve, G-mean, etc. to assess model performance, but other metrics could provide additional insights. The findings should be considered within the specific timeframe of the study and may not capture the latest dynamics or emerging trends in supply chain management.

## 1.7 Structure of the paper

This paper has been divided into five different chapters. The first chapter, “Introduction”, gives an overview of the research topic, presents the background and motivation, and highlights the significance and contribution of the study. It introduces the research objectives and research questions, setting the foundation for the rest of the paper. However, the remainder of the work is structured as follows:

The second chapter, “Examining literature and theoretical constructs”, provides a comprehensive review of the existing body of literature related to supply chain management, bullwhip effect, and backorder prediction. An overview on advanced analytics with machine learning has also been presented, where machine learning techniques, comprehensive process in machine learning, importance of data and challenges or issues associated with it, feature engineering techniques for prediction accuracy improvement and evaluation metrics for assessing model performance, etc. have been discussed. In addition, a thorough exploration has also been undertaken on various real-world applications of machine learning. After that, by reviewing previous studies and research papers on backorder prediction, valuable insights will be gained into the current state of knowledge in optimizing business operations and supply chain performance. The review synthesizes existing knowledge and establishes the theoretical framework for the current research. The chapter also discusses the rationale and uniqueness of this paper while at the same time it identifies gaps in the existing research.

The next chapter, “Research design and methodology”, consists of the research methodology adopted in this work. It describes the data collection process, details about the dataset used in the study, including its source, size, and attributes. It outlines the specific characteristics of the data that are relevant to the research objectives. Further to this, this chapter also covers the selection of machine learning algorithms and their training, as well as the process of tuning the hyperparameters of the selected machine learning algorithms to optimize their performance. It also discusses the evaluation metrics used to assess the performance of the predictive models.

In the “Experimental results and analysis” chapter, the collected data is thoroughly analyzed using the chosen machine learning algorithms. The chapter focuses on presenting and

interpreting the results obtained from the analysis, emphasizing the performance of the models in predicting backorders and benchmarking the top-performers against state-of-the-art solutions. The findings are then discussed in relation to the research objectives, providing insights into the effectiveness of the proposed approach. The chapter concludes with a discussion on the broader implications and a critical evaluation of model usability.

Finally, Chapter 5, "Conclusion and future work", presents a summary of the key findings and contributions of the research. It recaps the research questions and objectives and discusses the limitations encountered during the study. On top of it, it also explores potential areas for future research and provides concluding remarks on the significance of the study.

# 2 Examining literature and theoretical constructs

This chapter will review the existing body of literature related to supply chain management, bullwhip effect and backorder prediction, advanced analytics, and machine learning techniques. This literature review will help identify any gaps or areas that require further investigation, forming the basis for the research objectives.

## 2.1 Overview of supply chain management and its challenges

According to SAP (n.d.) and Perkins et al. (2021), supply chain management, SCM, entails the integration of activities that facilitate the conversion of raw materials into finished goods and their subsequent delivery to customers. These activities encompass sourcing, product design, manufacturing, inventory management, transportation, and distribution. The primary aim of supply chain management is to optimize operational efficiency, enhance product quality, increase productivity, and ultimately ensure customer satisfaction.

The concept of a supply chain extends beyond individual companies, often encompassing multiple facilities in different countries. Managing material, information, and financial flows within multinational corporations presents ongoing challenges. Streamlined decision-making can be achieved through the integration of facilities under a unified organizational structure. Collaboration among functional units: marketing, production, procurement, logistics, and finance, is essential for effective supply chain management (Stadler & Kilger, 2005).

However, supply chain management faces numerous challenges in today's dynamic business environment. These challenges arise from various factors, such as market dynamics, technological advancements, and customer expectations. Some of the key challenges include:

- **Increased costs:** Rising costs of raw materials, transportation, and labor put pressure on supply chain operations, affecting profitability and competitiveness (Lans, 2019).

- **Material scarcity:** The onset of the global COVID-19 pandemic led to a scarcity of essential inputs, intensifying challenges arising from an abrupt spike in consumer demand. Retailers and suppliers found it increasingly challenging to meet this demand due to the limited supply of various parts and materials. A recent survey by the Institute for Supply Management/ISM highlights the presence of record-long lead times, acute shortages of critical materials, increasing commodity prices, and transportation difficulties across industries. Given the scarcity of inputs, a brand's ability to maintain growth relies heavily on sufficient working capital to navigate this period and prepare for peak seasons (Brown, 2022).

- **Labor unrest:** Supply chains face increased pressure when industrial tension occurs. Recent examples include trucker strikes in South Korea that disrupted computer supply chains and railway strikes in the UK that affected the delivery of construction materials. Dock workers in Germany and the UK have also gone on strike, and strikes at the Port of Liverpool are expected to cause congestion at freight hubs in Ireland (World Economic Forum, 2022).

- **Geopolitical uncertainty**: Geopolitical uncertainty has a great impact on the supply chain. The invasion of Ukraine has caused global energy and food price inflation, leading to supply chain disruptions and a global food crisis. A shortage of fertilizers is also impacting agricultural output in many countries. Tensions between China and the US, exacerbated by recent military exercises and political visits, have the potential to disrupt supply chains for critical components, such as semiconductors. These factors collectively pose significant challenges to global supply chains and have far-reaching consequences for various industries (World Economic Forum, 2022).

- **Qualified personnel:** Lans (2019) claims that finding individuals who are both interested in and passionate about supply chain management has become increasingly difficult. It is crucial to recruit personnel who possess a comprehensive understanding of the roles and responsibilities inherent to the field.

- **Collaboration and syncing of data:** Efficient supply chain management relies on access to supply chain data. Yet, managing the vast number of data points across global supply chains poses a significant challenge in this field ("Key Challenges in Supply Chain Management", 2022).

- **Digital transformation:** Enhancing supply chain operations requires the adoption of digital transformation and technologies, such as IoT/Internet of Things, AI/Artificial Intelligence, drones, and robotics. However, the main obstacle in supply chain management is effectively integrating these technologies into existing operations ("Key Challenges in Supply Chain Management", 2022).

- **Difficult demand forecasting:** Fluctuating customer demands, seasonal variations, and market uncertainties make forecasting particularly challenging in the supply chain. These complexities directly impact inventory management (Brown, 2022).

- **Port congestion:** Port congestion arises when a port reaches its capacity and cannot accommodate the arrival of ships for cargo loading or unloading. This can be caused by various factors, such as bad weather, accidents, equipment damage, unpredictable trade demands, and inadequate port infrastructure. The consequences of port congestion include delivery delays, queues, increased travel time, additional costs, trade loss, reduced productivity, limited port access, and significant implications for the logistics and supply chain industry (Fathima, n.d.).

Therefore, disruptions in the supply chain, demand volatility, etc., are some of the key underlying challenges in SCM. Understanding and proactively addressing these challenges are crucial for developing effective strategies and solutions to optimize supply chain performance.

## 2.2 Bullwhip effect and Backorder prediction in supply chain management

The “bullwhip” effect, also known as “whiplash” or “whipsaw” or “Forrester” effect was coined by an MIT Sloan School of Management professor named Jay Wright Forrester in the year of 1961 (Lee et al., 1997; Holicki, 2022). It is a phenomenon in which an increase in demand variability occurs as one moves from downstream stages to upstream stages in a supply chain (Lee et al., 1997, as cited in Pillai & Pamulety, 2013). It has significant implications for supply chain management, leading to issues, such as excessive inventory, poor customer service, revenue loss, transportation inefficiencies, and more (Lee et al., 1997; Wright & Yuan, 2008; Shukla et al., 2009, as cited in Pillai & Pamulety, 2013).

The bullwhip effect has been observed in various industries, including Procter & Gamble, Hewlett-Packard, the machine tool industry, the clothing supply chain, grocery retailers, and more (Lee et al., 1997; Disney & Towill, 2003; Ge et al., 2004; Kok et al., 2005; Terwiesch et al., 2005; Odonnell et al., 2006; Bhattacharya & Bandyopadhyay, 2011, as cited in Pillai & Pamulety, 2013). Researchers have identified operational and behavioral causes of the bullwhip effect (Paik & Bagchi, 2007; Bhattacharya & Bandyopadhyay, 2011, as cited in Pillai & Pamulety, 2013).

Operational causes include factors, such as demand forecast updating, batching of orders, fluctuations in price, the gaming of rationing and shortages, lack of communication and coordination, information transparency, and more. Behavioral causes, identified through experiments such as the beer distribution game, involve misperception of feedback, lack of training and learning, overreaction to orders, and underestimating the value of information (Sterman, 1989; Croson & Donohue, 2003; Nienhaus et al., 2006, as cited in Pillai & Pamulety, 2013).

Backorders contribute to the bullwhip effect, as they introduce variation in order quantities (Croson & Donohue, 2002, as cited in Pillai & Pamulety, 2013). However, backorder and out of stock should not be mistaken for one another. While out of stock means the delivery date of goods cannot be guaranteed, backorder allows customers to browse and place orders for products. In essence, a backorder can be understood as an order that will be delivered at a later, postponed date. (Raja, 2021; Dahiwalkar, 2021).

The following figure, Figure 1, illustrates the bullwhip effect that is a result of a slight increase in consumer demand and the interaction within the supply chain.


***Figure 01: An example of bullwhip effect (Li, 2017, p. 4)***

Backorders are simply a result of temporary unavailability or out-of-stock situations, whereby customers place orders for future production and shipment. These instances commonly occur when there is a surge in demand or when a popular product is anticipated to be released soon.

For example, during the COVID-19 pandemic, the increased need for antiseptic products and indoor activities led to a significant surge in online purchases. This unexpected demand spike caused the bullwhip effect, leaving many industries unprepared and surprised by the sudden increase in demand. Consequently, product stocks were inadequate, but customers were willing to wait due to limited alternatives. Another scenario is when a renowned company announces the upcoming release of a new product. In such cases, the company accepts backorders from customers due to an inadequate initial production quantity in relation to the anticipated demand (Shajalal et al., 2021, as cited in Ntakolia et al., 2021).

Proper management of backorders is important in inventory control as it directly affects the overall production costs of the entire supply chain (Ntakolia et al., 2021). Effectively managing backorders significantly impacts a company's revenue, stock market performance, and customer trust (Islam & Amin, 2020, as cited in Ntakolia et al., 2021).

Backorder prediction involves using data analysis and forecasting techniques to estimate the likelihood of backorders occurring. Machine learning algorithms and statistical models can be employed to analyze historical data on orders, inventory levels, customer demand, and other relevant factors. These models can then generate predictions on the probability of backorders in the future (Raja, 2021).

However, backorders have been the subject of extensive study in supply chain management research. Experimental studies have primarily focused on backorder settings to examine the bullwhip effect and its underlying causes (Sterman, 1989; Croson & Donohue, 2003; Croson & Donohue, 2005; Wu & Katok, 2006; Cantor & Katok, 2012, as cited in Pillai & Pamulety, 2013). In light of these findings, it becomes imperative for companies to accurately forecast backorders. This practice allows them to optimize inventory levels and streamline production planning while enhancing customer satisfaction. To this end, implementing data-driven approaches, such as machine learning and statistical modeling, can offer reliable backorder predictions (Raja, 2021).

## 2.3 Challenges and complexities associated with backorder prediction

Although backorder prediction is a specific aspect of supply chain management, it often intersects with the general challenges encountered in the broader supply chain spectrum, as highlighted in Section 2.1. Nevertheless, due to its unique nature and specific requirements, it presents its own set of challenges and intricacies that mandate deeper exploration for accurate forecasting. These include:

- **The foggy crystal ball:** Backorder prediction often requires the use of machine learning algorithms. However, sometimes the process lacks clarity and flexibility which can be compared to “gazing into a foggy crystal ball”. Moreover, incorporating inaccurate or faulty data can diminish the accuracy, clarity, and adaptability of the prediction process (Islam & Amin, 2020; Gao et al., 2022).

- **The quest for reliable data:** Echoing insights from Section 2.1, backorder prediction relies on historical data related to orders, inventory levels, and customer demand. However, obtaining comprehensive and accurate data can be challenging.

Additionally, issues, such as fragmented, inconsistent, and incomplete data can negatively impact the reliability of backorder prediction model (Islam & Amin, 2020).

- **The imbalanced tango:** Backorders are relatively rare compared to items that do not go into backorder. This imbalance in class distribution poses a challenge in developing predictive models. Techniques, such as ensemble learning, sampling, and specific metrics are employed to address this issue (Santis et al., 2017; Raja, 2021; Dahiwalkar, 2021; Shajalal et al., 2022).

- **The demand rollercoaster:** Demand forecasting is a crucial aspect of backorder prediction. However, as discussed in Section 2.1, accurately predicting demand patterns can be challenging due to various factors, such as seasonality, market trends, and external events. The dynamic nature of demand introduces complexities in accurately forecasting backorders (Raja, 2021; Brown, 2022).

- **The whirlwind supply chain:** As touched upon in Section 2.1, supply chains are subject to constant changes, including supplier disruptions, production delays, and market fluctuations, etc. These dynamic factors can introduce volatility into the backorder prediction process, which is difficult to capture and incorporate into prediction models (Luther, 2022; Lutkevich, 2023).

Addressing these challenges requires a combination of technical expertise, robust data management practices, and collaborative efforts among supply chain partners. Overcoming these complexities can lead to improved backorder prediction, optimized inventory levels, enhanced customer satisfaction, and overall supply chain efficiency.

## 2.4 Understanding machine learning and its applications

This section provides an overview of Machine Learning/ML and its significance in backorder prediction. The applications, types of techniques, and key concepts in machine learning will be discussed to understand its relevance in backorder prediction. By gaining insights into the advantages and limitations of machine learning, the author aims to establish a foundation for further exploration of machine learning methods in achieving effective backorder prediction.

### *2.4.1* Machine learning- what it means

AI/Artificial Intelligence, ML/Machine Learning, NN/Neural Networks, and DL/Deep Learning are often considered as buzzwords. These terms have gained significant attention and popularity in various industries and discussions, often capturing the interest and curiosity of people due to their potential implications and advancements in technology. Kavlakoglu (2020) used the metaphor of Russian nesting dolls to demonstrate the relationship between AI, ML, NN, and DL. Analogous to the structure of these dolls, each term represents a component nested within the preceding one, where AI serves as the broader field, and ML is a specialized subfield within it. DL, in turn, is a subfield of ML, and it heavily relies on NN as its fundamental architecture.

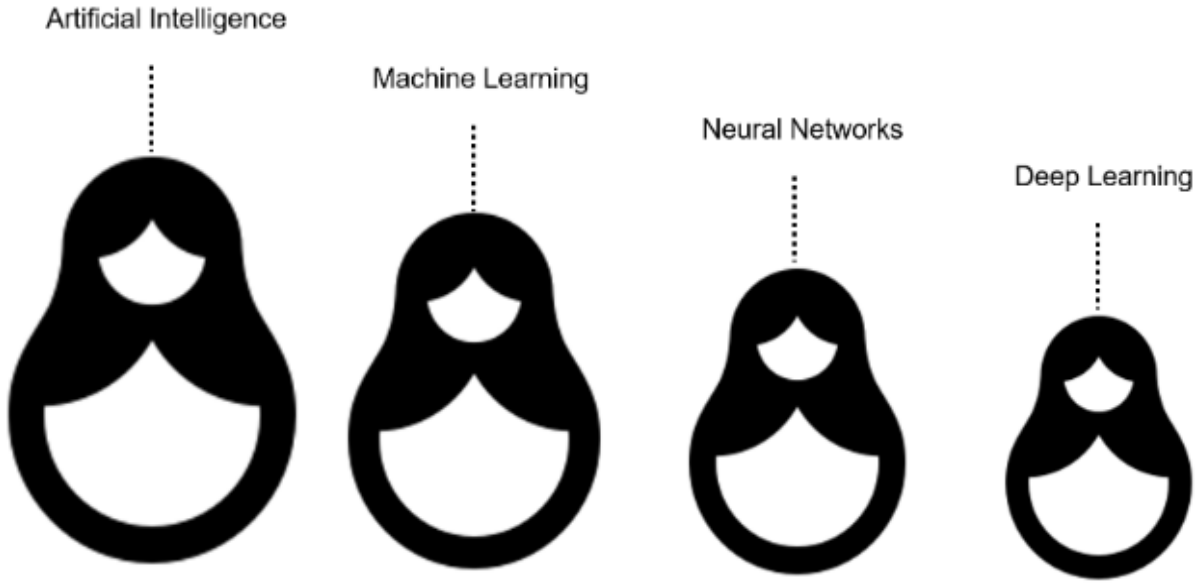



***Figure 02: Relationship between AI, ML, NN, and DL (Kavlakoglu, 2020, para. 3)***

According to Brown (2021), a news writer in renowned MIT Sloan, during the 1950s AI pioneer Arthur Samuel defined ML as the area of research that empowers computers to acquire knowledge and skills without explicit programming instructions. In contrast, Brown (2021) also argued that in certain situations, programming a machine to perform a specific task can be challenging or even impractical, particularly when it comes to complex tasks including image recognition. Humans can effortlessly recognize different individuals in pictures, but conveying this ability to a computer is challenging. Machine learning offers an alternative approach by enabling computers to learn and develop programming capabilities autonomously through experience.

However, Zhang (2017) defines ML as an automated approach to analyzing data by constructing analytical models. It utilizes adaptive algorithms that continuously learn and adapt from data and past computations. This allows the models to discover information and patterns without explicit instructions on where to look.

#### *2.4.1.1 Machine learning approaches: supervised versus unsupervised*

Generally, there are two main types of machine learning algorithms: supervised learning and unsupervised learning. Supervised learning involves a data scientist guiding the algorithm by providing labeled datasets with predefined outputs. Linear regression, logistic regression, multi-class classification, and support vector machines are examples of such approaches. In contrast, unsupervised learning allows the computer to independently identify patterns and processes without labeled data. Algorithms, such as k-means clustering, principal component analysis, and association rules fall under unsupervised learning. However, the choice between the two depends on factors, such as data structure and volume, as well as the specific use case. Machine learning is widely applicable across various industries and supports diverse business goals and use cases (Oracle, 2023).

#### *2.4.1.2 Machine learning process at a glance*

Machine learning enables systems to learn from data. It involves creating computer programs that can acquire knowledge independently by analyzing input data. ML professionals adhere to a standardized methodology to complete their tasks, irrespective of the model or training technique employed. These actions involve iteration that allows continuous evaluation and improvement. ML empowers machines to learn and adapt autonomously, leading to intelligent decision-making based on available information (Ahamed, 2022).

To say about the machine learning process, Brownlee (2016) emphasized that the initial stage of any project involves clearly identifying and defining the problem. While utilizing advanced and impressive algorithms may be tempting, their effectiveness becomes irrelevant if they are applied to the wrong problem. Therefore, it is crucial to ensure that the problem statement is accurate and well-defined before proceeding with any solution approach.

However, once the exact problem is defined, the remaining process of machine learning can be shown as below:

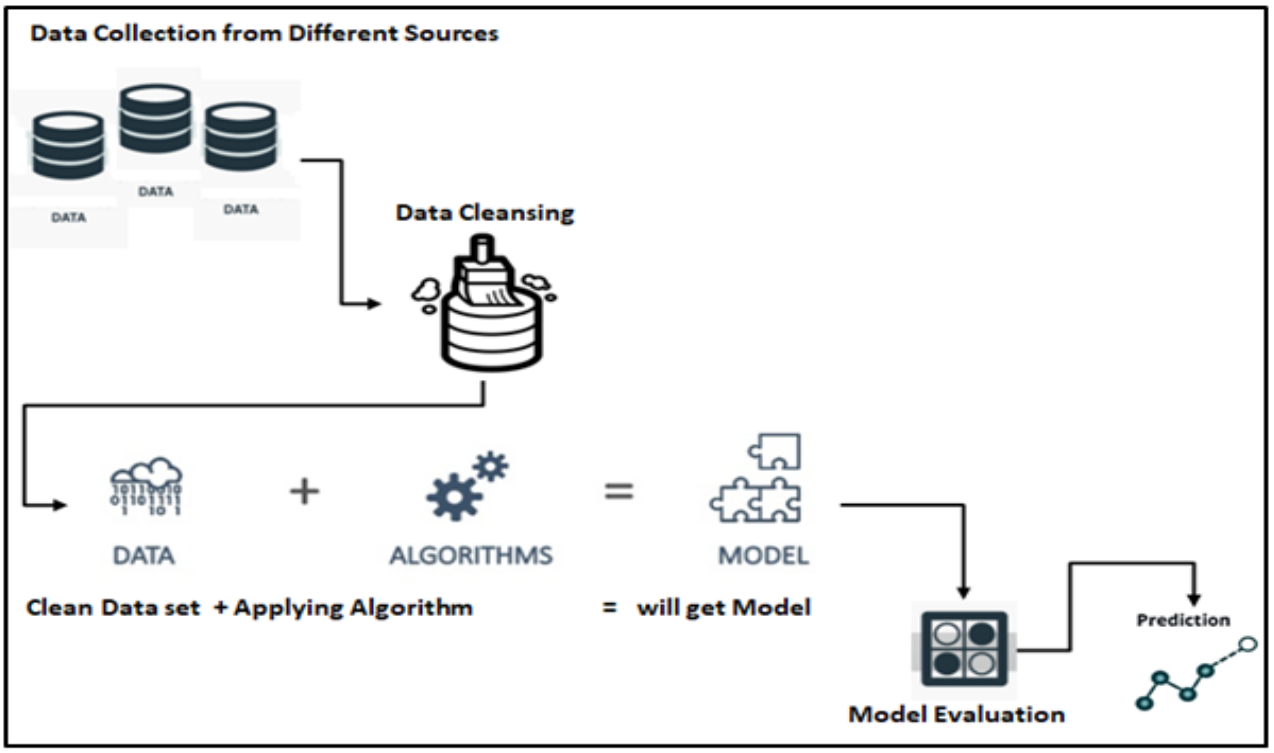



*Figure 03: ML process at a glance (Pandian, 2020, para. 6)*

- **Data Collection:** In machine learning, the learning process heavily relies on the data provided to the model. Thus, it is crucial to gather reliable data that enables the model to identify accurate patterns. The quality of the input data directly impacts the accuracy of the model's outcomes and predictions. Using incorrect or outdated data can lead to irrelevant results. Therefore, it is important to ensure the data is sourced from a reliable and trustworthy source. Good data is characterized by its relevance, minimal missing or duplicate values, and a comprehensive representation of the different subcategories or classes within the dataset (Banoula, 2023).

- **Data preparation:** After collecting the data, it undergoes data processing, including Exploratory Data Analysis /EDA and feature engineering. EDA helps understand and clean the dataset, while feature engineering involves handling missing values, converting categorical variables into numerical representations, addressing non-Gaussian distributions, finding outliers, and scaling features. The clean data then is

split into two sets, training set and testing set. The training data is used to recognize patterns and optimize the algorithm, while the test data evaluates the model's predictive capabilities on new data. However, it is important to judiciously split the data into separate training and test sets (usual range between 20 percent to 80 percent) to avoid overfitting and ensure accurate predictions (Pandian, 2020).

- **Model building:** Once the dataset is prepared, a suitable model can be selected to address the targeted problem. For tasks involving continuous outputs, such as predicting the waiting time for an order, a regression algorithm can be used. If the task involves classifying inputs, such as determining whether an order will be completed within a specific time frame, a classification algorithm is chosen. Different classification and regression algorithms exhibit varying performance depending on the dataset's characteristics. Models are selected based on whether the data is categorical or numerical and the number of features in the dataset. The interpretability of the models also varies, indicating the ease of understanding and interpreting the results obtained (Codeacademy, 2023).

- **Model evaluation:** Various evaluation methods exist for assessing the performance of different models. In regression, metrics, such as Sum of Squared Error /SSE, Mean Squared Error /MSE, Root Mean Squared Error /RMSE, Mean Absolute Error /MAE, Coefficient of determination /$R^2$, and Adjusted $R^2$ are commonly used (Pandian, 2020). For classification models, evaluation can be done through the Confusion Matrix, Accuracy score, Precision, Recall, F1 score, and metrics, such as AUC and ROC (Silwal, 2022). These evaluation techniques provide valuable insights into the accuracy and performance of regression and classification models.

  However, a classifier's effectiveness is determined by the evaluation metric applied. By using the wrong metric, one risks choosing an inefficient model or even underestimating the performance of the model. In machine learning, choosing the appropriate measure can be particularly challenging, especially for imbalanced classification. This is because errors in prediction do not have the same effect across imbalanced classes, contrary to the assumption made by standard metrics that class distributions are equal (Brownlee, 2020). Hence, Macro F1 score, AUC score, G-mean, and some graphical performance evaluation, such as ROC curve and Precision-

Recall curves, etc. play a vital role in the performance measurement in case of an imbalanced dataset (Bekkar et al., 2013; Brownlee, 2020; Allwright, 2022).

- **Model deployment:** Finally, ML deployment comes into action that involves integrating the finalized model into a production environment and obtaining results that can inform business decisions (Pandian, 2020).

However, one important aspect of machine learning is hyperparameter tuning. Hyperparameters are external configuration variables that data scientists manually set before training a model. They manually control features, such as model architecture, learning rate, and complexity. This process, known as hyperparameter tuning or optimization, involves experimenting with different combinations of hyperparameters to find the most effective ones. The iterative process involves trying out various parameter combinations and evaluating them based on metrics, such as accuracy as a base metric. Bayesian optimization, grid search, random search, etc. are the most commonly used algorithms for this purpose. Cross-validation techniques are often employed to ensure the model's generalizability across different subsets of the data. Overall, hyperparameter tuning is a crucial and computationally intensive process that significantly impacts the quality and effectiveness of machine learning models (AWS, 2023).

#### *2.4.1.3 Data in machine learning*

Brown (2021) claims that machine learning begins by collecting and preparing data, which can include various types, such as numerical data, images, or textual information, from different sources. However, data serves as the fuel for machine learning algorithms. The availability and quality of data significantly impact the performance and accuracy of the machine learning model (Domingos, 2012).

A survey conducted by CrowdFlower, a platform that offers data enrichment services for data scientists, data scientists allocate their time in the following manner (as cited in ProjectPro, 2023):

***Table 01: CrowdFlower survey on data scientists' time allocation***

| Task | Time spent |
|---|---|
| Organizing and cleaning data | 60% |
| Collecting datasets | 19% |
| Mining the data to draw patterns | 9% |
| Training the datasets | 3% |
| Refining the algorithms | 4% |
| Other tasks | 5% |

The survey findings suggest that a significant portion of a data scientist's time is dedicated to data preparation tasks (ProjectPro, 2023).

However, "Data preparation is an essential that can take place at various stages of the machine learning process" (Prof. Mezei, personal communication, October 12, 2021). Therefore, data preprocessing is a critical step in machine learning, aiming to prepare the data for analysis and modeling. Various techniques are employed to make the data more usable and reliable. For example, data cleaning involves identifying and addressing issues, such as incomplete, inaccurate, duplicated, irrelevant, outliers, or noisy data in the dataset. Dimensionality reduction is another technique used to simplify the data by reducing the number of features. This helps focus on the most important aspects and avoids unnecessary complexity and thus improve computational efficiency. It also helps in preventing overfitting and avoiding multicollinearity. Feature engineering involves creating new features based on domain knowledge to enhance the model's predictive power. Sampling techniques are employed when dealing with large datasets that may strain resources. Representative sampling data techniques, such as with/without replacement, stratified and progressive sampling can be used to capture the essence of the data without overwhelming computational capabilities (Azevedo, 2023).

Azevedo (2023) also argues about another technique, data transformation, that converts the data to the same structure to avoid poor model performance. This includes normalization and standardization, which are used to adjust the data's scale and distribution. Apart from this, transforming categorical variables also is considered as data transformation. Further to this, handling imbalanced data is also essential when dealing with datasets that have a

disproportionate distribution of classes. Techniques, such as oversampling, under-sampling, and hybrid approaches are employed to address this issue.

Overall, these data preprocessing techniques are vital for improving data quality, model performance, and the overall success of machine learning projects.

### *2.4.2* Machine learning use cases and the power of prediction

Machine learning plays a pivotal role in the business models of certain companies, such as Netflix, with its recommendation algorithm, and Google, with its search engine, while other companies are actively exploring its potential applications, even if it is not their primary focus. A recent survey by Deloitte discovered that 67% of companies are already leveraging machine learning whilst 97% will either be using or planning to use within a year. However, many organizations are still struggling to figure out the problems that can be addressed through machine learning (Brown, 2021).

Brown (2021) also mentioned that, based on a recent study from MIT, while no occupation will remain untouched by machine learning, complete replacement by machines is unlikely. Successful implementation of machine learning involves reorganizing jobs into discrete tasks, some of which can be automated, while others require human involvement. Nevertheless, various applications of machine learning can be observed across different industries. Recommendation algorithms powered by machine learning are utilized by different platforms, such as Netflix, YouTube, and other social media networks to personalize content based on user preferences. Machine learning algorithms are also employed in image analysis and object detection tasks, allowing for the identification and differentiation of objects and individuals in images. While facial recognition algorithms have raised concerns, machine learning finds application in other areas, such as finance, where hedge funds analyze parking lot occupancy to gain insights into company performance. Fraud detection benefits from machine learning as well, as algorithms can identify anomalies and patterns in financial transactions that help detect potentially fraudulent activities, such as unauthorized credit card usage or spam emails. Companies also leverage machine learning and NLP/ Natural Language Processing to deploy automatic helplines or chatbots. This approach enables them for automated customer assistance and learning from past conversations to deliver relevant

responses. Machine learning, particularly deep learning, forms the foundation of technology used in self-driving cars. Further to this, machine learning algorithms play a vital role in medical imaging and diagnostics, where they can be trained to analyze medical images and data, assist in diagnosis, and predict markers of diseases, such as assessing cancer risk using mammograms.

Machine learning offers predictive capabilities that allow organizations to make informed decisions based on advanced analytics. One example is predictive maintenance, where ML models identify equipment at risk of failure, enabling maintenance teams to take preventive action. This approach maximizes productivity, improves asset performance and longevity, reduces costs, and enhances regulatory compliance. Moreover, predictive maintenance aids in inventory control and management by accurately forecasting spare parts and repairs. This results in reduced expenses, and increased operational efficiency (Oracle, 2023).

Hence, ML has found applications in different business areas, including healthcare, finance, marketing, and now increasingly in supply chain management. By harnessing the power of ML, organizations can gain a deeper understanding of their supply chain dynamics to make well-informed decisions. This enables them to optimize inventory levels, improve customer satisfaction, reduce costs, and ultimately achieve their business goals.

## 2.5 Existing methods and techniques for backorder prediction

Backorder prediction, being a critical task in inventory and supply chain management systems, aims to minimize losses and optimize customer satisfaction. To achieve these objectives, along with the traditional approach, researchers have proposed several methods and techniques that leverage machine learning algorithms and data analysis techniques.

This section aims to provide a comprehensive overview of existing research in this domain, highlighting the different approaches employed and their implications for backorder prediction. However, it is worth noting that these findings, although not specifically targeting backorder prediction, provide valuable insights and serve as a reference for this project.

### *2.5.1* Traditional approach versus modern approach in backorder prediction

The traditional approach to backorder prediction is often considered subjective. There are varying views on what constitutes a conventional method, with some, such as Singh et al. (2021), contending that randomly manufacturing inventory without proper measures or detailed analysis of customer demand is a traditional approach in the supply chain. Ntakolia et al. (2021) and Hájek & Abedin (2020) stated that traditional methods described in the literature rely on stochastic approximations and do not consider insights from historical data.

In contrast, several studies suggest even some advanced analytics techniques can be categorized as traditional methods (Rheude, 2022; IBM, 2021; Hyndman & Athanasopoulos, 2021; Carbonneau et al., 2007; 2008). However, it is widely recognized that these conventional approaches may be insufficiently accurate and efficient for predicting backorders.

Both the traditional and advanced approaches are presented in an extensive manner below:

#### *2.5.1.1 Traditional approach*

Traditional models often involve a combination of qualitative and quantitative analysis, considering previous trends, current market situations, and expert insights. For example, qualitative techniques, such as expert judgment and the Delphi method involve gathering opinions and expertise from supply chain professionals or domain experts to estimate future demand and potential backorders. In the market research method, researchers use a variety of techniques that include conducting surveys, interviews, focus groups, and observations to gather customer feedback, preferences, and demographic data. This approach helps understand consumer behavior and market trends. Further to this, the sales force composite method relies on feedback from the sales team, who have direct contact with customers. Salespeople provide qualitative insights for demand forecasting by sharing their knowledge of client preferences, product trends, and competition actions. These approaches can provide insightful information, especially in situations where historical data is limited or unreliable. However, they are subjective and reliant on the expertise and biases of the individuals involved (Rheude, 2022).

On the other hand, quantitative analysis or economic method requires number crunching (Rheude, 2022). Traditionally, statistical regression techniques have been commonly used to analyze numerical inventory values over time and make future projections. These techniques compare the projected inventory values with the actual stock levels to evaluate the accuracy of the regression algorithms (IBM, 2021). However, traditional methods are commonly used as baseline methods for comparison in forecasting studies (Carbonneau et al., 2007).

Among the quantitative analyses, one commonly used traditional approach in backorder prediction is time-series analysis, which involves analyzing historical inventory data and using statistical methods, such as moving averages, exponential smoothing, and Autoregressive Integrated Moving Average/ ARIMA models. The data can be analyzed using time-series models to identify trends, seasonality, and other patterns that can be used to predict future inventory levels. However, they could have trouble capturing non-linear relationships and handling complex factors that impact backorders in the supply chain. Another conventional method is the use of statistical regression models, for instance, multiple linear regression, to determine the relationships between various factors and backorders. These models can analyze historical data and assess the impact of variables, such as lead time, demand, and inventory levels on the occurrence of backorders. These regression models often assume linearity and could miss complex relationships present in the data (Hyndman & Athanasopoulos, 2021; Carbonneau et al., 2007; 2008).

In addition to time-series analysis and regression models, traditional supply chain management techniques, such as Economic Order Quantity/ EOQ and Just-in-Time/JIT inventory management have also been used to manage backorders. JIT strives to lower inventory holding costs by coordinating production and delivery schedules, whereas EOQ assists in determining the ideal order quantity to minimize costs and maintain inventory levels (Sprague et al., 1990; Chopra & Meindl, 2013).

These techniques focus on optimizing inventory levels, even though they may not explicitly address backorder prediction and proactive management.

#### *2.5.1.2 Modern/ML approach*

Modern techniques, such as machine learning/ ML algorithms have also been extensively utilized in backorder prediction to provide flexibility and clarity in the decision-making process. Researchers have developed predictive models employing ML techniques that can predict probable backorder scenarios in the supply chain to help decision-makers understand and optimize the accuracy of predictions. These models can manage vast amounts of data, identify relevant trends and patterns, and produce forecasts that assist in making informed business decisions. To explore the effectiveness of ML-based and traditional forecasting methods in predicting manufacturers' uncertain demands, researchers conducted a comparative analysis (Islam & Amin, 2020).

For example, Carbonneau et al. in 2006[1] (2008, as cited in Li, 2017; Islam & Amin, 2020) conducted a study focusing on comparing traditional methods, such as Naïve forecasting, average, moving average, trend analysis and multiple linear regression with advanced prediction methods, such as Neural Networks/ NN, Recurrent Neural Networks/ RNN, Support Vector Machines/ SVM. Surprisingly, the study discovered that while the average performance of ML methods did not outperform that of conventional approaches, an SVM that was trained on a variety of demand-series produced extremely accurate forecasts, offering a glimpse of possibility. Encouraged by these findings, the same researchers, Carbonneau et al. (2007, as cited in Li, 2017; Islam & Amin, 2020), further expanded their investigation by incorporating 22 different tools in total, including both conventional methodologies and cutting-edge ML techniques. This extended research revealed noticeable improvements in prediction accuracy compared to traditional models (Islam & Amin, 2020).

Furthermore, Guanghui (2011, as cited in Li, 2017; Islam & Amin, 2020) conducted a study where Support Vector Regression/ SVR method was applied to predict the demand in a supply chain, comparing it with the Radial Basis Function/ RBF neural network method. The results indicated that SVR outperformed RBF in terms of prediction performance, making it a suitable and efficient method for demand forecasting in the supply chain.

[1] The work was accepted in 2006 but published in 2008. Source: https://www.sciencedirect.com/science/article/pii/S0377221706012057

#### *2.5.1.2.1 Enhancement in backorder prediction*

Companies face the dilemma of deciding whether to produce or acquire backordered products, while customers may cancel their orders if the wait is too long, resulting in unsold inventory. To improve strategic inventory management decisions, Shajalal et al. (2022) argued for the inclusion of explanations for AI recommendations. Moreover, explainable machine learning models have been proposed to enhance backorder prediction by considering the costs associated with backorders. Ntakolia (2021) developed an explainable machine learning model that helps identify material backorders and optimize production and shipment processes, contributing to an effective and cost-efficient supply chain. This approach considers the unavailability of stock and delays in product delivery, which can lead to additional production costs and dissatisfied customers.

Later, Shajalal et al. (2022) proposed a novel Convolutional Neural Network/ CNN-based predictive model for predicting product backorders in inventory management. They investigated the provision of explanations using Shapley additive explanations, which elucidate the overall priority of the models in decision-making. Additionally, they introduced locally interpretable surrogate models that can explain individual predictions made by the model. Shin et al. (2012, as cited in Islam & Amin, 2020) proposed a dynamic backorder replenishment planning framework based on risk analysis, aiming to reduce supply chain and inventory control expenses. This framework utilized the Bayesian Belief Network to make informed decisions. Acar & Gardner (2012, as cited in Islam & Amin, 2020) also recommended a similar framework, employing optimization and simulation techniques to optimize backorder replenishment planning. In a different approach, Rodger (2014, as cited in Islam & Amin, 2020) introduced a risk-triggering model that incorporated fuzzy feasibility and Bayesian probabilistic evaluation to assess backorder risks.

To address the issue of imbalanced classes in backorder prediction, researchers have explored various techniques, such as ML classifiers, sampling techniques, and ensemble learning. These techniques aim to improve the accuracy of backorder prediction models by handling imbalanced datasets and leveraging the collective predictions of multiple models. One approach that has gained significant attention is the utilization of deep neural networks for backorder prediction. Shajalal et al. (2021) developed a deep neural network model to handle the data imbalance issue between backorders and filled orders. Techniques, such as minority

class weight boosting, random oversampling, and the synthetic minority oversampling technique/ SMOTE have been employed to address this challenge and balance the dataset. Deep neural networks are well-suited for capturing complex patterns and dependencies in the data, making them effective for predicting backorders. These models are trained on historical data that incorporates product information, customer details, and inventory status. They help them make accurate predictions by identifying patterns that indicate backorders. However, evaluation metrics, such as the area under the Receiver Operator Characteristic/ ROC curve and Precision-Recall curves are commonly used to assess the performance of predictive models (Raja, 2021).

### *2.5.2* Modern/ML approach is better

Modern machine learning/ ML techniques have drawn interest because they have the potential to perform better than conventional approaches in backorder prediction, despite the advantages that conventional methods may offer.

The following is a detailed discussion of how the ML approach is better than the traditional approach:

- **Improved accuracy:** Numerous studies have demonstrated that ML algorithms are more accurate at making predictions than conventional techniques. A notable example of this is the study of Carbonneau et al. (2007), where ML algorithms outperformed conventional approaches in terms of prediction accuracy.

- **Handling non-linearity:** Traditional approaches may have trouble capturing and learning from non-linear patterns in the data, whereas ML models excel in this aspect. IBM (2021) points out that although regression models are commonly used and straightforward for time-series data analysis, they may not be appropriate for reliably predicting complex non-linear correlations in more intricate issues.

- **Complex pattern identification:** One advantage of machine learning is its capacity to uncover complex patterns that surpass human cognitive capabilities. While machine learning can swiftly analyze enormous datasets, find hidden patterns, and

enable more accurate forecasts, traditional forecasting approaches are limited by the volume of data that humans can process and analyze properly (Wisneski, 2022).

- **Scalability and adaptability:** ML models are highly scalable and adaptable to different datasets and supply chain scenarios. They are scalable because they can handle and process large volumes of data effectively. Their adaptability lies in their capacity to incorporate various types of data, including non-traditional sources, and adjust their predictions accordingly (Wisneski, 2022).

- **Handling uncertainty:** The inherent unpredictability of backorder forecasting is caused by various external factors. By adding probabilistic strategies and ensemble techniques, ML models may manage this uncertainty. Raja (2021) emphasized how ensemble learning strategies might enhance backorder prediction by aggregating the predictions of multiple models.

- **More accessibility:** Wisneski (2022) asserted that machine learning offers more accessibility compared to conventional methods, which frequently require specialized knowledge and training. As technology advances, ML is becoming more readily available, enabling users to build models on various software platforms without requiring extensive training or experience.

- **Unbiased predictions:** By reducing the impact of human biases and subjective viewpoints, machine learning offers an edge over conventional forecasting techniques, resulting in more accurate predictions. Machine learning algorithms do not have personal biases or emotions, in contrast to humans who can be affected by these things. This makes decision-making impartial, especially in situations, such as launching new stores when conventional approaches could be exposed to subjective biases. However, it is worth noting that machine learning models can still manifest bias if trained on biased data, but employing unbiased data and cross-validation approaches can assist in ensuring accuracy (Wisneski, 2022).

Furthermore, incorporating big data analytics into backorder prediction has shown promising results. By integrating a profit-based measure into the prediction model and optimizing the

decision threshold, the expected profit of backorder decisions can be maximized. This data-driven approach leverages valuable insights from historical inventory data, enabling more precise predictions (Hájek & Abedin, 2020). However, the selection and effectiveness of these methods and techniques may vary depending on the specific context and dataset. Continuous research is underway to explore new approaches and algorithms to further enhance the accuracy and efficiency of backorder prediction models.

### *2.5.3* Recent relevant studies at a glance

Machine learning algorithms have gained significant traction in backorder prediction due to their ability to handle complex patterns and large volumes of data. These algorithms leverage historical data and employ various techniques to predict backorder scenarios accurately. Some commonly used ML algorithms in backorder prediction in most recent studies include the followings:

Santis et al. (2017) conducted a study on backorder prediction using classifiers and ensemble methods. They incorporated balancing techniques, such as RUS/ Random Under-Sampling and SMOTE/ Synthetic Minority Over-sampling Technique to address the imbalanced nature of the dataset. Among the classifiers, LR/ Logistic Regression achieved an AUC/Area Under the ROC Curve score of 0.92, while CART/ Classification Tree performed slightly better with a score of 0.94. Moving on to the ensemble methods, RF/ Random Forest achieved a score of 0.94, indicating strong performance. However, both GBOOST/ Gradient Tree Boosting and BLAG/ Blagging outperformed the other models with impressive AUC scores of 0.95. These results highlight the effectiveness of ensemble methods, particularly GBOOST and BLAG, in predicting backorder accurately.

In the study conducted by Hájek and Abedin (2020), in which they proposed the profit-max CBUS technique, the results showed that the RF classifier achieved the highest AUC score of 0.92, indicating strong predictive performance. LR and SVM also performed relatively well with AUC scores of 0.77 and 0.78, respectively. However, the KNN/ K-Nearest Neighbor classifier and NN/ Neural Network had lower AUC scores of 0.74 and 0.60, respectively, indicating weaker performance in predicting backorders. These findings suggest that the RF classifier is the most effective among the evaluated classifiers for the given task.

Islam and Amin (2020) assessed the performance of GBM/ Gradient Boosting Machine and DRF/ Distributed Random Forest models using SMOTE and ROS/ Random Oversampling techniques on actual and modified (ranged) data. The ranged data is derived from the actual data by converting certain features into different class ranges and then multiplying by correlational factors. When trained with real data, these models showed increased mean classification errors and lower AUC values, indicating a potential overfitting issue. AUC values decreased somewhat throughout the testing phase, with DRF recording an AUC of 0.79 and GBM recording an AUC of 0.80.

Ntakolia et al. (2021) aimed to evaluate the performance of various machine learning models in predicting backorders of products. Eight machine learning models were compared in the study. RF, XGBoost, LightGBM, and Balanced Blagging/ BB exhibited excellent performance with an AUC score of 0.95. These models show promise for effective inventory management and preventing stockouts. In contrast, KNN, LR, and SVM achieved lower accuracy with AUC scores of 0.82, 0.79, and 0.83, respectively. To enhance the performance of the identified models, calibration techniques, such as Isotonic Regression and Platt Scaling were applied. The performance of these models was further evaluated using additional metrics, such as accuracy, recall, precision, and F1-score. The majority of the classifiers achieved accuracy rates of up to 88.85%, except for KNN, LR, and SVM, which had lower accuracy rates (up to 75.93%). Additionally, RF, XGBoost, LightGBM, and BB models performed well in terms of recall (up to 90.69%), precision (up to 88.10%), and F1-score (up to 89.12%). The results revealed that Isotonic Regression provided better calibration results for RF, XGBoost, and LightGBM, while Platt Scaling yielded superior performance for the BB classifier. Notably, the LightGBM classifier calibrated with Isotonic Regression demonstrated the best overall performance, approaching the perfectly calibrated model.

Another work done by Dahilwalkar (2021) compares the performance of different machine learning models for backorder prediction. The AUC scores of the models used in the study were LR (0.90), RF (0.95), Adaboost/ Adaptive Boosting (0.94), GBDT/ Gradient Boosted Decision Trees (0.95), and Stacking Classifier (0.88). These findings highlight the strong performance of RF, Adaboost, and GBDT models in achieving high AUC scores. LR also exhibited good performance, with a score of 0.9. The Stacking Classifier had a slightly lower AUC score of 0.88, but still demonstrated decent performance. Additionally, considering the F1 score, LR had the lowest performance among all models. RF achieved a good F1 score of

0.63, while Adaboost outperformed RF with an F1 score of 0.53. GBDT showed strong performance with an F1 score of up to 0.56. Stacking classifiers demonstrated similar performance, slightly lower than GBDT. Custom ensembles utilizing the DecisionTree Regressor achieved an F1 score of 0.62. Overall, these results provide valuable insights into the performance of different models in backorder prediction tasks.

In the most recent study, Shajalal et al. (2022) aimed to introduce explainable models in backorder prediction. Different dataset balancing techniques, such as Adaptive Synthetic Sampling/ ADASYN and SMOTE, were used in the experimental setups. Based on accuracy and AUC, the effectiveness of several classifiers was assessed. The findings demonstrated that in the majority of experimental setups, the ADASYN balancing strategy surpassed SMOTE in terms of accuracy and AUC. This shows that using ADASYN is preferable when putting real-time backorder prediction systems into practice. With a score of 0.95, the Gradient Boosting classifier had the highest accuracy among the tested traditional machine learning models. The SVM outperformed other models in terms of AUC, earning a score of 0.87. The Decision Tree, SVM, and KNN classifiers all showed strong performance in backorder prediction, with accuracy scores of 0.94, 0.85, and 0.90, respectively, and AUC scores ranging from 0.81 to 0.87. In contrast, the convolutional neural network/CNN based approach with max-pooling layers (MxCNN_100 and MxCNN_50) significantly outperformed traditional machine learning models in predicting future backorders, particularly in terms of AUC. The Gaussian Naive Bayes classifier, on the other hand, had relatively lower accuracy (0.79) and AUC (0.82) scores compared to the other models. To alleviate the overfitting issue and enhance performance, dropout layers were added to the CNN-based model. On the training set of data, the model's accuracy was 0.88 and its AUC was 0.95; on the testing set, it was 0.89 and 0.95. These results show the potential of diverse classifiers, CNN-based models, and the value of dataset balancing methods in backorder prediction tasks.

To sum up, recent studies underscore the power of different machine learning algorithms in backorder prediction, with various balancing techniques proving beneficial for imbalanced datasets. Additionally, AUC remains a vital performance metric, with models often achieving scores around 0.90 or higher. The exploration of deep learning via CNNs has shown promising results. However, there is a growing emphasis on not only predicting accurately but also optimizing for business outcomes.

*Table 02: Performance comparison among known related works on similar dataset*

| Researchers | ML techniques | AUC |
|---|---|---|
| **Santis et al. (2017)** | LR/Logistic Regression | 0.92 |
| | CART/Classification Tree | 0.94 |
| | RF/Random Forest | 0.94 |
| | GBOOST/Gradient Tree Boosting | 0.95 |
| | BLAG/Blagging | 0.95 |
| **Hájek & Abedin (2020)** | LR/Logistic Regression | 0.77 |
| | KNN/K-Nearest Neighbor | 0.74 |
| | RF/Random Forest | 0.92 |
| | NN/Neural Networks | 0.60 |
| | SVM/Support Vector Machines | 0.78 |
| **Islam & Amin (2020)** | GBM/Gradient Boosting Machine | 0.80 |
| | DRF/Distributed Random Forest | 0.79 |
| **Ntakolia et al. (2021)** | RF/Random Forest | 0.95 |
| | KNN/K-Nearest Neighbor | 0.82 |
| | NN (MLP)/Neural Networks (Multilayer perceptron) | 0.92 |
| | LR/Logistic Regression | 0.79 |
| | SVM/Support Vector Machines | 0.83 |
| | XGBoost/Extreme Gradient Boosting | 0.95 |
| | LightGBM/Light Gradient Boosting Machine | 0.95 |
| | BB/Balanced Blagging | 0.95 |
| **Dahilwalkar (2021)** | LR/Logistic Regression | 0.90 |
| | RF/Random Forest | 0.95 |
| | Adaboost/Adaptive Boosting | 0.94 |
| | GBDT/Gradient Boosted Decision Trees | 0.95 |
| | Stacking Classifier | 0.88 |
| **Shajalal et al. (2022)** | DT/Decision Tree | 0.81 |
| | SVM/Support Vector Machines | 0.87 |
| | GBM/Gradient Boosting Machine | 0.83 |
| | Gaussian Naïve Bayes | 0.82 |
| | KNN/K-Nearest Neighbor | 0.85 |
| | CNN_50/ Conventional Neural Network with ReLU | 0.94 |
| | CNN_100 | 0.95 |
| | MxCNN_50/ CNN with maximum pooling layers | 0.95 |
| | MxCNN_100 | 0.95 |

Considering the imbalanced class distribution, the evaluation of these studies is shown based on the AUC/ Area Under the Curve score, a common metric used to assess the accuracy of the models.

## 2.6 Rationale and uniqueness of the paper

Research rationale involves establishing the relevance of the research purpose by considering the current body of knowledge, including relevant theories, and acknowledging the potential practical implications (Rojon & Sauders, 2012). This sub-chapter aims to highlight the necessity and originality of the current research study in the context of backorder prediction. It highlights the constraints and gaps found in earlier studies along with the unique aspects and contributions of the current study.

Although there have been many studies on backorder prediction, there are still many problems and areas that might use more research. Existing research has developed various valuable prediction models and provided insightful information on the variables causing backorders. However, there is still a need for more comprehensive and accurate approaches that account for the complexity of contemporary supply chain dynamics. The following reasons will establish the rationale and uniqueness of this paper:

- **Integration of relevant theories and concepts:** Many of the previous research studies in backorder prediction have not thoroughly investigated the underlying theories and concepts. These studies might have emphasized on empirical analysis or practical applications without offering a thorough theoretical framework. However, this paper addresses this gap by incorporating detailed theories to enhance the understanding and interpretation of backorder prediction models.

- **Up-to-date dataset:** Although the same or similar dataset has been used in previous research, it is crucial to emphasize that the version selected for this study is the most recent one that is currently available. With the advancements in data availability, this research makes use of the comprehensive and up-to-date dataset, enabling a more accurate and realistic representation of the backorder prediction problem.

- **Rich information about the dataset:** One notable gap in previous research studies is the lack of adequate information about the dataset used for backorder prediction. The current research addresses this gap by providing detailed information about the dataset, including the attributes, data types, and their respective meanings. This

improves the research transparency and reproducibility and enables readers to gain a deeper understanding of the data and its implications for backorder prediction.

- **Dataset handling and data leakage prevention:** In the datasets provided for this study, the data has already been partitioned into distinct training and test sets in the given source. It is important to highlight that some earlier studies chose to merge these sets into a single data frame, a choice that lacked a clear rationale. Such a method raises questions concerning possible data leakage, where data from the test set may unintentionally affect the learning process. This can lead to overly optimistic performance estimations, potentially not generalizing well to new, unseen data (Krish Naik, 2020a). On the contrary, this research maintains a systematic approach and assures comparability with prior studies by handling the datasets and performing data preparation methods separately. By using this method, the analysis gains credibility, producing backorder prediction models that are both accurate and reliable.

- **Handling missing values and imbalanced datasets:** In earlier studies, a variety of methods have been used to handle missing values and imbalanced datasets. While some studies dropped all the missing values or utilized mean values for imputation, others employed the median values. Similarly, a range of different techniques were applied to address the issue of imbalanced datasets. Some earlier studies have addressed data imbalance likely to improve their findings. Preprocessing is a vital stage in machine learning tasks, but it is important to follow a systematic approach to ensure the model's robustness to the unseen data. The results in the Confusion Matrix, as shown in some earlier publications, suggest that the data may have been overly tailored to yield favorable findings, potentially compromising the model's ability to generalize to real-world scenarios. In contrast, the current research suggests a novel approach for handling missing values and addressing class imbalance, providing more robust and accurate backorder prediction models.

- **Scaling the dataset:** The dataset exhibited a significant number of outliers. Given the presence of outliers, robust scaling would be the best option, although it was found that MinMax scaling or standard scaling were mostly used in earlier studies. Such decisions could result in skewed model performance and may not be the best option

for datasets with noticeable outliers. However, in this study, Robust scaling is employed to account for these outliers and guarantee more reliable and effective model training (Singh, 2022).

- **Feature selection and correlation metrics:** Some of the previous studies in the field of backorder prediction have shown inconsistencies in the approach to feature selection. While some studies did not explicitly mention the use of correlation metrics for feature selection, others acknowledged its importance. To enable a robust analysis, this paper recognizes the significance of correlation metrics in feature selection. Along with other techniques, the research uses correlation measures as well to discover and prioritize the most influential factors that are highly correlated with the occurrence of backorders. This method improves the reliability and efficacy of the ML models used to predict backorders, providing supply chain management with more useful information and outcomes.

- **Focus on machine learning models:** The current research emphasizes the use of pure machine learning models, while some of the previous studies have investigated backorder prediction extending to neural network models. The study intends to assess the performance of different machine learning algorithms, including their hyperparameter tuning, and determine the most effective models for backorder prediction tasks. This approach offers a novel viewpoint and potentially uncovers alternative models that might rival, if not surpass, the performance of neural networks in certain scenarios.

- **Comprehensive performance evaluation metrics:** Only a few prior studies evaluated backorder prediction algorithms using an extensive set of performance evaluation parameters. Conversely, considering the imbalanced nature of the product backorder dataset, this research aims to address this limitation by incorporating multiple metrics, such as precision, recall, accuracy, F1 score, ROC/AUC score, ROC/AUC curve, Precision-Recall curve, G-mean together. This comprehensive evaluation provides a more holistic assessment of the model's performance, enabling better-informed decision-making in the supply chain.

Moreover, it is necessary to stress the importance of ensuring the thesis is easily readable and comprehensible, which is an essential aspect of knowledge dissemination and practical application. This thesis achieves these qualities by incorporating clear explanations of relevant theories, logical structure, reader-friendly presentation style, effective visual aids, and detailed information about the dataset. Along with this, the research also aims to eliminate unnecessary jargon and technical complexities that will enable readers to follow the research smoothly and interact with the findings. Therefore, this paper is not only academically rigorous but also approachable and engaging to a wider audience.

In the subsequent chapters, the research methodology, experimental setup, and result analysis will be covered. Based on the overall analysis, conclusions and necessary recommendations will also be presented. By focusing on future directions, this work will help increase supply chain management's ability to predict backorders.

# 3 Research design and methodology

A research design is the methodical organization of parameters for data collection and analysis with the goal of extrapolating findings from a sample to the entire population. It accomplishes a variety of tasks, such as minimizing expenditure, facilitating smooth scaling of research operations, collecting relevant data and techniques, providing a blueprint for plans, offering an overview to other experts, and providing direction to the research process. The characteristics of a good research design include objectivity, reliability, validity, generalizability, adequate information, flexibility, and adaptability, each of which enhances the overall quality of the study (Pandey & Pandey, 2015).

Nevertheless, research methodology, a philosophy of how an investigation should be conducted, entails a methodological approach employed in a particular area of inquiry to address the aims, objectives, and questions of the research. The methodologies explain and define the kinds of problems that are worthwhile investigating. They also provide guidelines on how to frame a problem so that it can be investigated using specific designs and procedures, as well as how to choose and develop suitable data collection and analysis methods (University of Pretoria, n.d.; Jansen & Warren, 2023).

Therefore, this design is created to ensure that the research problem is effectively handled and salient. It also explains to the readers how the researcher conducted the study by outlining the methodologies and the procedures used for data collection, analysis, and measurement, as well as the specific quantification of the variables involved.

The research methods and approach that will be used for the study are explained in this chapter. As part of the research design, the ethical issues of the study are also presented. Tools, software, and libraries employed in the research are detailed subsequently. Finally, the tactics that will be used for the processes of data gathering, sampling, and analysis are then presented.

## 3.1 Methodological approach

The two primary categories of research methods are quantitative and qualitative. The main area where these strategies diverge is in the sort of data that they gather and examine. Qualitative research concentrates on gathering non-numerical information, such as words, images, and sounds, through various techniques. It seeks to investigate subjective experiences, beliefs, and attitudes. Qualitative research aims to provide rich and complete descriptions of the phenomenon being studied in order to discover fresh insights and interpretations. In contrast, quantitative research relies on collecting numerical data and conducting statistical analysis. Its goal is to produce quantifiable, measurable facts that can be expressed numerically. Quantitative research is frequently utilized to test hypotheses, spot patterns, and make predictions based on actual data (Jansen & Warren, 2020).

Therefore, this thesis follows a quantitative research approach. This involves collecting numerical/quantitative data, advanced statistical analysis, and application of machine learning algorithms to make accurate backorder predictions in the supply chain domain. This study aims to produce objective and empirical findings that can be measured and expressed in numerical terms. The quantitative nature of this thesis allows for the potential replication of the study by other researchers, as the data and analysis can be objectively presented and interpreted.

## 3.2 Ethical considerations

The American Psychological Association/APA (2009, p. 11) highlighted the existing ethical norms to ensure accuracy, preserve participant rights, and safeguard intellectual property rights. This study will abide by ethical rules and guidelines that distinguish between legitimate and unlawful research methods. Its goal is to advance knowledge and the truth while preserving accuracy. Throughout the process of gathering data, analyzing it, and presenting the findings, the research will be conducted with the utmost objectivity and honesty. The methodology and processes of this study will be fully documented to ensure transparency and to provide readers with a clear understanding of how the study was performed. The researcher will work to maintain consistency in ideas and actions while using open and transparent methods and procedures. The appropriate credit will be given to the information's original creators and all applicable copyrights, patents, and intellectual property rights will be upheld. By providing data and findings in an impartial and responsible manner, the research will recognize and avoid plagiarism while upholding ethical standards for advancing social good (Walliman, 2017).

Therefore, by following ethical standards and principles, the research will strive to make a meaningful and ethical contribution to the field.

## 3.3 Tools/software and libraries

The study employed the Jupyter Notebook as a programming environment for Python. To conduct the data preparation, analysis, and modeling, several software libraries were utilized. For activities ranging from operating system interactions to data analysis, foundational libraries including “os”, “numpy”, and “pandas” were used. Additionally, visualization was accomplished using libraries, such as “matplotlib” and “seaborn”. For algorithm performance optimization and data imputation, libraries from “sklearn” and its extensions were used. Furthermore, resources for statistical analysis, feature processing, and model creation came from both “sklearn” and other specialized libraries. As a result of some deprecations in more recent versions of “numpy”, aliasing techniques were used to ensure smooth code execution and avoid potential reference issues.

## 3.4 Data collection

Data collection for research requires conducting systematic observations or measurements. Data gathering allows a researcher to obtain firsthand knowledge and thus gain unique insights into the study challenge, regardless of whether the research is intended for business, governmental, or academic purposes (Bhandari, 2023).

However, understanding the distinction between primary and secondary sources is essential for academic research in a variety of fields. Interview transcripts, statistical data, memoirs, and pieces of art are examples of primary sources that offer personal proof and unprocessed information. They help produce innovative and appealing scholarly research. On the other hand, the facts and commentary offered by other researchers are included in secondary sources. Journal articles, academic books, and reviews are a few instances of secondary sources. Innovative, interesting, and powerful scholarly writing and arguments require the utilization of both primary and secondary materials (Western Governors University, 2023).

Nevertheless, the dataset for this study will be gathered by the researcher from the link supplied (https://data.world/amitkishore/can-you-predict-products-back-order). It is publicly available under the heading "Predict products back-order to manage service level". This shows that the dataset supplier has already made the data accessible. As a result, the researcher will use secondary data by examining and processing this dataset in order to create machine learning models and derive insightful information for supply chain management's backorder prediction. The dataset comprises two Comma-Separated Values/CSV files, namely the training and test datasets.

### *3.4.1* Dataset overview

The data was captured at the beginning of each week as weekly snapshots (Li, 2017). Both the datasets are extensive, and each contains a range of relevant variables (23 columns) for backorder prediction, including SKU/Stock Keeping Unit, national inventory levels, lead time, in-transit quantity, forecasted sales, past sales, and other factors known to influence the occurrence of backorders. The names of the variables of both the datasets are same. While the training dataset has 1,687,861 rows, the test dataset has 242,076 rows.

*Table 03: A quick overview of the dataset*

| Count | Column name | Description | Data type |
|---|---|---|---|
| **1** | Sku | Random ID for the product | Nominal |
| **2** | national_inv | Current inventory level for the part | Numeric |
| **3** | lead_time | Transit time for product (if available) | Numeric |
| **4** | in_transit_qty | Amount of product in transit from source | Numeric |
| **5** | forecast_3_month | Forecast sales for the next 3 months | Numeric |
| **6** | forecast_6_month | Forecast sales for the next 6 months | Numeric |
| **7** | forecast_9_month | Forecast sales for the next 9 months | Numeric |
| **8** | sales_1_month | Sales quantity for the prior 1 month time period | Numeric |
| **9** | sales_3_month | Sales quantity for the prior 3 month time period | Numeric |
| **10** | sales_6_month | Sales quantity for the prior 6 month time period | Numeric |
| **11** | sales_9_month | Sales quantity for the prior 9 month time period | Numeric |
| **12** | min_bank | Minimum recommend amount to stock | Numeric |
| **13** | potential_issue | Source issue for part identified | Categorical |
| **14** | pieces_past_due | Parts overdue from source | Numeric |
| **15** | perf_6_month_avg | Source performance for prior 6 month period | Numeric |
| **16** | perf_12_month_avg | Source performance for prior 12 month period | Numeric |
| **17** | local_bo_qty | Amount of stock orders overdue | Numeric |
| **18** | deck_risk | Part risk flag | Categorical |
| **19** | oe_constraint | Part risk flag | Categorical |
| **20** | ppap_risk | Part risk flag | Categorical |
| **21** | stop_auto_buy | Part risk flag | Categorical |
| **22** | rev_stop | Part risk flag | Categorical |
| **23** | went_on_backorder | Product actually went on backorder | Categorical |

However, out of the 23 attributes, 16 are numerical data types. The “sku” column contains the nominal data type, which represents a random ID for each corresponding product. The remaining six attributes are categorical data types, with values of “Yes” or “No”, which can be treated as binary values (Li, 2017).

Among the next set of columns, the “national_inv” column indicates the current inventory level for the specific part or product. It provides information about the quantity of stock available at a national level. The column “lead_time” refers to the length of time it takes to process an order. It denotes the duration it takes for the product to be delivered or transported from the source to its destination. The quantity of the product that is currently being transported from the source is shown in the “in_transit_qty” column. It shows how many goods are being transported to their final location (Li, 2017).

Apart from this, certain attributes refer to projected sales while others refer to actual sales that have already occurred, for example, “forecast_3_month” refers to projected sales for the following three months, while “sales_9_month” shows the sales quantity for the previous nine-month period (Li, 2017).

In addition to this, the “min_bank” column indicates the minimum quantity of stock that should be kept on hand for the related product. It helps in determining the optimal stock level to meet demand and avoid shortages. The “potential_issue” column shows whether a problem or concern has been suspected with the given component or item. It acts as a warning sign to draw attention to any potential issues or difficulties that might impair the product's availability. Then the “pieces_past_due” column represents the number of parts that are overdue from the source. It lists the item quantities that have not been delivered or received by the due date, thereby causing supply chain delays or interruptions (Li, 2017).

Moving further, the “perf_6_month_avg”, and “perf_12_month_avg” columns provide information on the performance of the source or supplier over the past six and twelve months, respectively. They indicate the average performance ratings or metrics associated with the source's ability to meet delivery schedules, quality standards, and other performance criteria. However, some datapoints in these two columns are observed with a loaded dummy value of -99 as they have not been scored. The next column “local_bo_qty” represents the number of stock orders that are currently overdue at the local level. It provides insights into the item quantity that has not been fulfilled or delivered within the expected timeframe at the local distribution or storage locations (Li, 2017).

Moreover, the columns “deck_risk”, “oe_constraint”, “ppap_risk”, “stop_auto_buy”, and “rev_stop” describe various risk factors and limitations related to the product or its

procurement process. They act as warning signs to indicate whether the product carries certain risks, such as being a deck risk (prone to obsolescence), having operational constraints, exhibiting production part approval process/PPAP risks, the need for manual intervention to stop automatic buying, or experiencing revenue interruptions (Li, 2017).

Finally, the "went_on_backorder" column indicates whether a product went on backorder or not, meaning it was available or unavailable for immediate delivery or supply. It serves as the target variable for backorder prediction (Li, 2017).

Overall, the dataset offers a wealth of data for investigating the connections and trends between predictor variables and the occurrence of backorders. It is possible to gain important insights to improve supply chain management and service levels through rigorous analysis and modeling. The dataset source and other pertinent information are clearly stated in the thesis to ensure transparency and encourage replication and to make it possible for other researchers to access and verify the results.

## 3.5 Data preprocessing techniques

Real-world data often require preprocessing to ensure their quality for analysis. Data preprocessing addresses various issues present in datasets to maintain their integrity and dependability (Bhagat, 2022).

To assure the dataset's quality and consistency throughout the study, both the training and test datasets have undergone meticulous curation. Every action in this study has been performed with a careful awareness of the potential for data leakage. The same procedures have been used to pre-process the training and test sets individually from the beginning. This standardized and distinct processing ensures that the model's performance on the test set is accurately representative of how well it performs on unobserved data. During analysis, it is crucial to preserve data dependability and integrity. To guarantee the quality and validity of the findings, any inconsistencies in the dataset have been addressed using the proper data preprocessing techniques. The project intends to create reliable machine learning models for backorder prediction in the supply chain sector using this dataset.

For example, both the datasets collected for this research contain null values, outliers and other inconsistencies that require proper techniques to handle. In addition to that, choosing important features, rather than having all 23 columns, for the machine learning models is also crucial as unimportant or redundant ones can degrade an algorithm's performance, accuracy, and efficiency (Heavy.AI, n.d.). Hence, in the data preparation phase, a series of preprocessing techniques have been applied to both the datasets. These techniques include replacing and transforming values, missing values imputation, outlier handling, choosing the important features through different tests, scaling, etc.

However, an exception has been made in resampling only the training data to address class imbalance in the target column. The test set remained unaltered in this respect, ensuring it mirrors the population of interest, as modifying it could yield misleading results (Prof. Mezei, personal communication, August 08, 2023).

This section largely highlights the preprocessing methods used on the training dataset, which are concisely and clearly described by the provided code snippets and supplementary images. This choice has been made for the following reasons:

- **Preprocessing step consistency:** The test set's preprocessing steps mimic the training set's preprocessing steps exactly. As a result, outlining the training set's procedures effectively explains the strategy without becoming tedious.

- **Emphasis on methodology over repetition:** By drawing attention to the preprocessing of the training set, the emphasis is put on methodology and methods rather than code repetition. This makes sure that readers can understand the crucial preprocessing techniques without being overloaded with unnecessary details.

- **Effective presentation:** Aligning with academic conventions, the choice to detail only the training set's preprocessing improves readability and concision.

Moreover, the test set has been treated to the exact identical methods, except resampling, assuring scientific rigor and consistency throughout the investigation, despite the decision to only discuss the training set's preprocessing stages.

### *3.5.1* Data descriptives- key insights

Before starting any preprocessing or modeling operations, it is essential to thoroughly understand the dataset. The primary patterns and variability of dataset properties are revealed by descriptive statistics. The following figure shows key insights from the numerical columns:

```
### Some descriptive statistics

train_df.describe().transpose()
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| national_inv | 1687860.0 | 496.111782 | 29615.233831 | -27256.0 | 4.00 | 15.00 | 80.00 | 12334404.0 |
| lead_time | 1586967.0 | 7.872267 | 7.056024 | 0.0 | 4.00 | 8.00 | 9.00 | 52.0 |
| in_transit_qty | 1687860.0 | 44.052022 | 1342.741731 | 0.0 | 0.00 | 0.00 | 0.00 | 489408.0 |
| forecast_3_month | 1687860.0 | 178.119284 | 5026.553102 | 0.0 | 0.00 | 0.00 | 4.00 | 1427612.0 |
| forecast_6_month | 1687860.0 | 344.986664 | 9795.151861 | 0.0 | 0.00 | 0.00 | 12.00 | 2461360.0 |
| forecast_9_month | 1687860.0 | 506.364431 | 14378.923562 | 0.0 | 0.00 | 0.00 | 20.00 | 3777304.0 |
| sales_1_month | 1687860.0 | 55.926069 | 1928.195879 | 0.0 | 0.00 | 0.00 | 4.00 | 741774.0 |
| sales_3_month | 1687860.0 | 175.025930 | 5192.377625 | 0.0 | 0.00 | 1.00 | 15.00 | 1105478.0 |
| sales_6_month | 1687860.0 | 341.728839 | 9613.167104 | 0.0 | 0.00 | 2.00 | 31.00 | 2146625.0 |
| sales_9_month | 1687860.0 | 525.269701 | 14838.613523 | 0.0 | 0.00 | 4.00 | 47.00 | 3205172.0 |
| min_bank | 1687860.0 | 52.772303 | 1254.983089 | 0.0 | 0.00 | 0.00 | 3.00 | 313319.0 |
| pieces_past_due | 1687860.0 | 2.043724 | 236.016500 | 0.0 | 0.00 | 0.00 | 0.00 | 146496.0 |
| perf_6_month_avg | 1687860.0 | -6.872059 | 26.556357 | -99.0 | 0.63 | 0.82 | 0.97 | 1.0 |
| perf_12_month_avg | 1687860.0 | -6.437947 | 25.843331 | -99.0 | 0.66 | 0.81 | 0.95 | 1.0 |
| local_bo_qty | 1687860.0 | 0.626451 | 33.722242 | 0.0 | 0.00 | 0.00 | 0.00 | 12530.0 |

***Figure 04: Summary statistics***

Here some summary statistics are visible including row counts, each column's minimum and maximum values, mean and standard deviations, percentile-wise data distributions. Along with these, the following observations are also made:

- Due to their wide value ranges, columns, such as "national_inv" and "in_transit_qty" may indicate probable outliers.
- The average lead time for items is a week, however some might take up to 52 days.
- Values of -99 in the Performance columns suggest placeholders for unrecorded data.
- Higher outliers in longer-term Forecasts and Sales columns could signify seasonality or exceptional occurrences.
- Variation in the "local_bo_qty" reveals occasional, considerable local backorders.

These insights will guide subsequent data preparations and modeling decisions.

### *3.5.2* Handling loaded dummy values and missing values

As discussed above, in the performance columns ("perf_6_month_avg" and "perf_12_month_avg"), values that have not been evaluated are shown as -99 which can be treated as null/missing values (Dalvi, 2021). The author of this study produced codes to transform these dummy values into null values.

However, now that the dummy values are transformed into null values, all the null values of the entire dataset can be treated in a systematic approach. The "lead_time" column in the training dataset has 40,837 (4.00%) missing values, while some columns have one missing value each. On the other hand, regarding the transformed performance columns, the "perf_6_month_avg" and "perf_12_month_avg" have now 55,761 (5.47%) and 50,658 (4.96%) null values, respectively.


***Figure 05: Missing values in "lead_time" and Performance columns***

Imputation entails replacing missing values with approximations, whereas deletion is eliminating any rows or columns that include any missing values. The technique that will be used will rely on the type of data and the specifications of the study (Linkedin, 2023).

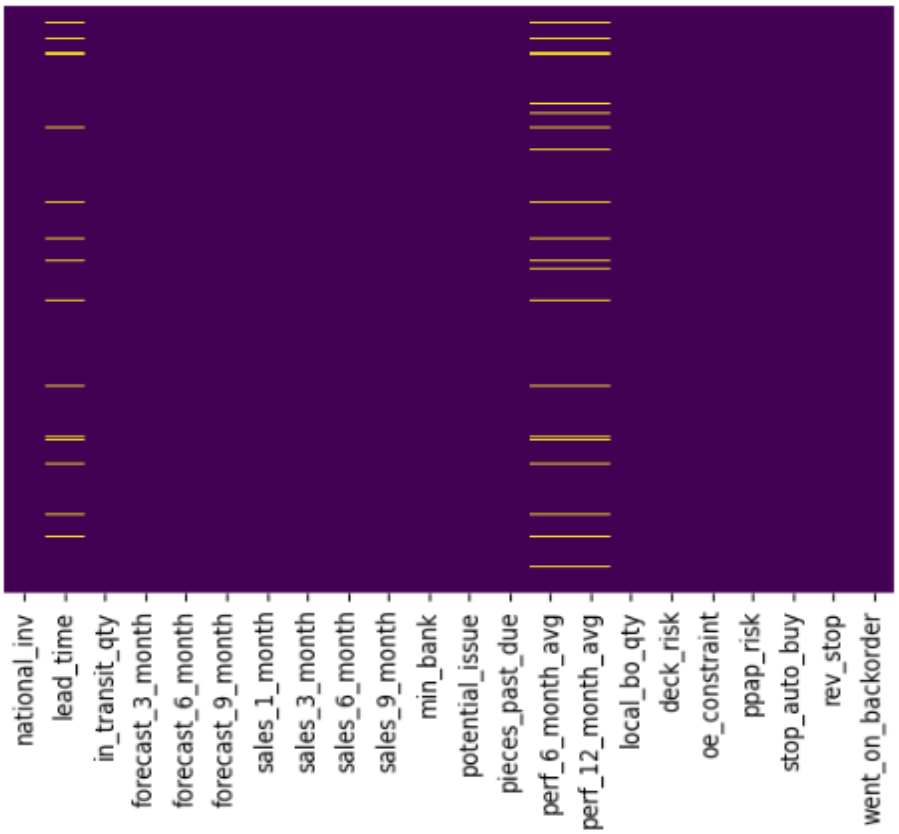



***Figure 06: Examining the randomness of the missing values***

As evident from the above figure, the missing values in the "national_inv" column and the Performance columns coincide. This indicates that the missing values in these three columns are MNAR/Missing Not At Random. According to Little & Rubin (2002), when data is MNAR, the fact that a specific piece of data is missing by itself conveys information, and simply discarding these observations could lead to incorrect or biased inferences. To ensure data integrity, the author of this research has decided to handle these missing values in the following manner:

- The associated rows from the dataset for any columns with only one missing value will be deleted. This strategy aids in maintaining overall data completeness.
- On the other hand, For the "lead_time" and Performance columns, each has a large number of MNAR missing values. Simply eliminating such a sizable amount of data would result in a major loss of information. In this situation, the author will take a different tack and use the proper technique to fill in the missing numbers.

The Multivariate Imputation by Chained Equations/MICE method is chosen by the thesis to fill in these missing data. Notably, MICE is a preferred method for handling missing data, especially when the data is MNAR. In order to impute the missing values using multiple imputations, this method takes advantage of the relationships between the variables. Thus, it lessens the impact of missing values while preserving the statistical properties of the data (Soni, 2023). The MICE process is outlined below using lucid chart:

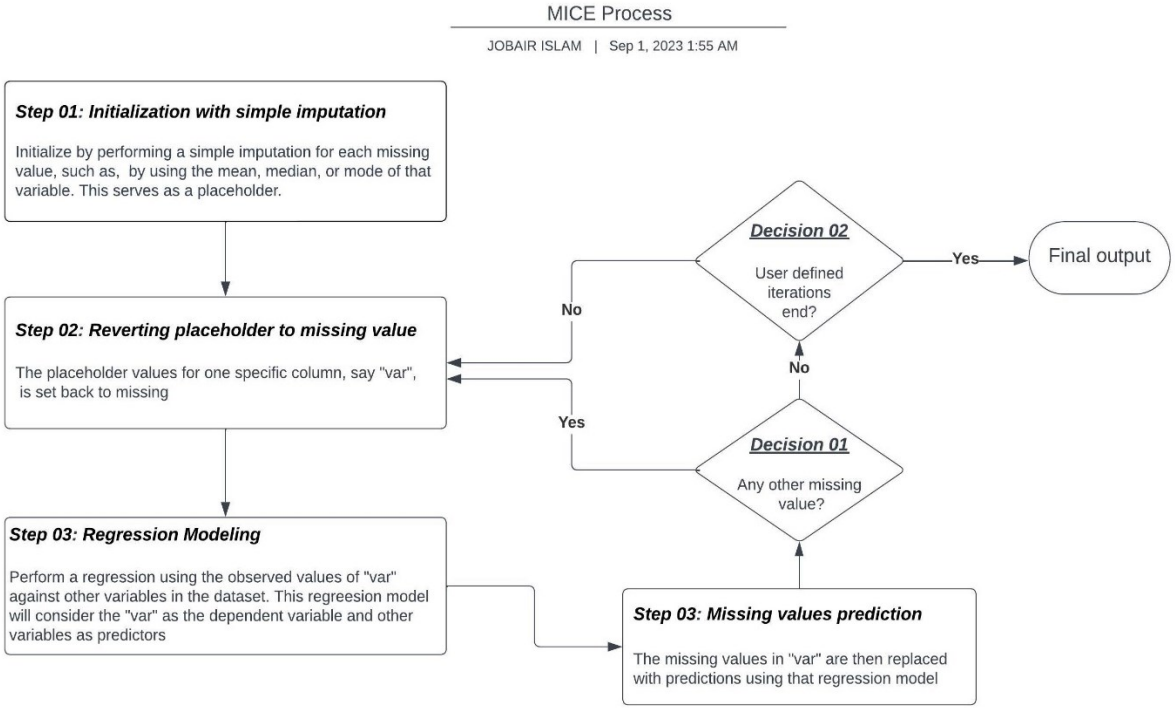



***Figure 07: MICE process (Shalev, 2018)***

The MICE method imputes missing values through a series of iterative predictive models. Every variable that has missing values is treated as the dependent variable in MICE, with all other variables serving as its predictors. The iterations continue until the imputed values seem to have converged (they no longer change significantly with further iterations). Usually, the process iterates until imputation for each of the required variables is accomplished. Even though the required iterations may differ, achieving convergence often does not require more than five iterations (Wilson, 2021). These techniques are expected to effectively address missing values, thereby preserving the quality and utility of the dataset for further research and modeling.

***NB:** Relevant Codes for "Handling loaded dummy values and missing values" are provided in the 'Appendix 01'.*

### *3.5.3* Outlier detection and handling

According to the NIST/National Institute of Standards and Technology (n.d.), an outlier is a data point that significantly differs from others in a dataset. Outliers may represent errors, such as mistakes in data coding or experimental procedures. If outliers are errors, these values should either be corrected or removed from the analysis. However, if they are not errors, they could arise from natural variations or reveal valuable scientific insights. Instead of outright deletion, employing robust statistical methods to handle them effectively is more appropriate. Hence, in the preliminary stages of data exploration, it is essential to identify and address potential outliers, as they can significantly influence the statistical measures and the machine learning models.

To achieve this for the numerical columns, box plots and histograms were initially utilized as visualization tools. These plots, however, turned out to be less informative due to the presence of extreme outliers, which often distorted the visual representation.

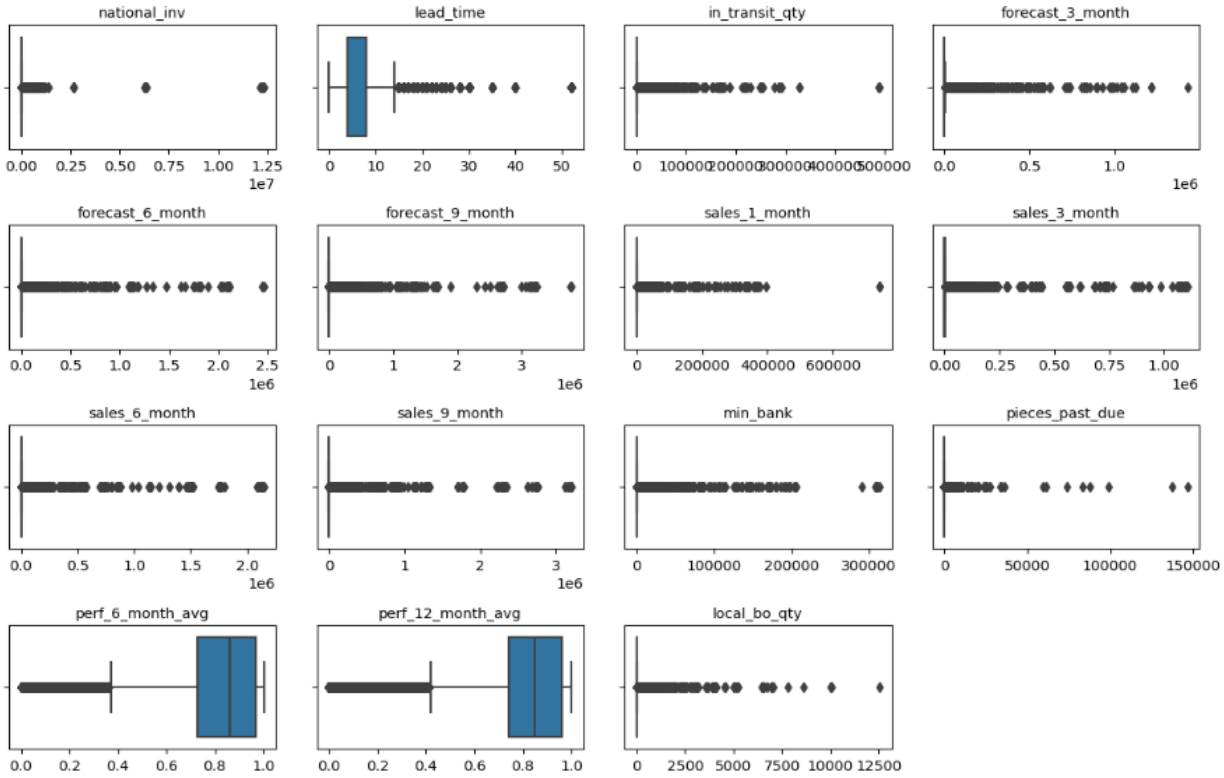



***Figure 08: Boxplots of the numerical columns***

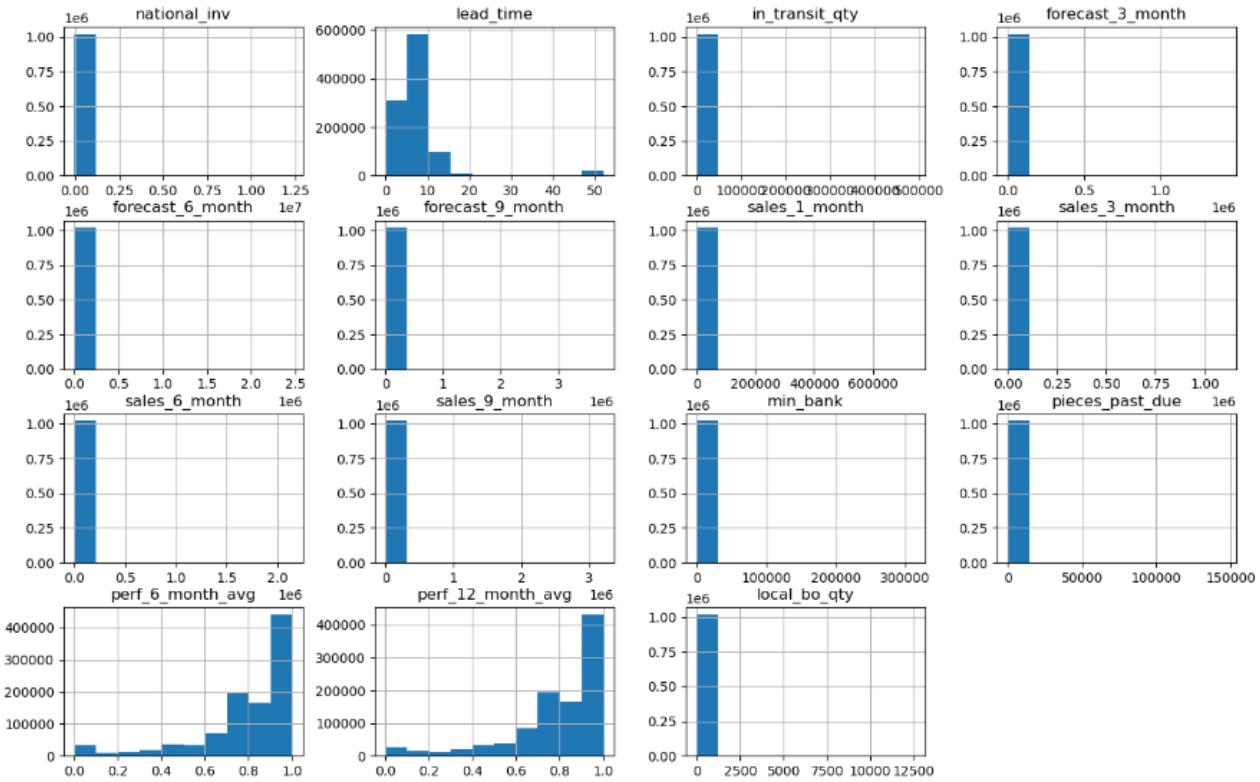



***Figure 09: Histogram of the numerical columns***

Recognizing the limitations of the box plots and histogram in this particular dataset, to further understand the extent and nature of these outliers, an additional method, percentile testing, has been employed. This method provides a more quantitative assessment with a more detailed perspective by examining the data distribution at specific percentile levels and thus helps identify the potential outliers.

For example, through the percentile test, it is now observed that for most of the columns, values from the 99th percentile are extremely larger compared to other percentiles. Therefore, these can be treated as extreme outliers. Thus, after a thorough analysis using this approach, it became evident that there were substantial deviations in the extreme values.

Additionally, it has also been observed that values of the “pieces_past_due” and “local_bo_qty” columns remained at zero up to the 97th percentile, questioning their influence on the target variable. This will be helpful in feature selection for the ML models.

The percentile test results are provided in “Table 04” below:

***Table 04: Percentile test result of the numerical columns***

| Count | Column name | Percentiles | | | | |
|---|---|---|---|---|---|---|
| | | $0^{th}$-$96^{th}$ | $97^{th}$ | $98^{th}$ | $99^{th}$ | $100^{th}$ |
| **1** | national_inv | -27256.0, 1960.0 | 2780.0 | 4418.0 | 8241.79 | 12334404.0 |
| **2** | lead_time | 0.0, 13.0 | 15.0 | 21.0 | 52.0 | 52.0 |
| **3** | in_transit_qty | 0.0, 200.0 | 285.0 | 462.0 | 985.0 | 489408.0 |
| **4** | forecast_3_month | 0.0, 835.0 | 1179.0 | 1872.0 | 3804.0 | 1427612.0 |
| **5** | forecast_6_month | 0.0, 1600.0 | 2230.0 | 3555.0 | 7146.0 | 2461360.0 |
| **6** | forecast_9_month | 0.0, 2350.0 | 3300.0 | 5193.86 | 10500.0 | 3777304.0 |
| **7** | sales_1_month | 0.0, 275.0 | 378.0 | 578.0 | 1105.0 | 741774.0 |
| **8** | sales_3_month | 0.0, 884.0 | 1208.0 | 1855.0 | 3542.0 | 1105478.0 |
| **9** | sales_6_month | 0.0, 1728.0 | 2390.0 | 3651.0 | 6976.0 | 2146625.0 |
| **10** | sales_9_month | 0.0, 2646.0 | 3621.0 | 5572.0 | 10563.79 | 3205172.0 |
| **11** | min_bank | 0.0, 280.0 | 375.0 | 562.0 | 1074.0 | 313319.0 |
| **12** | pieces_past_due | 0.0, 0.0 | 0.0 | 1.0 | 16.0 | 146496.0 |
| **13** | perf_6_month_avg | 0.0, 1.0 | 1.0 | 1.0 | 1.0 | 1.002 |
| **14** | perf_12_month_avg | 0.0, 0.99 | 1.0 | 1.0 | 1.0 | 1.0 |
| **15** | local_bo_qty | 0.0, 0.0 | 0.0 | 1.0 | 4.0 | 12530.0 |

In light of these findings, the decision has been made to eliminate the top 1% of outliers from the entire training set to maintain data integrity and ensure a robust modeling process.

***NB:*** *Relevant codes for "Outlier detection and handling" are provided in the 'Appendix 02'*

### *3.5.4* Binarization of the categorical features

According to Brownlee (2017), categorical variables consist of label values instead of numerical ones. Numerous machine learning techniques need numerical input data. Hence, categorical data must be transformed into a numerical representation. One of these methods is called "binarization" which converts categorical data into binary (0 or 1) values to make it compatible with algorithms that require numerical input. A categorical feature, especially one with two categories, is binarized when it is converted into a binary number format.

The benefits of binarization are as follows:

- **Model compatibility:** Converts categorical data to a numerical representation that is easily processed by algorithms.

- **Data simplification:** By converting data to a binary format, it is often possible to minimize the complexity of the data, hence facilitating its examination.

```
### Checking unique values in each categorical variable

for col in train_cat:
    print(f'{col}: {train_cat[col].unique()}')

## We can see here, all our catergorical features have 02 values each.
```

```
potential_issue: ['No' 'Yes']
deck_risk: ['No' 'Yes']
oe_constraint: ['No' 'Yes']
ppap_risk: ['No' 'Yes']
stop_auto_buy: ['Yes' 'No']
rev_stop: ['No' 'Yes']
went_on_backorder: ['No' 'Yes']
```

***Figure 10: Unique values of the categorical features***

Figure 10 illustrates that the categorical variables of the dataset have two options “Yes” and “No”. The author of this thesis has decided to transform those values into binary format so that “Yes” is represented by 1 and “No” by 0.

***NB:*** *Relevant codes for “Binarization of the categorical features” are provided in the ‘Appendix 03’*

### *3.5.5* Feature selection process for ML models

According to Gupta (2020), the objective of feature selection techniques in machine learning is to locate the ideal set of features that allows for the development of accurate models of the phenomenon being studied. However, a comprehensive set of methods has been utilized to select the final features for machine learning models in this study.

#### *3.5.5.1 Cardinality checking and dropping the "sku" column*

When a categorical feature has an excessive number of unique values, this is referred to as high cardinality (Sangani, 2021). In the initial EDA/Exploratory Data Analysis phase, a cardinality check has been conducted that involves evaluating the number of distinct values in each feature. The "sku" column, upon this cardinality test, has been found to have a unique value for each row, indicating high cardinality. Given its nature as an identifier and the potential issues associated with high cardinality in machine learning models, this column has subsequently been dropped from the dataset.

***NB:*** *Relevant codes for "Cardinality checking and dropping the 'sku' column" are provided in the 'Appendix 04'*

#### *3.5.5.2 Bivariate analysis of categorical features with the target column*

At this stage, each of the categorical features has been evaluated in relation to the target variable. The observations from this analysis are as follows:

- Products identified with "potential_issue" as "Yes" are considerably more likely to go into backorder compared to those labeled as "No".
- Products labeled with "deck_risk" as "Yes" are marginally less likely to go into backorder compared to those labeled as "No".
- Products identified with "oe_constraint" as "Yes" notably more tend to go into backorder compared to those labeled as "No"
- Products identified with "ppap_risk" as "Yes" are slightly more likely to go into backorder compared to those labeled as "No".
- Products with "stop_auto_buy" set to "No" or "Yes" have roughly the same likelihood of going into backorder.
- When "rev_stop" is "No", there is a 100% chance that the product did not go into backorder. However, domain knowledge suggests that "rev_stop" does not determine a product's backorder status; instead, backorders influence revenue generation.

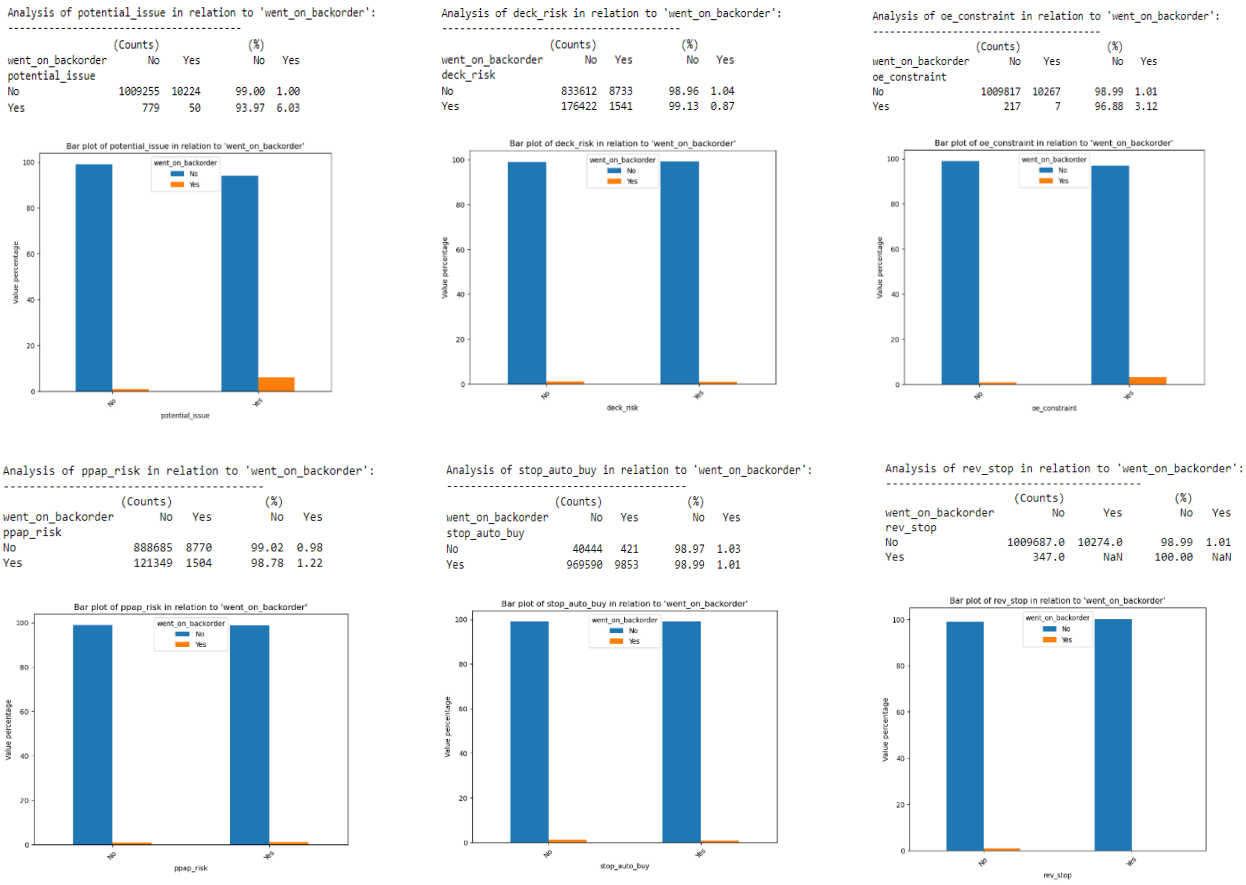



***Figure 11: Bivariate analysis of categorical features with the target column***

Overall, the proportion of products moving into backorder in each category closely matches the relevant class ratios. The possibility of the categorical variable's significance in forecasting backorder status is suggested by this alignment. The analysis will continue as it moves forward to explore these connections further.

***NB:*** *Relevant codes for "Bivariate analysis of categorical features with the target column" are provided in the 'Appendix 05'*

### *3.5.5.3 Chi-squared test*

The Pearson's Chi-squared test, often known as the Chi-squared test, is a statistical method used to determine whether there is a statistically significant relationship between two categorical variables. It contrasts predicted frequencies (the expected counts if there was no association between the variables) with observed frequencies (the actual counts).

Mathematically, the expected frequency for each cell in a contingency table is calculated as:

$$E_{ij} = \frac{Row\ i\ Total\ \times Column\ j\ Total}{Grand\ Total}$$

Where, $O_{ij}$ denotes the observed frequency, and $E_{ij}$ denotes the expected frequency.

Then, the Chi-squared statistic, $X^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$

The objective of this test is to determine whether a disparity between projected and actual data is caused by coincidence or if there is a strong connection between the variables being examined. Hence, the Chi-squared test is a great technique for helping the researchers understand and evaluate the relationship between two categorical variables (Biswal, 2023).

In this study, the test is applied to determine the relationship between all the categorical variables and the target column. The hypotheses set forth are:

- **Null hypothesis, $H_0$ :** There is no association between the categorical variable and the target column.
- **Alternative hypothesis, $H_1$:** There is a significant relationship between the categorical variable and the target column.

The p-value serves as the criterion for hypothesis testing, where the significance level, often represented as alpha (α), is typically set at 0.05. The decision rule is as follows:

- If the p-value exceeds the significance level, $H_0$ will not be rejected.
- Alternatively, if the p-value is less than or equal to the significance level, $H_0$ will be rejected in favor of $H_1$.

However, results from the Chi-squared test reveal that almost all the categorical variables, except "rev_stop" and "stop_auto_buy", have a statistically significant impact on the likelihood of products going into backorder. Further exploration of these relationships will be conducted in subsequent Sub-subsections.

***NB:*** *Relevant codes for "Chi-squared test" are provided in the 'Appendix 06'*

### *3.5.5.4 Bivariate analysis of numerical features with the target column*

At this stage, each of the numerical features has been evaluated in relation to the target variable. The initial approach involved the usage of box plots. However, even though top 1% of outliers from the entire dataset were removed earlier, the dataset still contains outliers, and their presence limited the clarity and interpretability of these plots and did not offer that much information about the relationship.

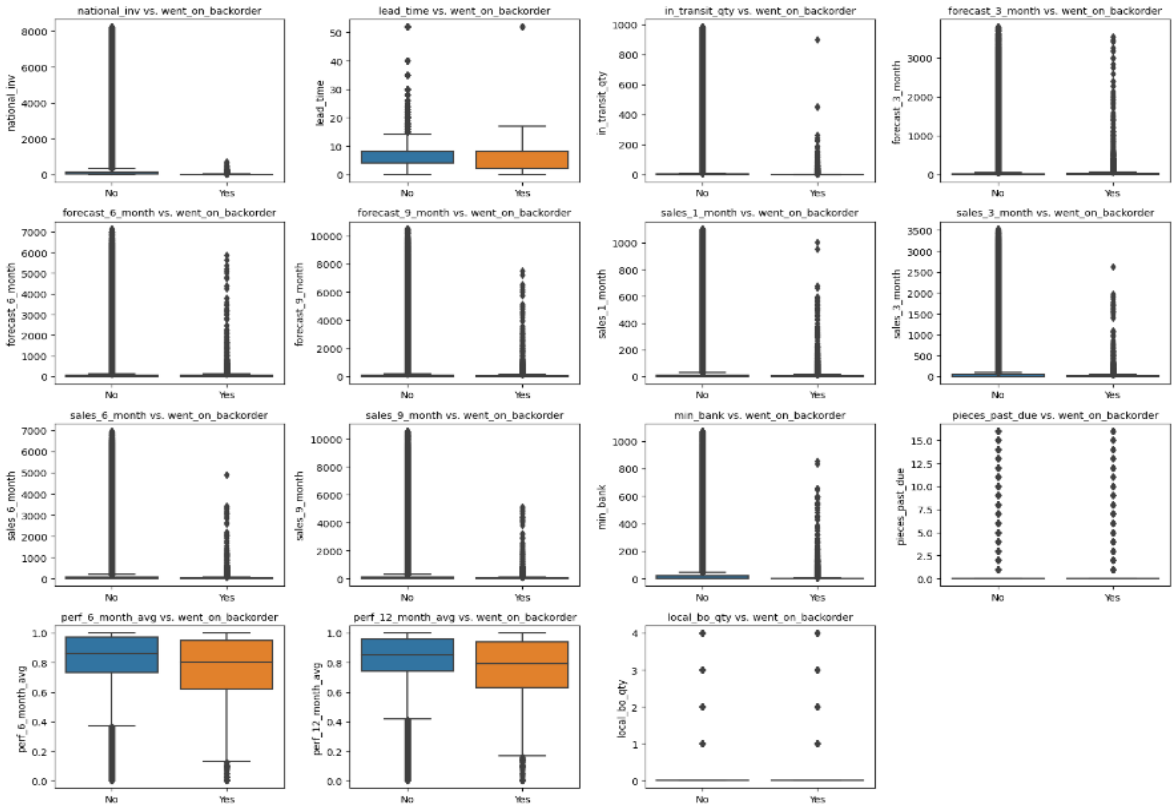
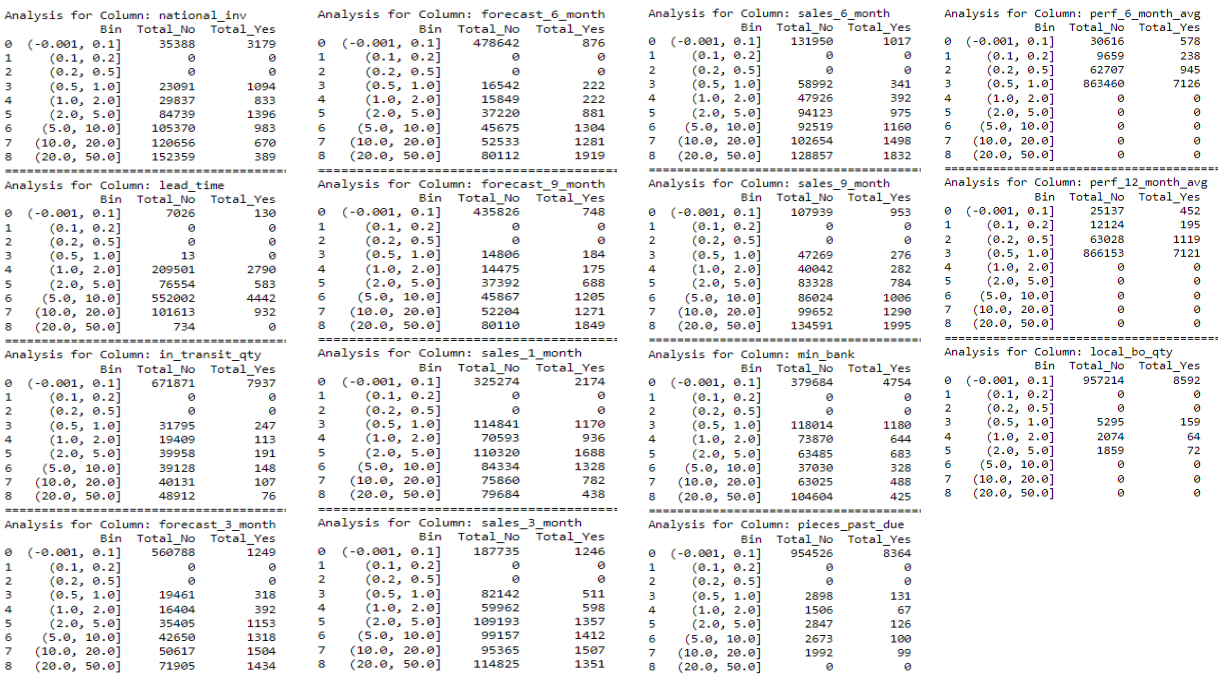



***Figure 12: Box plots for bivariate analysis of numerical features with the target column***

Subsequently, a detailed bivariate analysis has been conducted by binning the numerical values of the respective columns into specific intervals, ranging from 0 to 50. The objective was to observe how different numerical ranges influence the likelihood of products transitioning into backorder. This analysis resulted in the following key observations:

- Products that went on backorder tend to have lower “national_inv” values. Conversely, those who did not go on backorder exhibited higher 'national_inv' values.
- Products that went on backorder experienced longer “lead_time” compared to products that did not go on backorder.

- "in_transit_qty", Forecast and Sales columns, all have similar analysis results. Products with lower values in these columns tend to be more likely to go on backorder, while products with higher values tend to avoid backorders.
- Products with higher "min_bank" values are less likely to go on backorder.
- Products with higher values in the "pieces_past_due" column are more likely to go on backorder. However, percentile testing result in "Table 04" showed that the value of this column remained at zero up to the 97th percentile, suggesting its limited impact.
- Products with lower average performance in the past 6 or 12 months tend to be more backordered, while those with higher average performance tend to be less likely.
- Products with higher values of "local_bo_qty" are less likely to go on backorder. Yet, similar to "pieces_past_due", the result from percentile testing in "Table 04" revealed that values remained at zero up to the 97th percentile, questioning its influence.



Analysis for Column: national_inv

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 35388 | 3179 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 23091 | 1094 |
| 4 | (1.0, 2.0] | 29837 | 833 |
| 5 | (2.0, 5.0] | 84739 | 1396 |
| 6 | (5.0, 10.0] | 105370 | 983 |
| 7 | (10.0, 20.0] | 120656 | 670 |
| 8 | (20.0, 50.0] | 152359 | 389 |

Analysis for Column: lead_time

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 7026 | 130 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 13 | 0 |
| 4 | (1.0, 2.0] | 209501 | 2790 |
| 5 | (2.0, 5.0] | 76554 | 583 |
| 6 | (5.0, 10.0] | 552002 | 4442 |
| 7 | (10.0, 20.0] | 101613 | 932 |
| 8 | (20.0, 50.0] | 734 | 0 |

Analysis for Column: in_transit_qty

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 671871 | 7937 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 31795 | 247 |
| 4 | (1.0, 2.0] | 19409 | 113 |
| 5 | (2.0, 5.0] | 39958 | 191 |
| 6 | (5.0, 10.0] | 39128 | 148 |
| 7 | (10.0, 20.0] | 40131 | 107 |
| 8 | (20.0, 50.0] | 48912 | 76 |

Analysis for Column: forecast_3_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 560788 | 1249 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 19461 | 318 |
| 4 | (1.0, 2.0] | 16404 | 392 |
| 5 | (2.0, 5.0] | 35405 | 1153 |
| 6 | (5.0, 10.0] | 42650 | 1318 |
| 7 | (10.0, 20.0] | 50617 | 1504 |
| 8 | (20.0, 50.0] | 71905 | 1434 |

Analysis for Column: forecast_6_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 478642 | 876 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 16542 | 222 |
| 4 | (1.0, 2.0] | 15849 | 222 |
| 5 | (2.0, 5.0] | 37220 | 881 |
| 6 | (5.0, 10.0] | 45675 | 1304 |
| 7 | (10.0, 20.0] | 52533 | 1281 |
| 8 | (20.0, 50.0] | 80112 | 1919 |

Analysis for Column: forecast_9_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 435826 | 748 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 14806 | 184 |
| 4 | (1.0, 2.0] | 14475 | 175 |
| 5 | (2.0, 5.0] | 37392 | 688 |
| 6 | (5.0, 10.0] | 45867 | 1205 |
| 7 | (10.0, 20.0] | 52204 | 1271 |
| 8 | (20.0, 50.0] | 80110 | 1849 |

Analysis for Column: sales_1_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 325274 | 2174 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 114841 | 1170 |
| 4 | (1.0, 2.0] | 70593 | 936 |
| 5 | (2.0, 5.0] | 110320 | 1688 |
| 6 | (5.0, 10.0] | 84334 | 1328 |
| 7 | (10.0, 20.0] | 75860 | 782 |
| 8 | (20.0, 50.0] | 79684 | 438 |

Analysis for Column: sales_3_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 187735 | 1246 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 82142 | 511 |
| 4 | (1.0, 2.0] | 59962 | 598 |
| 5 | (2.0, 5.0] | 109193 | 1357 |
| 6 | (5.0, 10.0] | 99157 | 1412 |
| 7 | (10.0, 20.0] | 95365 | 1507 |
| 8 | (20.0, 50.0] | 114825 | 1351 |

Analysis for Column: sales_6_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 131950 | 1017 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 58992 | 341 |
| 4 | (1.0, 2.0] | 47926 | 392 |
| 5 | (2.0, 5.0] | 94123 | 975 |
| 6 | (5.0, 10.0] | 92519 | 1160 |
| 7 | (10.0, 20.0] | 102654 | 1498 |
| 8 | (20.0, 50.0] | 128857 | 1832 |

Analysis for Column: sales_9_month

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 107939 | 953 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 47269 | 276 |
| 4 | (1.0, 2.0] | 40042 | 282 |
| 5 | (2.0, 5.0] | 83328 | 784 |
| 6 | (5.0, 10.0] | 86024 | 1006 |
| 7 | (10.0, 20.0] | 99652 | 1290 |
| 8 | (20.0, 50.0] | 134591 | 1995 |

Analysis for Column: min_bank

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 379684 | 4754 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 118014 | 1180 |
| 4 | (1.0, 2.0] | 73870 | 644 |
| 5 | (2.0, 5.0] | 63485 | 683 |
| 6 | (5.0, 10.0] | 37030 | 328 |
| 7 | (10.0, 20.0] | 63025 | 488 |
| 8 | (20.0, 50.0] | 104604 | 425 |

Analysis for Column: pieces_past_due

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 954526 | 8364 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 2898 | 131 |
| 4 | (1.0, 2.0] | 1506 | 67 |
| 5 | (2.0, 5.0] | 2847 | 126 |
| 6 | (5.0, 10.0] | 2673 | 100 |
| 7 | (10.0, 20.0] | 1992 | 99 |
| 8 | (20.0, 50.0] | 0 | 0 |

Analysis for Column: perf_6_month_avg

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 30616 | 578 |
| 1 | (0.1, 0.2] | 9659 | 238 |
| 2 | (0.2, 0.5] | 62707 | 945 |
| 3 | (0.5, 1.0] | 863460 | 7126 |
| 4 | (1.0, 2.0] | 0 | 0 |
| 5 | (2.0, 5.0] | 0 | 0 |
| 6 | (5.0, 10.0] | 0 | 0 |
| 7 | (10.0, 20.0] | 0 | 0 |
| 8 | (20.0, 50.0] | 0 | 0 |

Analysis for Column: perf_12_month_avg

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 25137 | 452 |
| 1 | (0.1, 0.2] | 12124 | 195 |
| 2 | (0.2, 0.5] | 63028 | 1119 |
| 3 | (0.5, 1.0] | 866153 | 7121 |
| 4 | (1.0, 2.0] | 0 | 0 |
| 5 | (2.0, 5.0] | 0 | 0 |
| 6 | (5.0, 10.0] | 0 | 0 |
| 7 | (10.0, 20.0] | 0 | 0 |
| 8 | (20.0, 50.0] | 0 | 0 |

Analysis for Column: local_bo_qty

| | Bin | Total_No | Total_Yes |
|---|---|---|---|
| 0 | (-0.001, 0.1] | 957214 | 8592 |
| 1 | (0.1, 0.2] | 0 | 0 |
| 2 | (0.2, 0.5] | 0 | 0 |
| 3 | (0.5, 1.0] | 5295 | 159 |
| 4 | (1.0, 2.0] | 2074 | 64 |
| 5 | (2.0, 5.0] | 1859 | 72 |
| 6 | (5.0, 10.0] | 0 | 0 |
| 7 | (10.0, 20.0] | 0 | 0 |
| 8 | (20.0, 50.0] | 0 | 0 |

***Figure 13: Detailed bivariate analysis by binning the numerical values of the columns***

However, these findings are preliminary even though they offer insightful information. To validate these results and understand their implications in a prediction model, more in-depth investigation has been conducted.

***NB:*** *Relevant codes for "Bivariate analysis of numerical features with the target column" are provided in the 'Appendix 07'*

#### *3.5.5.5 Correlation matrix*

The Correlation Coefficient is a statistical measure of the association or the relationship that exists between two variables. In other words, it is only a single number (between -1 and +1) that indicates how closely two variables are connected and the extent to which changes in one affect the other. The interpretation of correlation strength varies across fields (Fernando, 2023). By conducting a correlation analysis between the target variable and the other features in the dataset, the researcher aimed to identify relevant variables that exhibit significant relationships with the target column. The correlation analysis is a fundamental step in feature selection, as it provides insights into the associations between variables (Krish Naik, 2020b).

**a) Pearson correlation:** According to Guo (2021), Pearson correlation, often represented as $r$, measures the linear relationship between two continuous variables. Mathematically, the Pearson correlation coefficient is represented as:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

Where,

$x_i$ and $y_i$ are the individual data points of x-variable and y-variable respectively,

$\bar{x}$ and $\bar{y}$ are the means of x-variable and y-variable respectively.

After binarizing the target column, the Pearson correlation test has been conducted with all the numerical columns. The focus on the "went_on_backorder" target variable has revealed an absence of any evident linear relationship with the other variables. Additional observations include:

- A distinct positive correlation between the Sales and Forecast columns.
- "in_transit_qty" has a strong positive correlation with the "min_bank", Forecast and Sales columns.
- Performance columns, "pieces_past_due", "lead_time", "local_bo_qty", and "national_inv" all show rather weak associations.
- All the Sales columns, Forecast columns and Performance columns showed

internally strong positive correlation. Sales columns have coefficients ranging from 0.93 to 0.99, while the Forecast columns have between 0.92 and 0.98, and Performance columns have a score of 0.94.

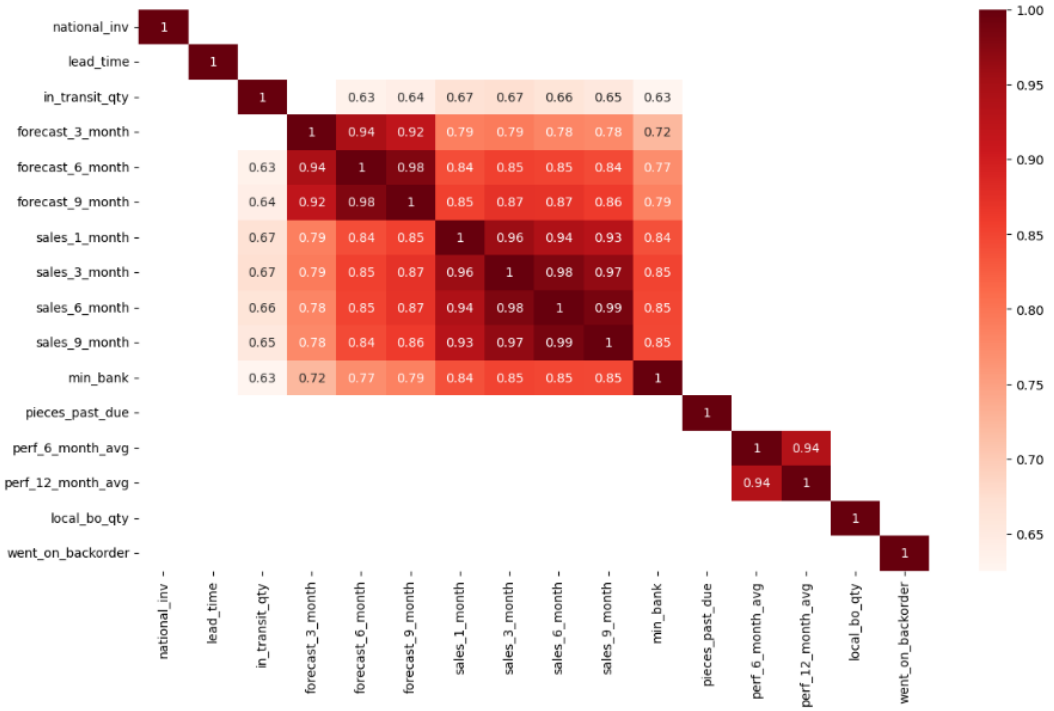



***Figure 14: Pearson correlation matrix***

In general, the analysis shows strong positive correlations between several features, suggesting that there are dependencies within the dataset. These relationships must be taken into consideration when creating predictive models to avoid multicollinearity issues.

**b) Spearman correlation:** Guo (2021), also describes how Spearman's rank correlation coefficient works. It, often represented as $\rho$, measures the monotonic relationships between variables based on their ranked values. It is defined as:

$$\rho = 1 - \frac{6\Sigma\, d_i^2}{n(n^2-1)}$$

Where, $d_i$ is the difference between two ranks of each observation and
n is number of observations.

After binarizing all the categorical features, the test has been applied to the entire dataset.

The Spearman correlation matrix indicates a rather weak monotonic relationship between the dataset's features and the target variable. This strengthens the reason for dropping those previously identified four columns ("local_bo_qty", "pieces_past_due", "rev_stop", and "stop_auto_buy") through percentile test and Chi-squared test.

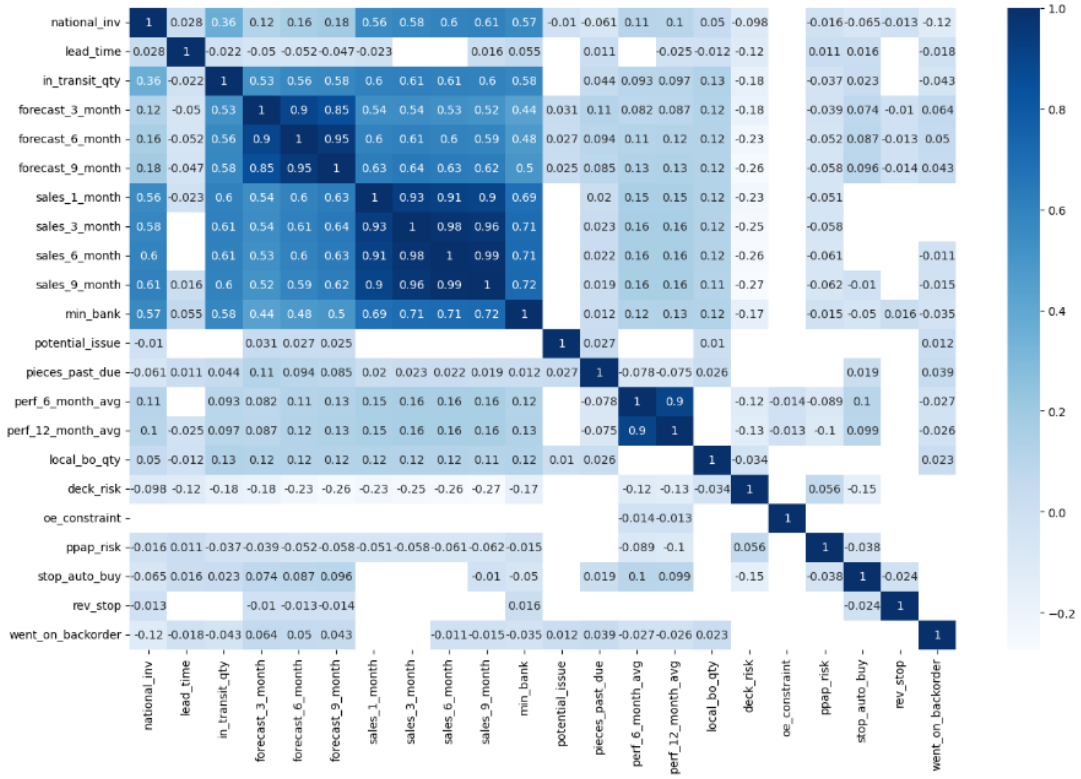



***Figure 15: Spearman correlation matrix***

Furthermore, here also it is clearly visible that, the Sales columns exhibit strong positive correlations in the group, with correlation coefficients ranging from 0.90 to 0.99, while the Forecast columns demonstrate strong positive correlations between them with coefficients from 0.85 to 0.95, and the Performance columns are highly correlated with a coefficient of 0.90. Given the substantial intercorrelations between these groups, a strategic approach might involve choosing a representative column from each of these three groups to avoid multicollinearity problems in subsequent predictive modeling. From a data perspective, considering columns with longer time frames, such as “forecast_9_month”, “sales_12_month”, and “perf_12_month” would be advantageous. This would facilitate a comprehensive view of historical data, enabling models to account for extended trends and patterns.

However, it is important to recognize that these correlation analyses only capture linear and monotonic relationships between variables. While features with strong correlations have been identified, there might still be other intricate dependencies or interactions between variables that not fully captured by these methods alone. Hence, to ensure a comprehensive feature selection process, the researcher utilized the SelectKBest method, employing the Mutual Information/ MI score as a metric. These additional techniques will allow this research to explore complex relationships and gain a more holistic understanding of feature relevance in the context of building accurate and reliable predictive models.

***NB:*** *Relevant codes for "Correlation matrix" are provided in the 'Appendix 08'*

#### *3.5.5.6 SelectKbest with Mutual Information / MI Score*

The SelectKBest method is a feature selection strategy that ranks and chooses a predefined number of top characteristics according to how important each one is to the target variable. This approach relies on a scoring function to evaluate and rate the relevance of each attribute. Among the available scoring functions, the Mutual Information / MI score stands out. MI primarily assesses the relationship between the dependencies of two variables. When one variable is observed in relation to another, it quantifies the knowledge or information obtained about the first variable. Its ability to detect any kind of association between the variables makes it especially valuable for complex datasets (Scikit-learn Documentation, n.d.a; Krish Naik, 2021). For a dataset with n features, $X_1$, $X_2$,…, $X_n$, and a target variable Y, the MI score for each feature $X_i$ in relation to Y can be described mathematically as:

$$MI(X_i, Y) = \sum x \in X_i \sum y \in Y\ p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Where, $p(x,y)$ is the joint probability distribution of $X_i$ and $Y$,

$p(x)$ and $p(y)$ are the marginal probability distribution of $X_i$ and $Y$ respectively,

Here, the double summation, $\sum x \in X_i \sum y \in Y$ iterates over all possible values of $X_i$ and $Y$ and effectively weighs each possible combination with its corresponding logarithmic term and then sums all these contributions up to get the overall mutual information.

However, the SelectKBest method has been employed for this analysis with MI as the scoring function, set to examine all features initially.

```
[('stop_auto_buy', '0.2150'), ('lead_time', '0.0697'), ('deck_risk', '0.0208'), ('perf_6_month_avg', '0.0118'), ('perf_12_month
_avg', '0.0096'), ('national_inv', '0.0095'), ('ppap_risk', '0.0093'), ('forecast_3_month', '0.0051'), ('forecast_6_month', '0.
0047'), ('forecast_9_month', '0.0043'), ('in_transit_qty', '0.0013'), ('sales_3_month', '0.0012'), ('min_bank', '0.0011'), ('sa
les_9_month', '0.0011'), ('sales_1_month', '0.0009'), ('sales_6_month', '0.0009'), ('pieces_past_due', '0.0003'), ('local_bo_qt
y', '0.0003'), ('potential_issue', '0.0000'), ('oe_constraint', '0.0000'), ('rev_stop', '0.0000')]
```

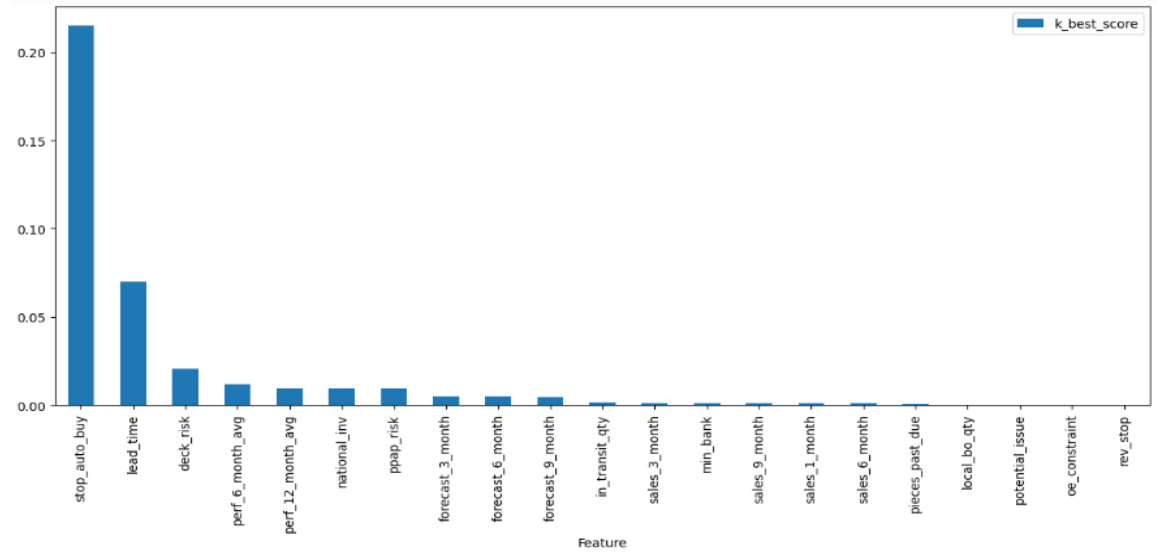



***Figure 16: SelectKBest feature ranking using MI score***

Based on the SelectKBest method using the Mutual Information score the following observations have been made:

- The top features include "stop_auto_buy", "lead_time", and "deck_risk", with significant MI scores and appear to be crucial.
- The Performance columns, Forecast columns and Sales columns have moderate scores that suggest their importance on the target column. However, there is a special consideration regarding them based on the correlation tests.
- Certain features, including "in_transit_qty", "pieces_past_due", "local_bo_qty", "oe_constraint", "potential_issue", and "rev_stop" have close to zero or very low MI scores, indicating that they might not be providing much information about the target variable compared to the other features.

***NB:** Relevant codes for "SelectKbest with Mutual Information / MI Score" are provided in the 'Appendix 09'*

#### *3.5.5.7 Summary of all observations / Finalizing the features for ML model*

As all the analyses have produced varied results, a balanced strategy for feature selection will be followed. A variable will be kept even if a single test determines it to be significant. However, any variable that is consistently labelled as insignificant throughout several tests will be ignored. A variable will also be eliminated if one test finds it to be insignificant and no other tests provide a compelling reason for its inclusion. On top of it, professional judgment will also serve as an additional guiding factor. In light of this strategy, the following conclusions have been reached:

- The longest timeframe columns from the Performance, Forecast, and Sales categories will be retained to ensure thorough trend analysis.
- Features with unusually low MI scores will be excluded, particularly when supported by correlation analyses. However, the column 'min_bank' will remain due to its significance as deduced from domain knowledge.

Therefore, the final features for the ML models are- “national_inv”, “lead_time”, “forecast_9_month”, “sales_9_month”, “min_bank”, “perf_12_month_avg”, “deck_risk”, “ppap_risk”, “stop_auto_buy” as predictors and “went_on_backorder” as the target column.

***NB:*** *Relevant codes for “Summary of all observations / Finalizing the features for ML model” are provided in the ‘Appendix 10’*

### *3.5.6* Handling duplicates and resetting indices

In real-world datasets, it is common to encounter disordered data containing numerous duplicate entries. These repetitive records do not contribute any new insights and can hinder computational efficiency. Thus, removing duplicates is a recommended step prior to model training (Durgapal, 2023). In this study, the dataset has been thoroughly scanned for any identical rows. Whenever duplicates have been detected, they have promptly been removed to ensure each data entry is unique and representative.

However, the indices were reset after any rows were removed, whether because of duplication or other data-cleaning procedures. This procedure guarantees a continuous and consistent numbering sequence that improves the readability and traceability of the dataset.

***NB:** Relevant codes for "Handling duplicates and resetting indices" are provided in the 'Appendix 11'*

### *3.5.7* Scaling the dataset

In the field of machine learning, feature scaling during the data pre-processing phase is crucial. Its impact may make the difference between a weak machine learning model and a powerful one (Roy, 20020). In essence, feature scaling becomes crucial for machine learning techniques that calculates the separations between data points. In the absence of scaling, characteristics with wider numerical ranges could overshadow those with smaller ranges, skewing how the model interprets the dataset (Patil, 2022).

However, when dealing with datasets containing outliers, it is important to choose right scaling technique. This is because outliers can profoundly affect statistical measures, such as the mean and standard deviation. In such scenarios, robust scaling is recommended. Robust scaling calculates the scaled value $x_{scaled}$ for an input value x as:

$$x_{scaled} = \frac{x - median(X)}{IQR(X)}$$

Where,

X is the feature,

median(X) is the median value of the feature column X,

IQR(X) is the interquartile range of the feature column X, which is the difference between the 75$^{th}$ percentile and the 25th percentile.

Robust scaling employs the median and Interquartile Range / IQR for scaling input values. Since both these metrics are resistant to outliers, robust scaling can effectively mitigate the undue influence of outliers, ensuring that the model remains unbiased (Singh, 2022).

Therefore, given the presence of outliers in the dataset for this study, robust scaling has been selected to provide an even and uniform transformation of all features. Due to its wide range of values, this method ensures that no feature dominates others.

***NB:*** *Relevant codes for "Scaling the dataset" are provided in the 'Appendix 12'*

### *3.5.8* Handling imbalanced training set / Resampling training set

The machine learning field frequently faces problems with imbalanced datasets, particularly when performing classification tasks. The term "imbalanced data" refers to datasets where the target class exhibits a disproportionate distribution of records, such as one class label significantly outnumbers the other in terms of observations. If the dataset is highly imbalanced, it affects the ML models' performance badly as it performs worse on the minority class, which is often the class of more interest. Hence, resampling techniques are often utilized to either raise the minority class instances or decrease the majority class instances (Mazumder, 2021).

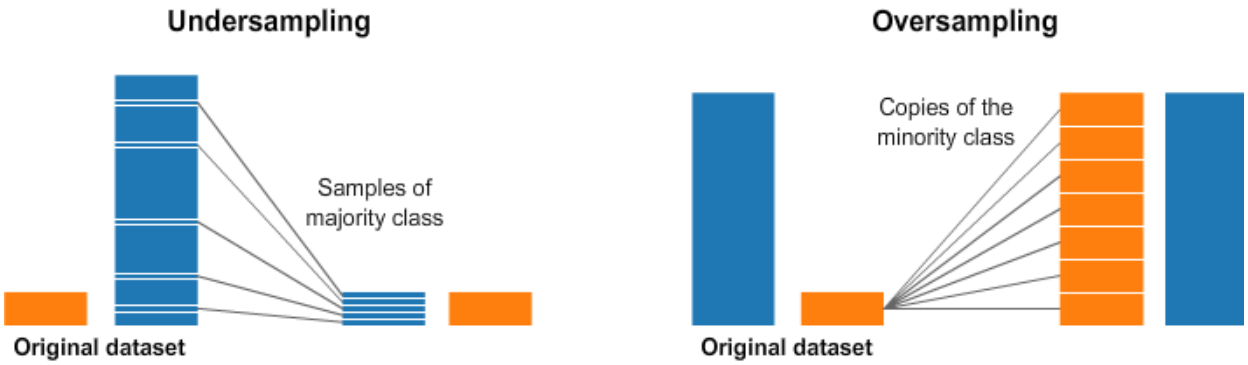



***Figure 17: Under-sampling and Oversampling (Aguiar, 2023, para. 12)***

There is a definite class imbalance in the training dataset of this study. Specifically, the majority class (labeled as 0) contains 795,206 observations, substantially outnumbering the minority class (labeled as 1), which has only 8,196 observations (98.98% versus 1.02% only). This significant disparity can lead to models that have a strong bias toward forecasting the majority class. Hence, among various sampling techniques, this work specifically used RUS and SMOTENC, two sampling techniques, to address the class imbalance, ensure a more equal representation, and encourage the development of objective predictive models.

- **RUS:** Random Under-sampling / RUS is arguably one of the most common sampling techniques. RUS seeks to achieve a balanced class distribution by eliminating majority class instances at random. Even though RUS uses subsets of the original dataset to protect data integrity, it might cause significant data loss for extremely unbalanced datasets, which could jeopardize model performance (Aguiar, 2023).

- **SMOTENC:** To understand Synthetic Minority Oversampling Technique for Nominal and Continuous / SMOTENC it is important to understand how SMOTE / Synthetic Minority Oversampling Technique works because SMOTENC is an extension of the SMOTE algorithm. However, SMOTE is a method that relies on the Euclidean Distance to determine nearest neighbors among data points in the feature space. SMOTENC expands the SMOTE algorithm to handle datasets with a mix of continuous and categorical variables. SMOTENC considers the nature of the features rather than merely interpolating between feature values to produce fictitious data points. It determines a mode of the categorical values of the closest neighbors for categorical features. In this way, SMOTENC creates artificial samples that are consistent with the organization of the data (Aguilar, 2019).

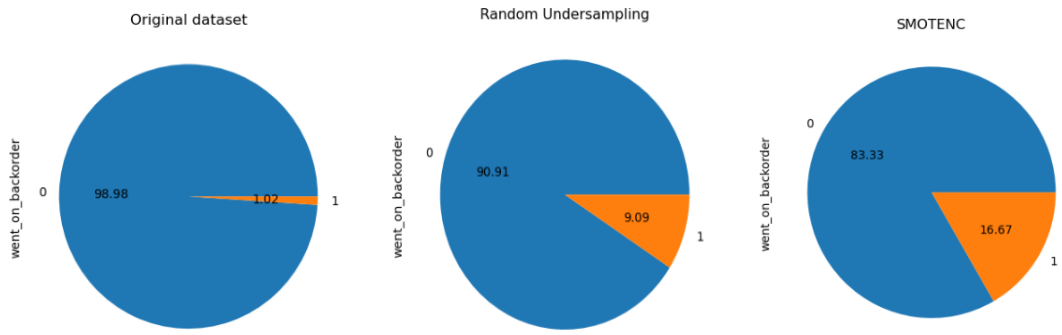



***Figure 18: Class distribution before and after resampling***

Nevertheless, to ensure a representative dataset without significantly reducing the amount of data, a modest 10% under-sampling has been applied with RUS technique. With this adjustment, the class distribution is changed to 81,960 observations for class 0 (90.91%) and 8,196 for class 1 (9.09%). This selective under-sampling is designed to maintain the dataset's integrity and robustness and reduce the risk of overfitting. Subsequently, SMOTENC has also been applied to achieve a more balanced distribution, yielding class ratios of 83.33% for class 0 (795,206 observations) and 16.67% for class 1 (159,041 observations). The initial

imbalance found in the dataset, coupled with resource constraints, influenced the choice to modestly oversample by 20% as opposed to taking a more aggressive strategy. This approach not only accommodates limited computational resources but also minimizes the risk of introducing noise or over-representing the minority class. By ensuring that the synthetic observations generated by SMOTENC are coherent and fit well with the original data, the integrity of the dataset is maintained while addressing its imbalance.

***NB:*** *Relevant codes for "Handling imbalanced training set / Resampling training set" are provided in the 'Appendix 13'*

## 3.6 Model building

According to Manika (2023), the process of model building is central to any machine learning-related research. While selecting the optimal model for a project can be a complex task, understanding each model's strengths and limitations can streamline the process. Having completed all the data preparation steps, the stage is now set for the crucial phase of model building. This is where theoretical knowledge and prepared data intertwine, turning insights into actionable predictions. The model's efficacy directly affects the decision-making and supply chain efficiency in the context of backorder prediction.

This section examines the intricacies of selecting algorithms, understanding their mechanics, and fine-tuning them for optimal performance. The objective is to build reliable predictive models that demonstrate machine learning's ability to handle complex business problems in the real world. The section also incorporates the appropriate performance measurement considering the imbalanced nature of the datasets.

### *3.6.1* Model selection

Manika (2023) also defines model selection as the process of selecting the best-fit model out of all potential options for a particular problem. She argues that this process ensures that the selected model not only performs well on training data but is also generalizable to new data. A well-chosen model improves accuracy, reduces overfitting, improves interpretability, and saves computational resources.

Drawing upon the comprehensive “Table 02”, presented in Chapter 02, several algorithms frequently appeared and demonstrated noteworthy performance across various studies. However, this research emphasizes the exploration of diverse classifiers to identify the most optimal model. Consequently, Neural Networks will not be a part of the training process. The literature review shed light on the following classifiers:

- Top-performing studies, such as Santis et al. (2017) and Hájek & Abedin (2020) attested to the effectiveness of Logistic Regression/LR.
- The interpretative aspect of the fundamental Decision Trees/DT was evident in the works of Santis et al. (2017) and Shajalal et al. (2022).
- Several major research consistently preferred the ensemble approach Random Forest/RF, obtaining notable AUC scores.
- In the study of Ntakolia et al. (2021), the potent gradient boosting algorithm XGBoost/XGB, demonstrated its formidable capabilities, solidifying its leading position among the best algorithms for backorder predictions.

Therefore, incorporating knowledge from the literature review and on the research supervisor's recommendation, this study will largely concentrate on the four models: Logistic Regression, Decision Tree, Random Forest, and XGBoost. This choice is supported by their consistent performance throughout earlier studies and flexibility in dealing with backorder forecast difficulties.

### *3.6.2* Model training

According to Weedmark (2021), model training is a crucial stage in machine learning, turning raw data insights into intelligence by enabling algorithms to learn and form connections. To capture patterns, identify anomalies and test correlation the process entails feeding the selected algorithm with training data. In supervised learning, model training yields a mathematical representation of the relationship between data features and a target label. Conversely, in unsupervised learning, the focus shifts to building representations solely from data features. The overall performance during the training phase provides an insight into the model's future effectiveness.

To ensure optimal performance of the model, continuous model refinement is crucial once the first model training is over. This involves several specialized techniques that concentrate on specific model components, enhancing the model's performance on new, unseen data. These include, among others:

- **Threshold optimization:** The underlying mathematics of classification tasks often include assigning probabilities (usually with a default threshold value of 0.5) to each class. This indicates that if the expected probability that an instance belongs to the positive class is greater than or equal to 0.5, it is considered as positive; otherwise, it is considered as negative. For example, when using logistic regression to detect spam, the two classes can be used to distinguish between spam and legitimate emails. A probability spectrum between 0 and 1 is produced by logistic regression when used with the sigmoid function, indicating the likelihood that an input sample is spam. Here, a probability of 0.99, implies that there is a high likelihood that the email is spam, whereas a probability of 0.003 suggests otherwise. The identification of the email's type, however, becomes less certain when the likelihood is close to the threshold, such as 0.51. This uncertainty highlights how important threshold optimization is. Better classification results may result from adjusting the threshold based on the type of data and specific requirements of a task. It is vital to establish an ideal threshold that balances these errors when false positives and false negatives have different consequences. This will guarantee that the model's predictions are both accurate and pertinent to the task at hand (Iguazio, n.d.).

- **Hyperparameter optimization**: As established in the theoretical review Chapter 02, default configurations known as hyperparameters are essential for model training. Model performance on unknown data is considerably improved by identifying and utilizing the best hyperparameters.

  According to Brownlee (2019), a sophisticated approach for hyperparameter optimization is Bayesian optimization which efficiently searches the hyperparameter space to find optimal values. In contrast to conventional techniques, which view hyperparameter tuning as a form of black-box optimization, Bayesian optimization ensures a more focused search based on probabilistic models during the hyperparameter tuning phase. It is based on the idea of a surrogate probability model,

a statistical model that calculates the function connecting the input hyperparameters to the expected model performance. Gaussian Processes/GPs are typically used as the surrogate. The distribution across functions in a GP, a non-parametric technique, is specified by points, and the multivariate normal distribution of these functions is assumed. A kernel function is used to define the relationship between the function values. The Bayesian optimization's performance is substantially impacted by the chosen kernel and its parameters. An acquisition function is also employed to choose the subsequent set of hyperparameters to be evaluated. This function strikes a balance between exploitation and exploration, or locations where the surrogate function predicts a good objective value. It assists in weighing the trade-offs between exploring areas with high uncertainty and experimenting in those that are near to the most well-known solution.

- **Cross-validation:** As highlighted in Chapter 02, cross-validation is integral to hyperparameter tuning ensuring model generalizability. In accordance with Gupta (2017), this technique evaluates the performance of models to see whether they can handle unobserved data. It mainly prevents overfitting, where a model impressively deciphers training data but struggles with new data. While dividing datasets into training and testing sets is standard procedure, relying solely on this distinction can occasionally be deceptive. Cross-validation addresses this limitation by dividing the data into several subsets or folds. A popular variation is K-Fold Cross-validation, in which the data is divided into “k” different subgroups. K-1 folds are assigned to training for each iteration, and the final fold is used for validation. Once each fold has been used as the validation set, the cycle is repeated. The performance measures from each iteration are then averaged to provide a comprehensive evaluation of the model's capabilities. A subtle strategy, stratified cross-validation, is particularly useful when the distribution of the target variable is imbalanced. In this method, each fold is formed by maintaining the percentage of samples from each class, ensuring that each subset replicates the overall sample distribution. Hence, this approach is especially beneficial for datasets where some classes are underrepresented.

For this research, the model training phase followed a systematic approach. After building the base models, an organized procedure was used to improve the overall models’ performance by determining the best choice threshold and fine-tuning hyperparameters. To enable a

thorough investigation of probable decision boundaries, an array of 200 evenly spaced values between 0 and 1 was created. The model's projected probabilities were binarized for each threshold. The balance between precision and recall was then assessed using the F1 score for each threshold. The threshold with the highest F1 score was chosen as the best decision boundary, according to the optimal threshold selection method. A rigorous hyperparameter tuning procedure was used for all the models in this research. The extensive search in the hyperparameter space was performed by utilizing the capabilities of Bayesian optimization. Along with that, StratifiedKFold for cross-validation was employed and this fine-tuning ensured a balanced representation of classes during each fold. However, during this tuning phase, a new optimal threshold was discovered and applied to refine the model's classification. This thorough investigation's conclusion produced the ideal collection of parameters, which prepared the ground for improved model performance. The following Sub-subsections provide more information on each model's training process, including its base and tuned configurations.

#### *3.6.2.1 Logistic Regression*

According to Datacamp (2019); IBM (n.d.b), Logistic Regression/LR, fundamentally a statistical model, is an extension of linear regression. While Linear Regression predicts a continuous outcome, Logistic Regression predicts the likelihood that a given instance would fall into a specific category in a classification problem based on one or more independent variables. The core of LR is the logistic function, often referred to as the sigmoid function. This S-shaped curve can transform any real-valued number into a value between 0 and 1.

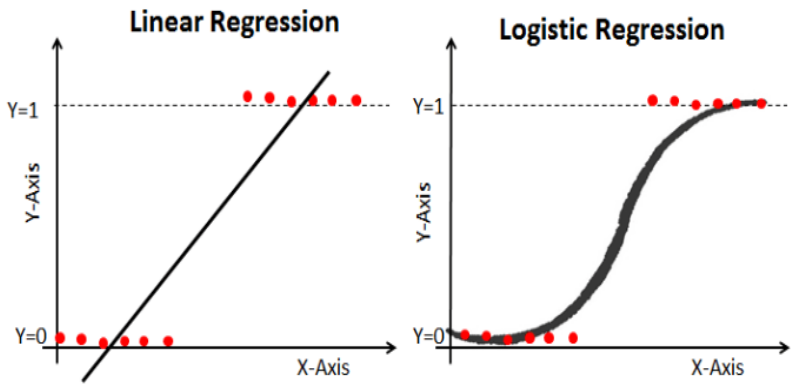



***Figure 19: Understanding logistic regression (Datacamp, 2019, para. 10)***

The sigmoid function is especially used in situations where the probability of an outcome is not linearly related to the independent variables. It is expressed in as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where, $\sigma(z)$ denotes the probability, Euler's number $e = 2.71828$, and $z$ is weighted sum of the input features which is calculated by:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots\ldots\ldots + \beta_n x_n$$

Here, $\beta_0 + \beta_1 + \beta_2 + \ \ldots\ldots\ldots + \beta_n$ are the coefficients of the model and

$x_1 + x_2 + \ \ldots\ldots\ldots + x_n$ are the input features or independent variables.

The likelihood that an event will occur divided by the probability that it will not occur is known as the odds of the event. The odds ratio is shown as follows for logistic regression:

$$Odds = \frac{p}{1 - p}$$

Where, $p$ is the predicted probability. The logarithm of the odds gives the following:

$$ln\left(\frac{p}{1 - p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots\ldots\ldots + \beta_n x_n$$

In this case, the outcome's logarithm of odds (also known as log-odds or logit) is shown on the left side ($ln$ stands for the natural logarithm) and is transformed into a linear relationship with the predictors. Logistic Regression employs the method of Maximum Likelihood Estimation/MLE to estimate the parameters or weights (Saini, 2021a). However, LR is effective and uses little processing power. It is widely favored by data professionals and is easy to apply and analyze. It provides probability scores for observations and functions without the need for feature scaling. The algorithm has inherent limitations as well. It is prone to overfitting and has trouble with many category variables. It also requires feature transformations for nonlinear issues. Performance may also decline when independent

variables either do not correlate with the target variable or are highly associated among themselves (Datacamp, 2019).

With this foundational understanding of Logistic Regression, this study employed the algorithm in the following manner:

- **Initial model development:** Training the Logistic Regression model with default parameters on the dataset was the initial step. The model was using "l2" regularization, the "liblinear" solver, and no explicit class weighting at this point. This initial modeling served as a baseline, facilitating subsequent optimization.

- **Hyperparameter tuning:** As it was discussed earlier, through the Bayesian optimization the investigation of hyperparameters was meticulous at this stage:

***Table 05: Hyperparameter tuning of Logistic Regression***

| Parameter Space | | Description | Best parameter | | |
|---|---|---|---|---|---|
| | | | Original | RUS | SMOTENC |
| C | 0.001 to 10,000 | Regularization strength, Spanning a log-uniform distribution | 9154.7 | 10000.0 | 10000.0 |
| class_weight | "None", "balanced" | This investigated the trade-offs between the default and a balanced approach | "balanced" | "balanced" | "balanced" |
| penalty | "l1", "l2" | Exploring both Lasso and Ridge regularization techniques | "l2" | "l2" | "l2" |
| solver | "liblinear", "saga" | Two algorithmic methods for optimization were considered. | "liblinear" | "liblinear" | "liblinear" |

After careful hyperparameter adjustment for the Logistic Regression, it is clear how customized setups may improve the algorithm's capacity for specific datasets. The options, as shown in the table, demonstrate both the model's adaptability and the breadth of this research's methodology.

### *3.6.2.2 Decision/Classification Trees*

According to Singh (2018), Decision or Classification Trees are a common supervised machine learning approach for both classification and regression problems. At their core, they make decisions based on a series of inquiries. A Decision Tree/DT consists of nodes, where each node stands for a characteristic or attribute, branches for a set of rules, and leaves that represent the results. Understanding various terminology in the context of Decision Trees is essential to understanding their structure and workings. The "Root Node" represents the full dataset or sample, which eventually divides into one or more cohesive sets. "Splitting" is the process of separating a node into several sub-nodes. A "Decision Node" is any sub-node that splits even more. On the other hand, nodes that stop splitting are known as "Leaf or Terminal Nodes". "Pruning", a technique that works against "Splitting" to streamline and optimize Decision Trees, primarily entails trimming the tree by removing nodes. Any specific section of the Decision Trees can be called as a "Branch" or "Sub-Tree". Terms are also used to describe the connectivity between nodes: a node that produces sub-nodes is the "Parent Node", and the resulting sub-nodes are its "Child Nodes".

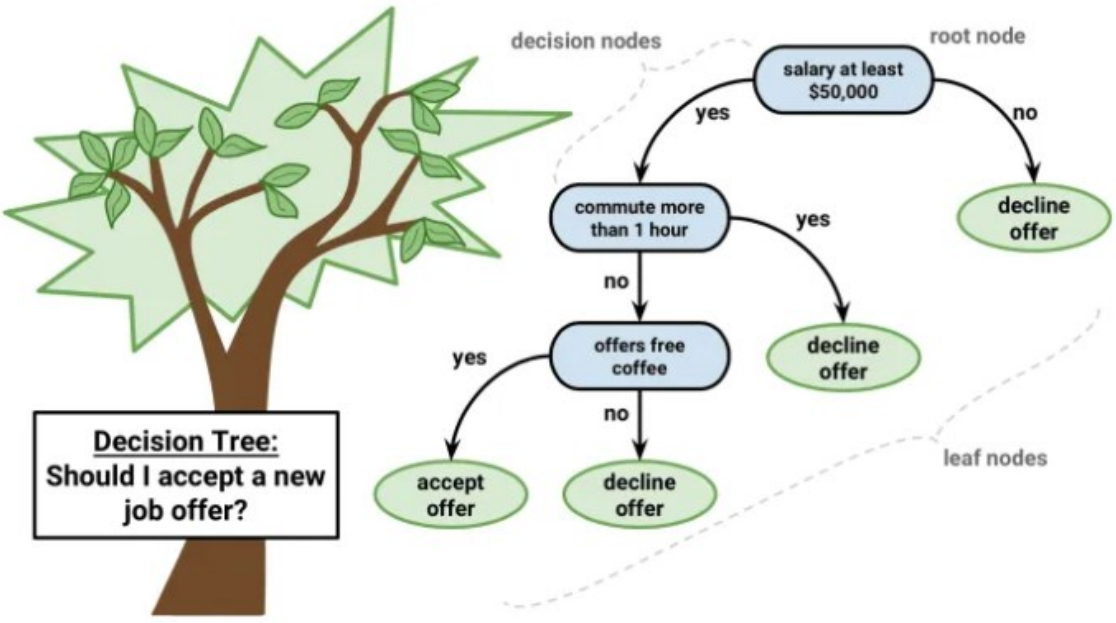



***Figure 20: Decision Trees (Singh, 2018, para. 1)***

However, Decision Trees use various measures to determine the appropriate split. For classification problems, these measures include:

a) **Entropy:** In a dataset, Entropy quantifies randomness or unpredictability. When a set is totally homogeneous, it becomes zero. Entropy $E$ of a set $S$ with respect to binary categorization into positive and negative classes can be calculated mathematically by:

$$E(S) = -\sum_{i=1}^{c} p_i log_2 (p_i)$$

Where, $c$ is the number of classes and $p_i$ represents the percentage of instances that belong to class $i$.

b) **Gini impurity:** It shows how frequently a randomly selected element would be misclassified. It drops to its lowest possible value (zero) when all cases in the node fit into a single target category. The Gini impurity for a set $S$ is:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

Even though both Entropy and Gini impurity aim to measure the disorder or impurity in data, the specific metric chosen often depends on practical considerations.

c) **Information Gain/IG:** IG plays a vital role in Decision Trees. It directs the algorithm's split decisions by determining how much information a given feature offers about the outcome. IG can be calculated using the following formula:

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N} I(D_{left}) - \frac{N_{right}}{N} I(D_{right})$$

Where, $f$ represents feature to split on and $D_p$ represents dataset of the parent node,

$D_{left}$ and $D_{right}$ represent dataset of the left child node and right child node, respectively,

$I$ represents Impurity criterion (Entropy or Gini),

$N$ represents total number of samples,

$N_{left}$ and $N_{right}$ represent number of samples at left child node and right child node, respectively.

For regression tasks, Decision Trees use Mean Squared Error/MSE. The approach determines the weighted average of the MSEs for the left and right child nodes at each potential split. The split that yields the lowest overall MSE is chosen (Singh, 2018).

Decision Trees are recognized for their simplicity of interpretation since they are simple to visualize. Additionally, they only need a minimal amount of data preparation; feature scaling, for example, is less important than it is for some other techniques. They do not make assumptions about the distribution of the underlying data because they are non-parametric. Decision Trees provide a variety of advantages, making them suitable for both classification and regression tasks. They are skilled at handling datasets with potentially imbalanced classes and can effectively capture non-linear decision boundaries. They do, however, have some drawbacks. Although this may be avoided with the correct parameters, they can occasionally overfit (performing well only on training set but not on test set), producing excessively complex trees that do not generalize well to new data. They can also be unreliable since even small changes in the data might produce a dramatically altered tree structure. To lessen this volatility, ensemble approaches might be used. Further to this, Decision Trees tend to use heuristic techniques, specifically greedy algorithms. These algorithms make the best split for the current node they are evaluating, without considering the impact of the decision on future splits. This can sometimes lead to solutions that do not always guarantee a globally optimal tree structure (Singh, 2018).

With this foundational knowledge of Decision Trees, the algorithm was employed in the study in the following manner:

- **Initial model development:** The Decision Tree method was first trained on the dataset using its default parameters. For the future optimization procedure, this initial modeling phase served as the baseline.

- **Hyperparameter tuning:** At this point, the rigorous investigation of the hyperparameters was performed using the previously discussed Bayesian optimization technique:

*Table 06: Hyperparameter tuning of Decision Trees*

| Parameter Space | | Description | Best parameter | | |
|---|---|---|---|---|---|
| | | | Original | RUS | SMOTENC |
| class_weight | "None", "balanced" | This investigated the trade-offs between the default and a balanced approach | "balanced" | "None" | "None" |
| criterion | "gini", "entropy" | Function to measure split quality | "entropy" | "entropy" | "entropy" |
| max_depth | 3 to 15 | Maximum allowed depth of tree. | 7 | 7 | 15 |
| max_features | "None", "sqrt" | Number of features for best split. | "None" | "None" | "None" |
| min_samples_leaf | 1 to 7 | Minimum samples required at a leaf node. | 7 | 1 | 7 |
| min_samples_split | 2 to 10 | Minimum samples to split a node. | 2 | 10 | 2 |
| splitter | "best", "random" | Strategy for node split | "best" | "best" | "best" |

It is obvious that meticulous hyperparameter tuning of the Decision Tree model yields diverse effects across different data treatments. The table shows how thoroughly the parameter space of the DT was investigated, as well as the best parameters that could be found for each data treatment. The versatility of the DT algorithm and the depth of this research's methodology are demonstrated by these outcomes.

#### *3.6.2.3 Random Forest*

According to Saini (2021b); IBM (n.d.c), Random Forest/RF is a widely used supervised machine learning algorithm. It serves as an ensemble technique, as at its core, it creates multiple decision trees during the training phase and merges them to produce a more accurate and stable prediction. Decision Trees and Random Forests share many similarities in terms of their fundamental concepts. In both models, the concepts of "Node", "Branch", "Leaf or Terminal Nodes", "Root Node", "Splitting", "Decision Node" and the parent-child

relationships between nodes are consistent. They also use various measures to determine the best split, such as Entropy, Gini Impurity, Information Gain for classification and MSE for regression. While DT and RF have a lot in common, they have distinctions as well. For regression tasks, DT predicts based on the average target value in the terminal node, whereas RF averages predictions from all trees in the ensemble. For classification problems, even though both use majority voting, DT uses it at the individual nodes level, while RF uses it at the ensemble level to aggregate predictions from multiple trees.

$$\hat{y}(x, D) = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m(x)$$

Where, $\hat{y}(x, D)$ represents the predicted output for an input $x$ based on dataset $D$

$M$ represents the total number of trees in the Random Forest

$\hat{y}_m(x)$ represents the prediction of the $m^{th}$ tree for the input $x$

A simple Random Forest classifier is shown below:

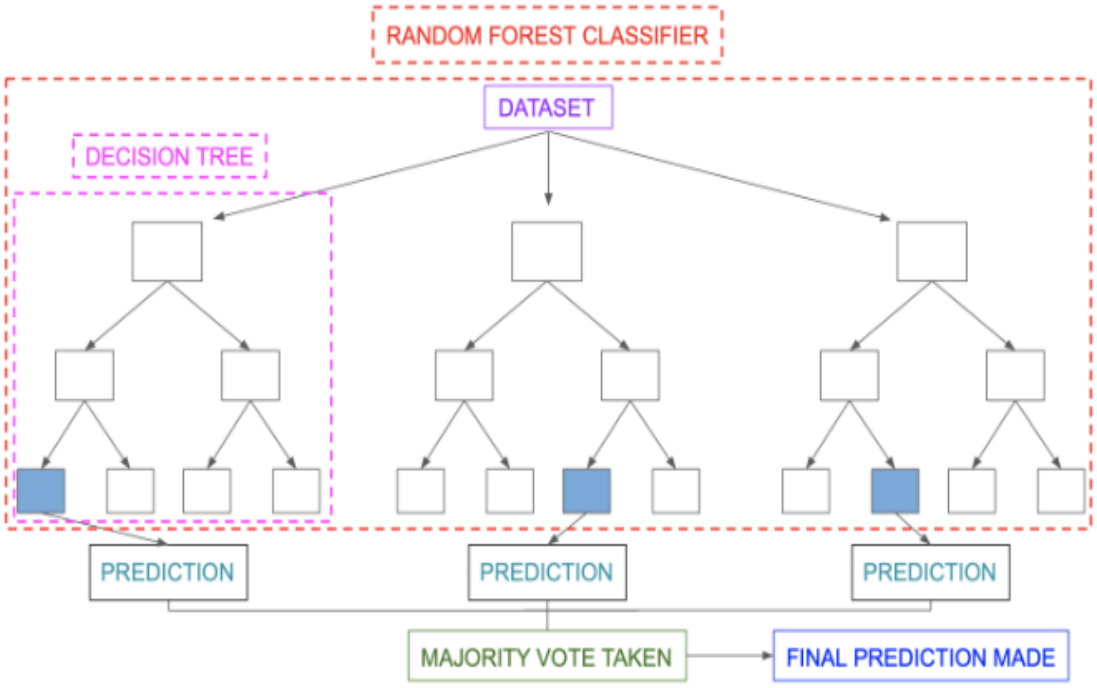



***Figure 21: Simple Random Forest Classifier (Saini, 2021b, Applying DT in RF algorithm)***

However, Random Forest is essentially an application of the bagging technique to Decision Trees, where each tree is constructed using a subset of data (also known as a bootstrap sample) drawn from the training dataset with replacement. Approximately one-third of this sample is reserved as the Out-of-Bag/OOB sample. The OOB sample plays a crucial role in cross-validation, refining the final prediction. Unlike conventional Decision Trees which may consider every feature for a split at each node, Random Forests only allow a random subset of features to be considered at each split. This randomness ensures diversity among the trees and minimizes correlations that make the ensemble resilient (IBM, n.d.c).

Due to its versatility, Random Forest is one of the most popular algorithms in the data science field. Even when hyperparameter tuning is not done meticulously, it often produces greater results. The parameters involved in Random Forests are simple and few which are easy to understand. By combining numerous Decision Trees that finally provide low bias and low variance, RF helps to avoid overfitting which is a persistent problem in machine learning tasks. Nevertheless, a major disadvantage of RF is the lengthy training period required due to the large number of trees involved. While training is typically quick, making predictions can be too slow. Although the algorithm works well in most cases, there are some situations where quick real-time predictions are essential, prompting consideration of alternative strategies (Saini, 2021b).

Drawing on the fundamental knowledge of Decision Trees and their ensemble application in Random Forests, and recognizing the inherent strengths and potential drawbacks, this study adopted a structured approach to employ the RF algorithm in the following manner:

- **Initial model development:** A Random Forest classifier was first built using default parameters to provide a baseline. This initial model provided a performance standard by which all later improvements were evaluated.

- **Hyperparameter tuning:** Given that specialized configurations can unleash Random Forest's full potential, a thorough hyperparameter tuning process was employed. The Bayesian optimization strategy helped in effective navigation in the hyperparameter space:

***Table 07: Hyperparameter tuning of Random Forests***

| Parameter Space | | Description | Best parameter | | |
|---|---|---|---|---|---|
| | | | Original | RUS | SMOTENC |
| bootstrap | "True", "False" | Whether bootstrap samples are used when building trees. | "False" | "False" | "True" |
| class_weight | "None", "balanced" | This investigated the trade-offs between the default and a balanced approach | "None" | "None" | "None" |
| criterion | "gini", "entropy" | Function to measure split quality | "entropy" | "entropy" | "entropy" |
| max_depth | 1 to 25 | Maximum allowed depth of tree. | 18 | 18 | 25 |
| min_samples_leaf | 1 to 7 | Minimum samples required at a leaf node. | 1 | 1 | 1 |
| min_samples_split | 2 to 15 | Minimum samples to split a node. | 2 | 2 | 2 |
| n_estimators | 50 to 300 | The number of trees in the forest. | 223 | 281 | 300 |

By using this organized methodology, the study not only took advantage of the Random Forest's powerful capabilities but also meticulously shaped the model to meet the unique nuances of the under-investigation dataset. The selected hyperparameters highlight the breadth and rigor of the research methodology, which demonstrates a deliberate balance between adaptability and performance.

#### *3.6.2.4 XGBOOST*

According to Chen & Guestrin (2016); Analytics Vidhya (2018); Geeksforgeeks (2023), Extreme Gradient Boosting/XGBoost is a powerful ensemble learning technique specifically designed to improve the efficiency and performance of a machine learning model. At its essence, gradient boosting optimizes a differentiable loss function by iteratively adding weak learners to the model. These weak learners are often decision trees in the context of XGBoost. The main differences between XGBoost and conventional gradient boosting are its

regularization, optimization of the algorithm, and capacity to tolerate missing information. Similar to the Random Forest described earlier in this study, XGBoost also operates with Decision Trees as its basis models. However, XGBoost uses somewhat different trees. A comparison of XGBoost to Decision Trees and Random Forests can help to clarify the uniqueness of the algorithm. While a Decision Tree operates as a single, stand-alone model and a Random Forest builds multiple trees in parallel from bootstrapped samples with "bagging" strategy, XGBoost takes a sequential approach. This strategy is based on the idea of "boosting" in which several weak prediction models are combined to create a powerful one. In an iterative refinement process, the trees in the XGBoost forecast residuals or errors rather than the direct target value, where new trees attempt to fix the mistakes caused by older ones. This results in sequential model improvement.

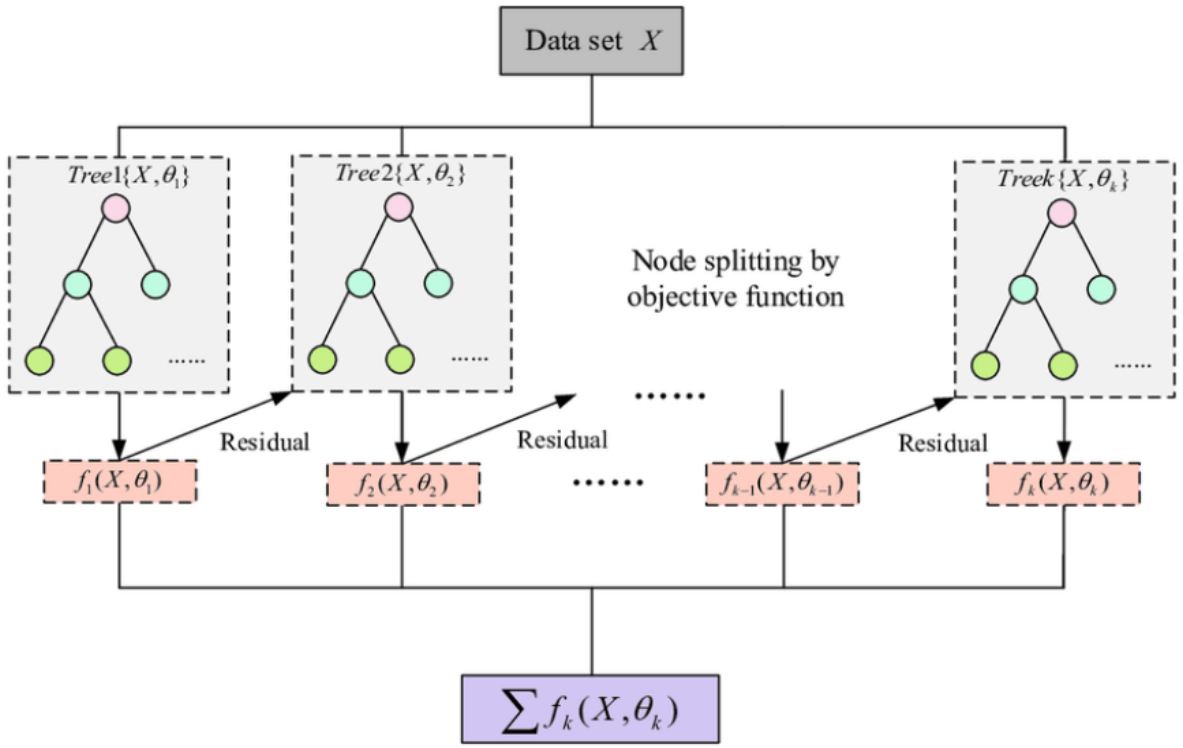



***Figure 22: Flow chart of XGBoost (Guo et al., 2020, p. 06)***

In the Figure 22, $f_k$ is the prediction made by the $k^{th}$ Decision Tree for input data $X$ with $\theta$ indicating the tree parameter. As detailed above, instead of relying on only one model, XGBoost sums the predictions cumulatively of each tree built up to the current iteration to make its predictions. Each tree (function $f_k$) aims to rectify the errors made by the preceding trees.

Mathematically, the model seeks to minimize the loss function through gradient descent, with each step being modified by the residuals:

$$New\ prediction = Old\ prediction + \eta \times Residual$$

Where, $\eta$ (eta) represents the learning rate.

To elaborate on the mathematical foundations, the objective function that XGBoost optimizes is specified as follows:

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{K} \Omega(T_k)$$

Where, $l$ is the differentiable loss function,

$y_i$ is the true label for instance $i$,

$\hat{y}_i^{(t)}$ is the predicted label for instance $i$ at iteration $t$,

$\Omega$ (Omega) is the regularization term and

$T_k$ is the $k^{th}$ Decision Tree.

The regularization term $\Omega$ in XGBoost is defined as:

$$\Omega(T) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2$$

Where, $T$ is the number of leaves in the tree,

$w$ is the vector of scores on leaves,

$\gamma$ (gamma) and $\lambda$ (lambda) are regularization parameters.

The versatility of XGBoost is admirable. For classification tasks, XGBoost predicts the probability of an instance belonging to a certain class; the probabilities are calculated using the logistic function, similar to Logistic Regression. For regression scenarios, XGBoost predicts continuous values, similar to Random Forest but the method of constructing the prediction model is different as the trees (functions) are iteratively created in XGBoost (Analytics Vidhya, 2018; Geeksforgeeks, 2023).

Nonetheless, one of the most notable qualities of XGBoost is its excellent performance and speed. XGBoost is not only accurate but also faster than many other boosting algorithms because of its effective implementation and capability to parallelize the tree-building process. It naturally has the capacity to control against overfitting, providing it an advantage over other boosting methods. Its ability to automatically handle missing data during training eliminates the frequently laborious phase of manual imputation, which is another key benefit. Additionally, XGBoost's method of building and pruning trees is distinct; it builds trees depth-first and then prunes them using max_depth to ensure optimal structure and avoid overcomplexity. Despite its advantages, XGBoost has certain drawbacks. Due to its sequential structure, it can occasionally require more computing power than bagging methods, such as Random Forests, especially when working with very large datasets. Although the technique has regularization characteristics, if not tuned correctly, XGBoost models could overfit, especially if the data include noise. Furthermore, careful hyperparameter adjustment, which can take a lot of time, is frequently necessary for XGBoost to achieve its full potential (Analytics Vidhya, 2018; Geeksforgeeks, 2023).

Harnessing the power of XGBoost and appreciating the unique nuances of the algorithm, this study employed it in a structured, two-tier approach:

- **Initial model development:** The first phase entailed creating a fundamental XGBoost model that was designed exclusively for binary classification. The model used the evaluation metric “logloss” for the “binary:logistic” aim, ensuring a clear and straightforward initialization. This base model serves as a reference point for evaluating the benefits of future hyperparameter adjustment.

- **Hyperparameter tuning:** A sophisticated hyperparameter tuning phase was launched to fully utilize the capabilities of the XGBoost algorithm. A more intelligent and purposeful search over the hyperparameter domain was performed by leveraging the Bayesian optimization technique. This reduced the possibility of overfitting while simultaneously ensuring ideal configurations:

*Table 08: Hyperparameter tuning of XGBoost*

| Parameter Space | | Description | Best parameter | | |
|---|---|---|---|---|---|
| | | | Original | RUS | SMOTENC |
| colsample_bytree | 0.5 to 1.0 | Proportion of columns (features) to subsample for each tree. | 1.0 | 1.0 | 1.0 |
| gamma | 1e-5 to 0.8 | Minimum loss reduction required to make a split, contributing to regularization and pruning. | 1e-5 | 1e-5 | 0.06941 |
| learning_rate | 0.01 to 0.15 | Step size shrinkage to prevent overfitting. | 0.15 | 0.15 | 0.15 |
| max_depth | 1 to 8 | Maximum allowed depth of tree. | 8 | 8 | 8 |
| min_child_weight | 1 to 5 | Minimum sum of instance weights required in a child. | 1 | 1 | 1 |
| n_estimators | 10 to 200 | Number of boosting rounds, or the number of trees added to the model. | 200 | 200 | 200 |
| subsample | 0.5 to 1.0 | Proportion of the dataset to be randomly sampled for each tree. | 1.0 | 1.0 | 0.83602 |

This study took full advantage of XGBoost's powerful algorithm by carefully adjusting it to match the unique properties of the test dataset using the structured methodology described in the table. The selected hyperparameters demonstrate a wise balance between personalization and effectiveness, underscoring the thoroughness and accuracy of the study methods.

## 3.7 Model evaluation metrics

Once the machine learning models have been developed, understanding how well the models perform is essential for determining their accuracy and reliability. As highlighted in Chapter 02, various evaluation metrics exist. Bekkar et al. (2013) claimed that a Confusion Matrix is often used to assess the performance of a classifier in machine learning. For binary classification problems, this matrix is typically a 2x2 grid, as illustrated in the following table:

*Table 09: Confusion Matrix for two classes classification (Bekkar et al., 2020, p. 27)*

| | Predicted Negative | Predicted Positive |
|---|---|---|
| **Actual Negative** | TN (Number of True Negative) | FP (Number of False Positive) |
| **Actual Positive** | FN (Number of False Negative) | TP (Number of True Positive) |

The Confusion Matrix uses abbreviations TN, FN, FP, and TP, which stand for the following:

TN = True Negative, denoting the negative instances accurately classified as negative,
FN = False Negative, indicating the positive instances incorrectly labeled as negative,
FP = False Positive, which means the negative instances wrongly predicted as positive,
TP = True Positive, representing positive instances correctly predicted as positive.

The Confusion Matrix serves as the foundation from which several other common metrics are derived including Accuracy score, Precision, Recall, F1 score, etc. that are often utilized in evaluating classification models. However, when dealing with imbalanced datasets, as in this study, the conventional metrics might not provide a complete or comprehensive picture of a model's actual performance. Imbalanced datasets pose a unique challenge: since prediction errors between the majority and minority classes can result in misleading performance assessments. This is compounded by the general assumption made by many standard metrics that equal class distributions exist, as noted by Brownlee (2020). Due to these difficulties, it is essential to use evaluation metrics tailored for imbalanced scenarios.

Allwright (2022), emphasized on using multiple metrics as it is advantageous to track many metrics when creating a machine learning model since they demonstrate various performance facets. However, he further, specifically, focused on Macro F1 score as a primary choice of metrics due to its balance of precision and recall and its efficacy with imbalanced datasets. Macro F1 is the average of the F1 scores for each class whereas F1 score is the harmonic mean of precision and recall. Another popular metric for evaluating the accuracy of projected probabilities in binary classification is LogLoss, also known as cross-entropy or negative log likelihood. This metric effectively counts the average deviation between the probability distributions that happened and those that were projected. For a flawless classifier, the ideal value of log loss is 0.0. Poor classifiers, on the other hand, may have Log loss values that range from a positive value to infinity (Brownlee 2020).

Apart from this, AUC (Area Under the Curve) score, also found to be the most common metric that was utilized by all the researchers that this study has reviewed in Chapter 02, measures a model's aptitude for distinguishing between classes regardless of their distribution. The interpretation can be as follows: AUC value 0.5-0.6 is poor, 0.6-0.7 is fair, 0.7-0.8 is good, 0.8-0.9 is very good, and 0.9-1.0 is excellent. Other valuable measures for an imbalanced dataset are the Geometric Mean/G-mean and Balanced accuracy. G-mean focuses on the balance between sensitivity and specificity while the latter is the arithmetic mean of the two. For both metrics, a higher score indicates a better-performing model (Bekkar et al., 2013; Brownlee, 2020; Allwright, 2022). The following table is the representation of the discussed metrics so far:

***Table 10: Some metrics for two classes classification problems***

| Measure | Formula | Notations/Description |
|---|---|---|
| Accuracy | $\frac{TP+TN}{TP+TN+FP+FN}$ | Measures overall performance of model |
| Precision | $\frac{TP}{TP+FP}$ | Measures accuracy of the positive predictions |
| Recall or Sensitivity or TPR | $\frac{TP}{TP+FN}$ | True Positive Rate |
| Specificity or TNR | $\frac{TN}{TN+FP}$ | True Negative Rate |
| FPR | $\frac{FP}{TN+FP} = 1 - Specificity$ | False Positive Rate |
| FNR | $\frac{FN}{TP+FN} = 1 - Sensitivity$ | False Negative Rate |
| F1 score | $\frac{2 \times (precision \times recall)}{(precision + recall)}$ | Harmonic mean of Precision and Recall |
| Macro F1 score | $\frac{1}{N}\sum_{i=1}^{N} F1_i$ | $N$ is the number of classes and $F1_i$ is the F1 score of the $i^{th}$ class |
| Balanced accuracy | $\frac{Sensitivity + Specificity}{2}$ | Arithmetic mean of Sensitivity and Specificity |
| LoggLoss | $-((1-y) \times \log(1-\hat{y}) + y \times \log(\hat{y}))$ | $y$ represents expected values and $\hat{y}$ represents predicted values |
| G-mean | $\sqrt{Sensitivity \times Specificity}$ | Balances classification performances on both the majority and minority classes |

Additionally, graphical techniques that capture the nuances that conventional measures might miss, such as the ROC curve and Precision-Recall curves, can provide more in-depth insights into model performance. The ROC curve plots the trade-off between the FPR (x-axis) and TPR (y-axis) for various threshold values. An AUC of 0.50 denotes no discrimination (equivalent to random guessing), whereas an AUC of 1.0 denotes perfect discrimination. The Precision-Recall curve, also known as the PR curve, plots Recall (x-axis) against Precision (y-axis) for various threshold values. Unlike the ROC curve, which uses both true negatives and false positives, the Precision-Recall curve only considers the positive (often the minority) class. A no-skill classifier shows as a horizontal line on the figure, with precision matching the positive instance ratio of the dataset (0.5 for balanced datasets). Conversely, the top right point signifies a flawless classifier (Brownlee, 2020). Average Precision/AP provides a comprehensive summary of the Precision-Recall curve. It calculates the weighted average of the Precision values achieved at each threshold level and thus offers a single score to evaluate the models (Saxena, 2022). Therefore, it is evident that different techniques help evaluate machine learning models from different perspectives. While some offer broad, quantifiable measures, other may provide even deeper insights by shedding light on intricate nuances.

However, to summarize, the methodology outlined in this chapter provides a comprehensive framework for addressing the research questions posed at the commencement of this study. The methods used are based on the literature, best practices, and are adapted to the unique circumstances and difficulties of this research. The thorough explanation guarantees that the procedure is transparent, repeatable, and open to criticism, supporting the validity and reliability of the results. The rigor and robustness of these approaches will be crucial in ensuring that the results are relevant and reliable as the research transitions from methodological design to empirical analysis in the subsequent chapter.

# 4 Experimental results and analysis

As established in the preceding chapters, evaluating the effectiveness of machine learning models is crucial. In this study, to determine how well all four developed models (Logistic Regression, Decision Trees, Random Forests and XGBoost) work with the unseen data (test set), a range of different evaluation metrics have been leveraged. This chapter methodically presents and analyzes the performance of these models across different sampling techniques.

## 4.1 Base models evaluation

The initial experiments focused on understanding the fundamental performance of the machine learning algorithms without any parameter optimization. This serves as the basis and provides a preliminary view of the capability of each algorithm to predict backorders. The following table showcases the results of the base models. To provide a holistic view of each model's performance, notable metrics, such as Accuracy, Balanced accuracy, Precision, Recall, F1 Score, Macro F1 Score, G-mean, ROC/AUC Score, and LogLoss are included:

***Table 11: Base models' performance evaluation***

| Model | Sampling Technique | Threshold | Accuracy | Balanced accuracy | Precision | Recall | F1 Score | Macro F1 | G-mean | ROC/AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LR | ORIGINAL | Default | 0.98 | 0.50 | 0.15 | 0.0 | 0.01 | 0.5 | 0.06 | 0.85 | 0.07 |
| | | 0.04 | 0.96 | 0.59 | 0.1 | 0.21 | 0.14 | 0.56 | 0.46 | | |
| | RUS | Default | 0.98 | 0.52 | 0.16 | 0.04 | 0.06 | 0.53 | 0.19 | 0.87 | 0.12 |
| | | 0.29 | 0.97 | 0.60 | 0.13 | 0.23 | 0.16 | 0.57 | 0.47 | | |
| | SMOTENC | Default | 0.97 | 0.57 | 0.15 | 0.16 | 0.15 | 0.57 | 0.39 | 0.87 | 0.19 |
| | | 0.44 | 0.96 | 0.63 | 0.13 | 0.29 | 0.18 | 0.58 | 0.53 | | |
| DT | ORIGINAL | Default | 0.98 | 0.53 | 0.22 | 0.07 | 0.11 | 0.55 | 0.27 | 0.54 | 0.62 |
| | | 0.01 | 0.98 | 0.54 | 0.18 | 0.09 | 0.12 | 0.55 | 0.29 | | |
| | RUS | Default | 0.95 | 0.66 | 0.11 | 0.36 | 0.17 | 0.57 | 0.58 | 0.66 | 1.90 |
| | | 0.50 | 0.95 | 0.66 | 0.11 | 0.36 | 0.17 | 0.57 | 0.59 | | |
| | SMOTENC | Default | 0.98 | 0.54 | 0.17 | 0.08 | 0.11 | 0.55 | 0.27 | 0.54 | 0.68 |
| | | 0.01 | 0.98 | 0.54 | 0.15 | 0.09 | 0.11 | 0.55 | 0.30 | | |
| RF | ORIGINAL | Default | 0.98 | 0.51 | 0.24 | 0.01 | 0.02 | 0.51 | 0.11 | 0.72 | 0.18 |
| | | 0.06 | 0.96 | 0.67 | 0.17 | 0.36 | 0.23 | 0.61 | 0.59 | | |
| | RUS | Default | 0.97 | 0.65 | 0.19 | 0.32 | 0.24 | 0.61 | 0.56 | 0.89 | 0.11 |
| | | 0.46 | 0.97 | 0.67 | 0.18 | 0.36 | 0.24 | 0.61 | 0.59 | | |
| | SMOTENC | Default | 0.98 | 0.52 | 0.24 | 0.04 | 0.07 | 0.53 | 0.20 | 0.78 | 0.17 |
| | | 0.16 | 0.97 | 0.62 | 0.19 | 0.26 | 0.22 | 0.60 | 0.51 | | |
| XGB | ORIGINAL | Default | 0.99 | 0.50 | 0.36 | 0.01 | 0.01 | 0.5 | 0.08 | 0.91 | 0.06 |
| | | 0.09 | 0.97 | 0.66 | 0.23 | 0.34 | 0.27 | 0.63 | 0.58 | | |
| | RUS | Default | 0.97 | 0.66 | 0.23 | 0.34 | 0.27 | 0.63 | 0.57 | 0.91 | 0.08 |
| | | 0.53 | 0.98 | 0.65 | 0.24 | 0.31 | 0.27 | 0.63 | 0.55 | | |
| | SMOTENC | Default | 0.98 | 0.55 | 0.32 | 0.1 | 0.15 | 0.57 | 0.32 | 0.91 | 0.06 |
| | | 0.28 | 0.97 | 0.64 | 0.22 | 0.29 | 0.25 | 0.62 | 0.53 | | |

Even though the table showcases a holistic view of several metrics, given the complexity and specific challenges of the study, more informative metrics compared to their traditional counterparts, such as Balanced accuracy and Macro F1 are emphasized to streamline the result analysis. Balanced accuracy provides a more equitable measure of model performance across different classes than traditional Accuracy, while Macro F1 offers a holistic view of the model's performance across all classes, distinct from the standard F1 Score (Dalvi, 2021).

Initially, class imbalance with a default threshold value of 0.50 hindered model performance. However, introducing an optimized threshold improved Recall significantly, even though a tradeoff between Precision and Recall was observed. This optimization positively influenced several metrics, notably Balanced accuracy, Macro F1, and G-mean scores in most cases.

For Logistic Regression/ LR, utilizing the ORIGINAL sampling technique, the model registered an ROC/AUC score of 0.85 and a LogLoss of 0.07. An optimized threshold value of 0.04 led to an improvement in Balanced accuracy, Macro F1, and G-mean scores to 0.59, 0.56, and 0.46, respectively. When the RUS sampling technique was implemented, the model displayed an ROC/AUC score of 0.87 and a LogLoss of 0.12. Utilizing an optimized threshold value of 0.29, a more pronounced increment was observed in terms of Balanced accuracy, Macro F1, and G-mean, reaching scores of 0.60, 0.57, and 0.47, respectively. The most significant advancement, however, occurred with the SMOTENC sampling technique, paired with an optimized threshold value of 0.44. Here, the scores rose to 0.63, 0.58, and 0.53, respectively. The model achieved an ROC/AUC score of 0.87 and a LogLoss of 0.19.

For the Decision Trees/ DT model, using the ORIGINAL sampling technique, the ROC/AUC score was 0.54 with a LogLoss of 0.62. Introducing an optimized threshold value of 0.01 resulted in an improved Balanced accuracy and G-mean scores, 0.54 and 0.29, respectively. With the RUS sampling technique, the ROC/AUC score improved to 0.66, though it came with a significantly higher LogLoss of 1.9. However, the optimal threshold was discovered to be 0.50, which is the same as the default threshold. Despite this, the G-mean score saw a small improvement, rising to a value of 0.59. This implies that the model performance may have been improved by the RUS sampling technique itself. The SMOTENC sampling technique, coupled with the optimized threshold value of 0.01 also improved the G-mean to 0.30 which again shows the significance of the sampling technique itself. However, the model displayed an ROC/AUC score of 0.54 and a LogLoss of 0.68.

For Random Forests/ RF, using the ORIGINAL sampling technique, the model registered an ROC/AUC score of 0.72 and a LogLoss of 0.18. An optimized threshold value of 0.06 marked significant improvement in metrics: Balanced accuracy reached 0.67, while G-mean improved dramatically to 0.59. The Macro F1 Score also saw a commendable rise to 0.61. With the RUS sampling technique, the model displayed an impressive ROC/AUC score of 0.89 with a reduced LogLoss of 0.11. Even though the default threshold already showed improved metrics, such as Balanced accuracy of 0.65, Macro F1 of 0.61, and G-mean of 0.56, the optimized threshold of 0.46 improved its Balanced accuracy (0.67) and G-mean (0.59) while the Macro F1 remained the same. The SMOTENC sampling technique with an optimized threshold of 0.16 enhanced performance, with Balanced accuracy, Macro F1, and G-mean reaching 0.62, 0.60, and 0.51, respectively. In this setup, the model achieved an ROC/AUC score of 0.78 and a LogLoss of 0.17.

The XGBoost/XGB classifier with the ORIGINAL sampling technique, the model recorded an ROC/AUC score of 0.91 with a LogLoss of 0.06. Applying an optimized threshold value of 0.09 led to a notable increment in Balanced accuracy, Macro F1, and G-mean scores, reaching values of 0.66, 0.63, and 0.58, respectively. In the case of RUS sampling, the model achieved an ROC/AUC score of 0.91 and a LogLoss of 0.08. The default threshold already presented enhanced metrics, with a Balanced accuracy of 0.66, a Macro F1 of 0.63, and a G-mean of 0.57. However, when the optimized threshold of 0.53 was introduced, the Balanced accuracy and the G-mean decreased slightly to 0.65, and 0.55, respectively, while the Macro FI remained the same. Lastly, utilizing the SMOTENC sampling technique, the model displayed an ROC/AUC score of 0.91 and a LogLoss of 0.06. An optimized threshold of 0.28 further refined the Balanced accuracy, Macro F1, and G-mean scores to 0.64, 0.62 and 0.53, respectively.

It is evident upon the comprehensive evaluation of the base models across various metrics that different models have their strengths and potential weaknesses when faced with imbalanced datasets. While all the metrics offered valuable insights into the models' ability to distinguish between classes, some of the metrics including Balanced accuracy, Macro F1, and G-mean were enhanced in most cases when optimized threshold values were used. These findings serve as a foundation for the subsequent exploration of hyperparameter-tuned models. After gaining the foundational insights from the base models, the next pivotal step is assessing the performance of the hyperparameter-tuned models, using the similar metrics.

## 4.2 Hyperparameter tuned models evaluation

The process of hyperparameter tuning can significantly influence the performance of any machine learning model, optimizing it for the specific intricacies of the dataset in hand. This section provides a deeper evaluation of the model performance following meticulous hyperparameter tuning. With the optimal configurations, by addressing the difficulties presented by the imbalanced datasets, these models are expected to demonstrate improved predictive power. Similar to the base model evaluation, the following table utilizes notable metrics to provide a holistic view of tuned models' performance:

***Table 12: Hyperparameter tuned models' performance evaluation***

| Model | Sampling Technique | Threshold | Accuracy | Balanced accuracy | Precision | Recall | F1 Score | Macro F1 | G-mean | ROC/AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **LR** | ORIGINAL | Default | 0.62 | 0.77 | 0.04 | 0.93 | 0.07 | 0.42 | 0.76 | 0.87 | 0.52 |
| | | 0.74 | 0.95 | 0.65 | 0.11 | 0.35 | 0.16 | 0.57 | 0.58 | | |
| | RUS | Default | 0.62 | 0.77 | 0.04 | 0.93 | 0.07 | 0.42 | 0.76 | 0.87 | 0.52 |
| | | 0.74 | 0.95 | 0.65 | 0.11 | 0.35 | 0.16 | 0.57 | 0.58 | | |
| | SMOTENC | Default | 0.64 | 0.78 | 0.04 | 0.93 | 0.07 | 0.42 | 0.77 | 0.88 | 0.51 |
| | | 0.77 | 0.96 | 0.63 | 0.13 | 0.28 | 0.17 | 0.58 | 0.52 | | |
| **DT** | ORIGINAL | Default | 0.85 | 0.83 | 0.07 | 0.81 | 0.14 | 0.53 | 0.83 | 0.90 | 0.35 |
| | | 0.91 | 0.97 | 0.65 | 0.22 | 0.32 | 0.26 | 0.62 | 0.56 | | |
| | RUS | Default | 0.97 | 0.65 | 0.22 | 0.32 | 0.26 | 0.62 | 0.56 | 0.90 | 0.09 |
| | | 0.45 | 0.97 | 0.69 | 0.20 | 0.39 | 0.27 | 0.63 | 0.62 | | |
| | SMOTENC | Default | 0.97 | 0.61 | 0.18 | 0.24 | 0.21 | 0.60 | 0.49 | 0.82 | 0.99 |
| | | 0.30 | 0.96 | 0.66 | 0.16 | 0.34 | 0.22 | 0.60 | 0.57 | | |
| **RF** | ORIGINAL | Default | 0.99 | 0.50 | 0.36 | 0.01 | 0.01 | 0.50 | 0.08 | 0.90 | 0.06 |
| | | 0.08 | 0.97 | 0.66 | 0.21 | 0.34 | 0.26 | 0.62 | 0.58 | | |
| | RUS | Default | 0.97 | 0.65 | 0.21 | 0.32 | 0.26 | 0.62 | 0.56 | 0.91 | 0.08 |
| | | 0.49 | 0.97 | 0.66 | 0.21 | 0.33 | 0.26 | 0.62 | 0.57 | | |
| | SMOTENC | Default | 0.98 | 0.55 | 0.29 | 0.10 | 0.15 | 0.57 | 0.32 | 0.90 | 0.08 |
| | | 0.18 | 0.97 | 0.67 | 0.19 | 0.37 | 0.26 | 0.62 | 0.60 | | |
| **XGB** | ORIGINAL | Default | 0.99 | 0.50 | 0.40 | 0.01 | 0.02 | 0.50 | 0.09 | 0.91 | 0.06 |
| | | 0.09 | 0.97 | 0.65 | 0.23 | 0.32 | 0.26 | 0.63 | 0.56 | | |
| | RUS | Default | 0.97 | 0.66 | 0.22 | 0.33 | 0.27 | 0.63 | 0.57 | 0.90 | 0.08 |
| | | 0.48 | 0.97 | 0.67 | 0.22 | 0.35 | 0.27 | 0.63 | 0.59 | | |
| | SMOTENC | Default | 0.98 | 0.53 | 0.38 | 0.06 | 0.11 | 0.55 | 0.25 | 0.91 | 0.06 |
| | | 0.16 | 0.97 | 0.67 | 0.19 | 0.37 | 0.25 | 0.62 | 0.60 | | |

Post hyperparameter tuning, the models' performance profiles showed interesting changes. Class imbalance remained a difficult obstacle while dealing with the default threshold of 0.50. In particular, the Logistic Regression classifier displayed unusually high Recall values, often at the expense of Precision. This pattern suggests that the minority class has a high true positive rate but also a significant number of false positives. This observation was confirmed by the strikingly low Precision values for these models with default threshold. The majority of models and sampling strategies saw a significant boost in Precision after the introduction of optimized thresholds, however this came at the expense of a little decline in Recall. This illustrates a more balanced approach, where the models successfully distinguished between the classes without unduly favoring either one. Similar to the base models, this also led to an enhanced scores across several metrics, such as Balanced accuracy, Macro F1, and G-mean scores in most cases.

For the hyperparameter tuned Logistic Regression model, when the ORIGINAL sampling technique was employed, the ROC/AUC score stood at 0.87 and the LogLoss at 0.52. When the optimized threshold of 0.74 was introduced, Balanced accuracy, Macro F1, and G-mean dropped noticeably to 0.65, 0.57, and 0.58, respectively. Using the RUS sampling technique, the outcomes mirrored the previous set, with the model producing an ROC/AUC score of 0.87 and a LogLoss of 0.52. However, utilizing an optimal threshold of 0.74 had a mixed effect: Balanced accuracy and G-mean were reduced to 0.65 and 0.58, respectively, while the Macro F1 significantly increased to 0.57. Lastly, the model earned a ROC/AUC score of 0.88 and a LogLoss of 0.51 using the SMOTENC sampling strategy. Here also, the Balanced accuracy and the G-mean reduced to 0.63 and 0.52, respectively at the optimum threshold value of 0.77, but the Macro F1 increased to 0.58.

For the hyperparameter tuned Decision Trees model, with the ORIGINAL sampling technique, the model demonstrated an ROC/AUC score of 0.90 and a LogLoss of 0.35. Yet, when the threshold was raised to 0.91, the G-mean and the Balanced accuracy fell to 0.65 and 0.56, respectively. However, the Macro F1 score climbed to 0.62. The model, which used the RUS sampling method, displayed a favorable LogLoss of 0.09 while the ROC/AUC remained unchanged. With the introduction of an optimized threshold of 0.45, there was an improvement in the Balanced accuracy to 0.69, the Macro F1 score to 0.63, and the G-mean to 0.62. Lastly, for the SMOTENC sampling approach, the model recorded an ROC/AUC score of 0.82 and a LogLoss of 0.99. However, when an optimal threshold of 0.30 was

applied, the Balanced accuracy, the Macro F1, and the G-mean enhanced to 0.66, 0.60, and 0.57, respectively.

For the hyperparameter tuned Random Forests model, when utilizing the ORIGINAL sampling method, it achieved an ROC/AUC score of 0.90 and a LogLoss of 0.06. When an optimized threshold of 0.08 was applied, the Balanced accuracy and the Macro F1 scores jumped to 0.66 and 0.62, respectively, whereas the G-mean took declined to 0.58. While employing the RUS sampling technique, the model slightly increased a ROC/AUC score of 0.91 and a LogLoss of 0.08. An optimal threshold of 0.49 led to the Balanced accuracy, Macro F1, and G-mean scores reaching 0.66, 0.62, and 0.57, respectively. Lastly, using the SMOTENC sampling strategy, the model reported an ROC/AUC score of 0.90 and a LogLoss of 0.08. By applying a threshold value of 0.18, there was an increase in Balanced accuracy to 0.67, Macro F1 to 0.62 and G-mean to 0.60.

For the hyperparameter-tuned XGBoost model, when the ORIGINAL sampling method was applied, the model attained an ROC/AUC score of 0.91 and a LogLoss of 0.06. However, with the introduction of an optimized threshold of 0.09, there were notable changes in performance metrics: the Balanced accuracy, Macro F1, and G-mean scores rose to 0.65, 0.63, and 0.56 respectively. Using the RUS sampling technique, the model registered an ROC/AUC score of 0.90 and a LogLoss of 0.08. On adjusting to an optimal threshold of 0.48, the Balanced accuracy, Macro F1, and G-mean scores enhanced to 0.67, 0.63, and 0.59, respectively. Lastly, with the SMOTENC sampling approach in play, the model registered an ROC/AUC score of 0.91 and a LogLoss of 0.06. A threshold tweak to 0.16 led to the Balanced accuracy jumping to 0.67, Macro F1 to 0.62, and the G-mean to 0.60.

Upon the detailed assessment of the tuned models, it is evident that adjusting specific parameters has a significant effect. Here also, in most scenarios, results with optimized thresholds consistently outperformed the results with default thresholds in terms of a number of important metrics including Balanced accuracy, Macro F1, and G-mean. Hence, there will be an inclination to place more emphasis on the results with optimized thresholds in upcoming analysis.

The benefits of tuning are further illustrated by comparison of these findings with those obtained from the basic models in next sub-chapter .

## 4.3 Comparative analysis: base Vs. tuned models

The base models and the tuned models are compared head-to-head in this sub-chapter. The purpose of this comparison is to highlight the improvements achieved through tuning. In particular, for imbalanced datasets, a detailed examination of results offers insights into how model performance alterations arise from tuning. However, it is important to note that while tuning adjustments significantly impact various performance metrics, they do not affect metrics, such as AUC and LogLoss. Consequently, when other metrics produce outcomes that are comparable to those of these stable measures, it becomes imperative to include them in the juxtaposition table to ensure a more thorough and differentiated comparison between the models.

The following table compares the results of the base models and their tuned counterparts:

***Table 13: Juxtaposition of base model's and tuned model's performance***

| Model | Sampling Technique | Balanced accuracy | | Macro F1 | | G-mean | | ROC/AUC | | LogLoss | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Base | Tuned | Base | Tuned | Base | Tuned | Base | Tuned | Base | Tuned |
| LR | ORIGINAL | 0.59 | 0.65 | 0.56 | 0.57 | 0.46 | 0.58 | 0.85 | 0.87 | 0.07 | 0.52 |
| | RUS | 0.60 | 0.65 | 0.57 | 0.57 | 0.47 | 0.58 | 0.87 | 0.87 | 0.12 | 0.52 |
| | SMOTENC | 0.63 | 0.63 | 0.58 | 0.58 | 0.53 | 0.52 | 0.87 | 0.88 | 0.19 | 0.51 |
| DT | ORIGINAL | 0.54 | 0.65 | 0.55 | 0.62 | 0.29 | 0.56 | 0.54 | 0.90 | 0.62 | 0.35 |
| | RUS | 0.66 | 0.69 | 0.57 | 0.63 | 0.59 | 0.62 | 0.66 | 0.90 | 1.90 | 0.09 |
| | SMOTENC | 0.54 | 0.66 | 0.55 | 0.60 | 0.30 | 0.57 | 0.54 | 0.82 | 0.68 | 0.99 |
| RF | ORIGINAL | 0.67 | 0.66 | 0.61 | 0.62 | 0.59 | 0.58 | 0.72 | 0.90 | 0.18 | 0.06 |
| | RUS | 0.67 | 0.66 | 0.61 | 0.62 | 0.59 | 0.57 | 0.89 | 0.91 | 0.11 | 0.08 |
| | SMOTENC | 0.62 | 0.67 | 0.60 | 0.62 | 0.51 | 0.60 | 0.78 | 0.90 | 0.17 | 0.08 |
| XGB | ORIGINAL | 0.66 | 0.65 | 0.63 | 0.63 | 0.58 | 0.56 | 0.91 | 0.91 | 0.06 | 0.06 |
| | RUS | 0.65 | 0.67 | 0.63 | 0.63 | 0.55 | 0.59 | 0.91 | 0.90 | 0.08 | 0.08 |
| | SMOTENC | 0.64 | 0.67 | 0.62 | 0.62 | 0.53 | 0.60 | 0.91 | 0.91 | 0.06 | 0.06 |

It is evident from the "Table 13" that tuning provided enhanced results across the board. The analysis reveals:

- **Logistic Regression/ LR:** There were improvements in Balanced accuracy, Macro F1, G-mean, and ROC/AUC in most sampling techniques following tuning, with the exception of SMOTENC which showed minimal change. Despite the unfavorable rise in LogLoss for tuned models, other enhanced metrics point to a better performance.

- **Decision Trees/DT:** Tuning resulted in notable gains in Balanced accuracy, Macro F1, G-mean, and ROC/AUC. While LogLoss was generally in favor of the tuned models, there was an exception in the case of SMOTENC sampling.

- **Random Forests/RF:** The outcomes after tuning were mixed. Some sampling techniques showed slight declines in Balanced accuracy and G-mean, yet Macro F1 and ROC/AUC mostly increased. For SMOTENC, all metrics showed improvement. Additionally, the LogLoss consistently favored the tuned models across all sampling methods.

- **XGBoost/XGB:** Post-tuning, Macro F1, ROC/AUC and LogLoss remained consistent almost across all the sampling techniques. There were improvements in other metrics, except a slight decrease in Balances accuracy and G-mean for ORIGINAL sampling.

It is evident from the above analysis that while the effectiveness of tuning varied, it predominantly led to performance improvements or stability in metrics across the models evaluated. In essence, tuning had a generally positive impact on model performance across different metrics and sampling techniques. Hence, the tuned models will receive all of the attention in the following section. This approach will facilitate a more in-depth and targeted examination of their results without reference to their base counterparts.

Now that, it has been determined that the more proficient models predominantly reside within the realm of the tuned space, it becomes imperative to rank these models within each sampling technique. This step is pivotal for identifying the top-performing models. A thorough ranking of these tuned models is covered in next sub-chapter for a clearer picture.

## 4.4 Finding the best models

While the tabulated metrics offer a quantitative snapshot, the narrative remains incomplete without a visual representation. Graphical representations complement these metrics and promise to provide an intuitive visual insight into model performance. In this section, along with considering the tabulated metrics (Table 13), graphical representations, such as the ROC/AUC and PR curves are used to simultaneously display the results of all four models in the context of each distinct sampling technique. This method facilitates a direct comparison of the performance of each tuned model under identical sampling conditions.

### *4.4.1* With "ORIGINAL" sampling approach

The ROC/AUC and PR curves offer a visual validation of the model ranking. When looking at the graphical representations below, several key trends emerge which validate the quantitative metrics in Table 13:

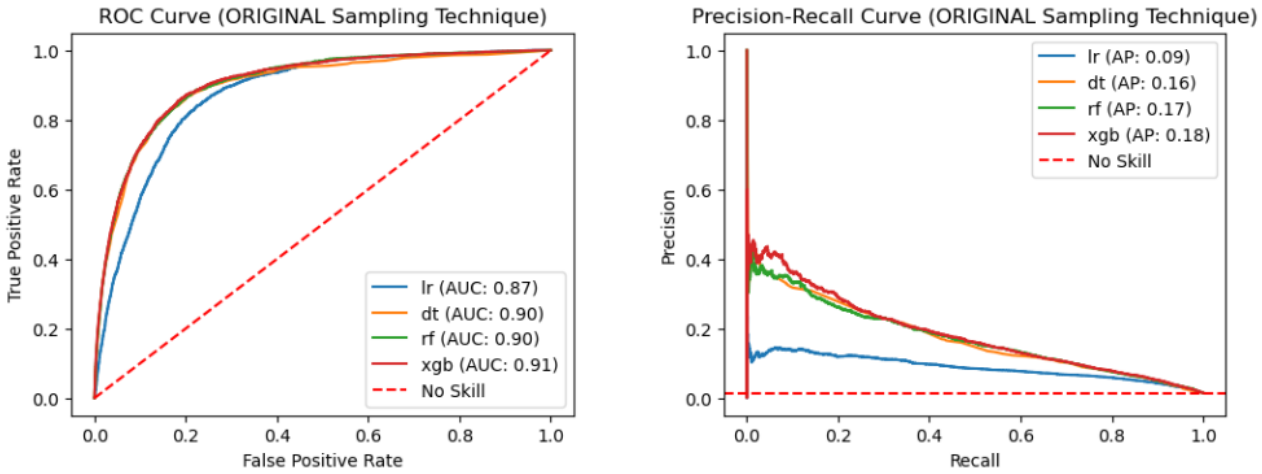



***Figure 23: ROC/AUC and PR curves for the models with ORIGINAL sampling strategy***

The ROC/AUC curve for XGBoost/XGB outperforms others, indicating its superior performance in terms of sensitivity and specificity. Additionally, its PR curve comparatively is closest to the top-right corner, reflecting its optimal precision and recall balance. These visuals complement the tabulated results, confirming XGB's rank at the top for Balanced accuracy, Macro F1, and ROC/AUC. Furthermore, it is also tied for the lowest LogLoss, making it the overall top performer with the ORIGINAL sampling when considering the

combined metrics. Random Forests/ RF comes closely behind, as shown by both the proximity of its ROC/AUC curve to XGB and the height of its PR curve, a performance that is also reflected in its favorable G-mean. Progress is seen by Decision Trees/ DT post-tuning, which places its ROC/AUC curve intermediate to RF and LR. Its PR curve indicates good precision and recall trade-offs, but when juxtaposed with table metrics, it also shows a LogLoss that is noticeably greater than XGB and RF. Despite obvious post-tuning improvements, Logistic Regression/ LR, particularly in the LogLoss domain, still trails behind its competitors.

Therefore, considering both graphical and quantitative evaluations under the ORIGINAL sampling technique, the models can be ordered as follows: XGB > RF > DT > LR

### *4.4.2* With "RUS" sampling approach

Moving on to the analysis utilizing the "RUS" sampling method, the graphical representations (ROC/AUC and PR curves) highlight several performance nuances that are not immediately apparent from the tabulated metrics alone:

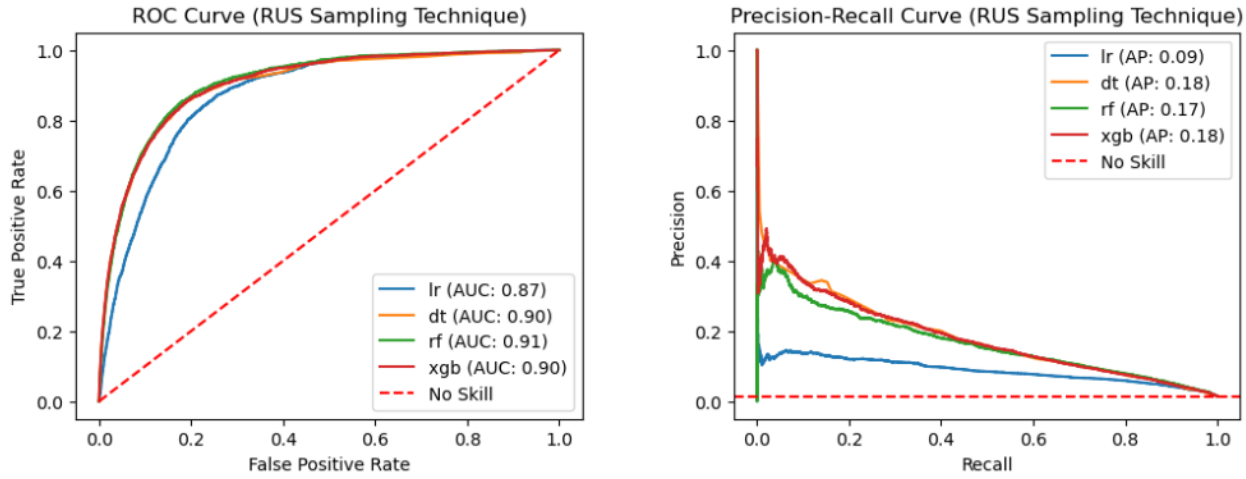



***Figure 24: ROC/AUC and PR curves for the models with RUS sampling strategy***

Comparatively, XGB's graphical representation closely resembles the top-right corners and attest to its top rankings in several tabulated metrics, justifying its exceptional performance. RF exhibits a dip in the PR curve, compared to both XGB and DT, although having an exceptional ROC/AUC score, indicating a trade-off in precision and recall. On the other

hand, DT exhibits curves that demonstrate its improved post-tuning performance and, in certain cases, goes head-to-head with XGB. Despite its advances, the graphical indicators for LR imply it lags behind its peers, especially when considering its tabulated metrics.

Therefore, by combining the quantitative measures and graphical insights, under the RUS sampling technique, the models can be organized as follows: XGB = DT > RF > LR.

### *4.4.3* With "SMOTENC" sampling approach

Transitioning to the analysis based on the "SMOTENC" sampling technique, a review of the associated ROC/AUC and PR curves offers more detail on the model performances, revealing subtler performance dynamics to supplement the tabulated metrics:

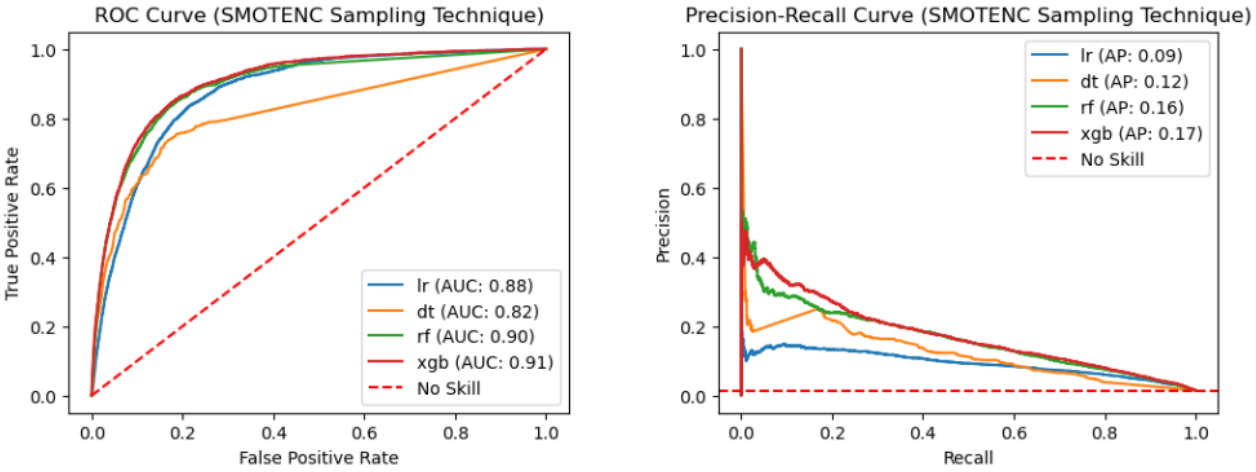



***Figure 25: ROC/AUC and PR curves for the models with SMOTENC sampling strategy***

With its curves attracting to the ideal points, XGB stands out clearly. With regard to the tabulated metrics, it has the highest ROC/AUC and the lowest LogLoss, further demonstrating its supremacy. While RF displays an admirable ROC/AUC curve and metrics that are nearly identical to those of XGB, a closer look reveals a somewhat muted PR curve and slightly worse AUC and LogLoss values. The ROC/AUC performance of DT occasionally falls short of even LR, despite post-tuning enhancements. Yet, DT has better metrics and curves than LR, barring its ROC/AUC and LogLoss. On the other hand, LR does perform better than DT in terms of AUC score and has a more favorable LogLoss, but its PR curve emphasizes how challenging it is to strike a balance between precision and recall. As a

result, when compared to other models, LR performs poorly overall, according to the measures used to evaluate the models.

Therefore, as a result of combining the quantitative measurements with the graphical insights under the SMOTENC sampling approach, the models are ranked as XGB > RF > DT > LR.

Now that the top performers across all three different sampling techniques are identified, comparing those with the available industry bests is both strategic and illuminating. Such a comparison not only highlights the relevance and robustness of the models developed in this study but also provides insights into the underlying questions regarding the effective utilization of predictive analytics in the supply chain and the comparative effectiveness of various machine learning algorithms. The comparison of the pinnacle performers from this study and the state-of-the-art is made in the following sub-chapter.

## 4.5 Elite model showdown: Best of current study Vs. best of state-of-the-art

While there are many indicators available to assess model performance, not all are suitable in every scenario. Apart from it, a thorough investigation of several measures may add needless complications and obscure the analysis's main objective. In the context of this study, the AUC score is a vital metric, considering the special challenges presented by imbalanced datasets. While it provides insights into a model's ability to differentiate between classes, it is important to recognize that it does not capture the entire performance landscape. Other metrics can offer different perspectives on model efficiency, precision, recall, and more. Nonetheless, due to the AUC score's widespread use in existing research, employing this metric facilitates direct comparisons with prior studies.

However, despite the fact that this study uses three different sampling strategies, it is crucial to recognize that some previous research has used sampling approaches that are not clearly detailed. Such discrepancies might lead to variations in model outcomes; hence, this distinction should be considered in any comparisons. The following table only shows the sampling strategies of those research which are specifically mentioned in the corresponding research:

*Table 14: Best of current study Vs. best of state-of-the-art*

| Researchers | ML techniques | AUC |
|---|---|---|
| **Santis et al. (2017)** | RF<br>GBOOST<br>BLAG | 0.94<br>0.95<br>0.95 |
| **Hájek & Abedin (2020)** | LR (Profit-max CBUS)<br>RF (Profit-max CBUS)<br>SVM (Profit-max CBUS) | 0.77<br>0.92<br>0.78 |
| **Islam & Amin (2020)** | GBM<br>DRF | 0.80<br>0.79 |
| **Ntakolia et al. (2021)** | RF (Under Sampling)<br>XGBoost (Under Sampling)<br>LightGBM (Under Sampling)<br>Balanced Blagging (Under Sampling) | 0.95<br>0.95<br>0.95<br>0.95 |
| **Dahilwalkar (2021)** | RF/Random Forest<br>Adaboost/Adaptive Boosting<br>GBDT/Gradient Boosted Decision Trees | 0.95<br>0.94<br>0.95 |
| **Shajalal et al. (2022)** | CNN_100 (ADASYN)<br>MxCNN_50 (ADASYN)<br>MxCNN_100 (ADASYN) | 0.95<br>0.95<br>0.95 |
| **Current study** | XGB (ORIGINAL)<br>DT and XGB (RUS)<br>XGB (SMOTENC) | 0.91<br>0.90<br>0.91 |

The majority of cutting-edge research models have AUC values in the 0.90+ range, demonstrating the effectiveness of contemporary machine learning methods in supply chain scenarios for backorder prediction. Some methods, such as Random Forest/RF and XGBoost, appear to be well-represented in the literature, indicating that they are robust in this application. The models used in this current study had excellent AUC values, up to 0.91. Even though they lag behind a few of the top-performing models, they nonetheless exhibit a high degree of effectiveness.

However, disparities in the datasets utilized in different studies can cause discrepancies in performance. The origin, size, quality, and intrinsic complexity of the dataset, etc. are a few examples of the variables that may affect the model outcomes. Even with the same dataset, variations in preprocessing steps or feature engineering techniques can lead to diverse model results. For example, some crucial elements that are stressed in one study may not be present or emphasized in another. Furthermore, as pointed out in the "Rationale and uniqueness of

the paper" section, certain studies might have resampled the "Test" set or explored Neural Network-based models (a deep learning approach that often produces superior outcomes), leading to higher AUC scores. Beyond the machine learning technique used, the training tactics, computational resources, and hyperparameters chosen can significantly influence model outcomes. All these fine subtleties are not captured by the AUC, even if it offers a useful performance snapshot.

In summary, this comparative analysis highlights the advancements achieved in supply chain operations using machine learning for backorder predictions. Although the models used in this study are competitive with some of the best in the field, there is still room for improvement. Future research opportunities are made possible by the modest changes in AUC values as well as factors, such as dataset discrepancies and model specificities. By investigating these possibilities, the models can be improved even more, getting closer to the highest level of prediction accuracy.

With an aim to provide a holistic view of the current study's achievements and possible directions, a more in-depth examination of model criticisms, usability, and future research directions will be conducted in the following sections.

## 4.6 Model criticism, usability, and further discussion

The quest for achieving the optimal predictive model often leads academics and practitioners to heavily rely on certain evaluation metrics. While the ROC/AUC score is a widely recognized metric, it does not fully represent the models' range of performance when used alone. All of the models in this study had decent or high AUC scores. Nonetheless, disparities in performance were discovered upon careful examination of other important metrics, including the Precision-Recall dynamics.

It is important to note that this was not an isolated incident associated with a particular model. However, one glaring example is the Random Forest model that uses the RUS sampling technique. It performed substantially worse on other measures than its peers within the RUS strategy, despite having the greatest AUC of 0.91. Consequently, it did not clinch a

leading spot based on other metrics within the RUS spectrum. The point is made clear by this striking contrast: a remarkable AUC does not guarantee holistic superiority across all evaluation criteria, as it does not always encapsulate a model's comprehensive performance landscape. Such disparities between the AUC and other metrics indicate the models' limitations in reliably distinguishing the positive from the negative classes. This divergence becomes paramount in situations where false positives have significant consequences. Hence, it is critical to evaluate models from a multi-metric standpoint to provide a more comprehensive evaluation of their strengths and weaknesses. Despite these observations, the models' usefulness is still evident for several reasons:

- **Utility over random guessing:** All of the models clearly outperform random guessing, as shown by the ROC/AUC and PR curves where the "No Skill" line refers to random guessing. This demonstrates their potential value and usefulness in real-world supply chain scenarios, especially for early backorder detection.

- **Nature of the problem and setting realistic expectations:** There are numerous uncontrollable elements that impact the intrinsic difficulty of backorder prediction. These can challenge even the most sophisticated models. While these models are not flawless, it is important to acknowledge their worth as they provide valuable insights. It is imperative to convey to stakeholders that machine learning models have limits and provide probabilistic projections based on past data. Thus, reasonable expectations ought to be established.

- **Business impact:** Even a little decrease in backorders can result in significant cost savings and higher customer satisfaction. Instead of concentrating only on metric scores, it is critical to consider the models in the context of their larger business impact.

- **Interpretable models:** Beyond its ability to anticipate outcomes, a model has other uses as well. Even when their absolute prediction accuracy is not the best in the industry, models that provide insights into feature importance can direct corporate strategies and decision-making procedures.

- **Feedback loop:** Setting up a feedback loop can be considered where predictions from the models are validated with actual outcomes. This input can be utilized to retrain and possibly enhance the model over time.

- **Human-in-the-loop systems:** Businesses can utilize the models as an initial filter or recommendation system, and then have domain experts assess the predictions made by the algorithms. In this manner, they can leverage both the models' capabilities and human expertise.

- **External factors:** Sometimes it is possible that some of the elements impacting the result are not captured by the data alone. There might be external elements (e.g., abrupt changes in the market, global events) that impact backorders but are not present in the dataset.

To summarize, although a highly effective predictive model would be optimal for supply chain backorder forecasting, even moderately effective models can provide useful insights, particularly when skillfully incorporated into the decision-making process. After all, the goal is not just high metric scores but tangible business impact and insights into the supply chain.

# 5 Conclusion and future work

Recent years have seen a growing focus on the study of the interaction between supply chain operations and predictive analytics. Increasingly, companies are realizing the importance of precise predictive models as they struggle with inventory control, backorder reduction, and demand forecasting. Hence, it is impossible to underestimate the importance of anticipating any backorders. The primary objective of this research was to investigate the accuracy and performance of predictive analytics models in identifying early warning signs of such backorders, thereby enabling businesses to adopt proactive inventory management approaches. This study meticulously evaluated the effectiveness of different machine learning algorithms in predicting backorders within the supply chain's scope. Moreover, the study explored how parameter adjustment affects these algorithms' accuracy and performance. A variety of performance criteria were also used to evaluate the models.

Upon reviewing the research process and its results, it is crucial to revisit the original goals, assess how well they were met, and consider the broader ramifications of the findings. This chapter outlines the lessons learned, addresses difficulties encountered, and recommends directions for further study in this important area.

## 5.1 Recapitulation of the study's outcomes

As this study's conclusion approaches, it is critical to summarize its main conclusions and ensure the research questions have been tackled with careful consideration and analytical rigor, echoing the sentiments of Lapan, Quartaroli & Riemer (2012). A succinct summary of the research process and outcomes will set the stage for answering the research questions.

A rigorous sequence of processes led to the development of this investigation:

The basis was established by a thorough analysis of relevant literature. Exploratory Data Analytics/ EDA technique was employed to explore the datasets, followed by extensive preprocessing measures, such as data cleaning, feature engineering, scaling, and resampling, etc. In the whole process, vigilant procedures were taken to ensure that no data leaked. An array of evaluation criteria, specifically designed for the imbalanced nature of the datasets, was used to analyze the outcomes of machine learning models that were methodically trained and fine-tuned. After comparing the performance of the outstanding models to industry norms, the models were critically examined, with an emphasis on their potential limitations and areas of applicability.

With this backdrop, this paper sought to answer:

**RQ-1.** How can predictive analytics be effectively utilized to identify early warning signs of potential backorders within the supply chain?

**Answer to RQ-1:** The application of diverse machine learning algorithms in predictive analytics has demonstrated its effectiveness in detecting preliminary indications of impending backorders. Businesses can foresee and take effective measures to handle future stockouts

with the help of the developed and tested models in this study, each of which has strengths and weaknesses of its own. Companies can put their proactive inventory management tactics into practice by using these models to enhance their supply chain resilience.

**RQ-2.** What is the comparative effectiveness of different machine learning algorithms in predicting backorders in the context of supply chain operations? Additionally, how does parameter tuning impact the performance and accuracy of these algorithms?

**Answer to RQ-2:** The effectiveness of different machine learning algorithms in making backorder forecasts varies among models. Even while some models demonstrated high AUC scores, it was discovered that these scores do not always necessarily correspond with other important performance metrics. This discrepancy highlights the importance of a multi-metric evaluation approach. Regarding parameter tuning, it is evident that adjusting parameters can have a substantial impact on the performance and accuracy of algorithms. A model's predictive power can be maximized with the right set of parameters. However, finding a balance is crucial to maintain generalizability and prevent overfitting.

To sum up, the findings emphasize the potential of predictive analytics in inventory management, while also stressing the significance of a well-informed and nuanced strategy. Predictive models have the capacity to be revolutionary, yet there is no universal solution or widely acknowledged “best” model. Rather, consistent with the outlined research objectives, success depends on understanding the specific context and the thoughtful application of these analytical instruments.

## 5.2 Reflection on contributions and limitations

Reflecting upon the journey of this research allows for a holistic understanding of its value and potential areas for improvement. It is essential to present a comparative analysis of the goals and actual results, which emphasizes the importance of the research and highlights any limitations. In addition to openly discussing the challenges faced, this section seeks to clarify the significant contributions made to the fields of supply chain management and predictive analytics.

### *5.2.1* Contributions revisited

The journey of this study unveiled several pivotal findings, which are worth revisiting:

- **Bridging theoretical and practical avenues:** This research not only provides scholarly insights but also bridges the gap between theoretical constructs and their actual application. The proposed strategies and overall findings in this study can help companies apply predictive analytics to real-world problems.

- **Enriching the academic landscape:** Building upon the "Rationale and uniqueness of the paper" in Chapter 2, this paper further solidifies its stand in the academic landscape by its in-depth and reader-friendly exploration of theoretical foundations. Furthermore, the methodologies utilized in this study throughout its various stages, from data collection to model building and interpretation, present a novel approach. By comparing these approaches with the established literature, future researchers can gain a fresh perspective to adopt and/or further refine the existing knowledge. Thus, this paper serves as a testament to the dynamic nature of theoretical and methodological rigor in the supply chain domain.

- **Optimization blueprint:** Businesses can use the findings as a blueprint to review and improve their inventory management procedures. These useful insights can result in more efficient operations and better resource management.

- **Customer-centric outcomes:** This research has led to a focus on proactive backorder management, which is evidence of a customer-centric approach. Customer loyalty and trust can be greatly increased by anticipating and controlling such hiccups in the supply chain industry.

- **A paradigm shift:** The study advocates for a transformative change in the way firms understand and handle backorders. By promoting proactive tactics over reactive ones, this study lays the framework for further research in this area.

### *5.2.2* Reflecting on limitations

It is imperative to acknowledge the limitations that this research was conducted under:

- **Dataset specificity:** Although the datasets provide valuable insights, it is worth noting that they only represent particular scenarios. When applied to a different dataset or environment, the suggested interpretations and techniques might need recalibration.

- **Assumptions in the spotlight:** The underlying assumptions of this study may have influenced the results to some extent. Future scholars and practitioners need to be aware of these nuances.

- **Narrow focus:** Despite being thorough, the study had some focal points, especially in relation to specific dataset preparation techniques, resampling strategies, prediction algorithms and tuning, as well as evaluation criteria. As a result, the insights have their locus within a pre-defined scope.

- **Product-specific backorder prediction:** While not explicitly stated as a limitation in the "Introduction" chapter, it is worth noting that this research broadly addresses backorder prediction without knowing and specifying which particular products or product types are at risk of going backorder. Different products have varying levels of importance to a business, and the implications of a backorder might vary based on the product type. Hence, it is important to understand the specific inventory and business needs when evaluating machine learning models, where especially metrics such as precision and recall might have distinct impacts.

In summary, although the research has undeniably advanced in addressing the gaps in the field of supply chain management, it had its limitations. Such a comprehensive view ensures that the knowledge gained from the study is contextualized appropriately and that the achievements and limitations encountered here can be applied to future research projects.

## 5.3 Personal research challenges

Having reflected upon the study's contributions and constraints, it is equally important to draw attention to the difficulties encountered during the research process. Several challenges were encountered while conducting this research, each playing a pivotal role in shaping the approach, methodology, and even the outcomes of the study:

- **Understanding dataset variables:** A primary obstacle that needed to be overcome was a thorough understanding of the variables in the datasets. Due to the lack of information in existing research with the same/similar dataset, domain knowledge and professional judgment had to be applied to understand the variables, as obtaining a complete understanding was essential for the next analytical procedures.

- **Handling missing values:** Preprocessing of the data introduced difficulties due to the significant missing values in the datasets. It was a complex task to determine the appropriate imputation techniques and ensure that they did not skew the results.

- **Outliers in the datasets:** In terms of outliers, the dataset posed a unique challenge. Conventional methods, such as box plots and histograms, proved ineffective for outlier detection as extreme values affected the visual representation. To manage this, an additional test (percentile testing) was conducted to better comprehend and address these outliers. However, substantial outliers remained in the prepared dataset even after the top 1% of the data were eliminated.

- **Feature importance for ML models:** A rigorous feature selection and importance analysis were needed to identify the variables that had the most predictive potential for the machine learning models.

- **Computational resources and training time:** Due to the huge-sized dataset, the training of the selected machine learning models required a massive amount of processing power. To illustrate, with decent computational capabilities, training some of the models required more than two days. Any requirement for retraining meant a comparable time commitment, whether due to any discrepancies identified after

training or other improvements. Apart from it, the percentage choice of sampling techniques was impacted by the computing needs; for example, lower percentages of SMOTENC had to be used due to limited resources. Furthermore, these constraints also limited the feasibility of exploring more computationally demanding methods, such as Support Vector Machines/ SVM and specific Neural Network/ NN architectures. Such constraints had numerous effects, affecting everything from data preparation and model selection to hyperparameter tuning.

- **Performance evaluation with suitable metrics:** The success of the study depended heavily on the use of suitable metrics. The challenge was not only about identifying them but also interpretation was troublesome as they displayed mixed behavior. There were instances where one metric indicated satisfactory model performance, while another suggested otherwise. Navigating this inconsistency and understanding the balance between metrics required careful consideration and judgment, which made the evaluation phase more intricate than initially anticipated.

Overall, these challenges prove that the intricacies involved in real-world data analysis and model training go beyond theoretical ideas. While this study overcame these obstacles to provide valuable insights, acknowledging these challenges not only clarifies the scope of this study but also paves the way for subsequent research endeavors.

Nevertheless, along with these above-mentioned personal challenges, it is crucial to recognize and address practical implementation challenges that are often overlooked in academic discussions. The following sub-subsection examines these challenges in greater detail and provides recommendations for their mitigation.

## 5.4 Addressing practical implementation challenges in SCM

The dynamics of the supply chain could be significantly transformed by utilizing advanced analytics, especially in the area of backorder prediction. While this study offers a solid basis, applying these models to real-world contexts can introduce a unique set of challenges. These difficulties can determine how useful these models are in real-world scenarios, from complex data to changing supply chain conditions.

The specific challenges and their possible solutions are presented below:

- **Data collection and integrity:** Collecting real-time and accurate data is essential for developing machine learning models. Data can be scattered across various systems in the field of supply chain management, especially concerning backorders.

  **Recommendation:** The implementation of centralized data warehousing solutions is recommended. Therefore, tools and/or platforms that offer seamless integration with existing ERP/Enterprise Resource Planning/ ERP and WMS/ Warehouse Management System software should be considered.

- **Model adaptability:** Due to the dynamic nature of supply chains and technology, models might become outdated as new trends and patterns evolve.

  **Recommendation:** It is advisable to schedule regular model retraining sessions as the developed model should update its weights on a regular basis to reflect the latest data. A hybrid approach can be utilized by combining batch and online learning.

- **Scalability concerns:** The volume of data to be processed and analyzed also increases with the expansion of business. This highly likely will affect the model's building and evaluation processes as well as performance.

  **Recommendation:** It is advisable to opt for scalable cloud-based machine learning solutions that can effectively process increasing volumes of data.

- **Feature evolution:** The significance of features may change over time. A formerly essential predictor might experience a decline in its importance while new vital predictors may evolve.

  **Recommendation:** It is advisable to continuously monitor the feature importance scores and integrate this monitoring within the model evaluation phase. In case of any changes in the feature importance scores, reengineering of the model should be considered.

- **Operational integration:** Having a predictive model serves only as an initial phase in the process. To ensure effectively impacting decision-making, a seamless integration of the system into daily operations is needed.

  **Recommendation:** It is advisable to develop user-friendly dashboards and interfaces, enabling stakeholders (with low/no technical expertise) to leverage model insights. Along with the accessibility of the prediction, it is also imperative to ensure its effective usage.

- **Stakeholder resistance:** Due to a lack of awareness about the benefits of advanced analytics and machine learning, fear of change, or even technophobia, traditional supply chain managers or personnel might oppose the idea of implementing predictive analytics. Such resistance can pose a substantial obstacle.

  **Recommendation:** It is advisable to conduct regular training and awareness sessions to promote predictive analytics by emphasizing its advantages. This will help stakeholders embrace it, realizing predictive analytics meant to complement the human decision-making process in the supply chain, not replace it.

All in all, it is evident that implementing advanced analytics for effectively predicting backorders is a multifaceted endeavor. Successful implementation requires a simultaneous evolution of both theory and practice. The insights and experiences gained from this study have set the stage for further exploration of prospective research and innovation paths, which will be discussed in the subsequent sub-chapter.

## 5.5 Future directions

The course of this study has been enlightening, revealing various complexities and insights within the domain of forecasting backorders by leveraging advanced analytics. Although this research offers a substantial foundation, there remains ample opportunity for further exploration, improvement, and expansion.

Some potential areas for future investigation include:

- **Interdisciplinary data augmentation:** Understanding consumer behavior, which directly impacts backorders, can be improved by an interdisciplinary approach. For example, supplementary datasets capturing consumer sentiments, purchasing behaviors, or market trends, which are derived from Behavioral economics, Psychology, or Sociology, could be merged with the core dataset. This interdisciplinary approach has the potential to offer more insights with richer features for model training. This can lead to improved prediction accuracy and enhance the model's ability to generalize effectively in a wide range of scenarios. Future researchers can explore the possibility of utilizing a multi-disciplinary approach.

- **Diversified methodological approaches:** While this study has produced significant advancements in backorder predictions utilizing advanced analytics, certain methodological choices were impacted by resource constraints. For example, there are models capable of inherently handling missing values, outliers, and class imbalances. The existing literature provides a variety of suggestions and recommendations for these scenarios. To ensure consistency and due to limitations, this study employed a uniform methodology across all four developed models. This decision, though justifiable given the circumstances, offers prospects for further investigation. Exploring diversified methodologies to leverage the unique strengths and mitigate the weaknesses of each model might yield more nuanced insights and potentially improve the predictive accuracy. This is a promising area for future research endeavors.

- **Product-specific feature selection and analysis:** Further to the discussion in "Reflecting on limitations", since product types influence the supply chain dynamics and the likelihood of backorders, future researchers should prioritize a comprehensive analysis to understand how the relative importance of different columns or features in the dataset might shift depending on the product type in question. Recognizing that different products or businesses may have varying priorities in terms of Precision and Recall, a tailored predictive approach for each product category would be beneficial. Along with providing refined predictive accuracy, such an approach also ensures that the analysis is more closely aligned with the unique challenges and intricacies of each product type.

- **Model-centric feature selection:** Different machine learning algorithms interpret and weigh features differently. Hence, an advantageous approach can be adopting a methodology for feature selection that considers the specific type of model being trained. For example, decision tree-based algorithms might benefit from a different subset of features compared to linear models. Future research could experiment with this model-centric feature selection to enhance performance optimization for the corresponding algorithms.

- **Cutting-edge modeling techniques:** Even though, due to being highly computationally demanding, the current study did not examine SVM and NN methods, they still offer promising outcomes, particularly with access to more reliable computational infrastructure. Additionally, the fields of Artificial Intelligence and Machine Learning are constantly evolving. The integration of the emerging algorithms and Deep Learning models has the potential to yield better prediction results for backorder management. Future researchers can explore these areas.

- **Multi-metric evaluation:** Drawing from established theories and the empirical evidence presented in this study, it becomes clear that a multi-metric evaluation approach is essential for highly imbalanced datasets. Further exploration in this direction remains crucial, as only a limited number of studies (including this one) have employed multi-metric evaluation criteria.

In conclusion, the integration of advanced analytics in backorder prediction highlights the dynamic nature of supply chain management. This paper examined the use of predictive analytics for early detection of potential backorders. It also evaluated different machine learning algorithms and emphasized the importance of parameter tuning in optimizing their predictive performance. Additionally, the research also demonstrated the necessity of utilizing multi-metric evaluation criteria for highly imbalanced datasets. Thus, the study has provided a framework for future investigations. Future research is expected to enhance current knowledge and transform the field of backorder predictions with the rapid technological advancements and the continuous evolution of data-driven strategies. However, it is important to consider that technology provides the required tools, but it is the human qualities of curiosity, adaptability, and collaboration that will have a significant impact on the future of backorder management and associated disciplines.

## References


Aguiar, H. (2023). *What Is Imbalanced Data and How to Handle It?* TurinTech. Retrieved September 01, 2023 from: https://www.turintech.ai/what-is-imbalanced-data-and-how-to-handle-it/

Aguilar, F. (2019, October 09). *SMOTE-NC in ML Categorization Models for Imbalanced Datasets.* Medium.com. Retrieved September 01, 2023 from: https://medium.com/analytics-vidhya/smote-nc-in-ml-categorization-models-fo-imbalanced-datasets-8adbdcf08c25

Ahamed, N. (2022, September 08). *Five Major Steps in the Machine Learning Process.* Medium.com. Retrieved July 07, 2023 from: https://medium.com/mlearning-ai/five-major-steps-in-the-machine-learning-process-4b4f1e28806

Allwright, S. (2022, August 09). *Metrics for imbalanced data (simply explained).* Retrieved August 24, 2023 from: https://stephenallwright.com/imbalanced-data-metric/

American Psychological Association/APA. (2009, p. 11). Publication Manual of the American Psychological Association (6th ed.). Washington, DC: American Psychological Association.

Analytics Vidhya (2018, September 06). *Introduction to XGBoost Algorithm in Machine Learning.* Retrieved September 12, 2023 from: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

Azevedo, N. (2023). *Data Preprocessing: 6 Techniques to Clean Data. Scalable Path.* Retrieved July 10, 2023 from: https://www.scalablepath.com/data-science/data-preprocessing-phase

Banoula, M. (2023, February 16). *Machine Learning Steps: A Complete Guide!* Simplilearn. Retrieved July 07, 2023 from: https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

Bates, M. J. (2010, Chapter 2). *Information Behavior In Encyclopedia of Library and Information Sciences*. (3rd ed.). University of California. New York: CRC Press. Retrieved June 22, 2023 from: https://pages.gseis.ucla.edu/faculty/bates/articles/information-behavior.html

Bekkar, M., Djemaa, K. H., & Alitouche, A. T. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications, 3*(10). Retrieved August 24, 2023 from: https://core.ac.uk/download/pdf/234677037.pdf

Bhagat, V. (2022, April 29). *Importance of Data Cleaning.* Topcoder. Retrieved July 13, 2023 from: https://www.topcoder.com/thrive/articles/importance-of-data-cleaning

Bhandari, P. (2023, June 21). *Data Collection: Definition, Methods & Examples.* Scribbr. Retrieved July 12, 2023 from: https://www.scribbr.com/methodology/data-collection/

Biswal, A. (2023, July 28). *What is a Chi-Square Test? Formula, Examples & Application. Simplilearn.* Retrieved August 28, 2023 from: https://www.simplilearn.com/tutorials/statistics-tutorial/chi-square-test

Brown, A. (2022, March 17). *Supply Chain Challenges in 2023 & How to Overcome Them.* Extensiv. Retrieved June 22, 2023 from: https://www.extensiv.com/blog/supply-chain-management/challenges

Brown, S. (2021, May 01). *Machine learning, explained.* MIT Sloan School of Management. Retrieved July 06, 2023 from: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Brownlee, J. (2016, December 23). *How to Define Your Machine Learning Problem.* Machine Learning Mastery. Retrieved July 06, 2023 from: https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Brownlee, J. (2017, July 28). *Why One-Hot Encode Data in Machine Learning?* Machine Learning Mastery. Retrieved August 26, 2023 from: https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Brownlee, J. (2019, October 09). *How to Implement Bayesian Optimization from Scratch in Python.* Machine Learning Mastery. Retrieved September 04, 2023 from: https://machinelearningmastery.com/what-is-bayesian-optimization/

Brownlee, J. (2020, January 08). *Tour of Evaluation Metrics for Imbalanced Classification.* Machine Learning Mastery. Retrieved August 23, 2023 from: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

Burtler, J. (2022, April 25). *What is the Bullwhip Effect?* Study.com. Retrieved June 22, 2023 from: https://study.com/learn/lesson/bullwhip-effect-causes-impacts.html

Carbonneau, R., Vahidov, R., & Laframboise, K. (2007). Machine Learning-Based Demand Forecasting in Supply Chains. *International Journal of Intelligent Information Technologies, 3*(4), 40-57. doi: http://dx.doi.org/10.4018/jiit.2007100103

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research, 184*(3), 1140-1154. doi: https://doi.org/10.1016/j.ejor.2006.12.004

Chen, T. & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* Retrieved September 12, 2023 from: https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf

Chopra, S., & Meindl, P. (2013). *Supply Chain Management: Strategy, Planning, and Operation.* (5th ed.). Pearson. Retrieved July 19, 2023 from: https://base-logistique-services.com/storage/app/media/Chopra_Meindl_SCM.pdf

Codeacademy. (2023). *The Machine Learning Process.* Retrieved July 07, 2023 from: https://www.codecademy.com/article/the-ml-process

Dahiwalkar, S. (2021, June 29). *Backorder Prediction With Machine Learning.* Medium.com. Retrieved June 26, 2023 from: https://shubhamdahiwalkar.medium.com/backorder-prediction-with-machine-learning-e1098a434815

Dalvi, C. (2021, June 15). *Backorder Prediction using Machine Learning.* Medium.com. Retrieved June 26, 2023 from: https://chinmaydalvi.medium.com/backorder-prediction-using-machine-learning-cbe2a7d2cfa4

Datacamp. (2019). *Understanding Logistic Regression in Python Tutorial.* Retrieved September 04, 2023 from: https://www.datacamp.com/tutorial/understanding-logistic-regression-python

Dehghan-Bonari, M., Bakhshi, A., Aghsami, A., & Jolai, F. (2021). Green supply chain management through call option contract and revenue-sharing contract to cope with demand uncertainty. *Cleaner Logistics and Supply Chain, 2,* 100010. doi: https://doi.org/10.1016/j.clscn.2021.100010

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87. doi: 10.1145/2347736.2347755

Durgapal, A. (2023, July 30). *Data Preprocessing — Handling Duplicate Values and Outliers in a dataset.* Medium.com. Retrieved August 22, 2023 from: https://medium.com/@ayushmandurgapal/handling-duplicate-values-and-outliers-in-a-dataset-b00ce130818e

Fathima, F. (n.d.). *Top 7 Supply Chain Management Challenges.* Logistics Brew By Stockarea. Retrieved June 22, 2023 from: https://stockarea.io/blogs/top-7-supply-chain-management-challenges/

Fernando, J. (2023, May 12). *The Correlation Coefficient: What It Is, What It Tells Investors.* Investopedia. Retrieved July 22, 2023 from: https://www.investopedia.com/terms/c/correlationcoefficient.asp

How backorders can impact your business. (n.d.). Four Winds. Retrieved June 22, 2023 from: https://www.fourwinds-ksa.com/how-backorder-can-impact-your-business/

Gao, H., Ren, Q., & Lv, C. (2022). *Supply Chain Management and Backorder Products Prediction Utilizing Neural Network and Naïve Bayes Machine Learning Techniques in Big Data Area: A Real- life Case Study.* doi: https://doi.org/10.21203/rs.3.rs-2020401/v1

Geeksforgeeks. (2023, February 06). *XGBoost.* Retrieved September 12, 2023 from: https://www.geeksforgeeks.org/xgboost/

Georgiev, N. (n.d.). *Bullwhip Effect In Supply Chain: Definition & Example.* Bluecart. Retrieved June 22, 2023 from: https://www.bluecart.com/blog/bullwhip-effect-definition

Guo, A. (2021, September 06). Pearson vs. Spearman Correlation: What's the difference? Medium.com. Retrieved August 22, 2023 from: https://anyi-guo.medium.com/correlation-pearson-vs-spearman-c15e581c12ce

Guo, R., Wang, T., Zhao, J., Zhao, Z., Liu, G,. & Gao, D. (2020). Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. *Applied Sciences, 10*(18):6593. doi: http://dx.doi.org/10.3390/app10186593

Gupta, P. (2017, June 05). *Cross-Validation in Machine Learning.* Medium.com. Retrieved September 04, 2023 from: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

Gupta, A. (2020). *Feature Selection Techniques in Machine Learning.* Analytics Vidhya. Retrieved August 22, 2023 from: https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

Hájek, P., & Abedin, M. (2020). A Profit Function-Maximizing Inventory Backorder Prediction System Using Big Data Analytics. *IEEE Access*. doi: 10.1109/ACCESS.2020.2983118

Heavy.AI. (n.d.). *Feature Selection.* Retrieved August 25, 2023 from: https://www.heavy.ai/technical-glossary/feature-selection#:~:text=Feature%20selection%20improves%20the%20machine,eliminating%20redundant%20and%20irrelevant%20features.

Holicki, R. (2022, May 10). *The Bullwhip Effect: What Is It and What Causes It?* Seeburger. Retrieved June 22, 2023 from: https://blog.seeburger.com/the-bullwhip-effect-what-is-it-and-what-causes-it/

Huovila, E. (2021). *Use of machine learning in supply chain management - case study with datarobot.*[Master's thesis, LUT University]. Retrieved June 22, 2023 from: https://lutpub.lut.fi/bitstream/handle/10024/162367/Masters_Thesis_Huovila_Emmi.pdf?sequence=1

Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice.* (3rd ed.). Melbourne: OTexts. Retrieved July 15, 2023 from: https://otexts.com/fpp3/

IBM. (2021, February 26). *Scientific and Technical Report Backorder Prediction. Defense Logistics Agency.* Retrieved June 22, 2023 from: https://apps.dtic.mil/sti/pdfs/AD1141471.pdf

IBM. (n.d.a). *What is supply chain management?* Retrieved June 22, 2023 from: https://www.ibm.com/topics/supply-chain-management

IBM. (n.d.b). *What is logistic regression?* Retrieved September 04, 2023 from: https://www.ibm.com/topics/logistic-regression

IBM. (n.d.c). *What is random forest?* Retrieved September 10, 2023 from: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

Iguazio (n.d.). *What is the Classification Threshold in Machine Learning?* McKinsey & company. Retrieved September 04, 2023 from: https://www.iguazio.com/glossary/classification-threshold/

Islam, S., & Amin, S. H. (2020). Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques. *Journal of Big Data, 7*(1). doi:10.1186/s40537-020-00345-2

Intellipaat. (2023, June 01). *Bullwhip Effect in Supply Chain.* Retrieved June 22, 2023 from: https://intellipaat.com/blog/bullwhip-effect-in-supply-chain/?US

Jansen, D. & Warren, K. (2020, April). *Research Design 101: Everything You Need To Get Started (With Examples).* Gradcoach. Retrieved July 12, 2023 from: https://gradcoach.com/research-design/

Jansen, D. & Warren, K. (2023, April). *What is research methodology?: A Plain-Language Explanation & Definition (With Examples).* Gradcoach. Retrieved July 12, 2023 from: https://gradcoach.com/what-is-research-methodology/

Kathuria, A. (n.d.). *How can manufacturers improve back-order predictions using machine learning?* Birlasoft. Retrieved June 22, 2023 from: https://www.birlasoft.com/articles/how-can-manufacturers-improve-back-order-predictions-using-machine-learning

Kavlakoglu, E. (2020, May 27). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* IBM. Retrieved July 06, 2023 from: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

Kenton, W. (2022). *Backorder: Definition, Causes, Example, Vs. Out-of-Stock.* Investopedia. Retrieved June 22, 2023 from: https://www.investopedia.com/terms/b/backorder.asp

Key Challenges in Supply Chain Management (And How To Overcome Them). (2022, October 18). *GEP.* Retrieved June 22, 2023 from: https://www.gep.com/blog/technology/supply-chain-management-how-to-overcome-challenges

Krish Naik. (2020a, January 20). *What is Data Leakage In Machine Learning?* [Video]. Retrieved July 22, 2023 from: https://www.youtube.com/watch?v=n9jz7G68pVg&t=375s

Krish Naik. (2020b, October 12). *Tutorial 2- Feature Selection-How To Drop Features Using Pearson Correlation* [Video]. Retrieved July 22, 2023 from: https://www.youtube.com/watch?v=FndwYNcVe0U&t=429s

Krish Naik. (2021, January 26). *Tutorial 3- Feature Selection-How To Select Features Using Information Gain For Classification In ML* [Video]. Retrieved August 22, 2023 from: https://www.youtube.com/watch?v=81JSbXZ26Ls&t=417s

Lans, D. (2019, February 05). 7 Main Challenges In Supply Chain Management And How You Can Workaround It. *Yourstory.* Retrieved June 22, 2023 from: https://yourstory.com/mystory/7-main-challenges-in-supply-chain-management-and-h-rdq2oy6mh9

Lapan, S., Quartaroli M., & Riemer, J. (2012). *Qualitative Research; an Introduction to Methods and Designs.* A Wiley Imprint, John Wiley & Sons.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). *The bullwhip effect in supply chains.* MIT Sloan Management Review. Retrieved June 22, 2023 from: https://sloanreview.mit.edu/article/the-bullwhip-effect-in-supply-chains/

Li, Y. (2017). *Backorder Prediction Using Machine Learning For Danish Craft Beer Breweries* [Master's thesis, Aalborg University]. Retrieved June 22, 2023 from: https://projekter.aau.dk/projekter/files/262657498/master_thesis.pdf

Linkedin. (2023). *What are some best practices for dealing with missing values and imputation methods?* Retrieved July 22, 2023 from: https://www.linkedin.com/advice/0/what-some-best-practices-dealing-missing-values-imputation

Little, R.J.A., & Rubin, D.B. (2002). *Statistical Analysis with Missing Data* (2nd ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Luther, D. (2022). *Backorders Defined: What It Is, Causes & Solutions.* Oracle Netsuite. Retrieved June 22, 2023 from: https://www.netsuite.com/portal/resource/articles/inventory-management/backorder.shtml

Lutkevich, B. (2023). *Supply chain.* Techtarget. Retrieved June 22, 2023 from: https://www.techtarget.com/whatis/definition/supply-chain

Mazumder, S. (2021, June 21). *5 Techniques to Handle Imbalanced Data For a Classification Problem.* Analytics Vidhya. Retrieved September 01, 2020 from: https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/

Manika. (2023, July 15). *Your 101 Guide to Model Selection In Machine Learning.* ProjectPro. Retrieved September 02, 2023 from: https://www.projectpro.io/article/model-selection-in-machine-learning/824

Morana, J. (2013). Sustainable Supply Chain Management. Hoboken, NJ: John Wiley & Sons, Inc.

NIST, National Institute of Standards and Technology. (n.d.). *Detection of Outliers.* Retrieved August 22, 2023 from: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm#:~:text=An%20outlier%20is%20an%20observation,not%20have%20been%20run%20correctly.

Ntakolia, C., Kokkotis, C., Kalsson, P., & Moustakidis, S. (2021). An Explainable Machine Learning Model for Material Backorder Prediction in Inventory Management. *Sensors, 21*(23). doi: https://doi.org/10.3390/s21237926

Oracle. (2023). *What is Machine Learning?* Retrieved July 07, 2023 from: https://www.oracle.com/hk/artificial-intelligence/machine-learning/what-is-machine-learning/

Pandey, P. & Pandey, M. M. (2015) *Research methodology: tools and techniques.* Buzau: Bridge center.

Pandian, S. (2020, December 18). *Understand Machine Learning and Its End-to-End Process.* Analytics Vidha. Retrieved July 07, 2023 from: https://www.analyticsvidhya.com/blog/2020/12/understand-machine-learning-and-its-end-to-end-process/

Patil, N. (2022, June 09). *Feature Scaling.* Medium.com. Retrieved August 29, 2023 from: https://medium.com/@nihal.patil1122/feature-scaling-a513aefda125

Phillips, W. (2022, November 11). *How to avoid the bullwhip effect on inventories.* CIPS. Retrieved June 23, 2023 from: https://www.cips.org/supply-management/news/2022/november/how-to-avoid-the-bullwhip-effect-on-inventories/

Pillai, V. M., & Pamulety, T. C. (2013, June 19-21). *Impact of Backorder on Supply Chain Performance – an Experimental Study.* [International Federation of Automatic Control]. 7th IFAC Conference on Manufacturing Modelling, Management, and Control, Saint Petersburg, Russia. Retrieved June 26, 2023 from: https://www.sciencedirect.com/science/article/pii/S1474667016345785

ProjectPro. (2023, July 10). *Why data preparation is an important part of data science?* Retrieved July 10, 2023 from: https://www.projectpro.io/article/why-data-preparation-is-an-important-part-of-data-science/242

Raja, S. (2021, March 4). *Backorder Prediction: Predicting Backorders using Machine Learning.* Analytics Vidhya. Retrieved June 26, 2023 from: https://medium.com/analytics-vidhya/backorder-prediction-d4f1c5362f18

Rheude (2022, August 08). *Demand Forecasting: Types, Methods, and Examples. Red Stag Fulfillment.* Retrieved July 15, 2023 from: https://redstagfulfillment.com/what-is-demand-forecasting/

Rojon, C., & Saunders, M. N. (2012). Formulating a convincing rationale for a research study. *Coaching An International Journal of Theory Research and Practice 5*(1):1-7. doi: http://dx.doi.org/10.1080/17521882.2011.648335

Roy, B. (2020, April 06). *All about Feature Scaling.* Medium.com. Retrieved August 29, 2023 from: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

Saini, A. (2021a, August 03). *Conceptual Understanding of Logistic Regression for Data Science Beginners.* Analytics Vidhya. Retrieved September 4, 2023 from: https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

Saini, A. (2021b, October 19). *An Introduction to Random Forest Algorithm for beginners.* Analytics Vidhya. Retrieved September 10, 2023 from: https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/

Santis, R., Aguiar, E. P. D., & Goliatt, L. (2017, November). *Predicting material backorders in inventory management using machine learning*. 4th IEEE Latin American Conference on Computational Intelligence, Arequipa, Peru. Retrieved June 26, 2023 from: https://www.researchgate.net/publication/319553365_Predicting_Material_Backorders_in_Inventory_Management_using_Machine_Learning

SAP. (n.d.). *What is SCM (supply chain management) and why is it?* Retrieved June 23, 2023 from: https://www.sap.com/products/scm/what-is-supply-chain-management-scm.html

Saxena, S. (2022, October 03). *Precision-Recall Curve.* Medium.com. Retrieved September 01, 2023 from: https://pub.towardsai.net/precision-recall-curve-26f9e7984add

Scikit-learn Documentation. (n.d.a). *Sklearn.feature_selection.SelectKBest.* Retrieved August 23, 2023 from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#

Shajalal, M., Hájek, P., & Abedin, M. Z. (2021). Product backorder prediction using deep neural network on imbalanced data. *International Journal of Production Research, 61*(1), 302-319. doi: https://doi.org/10.1080/00207543.2021.1901153

Shajalal, M., Boden, A., & Stevens, G. (2022). Explainable product backorder prediction exploiting CNN: Introducing explainable models in businesses. *Electron Markets 32, 2107–2122.* doi: https://doi.org/10.1007/s12525-022-00599-z

Shalev, O. (2018, June 12). *MICE is Nice, but why should you care?* Linkedin. Retrieved September 1, 2023 from: https://www.linkedin.com/pulse/mice-nice-why-should-you-care-ofir-shalev/

Silwal, D. (2022, January 05). *Confusion Matrix, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures.* Linkedin. Retrieved July 10, 2023 from: https://www.linkedin.com/pulse/confusion-matrix-accuracy-precision-recall-f1-score-measures-silwal/

Singh, A., Tharanum, L., Qureshi, M. M., Rachana, & Nivedha, S. (2021). *IJEC, 11*(7). Retrieved June 23, 2023 from: https://ijesc.org/upload/5edac79caaf3aeec6776703b28f473ac.Risk%20Management%20with%20Backorder%20in%20Supply%20Chain%20using%20Machine%20Learning%20Techniques.pdf

Singh, R. (2019, October 08). *Mathematics behind Decision Tree.* Medium.com. Retrieved September 08, 2023 from: https://ranasinghiitkgp.medium.com/mathematics-behind-decision-tree-73ee2ef82164

Singh, Y. (2022, March 22). *Robust Scaling: Why and How to Use It to Handle Outliers.* Proclus Academy. Retrieved August 24, 2023 from: https://proclusacademy.com/blog/robust-scaler-outliers/

Soni, B. (2023, February 27). *MICE or Multivariate Imputation with Chain-Equation.* Medium.com. Retrieved August 26, 2023 from: https://medium.com/@brijesh_soni/topic-9-mice-or-multivariate-imputation-with-chain-equation-f8fd435ca91

Sprague, L. G., Ritzman, L. R., & Krajewski, L. (1990). Production Planning, Inventory Management and Scheduling: Spanning the Boundaries. *Managerial and Decision Economics, 11*(5), 297-315. Retrieved July 18, 2023 from: https://www.jstor.org/stable/pdf/2487592.pdf?refreqid=excelsior%3A901b6bb9024a39b3f8b8bb6ec49d7c95&ab_segments=&origin=&initiator=&acceptTC=1

Stadler, H. & Kilger, C. (2005). *Supply Chain Management and Advanced Planning.* (3rd ed.). Berlin: Springer.

TimesPro. (2023, May 24). *Leveraging Data Analytics to Optimize Supply Chain Performance.* Retrieved June 22, 2023 from: https://timespro.com/blog/leveraging-data-analytics-to-optimize-supply-chain-performance

University of Maryland. (n.d.). *What Is Supply Chain Management, and Why Is It Important?* Retrieved June 22, 2023 from: https://onlinebusiness.umd.edu/mba/resources/what-is-supply-chain-management-and-why-is-it-important/

University of Pretoria. (n.d.). *Chapter 4:Research Design and Methodology.* Retrieved July 12, 2023 from: https://repository.up.ac.za/bitstream/handle/2263/24016/04chapter4.pdf

Walliman, N. (2017). *Research Methods: The Basics.* (2nd ed.).Routledge: Abingdon.

Weedmark, D. (2021, November 02). *Machine Learning Model Training: What It Is and Why It's Important.* Domino. Retrieved September 04, 2023 from: https://domino.ai/blog/what-is-machine-learning-model-training

Weisz, E., Herold, D.M., & Kummer, S. (2022). *Revisiting the bullwhip effect: how can AI smoothen the bullwhip phenomenon?* doi: 10.1108/IJLM-02-2022-0078

Western Governors University. (2023, April 06). *What Is the Difference Between a Primary and Secondary Source?* Retrieved July 12, 2023 from: https://www.wgu.edu/blog/what-difference-between-primary-secondary-source2304.html#close

Wilson, S. (2021, September 06). *The MICE Algorithm.* Retrieved September 1, 2023 from: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Wisneski, C. (2022, December 05). *7 Reasons Why Machine Learning Forecasting Is Better Than Traditional Methods.* Akkio. Retrieved July 14, 2023 from: https://www.akkio.com/post/5-reasons-why-machine-learning-forecasting-is-better-than-traditional-methods

WorldStreetMojo. 2023. *Bullwhip Effect.* Retrieved June 22, 2023 from: https://www.wallstreetmojo.com/bullwhip-effect/

World Economic Forum. (2022, September 7). *5 challenges facing global supply chains.* Retrieved June 22, 2023 from: https://www.weforum.org/agenda/2022/09/5-challenges-global-supply-chains-trade

Zhang, A. (2017). *Data Analytics: Practical Guide to Leveraging the Power of Algorithms, Data Science, Data Mining, Statistics, Big Data, and Predictive Analysis to Improve Business, Work, and Life.* (ISBN-10: 1544603975; ISBN-13 978-1544603971)

## List of Figures

FIGURE 01: AN EXAMPLE OF BULLWHIP EFFECT (LI, 2017, P. 4) ..... 12
FIGURE 02: RELATIONSHIP BETWEEN AI, ML, NN, AND DL (KAVLAKOGLU, 2020, PARA. 3) ..... 15
FIGURE 03: ML PROCESS AT A GLANCE (PANDIAN, 2020, PARA. 6) ..... 17
FIGURE 04: SUMMARY STATISTICS ..... 44
FIGURE 05: MISSING VALUES IN "LEAD_TIME" AND PERFORMANCE COLUMNS ..... 45
FIGURE 06: EXAMINING THE RANDOMNESS OF THE MISSING VALUES ..... 46
FIGURE 07: MICE PROCESS (SHALEV, 2018) ..... 47
FIGURE 08: BOXPLOTS OF THE NUMERICAL COLUMNS ..... 48
FIGURE 09: HISTOGRAM OF THE NUMERICAL COLUMNS ..... 49
FIGURE 10: UNIQUE VALUES OF THE CATEGORICAL FEATURES ..... 51
FIGURE 11: BIVARIATE ANALYSIS OF CATEGORICAL FEATURES WITH THE TARGET COLUMN ..... 53
FIGURE 12: BOX PLOTS FOR BIVARIATE ANALYSIS OF NUMERICAL FEATURES WITH THE TARGET COLUMN ..... 55
FIGURE 13: DETAILED BIVARIATE ANALYSIS BY BINNING THE NUMERICAL VALUES OF THE COLUMNS ..... 56
FIGURE 14: PEARSON CORRELATION MATRIX ..... 58
FIGURE 15: SPEARMAN CORRELATION MATRIX ..... 59
FIGURE 16: SELECTKBEST FEATURE RANKING USING MI SCORE ..... 61
FIGURE 17: UNDER-SAMPLING AND OVERSAMPLING (AGUIAR, 2023, PARA. 12) ..... 64
FIGURE 18: CLASS DISTRIBUTION BEFORE AND AFTER RESAMPLING ..... 65
FIGURE 19: UNDERSTANDING LOGISTIC REGRESSION (DATACAMP, 2019, PARA. 10) ..... 70
FIGURE 20: DECISION TREES (SINGH, 2018, PARA. 1) ..... 73
FIGURE 21: SIMPLE RANDOM FOREST CLASSIFIER (SAINI, 2021B, APPLYING DT IN RF ALGORITHM) ..... 77
FIGURE 22: FLOW CHART OF XGBOOST (GUO ET AL., 2020, P. 06) ..... 80
FIGURE 23: ROC/AUC AND PR CURVES FOR THE MODELS WITH ORIGINAL SAMPLING STRATEGY ..... 95
FIGURE 24: ROC/AUC AND PR CURVES FOR THE MODELS WITH RUS SAMPLING STRATEGY ..... 96
FIGURE 25: ROC/AUC AND PR CURVES FOR THE MODELS WITH SMOTENC SAMPLING STRATEGY ..... 97

## List of Tables

TABLE 01: CROWDFLOWER SURVEY ON DATA SCIENTISTS' TIME ALLOCATION ..... 20
TABLE 02: PERFORMANCE COMPARISON AMONG KNOWN RELATED WORKS ON SIMILAR DATASET ..... 32
TABLE 03: A QUICK OVERVIEW OF THE DATASET ..... 40
TABLE 04: PERCENTILE TEST RESULT OF THE NUMERICAL COLUMNS ..... 50
TABLE 05: HYPERPARAMETER TUNING OF LOGISTIC REGRESSION ..... 72
TABLE 06: HYPERPARAMETER TUNING OF DECISION TREES ..... 76
TABLE 07: HYPERPARAMETER TUNING OF RANDOM FORESTS ..... 79
TABLE 08: HYPERPARAMETER TUNING OF XGBOOST ..... 83
TABLE 09: CONFUSION MATRIX FOR TWO CLASSES CLASSIFICATION (BEKKAR ET AL., 2020, P. 27) ..... 84
TABLE 10: SOME METRICS FOR TWO CLASSES CLASSIFICATION PROBLEMS ..... 85
TABLE 11: BASE MODELS' PERFORMANCE EVALUATION ..... 87
TABLE 12: HYPERPARAMETER TUNED MODELS' PERFORMANCE EVALUATION ..... 90
TABLE 13: JUXTAPOSITION OF BASE MODEL'S AND TUNED MODEL'S PERFORMANCE ..... 93
TABLE 14: BEST OF CURRENT STUDY VS. BEST OF STATE-OF-THE-ART ..... 99

## List of Appendices

Appendix 01: Handling loaded dummy values and missing values ..... 131
Appendix 02: Outlier detection and handling ..... 132
Appendix 03: Binarization of the categorical features ..... 133
Appendix 04: Cardinality checking and dropping the 'sku' column ..... 134
Appendix 05: Bivariate analysis of categorical features with the target column ..... 135
Appendix 06: Chi-squared test ..... 136
Appendix 07: Bivariate analysis of numerical features with the target column ..... 137
Appendix 08: Correlation matrix ..... 138
Appendix 09: SelectKbest with Mutual Information / MI Score ..... 139
Appendix 10: Summary of all observations / Finalizing the features for ML model ..... 140
Appendix 11: Handling duplicates and resetting indices ..... 141
Appendix 12: Scaling the dataset ..... 142
Appendix 13: Handling imbalanced training set / Resampling training set ..... 143

***Appendix 01: Handling loaded dummy values and missing values***

a. Replacing loaded dummy values (-99) with Nan values:

```
### Replacing -99 by Nan values in the performance columns

train_df["perf_6_month_avg"].replace({-99.0 : np.nan},inplace= True)
train_df["perf_12_month_avg"].replace({-99.0 : np.nan},inplace= True)
```

b. Deleting rows with null values from columns that have only one missing entry:

```
### Dropping the Null values of the columns that have only 01 null value each

for col in train_df.columns:
    if train_df[col].isnull().sum() == 1:
        train_df = train_df.dropna(subset=[col], how='any')
```

c. Filling the missing values of the "lead_time" and performance columns:

```
### Filling the missing values with Iterative Imputation/MICE approach

## Creating the imputer object
imputer = IterativeImputer(max_iter= 10, random_state= 0)

## Fitting the imputer on the training data
imputer.fit(train_df[col_with_missing])
```

```
IterativeImputer(random_state=0)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
imputed_train_data = imputer.transform(train_df[col_with_missing])  ## Transforming the training data with the imputer

## Creating a new DataFrame with the imputed training data
imputed_train_df = pd.DataFrame(imputed_train_data, columns= train_df[col_with_missing].columns)

## Inserting the imputed numerical columns with the rest of the DataFrame
train_df[col_with_missing] = imputed_train_df
```

*Appendix 02: Outlier detection and handling*

a. Examining the percentile- an example with “local_bo_qty” column:

```
### 15.'local_bo_qty'

percentiles = [i for i in range(0, 110, 10)]

print("Percentiles of values:")
print("=" * 50)

for p in percentiles:
    print(f"{p} percentile of values is {train_df.local_bo_qty.quantile(p * 0.01)}")

print("=" * 100)
print("Between 90-100 percentile:")
print("=" * 50)

for p in range(90, 101):
    print(f"{p} percentile of values is {train_df.local_bo_qty.quantile(p * 0.01)}")

print("=" * 100)
print("Between 0-10 percentile:")
print("=" * 50)

for p in range(0, 10):
    print(f"{p} percentile of value is {train_df.local_bo_qty.quantile(p * 0.01)}")
```

b. Removing outliers- top 1% of values (0.99 quantile):

```
### Removing outliers: top 1% of values (0.99 quantile)

def goons(train_df):       ## Funcion for removing outliers

    for col in train_num.columns:
        outliers = train_num[col].quantile(0.99)
        train_df = train_df[(train_df[col] >= 0) & (train_df[col] <= outliers)]

    return train_df

train_df = goons(train_df)
```

*Appendix 03: Binarization of the categorical features*

a. Transforming categorical values of the target column:

```
### Replacing the values in the target column with Binary number-

## We replace all the "No" values by "0", and "Yes" values by "1" (binary format)

train_df["went_on_backorder"].replace({'No': 0, 'Yes': 1}, inplace=True)
train_df["went_on_backorder"].astype(int)
train_df["went_on_backorder"][:3]
```

```
0    0
1    0
2    0
```

b. Transforming categorical values of rest of the columns:

```
### Replacing the values of other categorical columns with Binary number-

## We replace all the "No" values by "0", and "Yes" values by "1" (binary format)

for col in otherTrain_cat:
    train_df[col].replace({'No': 0, 'Yes': 1}, inplace= True)
    train_df[col]= train_df[col].astype(int)

train_df.sample(3)
```

| min_bank | potential_issue | pieces_past_due | perf_6_month_avg | perf_12_month_avg | local_bo_qty | deck_risk | oe_constraint | ppap_risk | stop_auto_buy | rev_stop | v |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 62.0 | 0 | 0.0 | 1.00 | 0.99 | 0.0 | 0 | 0 | 0 | 1 | 0 | |
| 0.0 | 0 | 0.0 | 0.94 | 0.83 | 0.0 | 0 | 0 | 0 | 1 | 0 | |
| 1.0 | 0 | 0.0 | 0.93 | 0.95 | 0.0 | 0 | 0 | 0 | 1 | 0 | |

## *Appendix 04: Cardinality checking and dropping the 'sku' column*

a. Checking unique values of each column

```
### Checking unique values

train_df.nunique()
```

```
sku                  1687861
national_inv           14969
lead_time                 32
in_transit_qty          5230
forecast_3_month        7825
forecast_6_month       11114
forecast_9_month       13662
sales_1_month           5764
sales_3_month          10495
sales_6_month          14818
sales_9_month          18341
min_bank                5568
potential_issue            2
pieces_past_due          826
perf_6_month_avg         101
perf_12_month_avg        101
local_bo_qty             654
deck_risk                  2
oe_constraint              2
ppap_risk                  2
stop_auto_buy              2
rev_stop                   2
went_on_backorder          2
dtype: int64
```

**Observations:**

We can see here that all the values in 'sku' column are unique as the number of values is equal to the number of rows. Therefore, this column can be deleted.

b. Dropping the 'sku' column

```
## Dropping non-effective variable

train_df= train_df.drop(['sku'], axis = 1)
train_df.sample(3)
```

| | national_inv | lead_time | in_transit_qty | forecast_3_month | forecast_6_month | forecast_9_month | sales_1_month | sales_3_month | sales_6_month | sales_9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **126742** | 2.0 | 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | |
| **1208832** | 57.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **1353671** | 19.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

*Appendix 05: Bivariate analysis of categorical features with the target column*

```
### We will check each of our categorical column in relation with the Target column

def value_counts_and_plot(column):

    groupby_table = train_cat.groupby([column, 'went_on_backorder']).size().unstack()
    total_counts = groupby_table.sum(axis= 1)
    percentage_table = groupby_table.div(total_counts, axis= 0) * 100
    percentage_table = percentage_table.round(2)

    ## Concatenating count and percentage tables to show side by side
    result_table = pd.concat([groupby_table, percentage_table], axis= 1, keys= [('(Counts)'), ('       (%)')])

    print(f"Analysis of {column} in relation to 'went_on_backorder':")
    print("-" * 40)
    print(result_table)
    print("\n")

    fig, ax = plt.subplots(figsize=(8, 6))
    percentage_table.plot(kind='bar', ax= ax)

    ax.set_xlabel(f"{column}")
    ax.set_xticklabels(['No', 'Yes'], rotation= 45)

    ax.set_ylabel("Value percentage")
    ax.set_title(f"Bar plot of {column} in relation to 'went_on_backorder'")
    ax.legend(title='went_on_backorder', loc= 'upper center')

    plt.show()

    print("=" * 100)

for column in otherTrain_cat:
    value_counts_and_plot(column)
```

## *Appendix 06: Chi-squared test*

```
### We first determine our Null hoypothesis and Alternate hypothesis.

  # Our Null hypothesis, H0 = There is no association between the categorical variable and the target column
  # And the Alternate hypothesis, H1 = There is a significant relationship between them.

## We will use p-value to test our hypothesis. If the p-value > significance factor, we will fail to reject the H0.
 # Otherwise, we will reject it and automatically the H1 will be accepted.

## Signficance value is expressed as 'alpha', of which usual value is: 0.05


## Creating a contingency table and performing chi-square test for each categorical column

for column in otherTrain_cat.columns:

    contingency_table = pd.crosstab(train_cat[column], train_cat['went_on_backorder'])
    chi2, p, dof, expected = chi2_contingency(contingency_table)
    alpha= 0.05

    print(f"Chi-squared test for {column} vs. went_on_backorder:")
    print(f"P-value: {p}")

    if p<= alpha:
        print("These 02 have a significant relationship. We Reject the Null hypothesis")

    else:
        print("No association observed. We FAIL to reject the Null hypothesis")

    print("=" * 50)
```

```
Chi-squared test for potential_issue vs. went_on_backorder:
P-value: 1.6068136312342238e-46
These 02 have a significant relationship. We Reject the Null hypothesis
==================================================
Chi-squared test for deck_risk vs. went_on_backorder:
P-value: 5.921252566748727e-11
These 02 have a significant relationship. We Reject the Null hypothesis
==================================================
Chi-squared test for oe_constraint vs. went_on_backorder:
P-value: 0.0045004298543147355
These 02 have a significant relationship. We Reject the Null hypothesis
==================================================
Chi-squared test for ppap_risk vs. went_on_backorder:
P-value: 4.742846422211572e-16
These 02 have a significant relationship. We Reject the Null hypothesis
==================================================
Chi-squared test for stop_auto_buy vs. went_on_backorder:
P-value: 0.6486681017160183
No association observed. We FAIL to reject the Null hypothesis
==================================================
Chi-squared test for rev_stop vs. went_on_backorder:
P-value: 0.10736072631196185
No association observed. We FAIL to reject the Null hypothesis
==================================================
```

## *Appendix 07: Bivariate analysis of numerical features with the target column*

a. Box plots for bivariate analysis of numerical features with the target column:

```
### We will now visually inspect the relationship between the numerical columns and the target variable

## Creating plots for each numerical column with respect to the target column

plt.figure(figsize=(16, 12))

for i, column in enumerate(train_num, 1):
    plt.subplot(4, 4, i)
    sns.boxplot(data= train_df, x='went_on_backorder', y= column)
    plt.title(f'{column} vs. went_on_backorder', fontsize= 10)
    plt.xlabel('')

plt.tight_layout()
plt.show()
```

b. Detailed bivariate analysis by binning the numerical values of the columns:

```
### We will conduct a detailed analysis of the relationship between the numerical columns and the target variable

analysis_dict = {}  ## Defining the dictionary

## Iterating through each numeric column
for col in train_num:

    ## Binning the column
    bins = pd.cut(train_df[col], bins=[0, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50], include_lowest= True)

    ## Counting occurrences of 0s and 1s in each bin
    count_0s = train_df.loc[train_df['went_on_backorder'] == "No", col].groupby(bins).count()
    count_1s = train_df.loc[train_df['went_on_backorder'] == "Yes", col].groupby(bins).count()

    ## Creating separate DataFrames for the analysis of the current column
    col_analysis_df = pd.DataFrame({'Bin': count_0s.index, 'Total_No': count_0s.values, 'Total_Yes': count_1s.values})

    ## Adding the current analysis DataFrame to the dictionary with the column name as key
    analysis_dict[col] = col_analysis_df

## Printing the analysis DataFrames
for col, analysis_df in analysis_dict.items():
    print(f"Analysis for Column: {col}")
    print(analysis_df)
    print("=" * 50)
```

## *Appendix 08: Correlation matrix*

a. Pearson correlation test

```
plt.figure(figsize=(14, 8))

corr = train_df.corr()  ## Defining Pearson Correaltion matrix

## We assume that, Correlation Coefficient values greater than (- 0.6) or less than (+ 0.6) are not that significant.
sns.heatmap(corr[(corr <= -0.6) | (corr >= 0.6)], annot = True, cmap = 'Reds')
```

b. Spearman correlation test

```
plt.figure(figsize= (16, 10))

spearman_corr = train_df.corr(method='spearman') ## Defining Spearman Correlation matrix

## Here we are trying very small threshold value to see any possible correlation
sns.heatmap(spearman_corr[(spearman_corr <= -0.01) | (spearman_corr >= 0.01)], annot = True, cmap = 'Blues')
```

## *Appendix 09: SelectKbest with Mutual Information / MI Score*

```
### Splitting the dataset: We will first split the dataset into X_train and y_train variables for furthe analysis

X_train= train_df.drop(["went_on_backorder"], axis= 1)
y_train= train_df["went_on_backorder"]

X_train.shape, y_train.shape
```

```
((975329, 21), (975329,))
```

```
### SelectKBest with MI Score
  ## We will now use the Mutual Information as the scoring function within the SelectKBest method

## Initializing the SelectKBest method
k_best= SelectKBest(score_func= mutual_info_classif, k= "all")  ## We'll calculate scores for all features first

k_best.fit(X_train, y_train) ## Fitting on the training data

## Getting the scores
k_best_score = k_best.scores_

## Mapping these scores back to the feature names and creating a dataframe
k_best_df = pd.DataFrame({'Feature': X_train.columns, 'k_best_score': k_best_score})

k_best_df= k_best_df.sort_values(by='k_best_score', ascending= False) ## Sorting in descending order
k_best_df['k_best_formatted_score'] = k_best_df['k_best_score'].apply(lambda x: f"{x:.4f}")

print(list(zip(k_best_df['Feature'], k_best_df['k_best_formatted_score'])))
print()

k_best_df.set_index('Feature', inplace= True) ## Setting the Features as index

k_best_df.plot.bar(figsize= (15, 6))  ## Plotting the scores
```

## *Appendix 10: Summary of all observations / Finalizing the features for ML model*

### Summary of all observations / Selecting Features for ML models:

All our observations suggest we got somehow a mixed result. However we will follow a balancing strategy while selecting the coulnms for our ML models. If a single test is showing the importance of a certain variable but some other tests show no importance, we will still keep it. If multiple tests show a certain variable is less important then we will simply drop it. If a single test shows no importance of a variable and other tests also did not provide strong evidence for its inclusion, we will also drop it. On top of everything, we will try to utilize our professional judgement.

Hence, based on this strategy we opt for the followings-

- From the Performance columns, Forecast columns and Sales columns, we decided to retain the longest window for each group, as we want to capture broader trends.
- We will also remove the columns that had very low MI scores as we also saw they were not that signinificant during our correlation tests. There will be an exception to the 'min_bank' based on the domain knowledge.

```
### Dropping the less effective columns

less_effective_col= ['in_transit_qty', 'pieces_past_due', 'local_bo_qty', 'oe_constraint',
                     'potential_issue', 'rev_stop', 'forecast_3_month', 'forecast_6_month',
                     'sales_1_month', 'sales_3_month', 'sales_6_month', 'perf_6_month_avg']

train_df = train_df.drop(less_effective_col, axis = 1)
train_df.shape
```

```
(975329, 10)
```

## *Appendix 11: Handling duplicates and resetting indices*

```
### Removing duplicate rows (if any) from the dataset

train_df = train_df.drop_duplicates(keep = 'first')
train_df.shape
```

```
(803402, 10)
```

```
### Resetting index to maintain the serial

train_df.reset_index(drop= True, inplace= True)
train_df.tail(3)
```

| | national_inv | lead_time | forecast_9_month | sales_9_month | min_bank | perf_12_month_avg | deck_risk | ppap_risk | stop_auto_buy | went_on_backorder |
|---|---|---|---|---|---|---|---|---|---|---|
| 803399 | 488.0 | 2.0 | 2160.0 | 1733.0 | 189.0 | 0.96 | 0 | 0 | 1 | 0 |
| 803400 | 0.0 | 2.0 | 3412.0 | 765.0 | 657.0 | 0.99 | 0 | 0 | 0 | 1 |
| 803401 | 124.0 | 8.0 | 1240.0 | 1074.0 | 111.0 | 0.90 | 0 | 0 | 1 | 0 |

## *Appendix 12: Scaling the dataset*

```
### RobustScaler

scaler= RobustScaler()  ## Variable for RobustScaler

data_fit= scaler.fit(df_num) ## Fitting the scaler to the data
```

```
data_scaled= scaler.transform(df_num)   ## # Transforming the data using the scaler

df_scaled= pd.DataFrame(data_scaled, columns= df_num.columns)  ## Making the output array into the DataFrame
df_scaled.tail(3)
```

| | national_inv | lead_time | forecast_9_month | sales_9_month | min_bank | perf_12_month_avg |
|---|---|---|---|---|---|---|
| 803399 | 2.851613 | -1.5 | 17.322581 | 9.469274 | 7.192308 | 0.434783 |
| 803400 | -0.296774 | -1.5 | 27.419355 | 4.061453 | 25.192308 | 0.565217 |
| 803401 | 0.503226 | 0.0 | 9.903226 | 5.787709 | 4.192308 | 0.173913 |

```
### Now this new df contains all the selected features from our original dataset including the scaled data

newTrain_df= pd.concat([df_scaled, df_cat], axis= 1)
newTrain_df.tail(3)
```

| | national_inv | lead_time | forecast_9_month | sales_9_month | min_bank | perf_12_month_avg | deck_risk | ppap_risk | stop_auto_buy | went_on_backorder |
|---|---|---|---|---|---|---|---|---|---|---|
| 803399 | 2.851613 | -1.5 | 17.322581 | 9.469274 | 7.192308 | 0.434783 | 0 | 0 | 1 | 0 |
| 803400 | -0.296774 | -1.5 | 27.419355 | 4.061453 | 25.192308 | 0.565217 | 0 | 0 | 0 | 1 |
| 803401 | 0.503226 | 0.0 | 9.903226 | 5.787709 | 4.192308 | 0.173913 | 0 | 0 | 1 | 0 |

*Appendix 13: Handling imbalanced training set / Resampling training set*

a. Original dataset:

```
### Checking the value distribution in the target column

Value_Count= y_train.value_counts()

## Transforming into percentage form
Value_Percentage= y_train.value_counts(normalize= True).mul(100).round(2).astype(str) + '%'

## Combining the Value_Counts and percentages into a DataFrame
Value_Count_df = pd.DataFrame({'Value_Count': Value_Count, '   Value_Percentage': Value_Percentage})

print(Value_Count_df)

ax= y_train.value_counts().plot.pie(autopct= '%.2f')
_ax= ax.set_title('Original dataset')
```

b. RUS / Random under-sampling:

```
### Random Undersampling Technique

## Creating a variable for the RandomUnderSampler
rus= RandomUnderSampler(sampling_strategy= 0.1, random_state= 0)

## Providing our original data to tranform
X_train_rus, y_train_rus = rus.fit_resample(X_train, y_train)

print("New class distribution after Undersampling:\n")
print(y_train_rus.value_counts())
print()

ax= y_train_rus.value_counts().plot.pie(autopct= '%.2f')
_ax= ax.set_title('Random Undersampling')
```

c. SMOTENC:

```
### Random Undersampling Technique

## Creating a variable for the RandomUnderSampler
rus= RandomUnderSampler(sampling_strategy= 0.1, random_state= 0)

## Providing our original data to tranform
X_train_rus, y_train_rus = rus.fit_resample(X_train, y_train)

print("New class distribution after Undersampling:\n")
print(y_train_rus.value_counts())
print()

ax= y_train_rus.value_counts().plot.pie(autopct= '%.2f')
_ax= ax.set_title('Random Undersampling')
```