YARN SUPPLY CHAIN MANAGEMENT
AT
Deepak Spinners Limited

A PROJECT REPORT SUBMITTED TO
DEEN DAYAL UPADHYAYA COLLEGE, UNIVERSITY OF DELHI IN
PARTIAL FULFILLMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE IN MATHEMATICAL SCIENCES

By

Disha Pattnaik

(20015587009)

(20MTS5708)

Under the supervision of

Dr. Veena Jain

(Associate Professor)

DEPARTMENT OF OPERATIONAL RESEARCH

DEEN DAYAL UPADHYAYA COLLEGE

UNIVERSITY OF DELHI

NEW DELHI - 110078

MAY 2023

DECLARATION

I hereby declare that the project report entitled "Supply Chain Management at Deepak Spinners Limited" is being submitted to the Department of Operational Research, Deen Dayal Upadhyaya College, University of Delhi 110078 in partial fulfilment of the requirement for the award of the degree of Bachelor of Science in Mathematical Sciences. I have prepared this report under the guidance and supervision of Dr. Veena Jain, Associate Professor, Deen Dayal Upadhyaya College, University of Delhi, New Delhi-110078

The project is based on my original research and has not been submitted for the award of any degree or diploma from any other university or institution to the best of my knowledge.

Place: New Delhi **Disha Pattnaik**

Date: 22/05/2023 20MTS5708

CERTIFICATE

I hereby certify that the Project titled " Supply Chain Management at Deepak Spinners Limited " which has been submitted by Disha Pattnaik (20MTS5708), for the fulfilment of the requirement for awarding the degree of Bachelor of Science (B.Sc.) Mathematical Sciences is a record of the project work carried out by the student under my guidance and supervision. To the best of my knowledge, this work has not been submitted in any part or fulfilment for any Degree or Diploma to this University or elsewhere.

Place: New Delhi

Date: 22/5/2023

Dr. Veena Jain

(Supervisor)

Associate Professor

Department of Operational Research

Deen Dayal Upadhyaya College

University of Delhi

ACKNOWLEDGEMENT

I would like to express my heartfelt thanks to Dr. Veena Jain for providing me the valuable guidance, support, and encouragement throughout the project. Herexpertise and knowledge have been instrumental in shaping the project's direction and helping me to achieve the desired outcome.

I would also like to extend my gratitude to Mr. Rohit, Sales Head, DeepakSpinners Limited who helped me throughout the project. His insights andguidance were essential in the successful completion of the project.

Finally, I would like to express my appreciation to my family and friends fortheir unwavering support and encouragement. Their constant support, motivation, and understanding have been instrumental in helping me to complete this project successfully.

Disha Pattnaik
B.Sc. Mathematical Science
Deen Dayal Upadhyaya College
University of Delhi
NEW DELHI-110078
College Roll no.- 20MTS5708
Examination Roll no.- 20015587009

ABSTRACT

The transportation and logistics industry plays a vital role in the textile industry. Effective supply chain management is being crucial to the success of any business. This project is aimed to solve a transportation problem for a company and to find future sales of the product. SARIMA method has been usedto carry out forecasting.

The transportation problem was addressed by optimizing the allocation of goods from two warehouses to various retail stores. Sensitivity analysis wascarried out to identify the optimal solution to the transportation problem byvarying the constraints and analyzing the impact on the objective function.

Then, Historical sales data was analyzed, and the SARIMA model was used topredict future sales trends. This allowed the company to make informed decisions about production levels, inventory management, and marketing strategies.

The transportation optimization and sales forecasting models were integratedto develop a comprehensive supply chain management system. The system provides valuable insights into future sales trends and optimizes the allocationof goods from warehouses to retail stores.

Overall, the project aimed to address the challenges associated with transportation and supply chain management in the logistics industry, and to provide the company with the tools necessary to optimize profitability and staycompetitive in the market.

Index

Title	Page No.

List of Tables

S. No	Table	Page No.

List of Figures

S. No	Figure	Page No.

Chapter - 1 Introduction to Operations Research

1.1 Introduction

Operations research, abbreviated as OR, is a discipline concerned with the development and implementation of advanced analytical tools for better decision-making. It is sometimes considered a branch of the mathematical sciences.

Operations research finds optimal or near-optimal solutions to complicated decision-making problems by combining tools from other quantitative sciences including as modelling, statistics, and optimization. Operations research interacts with many other fields, including industrial engineering, due to its emphasis on practical applications. Operations research is frequently focused with determining the maximum (of profit, performance, or yield) or minimum values of some real-world objective (of loss, risk, or cost). Its methods have grown to address problems in a variety of industries, having originated in military activities before World War II.

1.2 History

McClosky and Trefthen coined the term "Operational Research" in 1940 in a small town in the United Kingdom called Bowdsey. This new science came as a result of research into military operations during WWII. There were strategic and tactical problems that were so complex that expecting adequate solutions from individuals or specialists was unrealistic. As a result, military leaders convened scientists from various disciplines and organised them into teams to help solve strategic and tactical problems, i.e., to discuss, evolve, and suggest ways and means to improve the execution of various military projects. They proposed certain approaches that demonstrated remarkable progress as a result of their collaborative efforts, experience, and deliberations. This new method of studying the system's operations in a systematic and scientific manner was known as Operations Research or Operational Research (abbreviated as O.R.).

The success of military teams drew the attention of industrial managers looking for solutions to their problems immediately after the war. Industrial operation research developed along different lines in the United Kingdom and the United States of America. The critical economic situation in the United Kingdom necessitated a drastic increase in production efficiency as well as the creation of new markets. The nationalisation of a few key industries expanded the potential field for OR. As a result, OR quickly spread from military to government, industrial, social, and economic planning.

India was one of the first countries to use O.R. The first operational research unit, known as the Regional Research Laboratory, was established in Hyderabad in 1949. Simultaneously, an additional unit was established in the Defence Science Laboratory to address the Stores, Purchase, and Planning issues.

An O.R. unit was established in the Indian Statistical Institute in Calcutta in 1953. The goal was to employ O.R. techniques in National Planning and Survey. The Operations Research Society of India was founded in 1955 and was one of the first members of the International Federation of Operations Research Societies. Today, the use of O.R. techniques has spread from the army to a diverse range of departments at all levels.

1.3 Scientific Method in Operations Research

The scientific method is the most important feature of Operations Research. It is divided into three phases:

1.3.1 Judgement Phase: This phase consists of the following steps:

- identifying the real-life problem,
- selecting an appropriate goal and the values of various variables related to the goals,
- selecting an appropriate scale of measurement, and
- formulating an appropriate model of the problem, abstracting the essential information so that a solution at the decision-goal makers can be sought.

1.3.2 Research Phase: This is the largest and longest of the two phases. However, the remaining two are equally important because they serve as the foundation for a scientific method. This phase includes:

- observations and data collection to gain a better understanding of the problem,
- hypothesis and model formulation,
- observation and experimentation to test the hypothesis with additional data,
- analysis of the available information and hypothesis verification using pre-established measures of effectiveness,
- predictions of various results from the hypothesis, and
- generalization of the results and conjectures.

1.3.3 Action Phase: This phase consists of making recommendations for the decision-making process by those who first posed the problem for consideration, or by anyone in a position to make a decision affecting the operation in which the problem occurred.

1.4 Advantage of Operations Research

1.4.1 Better Systems: Often, an O.R. approach is initiated to analyse a specific decision-making problem, such as the best location for factories, whether to open a new warehouse, and so on. It also aids in the selection of cost-effective modes of transportation, job sequencing, production scheduling, and the replacement of old machinery, among other things

1.4.2 Better Control: The management of large organisations recognises that providing continuous executive supervision to every routine task is a difficult and costly endeavour. An O.R. approach may provide an analytical and quantitative foundation for the executive to identify the problem area. The most commonly used applications in this category are those for production scheduling and inventory replenishment.

1.4.3 Better Decisions: O.R. models aid in decision making and reduce the possibility of making incorrect decisions. The O.R. approach provides the executive with a better understanding of how he makes decisions.

1.4.4 Better Coordination: An operations-research-oriented planning model aids in the coordination of a company's various divisions.

1.5 Limitations of Operations Research

1.5.1 Reliance on an electronic computer: O.R. techniques attempt to find an optimal solution while taking all factors into account. These factors are enormous in modern society, and quantifying them and establishing relationships between them necessitate voluminous calculations that can only be handled by computers.

1.5.2 Non-Quantifiable Factors: Only when all of the elements related to a problem can be quantified can O.R. techniques provide a solution. Quantification is not possible for all relevant variables. Unquantifiable factors have no place in O.R. models.

1.5.3 Money and Time Expenses: When the basic data is subject to frequent changes, incorporating it into the O.R. models is an expensive endeavour. Furthermore, a reasonably good solution now may be preferable to a perfect O.R. solution later on.

1.5.4 Implementation: Decision implementation is a delicate task. It must account for the complexities of human relationships and behaviour.

1.6 Applications of Operations Research

Aside from scientific advancement, O.R. is primarily concerned with the techniques of applying scientific knowledge. It provides an understanding that provides the expert/manager with new insights and capabilities to determine better solutions to his decision-making problems quickly, competently, and confidently. O.R. has successfully entered many different areas of research in recent years, including Defence, Government, Service Organizations, and Industry. We briefly describe some O.R. applications in management functional areas:

1.6.1 Project Allocation and Distribution

- Optimal project resource allocation, including men, materials, machines, time, and money.
- Identifying and deploying the appropriate workforce.
- Project planning, management, and control.

1.6.2 Production and Facility Planning

- Choosing the size and location of the factory.
- Estimate the number of facilities needed.
- Forecasting for various inventory items, as well as calculating economic order quantities and reorder levels.
- Scheduling and sequencing of production runs through proper machine allocation.
- Transportation loading and unloading; and

- Warehouse location determination.
- Decisions on maintenance policy.

1.6.3 Program Decisions

- What, when, and how to buy to reduce procurement costs.
- Policies for bidding and replacement.

1.6.4 Marketing

- The timing of product introduction.
- Advertising media selection.
- Product mix selection
- Customer preferences for product size, colour, and packaging.

1.6.5 Organizational Behaviour

- Personnel Selection, Determination of Retirement Age and Skills
- Job assignment and recruitment policies.
- Employee recruitment.
- Training programme scheduling

1.6.6 Finance

- Capital needs, cash flow analysis
- Credit policies, credit risks, and so on.
- Investment choice.
- The company's profit plan.

1.6.7 Research and Development

- Product launch planning
- Oversight of R&D projects.
- Identification of research and development priorities.
- Project selection and budget preparation.
- Development project dependability and control as a result, it is possible to conclude that operation research can be widely used in management decisions and can also be used as a corrective measure.

1.7 Some Operation Research Techniques

1.7.1 Linear Programming: Linear Programming (LP) is a mathematical technique for allocating a fixed number of resources to meet a number of demands in such a way that some objective is optimised while other defined conditions are also met.

1.7.2 Queuing Theory: Queuing theory helps in estimating number of people waiting in queue, the expected waiting time in the queue, the server's expected idle time, and so on. Thus, this theory can be applied in situations where decisions must be made to minimise the length and duration of the queue while spending the least amount of money.

1.7.3 Inventory Control Models: It is concerned with the acquisition, storage, and handling of inventories in order to ensure inventory availability whenever needed while minimising waste and losses. It assists managers in determining the best reordering time, level, and quantity.

1.7.4 Nonlinear Programming: These methods can be used when the optimization problem or some of the constraints are not linear. Non-linearity can be created by factors such as a discount on the purchase price of large quantities.

1.7.5 Network Scheduling-PERT and CPM: A network scheduling technique is used to plan, schedule, and monitor large projects. Such large projects are common in the fields of construction, maintenance, computer system installation, R&D design, and so on. Projects undergoing network analysis are divided into individual tasks, which are then arranged in a logical sequence by determining which activities should be performed concurrently and which should be performed sequentially.

1.7.6 Game Theory: It is used to make decisions in conflicting situations with one or more opponents (i.e., players). In game theory, we consider two or more people with different goals, each of whose actions influences the game's outcomes. The game theory provides solutions to such games, assuming that each player wishes to maximise his profits while minimising his losses.

1.7.7 Transportation Problem: The transportation problem is a subset of linear programming problems in which the goal is to minimise the cost of distributing a product from multiple sources to multiple destinations.

1.7.8 Simulation: It is a technique that entails creating a model of a real-world situation and then conducting experiments on it. When it is too dangerous, difficult, or time consuming to conduct a real study or experiment to learn more about a situation, simulation is used.

Chapter - 2 Supply Chain Management

2.1 Supply Chain

The supply chain is a network of organizations, individuals, and activities involved in the creation and delivery of a product or service to the end consumer. It includes all the steps involved in the production and distribution of goods and services, from the sourcing of raw materials to the delivery of finished products.

The supply chain can be divided into several key stages, including:

1. **Procurement:** The process of identifying and selecting suppliers, negotiating contracts, and acquiring the necessary raw materials and supplies for production.
2. **Production:** The process of transforming raw materials and supplies into finished products through manufacturing, assembly, or other production processes.
3. **Transportation:** The movement of goods and materials from one location to another, including the shipping, handling, and storage of products at various points in the supply chain.
4. **Warehousing:** The storage and management of goods and materials at various points in the supply chain, including distribution centres, warehouses, and other storage facilities.
5. **Distribution:** The delivery of products to customers, including the management of logistics, transportation, and delivery networks.

Overall, the supply chain is a complex and interconnected system that involves multiple stakeholders and activities. Efficient supply chain management is essential for businesses to ensure that they can meet customer demands, minimize costs, and remain competitive in the marketplace.

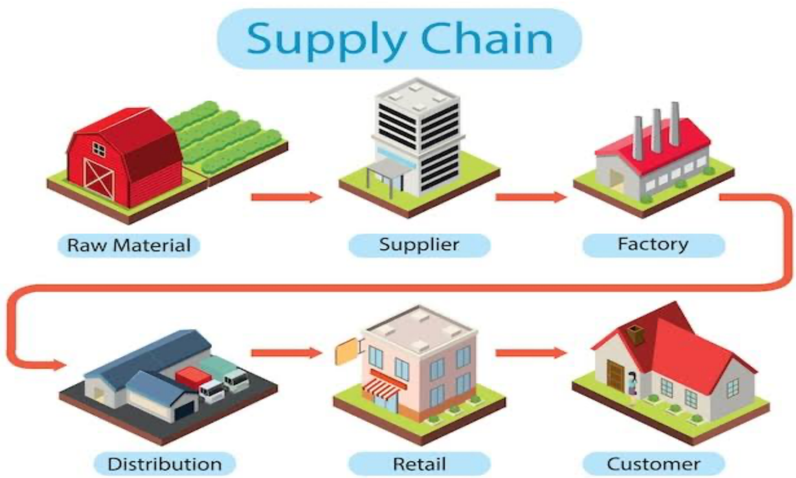


Figure 2.1: Supply Chain

2.2 Supply Chain Management (SCM)

Supply chain management is the management of a product's or service's full manufacturing cycle, from raw materials through ultimate delivery to the customer. A business establishes a network of suppliers (”links in the chain”) to transport products from raw material suppliers to organisations that sell products and services.

Suppliers strive to build and operate supply networks that are as efficient and cost-effective as feasible through supply chain management (SCM). Supply chains encompass every- thing from manufacturing to product development as well as the information systems required to coordinate these activities.

SCM seeks to centrally regulate or link a product’s production, shipment, and distribution. Companies may decrease costs and deliver items to customers faster by optimizing’s the supply chain. Internal inventories, internal manufacturing, distribution, sales, and business vendor stocks are all under tighter supervision. SCM is built on the principle that practically every product that reaches the market is the result of the work of many different elements that make up a supply chain. Although supply chains have been around for a long time, most businesses have only lately recognised them as a valuable addition to their operations.

2.3 How does Supply Chain Management works?

2.3.1 Planning

Planning is a critical component of supply chain management (SCM), as it helps to ensure that the supply chain is efficient, cost-effective, and responsive to customer demands. Effective planning involves developing strategies and processes to manage the flow of goods and services through the supply chain, from the point of origin to the point of consumption.

The practice of correctly arranging the course of a material or a product first from feedstock stage to the end consumer is known as supply chain planning. Supply forecasting, sales forecasting, production scheduling, distribution management, logistics, and sales forecasting are all part of this process.

Overall, planning is a critical component of SCM, as it helps to ensure that the supply chain can operate efficiently and effectively, while meeting the needs of customers and minimizing costs. Effective planning requires collaboration and coordination among all participants in the supply chain, as well as the use of technology and data analytics to support decision-making and improve visibility across the supply chain.

The beating heart of supply chain planning is supply chain forecasting. Businesses analyze previous and present supply to estimate future demand using a combination of historical facts, data, and a little intuition. Understanding how to forecast your supply chain accurately will enhance your relationship with its suppliers and ensure that you are scheduling the correct amount of cargo to minimize excessive costs. You’ll also know how to prepare for both expected and unforeseen disruptions.

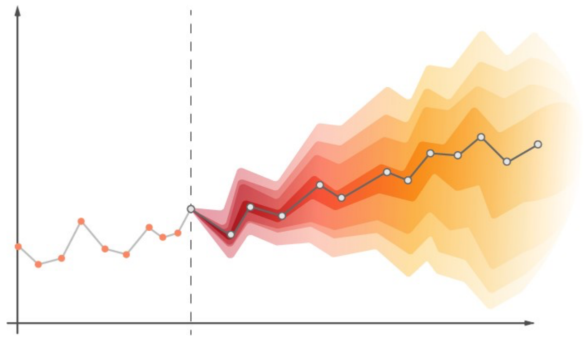


Figure 2.2: Supply Chain Forecasting

2.3.2 Sourcing

The sourcing process encompasses all activities centered on locating and explore potential suppliers, along with selecting and working with the best value provider. Establish mechanisms to plan and control supplier relationships between the buyer and the supplier on what will be acquired and the contract conditions at the conclusion of the sourcing process. Sourcing specialists assess the supply market, devise, and implement a plan, negotiate contract conditions, and draft a contract with suppliers. Ordering, receiving, inventory management, and supplier payments are all important procedures.

Consider the following sourcing strategies:

- Outsourcing: Using external vendors to offer services and goods that have been previously delivered inside.
- Insourcing: Offloading a task to someone within the company.
- Near sourcing: To save time and money, a company locates certain operations near where its final goods are sold. Integrating vertically. Merging enterprises in the same industry that are at various phases of production and/or distribution. Back- ward integration occurs when a firm buys its input provider; forward integration occurs when a company acquires companies in its distribution chain.
- Few or many suppliers: For commodity items, a many-supplier strategy is prevalent, and purchasing is often based on price. Even if other suppliers provide comparable items. Purchases made with only one supplier are known as sole-source procurement. Single-supplier partnerships might be risky, but they can also be profitable.
- Joint ventures: A joint venture is a commercial company formed by two or more partners that has shared ownership, rewards and risks, and governance.
- Virtual business: A network of independent businesses (suppliers, consumers, and rivals) connected by information technology to share talents, costs, and market access.

Sourcing is a critical component of supply chain management (SCM), as it involves identifying, selecting, and managing suppliers to ensure that the necessary materials and resources are available to support production and delivery. Effective sourcing can help to ensure that the supply chain is able to operate efficiently and effectively, while minimizing costs and maintaining quality.

Overall, sourcing is a critical component of SCM, as it helps to ensure that the necessary materials and resources are available to support production and delivery. Effective sourcing requires collaboration and coordination among all participants in the supply chain, as well as the use of technology and data analytics to support decision-making and improve visibility across the supply chain.

2.3.3 Inventory

Inventory is a critical component of supply chain management (SCM), as it involves managing the stock of raw materials, work-in-progress (WIP), and finished goods to ensure that products are available when needed, while minimizing excess inventory and associated costs. Effective inventory management can help to ensure that the supply chain operates efficiently and effectively, without experiencing stockouts or excess inventory.

Some of the key aspects of inventory management in SCM include:

- **Inventory optimization:** This involves determining the appropriate levels of inventory for each stage of the supply chain, based on factors such as demand variability, lead times, and supplier performance. Effective inventory optimization can help to ensure that products are available when needed, while minimizing excess inventory and associated costs.
- **Inventory tracking and visibility:** This involves tracking inventory levels and movements throughout the supply chain, to ensure that products are available when needed and to identify potential issues or bottlenecks. Effective inventory tracking and visibility can help to ensure that the supply chain operates efficiently and effectively, without experiencing stockouts or excess inventory.
- **Inventory replenishment:** This involves replenishing inventory levels as needed, to ensure that products are available when needed and to avoid stockouts. Effective inventory replenishment can help to ensure that the supply chain operates smoothly and responds effectively to unexpected events.
- **Safety stock management:** This involves maintaining a buffer of inventory to protect against unexpected demand spikes, supplier disruptions, or other issues that could impact the supply chain. Effective safety stock management can help to ensure that the supply chain is able to operate smoothly and respond effectively to unexpected events.
- **Inventory cost management:** This involves managing the costs associated with holding inventory, such as storage, handling, and obsolescence. Effective inventory cost management can help to minimize inventory-related costs, while ensuring that products are available when needed.

Overall, inventory management is a critical component of SCM, as it helps to ensure that products are available when needed, while minimizing excess inventory and associated costs. Effective inventory management requires collaboration and coordination among all participants in the supply chain, as well as the use of technology and data analytics to support decision-making and improve visibility across the supply chain.

2.3.4 Production

Many individuals confuse the terms manufacturing and production. As a result, while discussing industry processes, these terms are often interchanged. Manufacturing and production, while having identical basics, are two distinct processes. So, what actually is the gap between manufacturing and production?

2.3.4.1 What Does Manufacturing Actually involve?

Manufacturing is the process of converting raw materials or parts into completed things using tools, human labour, equipment, and chemical processes.

Manufacturing enables companies to sell completed goods for more than the cost of the raw materials required. Large-scale manufacturing enables the mass production of items employing assembly line procedures and technical innovation as primary assets. Manufacturers may take benefits of economies of scale by using efficient production techniques to produce more units for less money.

2.3.4.2 Common processes involved in manufacturing:

- Fabrication
- Prefabrication
- Manufacturing quickly
- Manufacturing agility
- Manufacturing lean
- Manufacturing flexibility

2.3.4.3 Production and it's aspects

Production is an important aspect of supply chain management (SCM) as it involves the actual manufacturing or assembly of products that are needed to meet customer demand. Production in SCM encompasses a range of activities, including planning, scheduling, sourcing of raw materials, and actual manufacturing or assembly.

Effective production management is essential for ensuring that products are produced efficiently, cost-effectively, and to the desired level of quality. This involves a number of considerations, such as optimizing production processes, minimizing waste and downtime, and ensuring that sufficient inventory is available to meet customer demand.

One important aspect of production in SCM is capacity planning. Manufacturers need to carefully manage their production capacity to ensure that they are able to meet customer demand without overproducing and creating excess inventory or underproducing and creating stockouts. Effective capacity planning requires careful consideration of factors such as production lead times, raw material availability, and demand forecasts.

Another key aspect of production in SCM is quality control. Manufacturers need to ensure that the products they produce meet the required quality standards, and that any defects or issues are identified and addressed quickly. Quality control involves a range of activities, such as inspection, testing, and continuous improvement initiatives.

To optimize production in SCM, manufacturers can use a range of tools and techniques, such as lean manufacturing, Six Sigma, and automation. These methodologies aim to eliminate waste, reduce cycle times, and improve quality, which can result in significant cost savings and improved customer satisfaction.

Overall, production is a critical component of SCM, and manufacturers must work closely with their supply chain partners to ensure that they can produce products efficiently, cost-effectively, and to the required level of quality. Effective production management can help to ensure that customer demand is met in a timely and efficient manner, while also minimizing costs and improving quality.

2.3.4.4 Methods of Production:

Job Creation

A single product is produced at a time in job manufacturing. Businesses that employ job production create one-of-a-kind, non-standard goods that are manufactured to order each time. Shipbuilding and bridge construction are examples of job production since each ship is built to the customer's specifications.

Job Production Characteristics

- The product is made to the customer's specifications.
- Multipurpose machinery is used in conjunction with skilled, adaptable labour.
- Workers are driven to complete a wide range of duties.
- Short manufacturing runs result in higher unit costs.

Production in batches

It entails producing many variations of the same fundamental product in batches. Producing soap in various scents, for example. There is a recurrence of production in contrast to job production.

Batch production characteristics

- Suitable for a large range of almost identical commodities that may be produced using the same technology in various locations.
- More diversity for employees means higher job satisfaction.
- Unit costs are higher when manufacturing runs are short.
- When switching between batches, supplies prove to be ideal at times.

Flow production

It includes high production. When one procedure is accomplished, the job continues without interruption to the next. It's a continual process in which pieces go from one stage to the next until they're finished. Chocolate bars production process is an example of flow production.

2.3.4.5 How technology has transformed production processes

Technology advancements can help cut expenses and increase product quality.

- Automation: is the use of computer-controlled technology to produce goods.
- Mechanization: Workers run machines to produce goods.
- CAD stands for computer-aided design: which uses 3D drawing software to create new items.
- Electronic point of sale :When an item's barcode is read once it is sold, stock records are automatically changed at the electronic point of sale. Consider supermarket stock.
- EFTPOS (Electronic Funds Transfer at Point of Sale): Bank-connected cash registers (The customer's card is swiped, and money is immediately transferred from the customer's bank account).

2.3.5 Transportation and Logistics

Transportation and logistics play a crucial role in supply chain management (SCM) as they are responsible for the movement of goods and materials between different stages of the supply chain. Transportation involves the physical movement of goods, while logistics involves the planning, coordination, and management of that movement.

Effective transportation and logistics management is essential for ensuring that products are delivered to customers on time and at the lowest possible cost. This involves a number of considerations, such as optimizing transport routes, selecting the most appropriate mode of transport (e.g. road, rail, sea, air), and managing inventory levels to ensure that sufficient stock is available to meet customer demand.

One important aspect of transportation and logistics in SCM is freight management. This involves the management of the transportation and delivery of goods, including the negotiation of contracts with carriers, the tracking of shipments, and the management of freight costs. Effective freight management requires careful consideration of factors such as transportation mode, delivery schedules, and transit times.

Another key aspect of transportation and logistics in SCM is inventory management. This involves the management of inventory levels to ensure that sufficient stock is available to meet customer demand, without creating unnecessary inventory holding costs. Effective inventory management requires careful coordination between different stages of the supply chain, including manufacturers, distributors, and retailers.

To optimize transportation and logistics in SCM, companies can use a range of tools and techniques, such as transportation management systems (TMS), warehouse management systems (WMS), and supply chain visibility platforms. These technologies can help to improve visibility and coordination across the supply chain, reduce costs, and improve customer satisfaction.

Overall, transportation and logistics are critical components of SCM, and companies must work closely with their supply chain partners to ensure that they can deliver products to customers efficiently and cost-effectively. Effective transportation and logistics management can help to ensure that products are delivered to customers on time, at the lowest possible cost, and with the highest level of quality.

The majority of modes of transportation are designed to move either passengers or freight, although they may transport both. A truck, for example, may move any type of cargo, but a passenger plane does have a belly hold that can accommodate luggage and passengers. Each mode is distinguished by a set of technical, organizational, and eco- nomic qualities. Speed limitations, safety requirements, and operating hours are examples of technical characteristics, whereas technical aspects include factors like horsepower, capacity, and driving technologies.

2.3.6 Return of Goods (Reverse Logistics)

Reverse logistics corresponds to tracking the products' lifecycle after they reach the final user. This might include things like how your product could be reused, how it can be properly disposed of after usage, and any other way your expired product can be useful.

The return of items from the end customer to the producer is the reverse logistics that has the greatest influence on supply chains. The practice of managing items that are returned to an organization is known as returns management, sometimes known as reverse logistics. When dealing with returns, quality control is crucial since faults in the item must be recognized to make improvements to production methods, building materials, or vendors.

Returns have an impact on the supply chain because they increase holding costs by necessitating more storage space and personnel. Not only will returned products take up space, but their value may degrade, reducing their marketability.

To overcome these concerns, an effective returns management mechanism must be implemented. Creating techniques to swiftly recoup value from returned items via resale or secondary markets might be part of this process.

Effective procedures will also enable businesses to provide better customer service, since returns will be handled fast and replacements will be supplied immediately.

Four essential supply chain metrics that aid in identifying the flow of returned products into the supply chain:

- **Quantity:** Are the same goods being returned repeatedly? Is this a widespread occurrence? If you answered yes to either of these questions, you're likely dealing with more than a few malfunctioning units. You might have to consider either recall or maybe a complete revamp of the manufacturing process.
- **Percentage of Sales:** How much of your revenue is lost due to product returns? How many of these items can you reintroduce into your production process using reverse logistics?
- **Product Return Condition:** Is the product malfunctioning after a certain operation? Can you see any failure trends among the returned goods? Quality assurance (QA) and failure repetition are critical in this situation. We need to figure out what's wrong so that we can adapt and fix it before it happens again.
- **Financial Value:** Its firm might be losing huge amounts of money in potential value if we don't monitor and manage reverse logistics. Take, for example, defective devices that are returned to the maker. Electronics sold in secondary marketplaces" represent an estimated $ 15 billion (sales) in the United States," according to" Recovering Lost Profits by Improving Reverse Logistics." By adopting reverse logistics, these electronic enterprises can transform product failure into fresh revenues.

Reason of return	Percentage of returns (%)
Product defective	60
Poor performance-does not meet user expectation	15
Improper marketing of tool	10
Buyer remorse	10
Tool used for a specific purpose, then returned	5

Table 2.1: Return Analysis

Chapter - 3 Transportation Problem and Sensitivity Analysis

3.1 Transportation Problem

The transportation problem is a classic optimization problem in operations research that deals with determining the least-cost way to transport goods from a set of sources to a set of destinations. It is a linear programming problem that can be used to solve a wide range of real-world logistics and supply chain problems, including transportation planning, inventory management, and network design.

In the transportation problem, the objective is to minimize the total cost of transporting a set of goods from a set of sources to a set of destinations, subject to various constraints such as capacity limits and demand requirements. The problem involves determining the optimal shipping quantities from each source to each destination, given the unit transportation costs and the availability and demand of goods.

The transportation problem can be formulated as a linear programming problem with a set of decision variables that represent the amount of goods shipped from each source to each destination. The objective function is typically the total transportation cost, which is the sum of the unit transportation costs multiplied by the shipping quantities. The constraints in the problem include capacity constraints, which limit the amount of goods that can be shipped from each source, and demand constraints, which specify the required amount of goods at each destination.

The transportation problem can be solved using a variety of algorithms, including the simplex method, the transportation simplex method, and the network flow algorithm. These algorithms are designed to find the optimal solution to the problem, which is the set of shipping quantities that minimize the total transportation cost while satisfying all of the constraints.

The transportation problem has numerous applications in logistics and supply chain management, including determining the optimal routing of goods between warehouses and retailers, determining the optimal allocation of inventory between production facilities and warehouses, and optimizing the distribution of goods across a network of suppliers and customers. By solving the transportation problem, companies can optimize their logistics and supply chain operations, reduce costs, and improve customer service.

3.1.1 Variations in Transportation Problem:

The transportation problem has several variations, depending on the specific constraints and requirements of the problem. For example, the problem can be extended to include multiple modes of transportation, such as air, sea, and land transport, with different costs and capacity constraints for each mode. The problem can also be modified to include time constraints, such as delivery deadlines, which can significantly impact the optimal shipping quantities and routes.

3.1.2 Types of Transportation Problem:

There are several types of transportation problems, each with its own set of constraints and requirements. The most common types of transportation problems are:

1. **Basic transportation problem:** This is the simplest form of the transportation problem, which involves determining the optimal shipping quantities from a set of sources to a set of destinations, subject to capacity and demand constraints.

2. **Transshipment problem:** This type of transportation problem involves intermediate locations, called transshipment points, where goods can be temporarily stored and transferred between different modes of transportation. The transshipment problem adds additional complexity to the optimization problem, as it involves determining the optimal routing of goods through multiple stages, with different costs and capacity constraints at each stage.

3. **Multi-commodity transportation problem:** This type of transportation problem involves multiple types of goods or commodities that need to be shipped from multiple sources to multiple destinations, subject to capacity and demand constraints for each commodity.

4. **Time-constrained transportation problem:** This type of transportation problem involves delivery deadlines or time constraints that must be met in addition to capacity and demand constraints. The objective is to minimize the total transportation cost while ensuring that all goods are delivered on time.

5. **Multi-objective transportation problem:** This type of transportation problem involves multiple conflicting objectives, such as minimizing transportation cost and maximizing customer satisfaction. The challenge is to find a trade-off between the different objectives and identify the optimal shipping quantities that satisfy all constraints.

6. **Stochastic transportation problem:** This type of transportation problem involves uncertain demand or supply, where the quantities of goods to be shipped are not known with certainty. The objective is to minimize the expected transportation cost while accounting for the uncertainty in demand or supply.

Each type of transportation problem requires different optimization techniques and algorithms to find the optimal shipping quantities that satisfy all constraints and minimize the total transportation cost. By understanding the specific constraints and requirements of the transportation problem, companies can make data-driven decisions on inventory allocation, shipping routes, and delivery schedules, leading to more effective supply chain management and increased profitability.

3.2 Sensitivity Analysis

Sensitivity analysis in transportation problem is a method used to evaluate how sensitive the optimal solution is to changes in the parameters of the problem. In the transportation problem, the sensitivity analysis is used to determine the impact of changes in the transportation costs, supply and demand quantities, and capacity constraints on the optimal shipping quantities and total transportation cost.

Sensitivity analysis involves computing the dual values, also known as shadow prices, of the constraints and the objective function. The dual values represent the change in the optimal objective value for a unit change in the right-hand side of the constraint or the objective coefficient. By analysing the dual values, we can determine the impact of changes in the parameters on the optimal solution.

3.2.1 Various Categories

The sensitivity analysis in transportation problem can be divided into two categories: sensitivity analysis of the objective function and sensitivity analysis of the constraints.

1. **Sensitivity analysis of the objective function:** The sensitivity analysis of the objective function involves evaluating how the optimal shipping quantities and the total transportation cost change when the unit transportation cost changes. This analysis is important for companies to determine the impact of changes in transportation costs on their supply chain operations.

2. **Sensitivity analysis of the constraints:** The sensitivity analysis of the constraints involves evaluating how the optimal shipping quantities and the total transportation cost change when the capacity constraints, supply or demand quantities change. This analysis is important for companies to determine the impact of changes in capacity constraints or supply and demand quantities on their supply chain operations.

The sensitivity analysis in transportation problem can be performed using various tools such as Excel Solver, LINGO, Python, R and Gurobi. These tools provide sensitivity reports that allow companies to analyse the impact of changes in the parameters on the optimal solution and make data-driven decisions on their supply chain operations.

Sensitivity analysis in transportation problem is an important tool that enables companies to evaluate the impact of changes in the transportation costs, supply and demand quantities, and capacity constraints on their supply chain operations. By performing sensitivity analysis, companies can identify the optimal shipping quantities and total transportation cost and make data-driven decisions to improve their supply chain performance.

In addition to evaluating the impact of changes in the transportation costs, supply and demand quantities, and capacity constraints, sensitivity analysis in transportation problem can also be used to identify the range of values within which the optimal solution remains valid. This range of values is called the range of optimality.

For example, if the capacity of a particular source increases or decreases by a certain amount, the optimal shipping quantities and total transportation cost may change. However, if the change in capacity is within the range of optimality, the optimal solution remains valid. If the change in capacity is outside the range of optimality, a new optimal solution needs to be computed.

The range of optimality can be determined by analysing the shadow prices of the constraints. The shadow prices represent the change in the optimal objective value for a unit change in the right-hand side of the constraint. If the shadow price is positive, it indicates that the constraint is binding, and the optimal solution remains valid within a certain range of values. If the shadow price is zero, it indicates that the constraint is not binding, and the optimal solution remains valid for any value within the constraint.

Sensitivity analysis in transportation problem can also be used to identify the critical constraints or parameters that have the most significant impact on the optimal solution.

Moreover, sensitivity analysis in transportation problem can be extended to multi-objective transportation problems, where the objective is to optimize multiple conflicting objectives, such as minimizing transportation cost and maximizing customer satisfaction. Sensitivity analysis can be used to identify the trade-off between the different objectives and the impact of changes in the parameters on the optimal solution.

In conclusion, sensitivity analysis in transportation problem is a powerful tool that enables companies to evaluate the impact of changes in the transportation costs, supply and demand quantities, and capacity constraints on their supply chain operations. By performing sensitivity analysis, companies can identify the range of optimality, the critical constraints or parameters, and the trade-off between conflicting objectives. This information can be used to optimize their transportation network, improve their supply chain performance, and increase their profitability.

Chapter - 4 Supply Chain Planning and Forecasting

Supply chain planning refers to the process of designing and optimizing a supply chain network to meet the demands of customers while minimizing costs and improving efficiency. Supply chain planning involves a series of interdependent processes, including demand planning, inventory planning, production planning, and transportation planning, among others.

1. Demand planning is the process of forecasting customer demand for a product or service. This information is used to determine the quantities of products that need to be produced or purchased from suppliers. Accurate demand planning is critical for efficient inventory management, as overstocking or understocking can lead to excess costs or lost sales, respectively.

2. Inventory planning involves determining the optimal inventory levels of each product in the supply chain network. The goal is to maintain enough inventory to meet customer demand while minimizing inventory holding costs. Inventory planning requires collaboration between different parts of the supply chain network, including suppliers, manufacturers, and distributors.

3. Production planning involves scheduling the production of goods in a way that maximizes efficiency and minimizes costs. Production planning considers factors such as available resources, production capacity, and production lead times to determine the optimal production schedule.

4. Transportation planning involves optimizing the movement of goods between different points in the supply chain network. This includes determining the best mode of transportation (e.g., truck, ship, air) and the most efficient routing to minimize transportation costs and lead times.

Other processes involved in supply chain planning include supplier selection and management, quality control, and risk management. Effective supply chain planning requires collaboration and communication between different stakeholders in the supply chain network, including suppliers, manufacturers, distributors, and retailers.

Advanced technologies such as artificial intelligence, machine learning, and big data analytics are increasingly being used to improve supply chain planning. These technologies can provide real-time insights into customer demand, inventory levels, production schedules, and transportation routes, enabling companies to make data-driven decisions and optimize their supply chain network.

In conclusion, supply chain planning is a critical process for ensuring the efficient and cost-effective movement of goods through the supply chain network. By optimizing processes such as demand planning, inventory planning, production planning, and transportation planning, companies can improve their supply chain performance and increase their competitiveness in the market.

The core of planning is forecasting. In addition, forecasting is so important to good planning that it is not an overstatement to say that the forecast’s validity and accuracy determine the plan’s success in major part.

4.1 What is Supply Chain Forecasting

Supply chain forecasting is the process of predicting future demand for a product or service in a supply chain network. Forecasting is a critical component of supply chain management because it helps companies plan and allocate resources efficiently to meet customer demand while minimizing costs.

Effective supply chain forecasting requires a comprehensive understanding of the market, including customer behaviour, market trends, and competitor actions. It involves analysing historical data and using statistical models, machine learning algorithms, and other analytical tools to forecast future demand accurately.

The forecasting process typically involves the following steps:

1. **Data collection:** Collecting relevant data from various sources, such as sales data, market data, and customer data.
2. **Data analysis:** Analyzing the data to identify patterns and trends that can help predict future demand.
3. **Forecast generation:** Generating a forecast using statistical models or other analytical techniques.
4. **Forecast evaluation:** Evaluating the accuracy of the forecast by comparing it to actual demand.
5. **Forecast adjustment:** Adjusting the forecast based on the evaluation results and incorporating additional data or insights.

4.1.1 Types of Forecasting

There are several types of forecasting methods used in supply chain management, including:

1. **Time-series forecasting:** This method uses historical data to predict future demand by identifying patterns and trends over time.
2. **Causal forecasting:** This method identifies the relationship between demand and other factors such as economic indicators, competitor actions, and marketing campaigns.
3. **Judgmental forecasting:** This method relies on expert opinions and insights to make predictions about future demand.
4. **Collaborative forecasting:** This method involves collaboration between different stakeholders in the supply chain network, such as suppliers, manufacturers, and retailers, to improve the accuracy of the forecast.

Advanced technologies such as artificial intelligence, machine learning, and big data analytics are increasingly being used to improve supply chain forecasting. These technologies can provide real-time insights into customer behaviour, market trends, and competitor actions, enabling companies to make more accurate predictions and optimize their supply chain operations.

Supply chain forecasting is a critical process for ensuring the efficient and cost-effective movement of goods through the supply chain network. By accurately predicting future demand, companies can plan and allocate resources efficiently, reduce inventory costs, and improve customer satisfaction.

4.1.2 Quantitative forecasting methods

Quantitative and qualitative approaches to supply chain forecasting are the most common. Quantitative forecasting uses complicated algorithms and computer programs to anticipate future sales based on previous data.

1. **Moving average forecasting:** It is a straightforward form of predicting that uses previous averages. However, it treats all data identically and ignores the fact that more recent data may be a better predictor of future trends than data from three or five years ago — and it ignores seasonality and patterns. The moving average model calculates the average of a fixed window of previous observations to forecast the next period. The window size, also known as the order of the model, determines how many previous periods are considered. The moving average model assumes that the future values will be similar to the average of the recent past values.

2. **Exponential smoothing:** It takes into consideration both historical and present data, as well as seasonality. It's perfect for making short-term projections because of this. Exponential smoothing assigns exponentially decreasing weights to past observations, with more weight placed on recent observations. This model assumes that recent observations are more informative and relevant for forecasting than older ones. There are different variations of exponential smoothing models.

3. **The auto-regressive integrated moving average (ARIMA):** It is a forecasting approach that is noted for its accuracy, but it is also time-consuming and expensive. It's best for predicting for up to 18 months. ARIMA is a widely used model for time series forecasting. It combines three components: autoregressive (AR), differencing (I), and moving average (MA). SARIMA extends the ARIMA model to include seasonality. It incorporates additional seasonal components to capture the seasonal patterns in the data.

4. **Neural Networks (LSTM):** This model can be used for time series forecasting by feeding it historical sales data as input and training it to predict future sales. Neural networks can be useful when the relationship between past and future values is complex. Neural networks, particularly recurrent neural networks (RNNs), can be employed for time series forecasting. RNNs have a unique ability to capture temporal dependencies and patterns in sequential data. Long Short-Term Memory (LSTM) networks are a popular variant of RNNs that

are effective in modelling time series data. These networks learn from historical sales data to identify underlying patterns and make predictions.

4.2 Understanding SARIMA

4.2.1 Overview

Time-series data is data that has been indexed in such a way that the data points represent the number of changes occurring over time. For example, a unit of commodity sales for a specific date, week, month, or year, or a temperature change over time. As a result, one of the most significant disciplines of data analysis is forecasting future value based on the history of a time series. We have numerous models in forecasting that assist us produce predictions and forecast values to fulfil our future aspects based on the requirement of the circumstance. AR, MA, ARIMA, SARIMA, VAR, SARIMAX, and other models are examples.

SARIMA (Seasonal Autoregressive Integrated Moving Average) is a statistical model used for time-series forecasting. It is an extension of the ARIMA model, which stands for Autoregressive Integrated Moving Average.

The SARIMA model takes into account the seasonality of a time series, which is a regular pattern that repeats over a fixed period. For example, sales of winter clothes may increase every year during the winter season, while sales of summer clothes may decrease during the same period.

The SARIMA model consists of the following components:

1. **Seasonal differencing:** This involves subtracting the value of the time series at a particular season from the value at the same season in the previous year. Seasonal differencing removes the seasonality from the time series and makes it stationary.
2. **Autoregression:** This involves predicting future values of the time series based on its past values. The autoregression component of the SARIMA model is based on the concept of lagged correlations, which means that the value of the time series at a particular time depends on its values at previous times.
3. **Moving average:** This involves predicting future values of the time series based on the difference between its past values and the mean of the time series. The moving average component of the SARIMA model helps to smooth out the noise in the time series.
4. **Integration:** This involves the differencing of the time series to make it stationary. Stationarity is necessary for accurate forecasting because it allows for the use of statistical methods that assume that the time series has constant mean and variance.

The parameters of the SARIMA model include the seasonality, the order of differencing, the autoregressive order, the moving average order, and the degree of seasonal differencing. These parameters can be estimated using statistical techniques such as maximum likelihood estimation.

The SARIMA model is widely used in industries such as finance, economics, and retail for forecasting sales, stock prices, and economic indicators. It is a powerful tool for predicting future values of a time series based on its past values and seasonality.

4.2.2 SARIMA's Forerunners

1. Autoregressive (AR) Models

Autoregressive (AR) models are a type of statistical model commonly used in time series analysis. They are used to model the behaviour of a time series by relating it to its own past values.

In an AR model, the value of the time series at a given point in time is modelled as a linear function of its past values. The order of the model, denoted by p, determines the number of past values used to predict the current value. For example, an AR(1) model uses only the immediate past value, an AR(2) model uses the two most recent past values, and so on. AR models can be useful for predicting future values of a time series, identifying trends and patterns, and detecting changes in behaviour over time.

2. Moving average (MA) Models.

Moving average (MA) models are another type of statistical model commonly used in time series analysis. They are used to model the behaviour of a time series by relating it to past forecast errors.

In an MA model, the value of the time series at a given point in time is modelled as a linear combination of past forecast errors. The order of the model, denoted by q, determines the number of past forecast errors used to predict the current value. For example, an MA(1) model uses only the immediate past forecast error, an MA(2) model uses the two most recent past forecast errors, and so on. MA models can be useful for predicting future values of a time series, identifying trends and patterns, and detecting changes in behaviour over time.

3. Autoregressive Moving Average (ARMA) Models

Autoregressive Moving Average (ARMA) models are a class of time series models used for forecasting future values based on past observations. The ARMA model combines two separate models, the Autoregressive (AR) model and the Moving Average (MA) model.

The ARMA model combines the AR and MA models to account for both the linear dependence on past values and short-term fluctuations in the data. The ARMA model assumes that the value of a variable at time 't' is a linear combination of its past values and past errors, with the addition of a random error term. The ARMA model is typically denoted as ARMA(p, q), where 'p' is the order of the AR model, and 'q' is the order of the MA model.

ARMA models are widely used in finance, economics, engineering, and other fields where time series analysis is required. They can be used to forecast future values, detect trends, and identify cyclical patterns in the data.

4. Autoregressive Integrated Moving Average (ARIMA) Models

Autoregressive Integrated Moving Average (ARIMA) model is a type of time series model used for forecasting. It is an extension of the Autoregressive Moving Average (ARMA) model, which includes a differencing step to make the time series stationary before fitting an ARMA model. The stationary condition is important for the model to work properly, as it ensures that the model's parameters are constant over time.

The ARIMA model consists of three main components: the autoregressive (AR) component, the differencing (I) component, and the moving average (MA) component. The AR component uses past values of the dependent variable to predict future values. The MA component uses past errors to predict future values. The I component involves differencing the data to make it stationary. The ARIMA model is denoted as ARIMA(p,d,q), where p is the order of the AR component, d is the order of differencing, and q is the order of the MA component.

The ARIMA model is particularly useful for analysing time series data that has a trend or seasonal pattern. It can capture trends and patterns in the data by adjusting the model's parameters accordingly. The ARIMA model is widely used in various fields, including economics, finance, and engineering, to forecast future values of time series data.

5. SARIMA

Seasonal Autoregressive Integrated Moving Average (SARIMA) is an extension of the ARIMA model that includes additional seasonal terms. It is useful for modelling time series data with trends and seasonal patterns.

SARIMA models are similar to ARIMA models, but with the addition of three additional seasonal parameters:

- P: seasonal autoregressive order
- D: seasonal differencing order
- Q: seasonal moving average order

The SARIMA model is denoted as SARIMA(p, d, q)(P, D, Q)m, where p, d, and q are the non-seasonal ARIMA parameters, P, D, and Q are the seasonal ARIMA parameters, and m is the number of time steps in each seasonal cycle.

The seasonal ARIMA terms allow for modelling the seasonal patterns in the data, such as monthly or quarterly seasonality. This can be useful for making more accurate predictions, especially for data with strong seasonal patterns.

4.3 What makes supply chain forecasting difficult?

The access to insights on future expectations, patterns, and supply data that supply chain forecasting gives opens up a world of options for the business. Nonetheless, there are numerous variables that might cause the system to disrupt. Following are the variables:

1. **Changes in regulations and worldwide events:** Changes in rules between countries and continents have caused forecasting problems as supply chains adjust to new laws and historical data becomes less relevant. Emergency laws were established throughout the world in the aftermath of COVID-19 to restrict borders, halt travel, and impede commerce — the consequences have been far-reaching and ongoing, with bottlenecks at borders and congestion at ports. It's simple to understand how regulatory changes, when combined with Brexit, might disrupt supply networks.

2. **Changing consumer habits and trends:** While changing consumer habits and trends are unavoidable in the realm of supply chains, the volatility with which things change makes forecasting difficult. For example, while the globe shut down last year, shoppers turned online and purchased almost $4.2 trillion dollars of products and services globally. While shopping online has only gotten bigger, the abrupt transition prompted many small firms to adjust fast to prevent supply shortages or delays.

3. **Seasonality and lead times from suppliers:** In supply chain forecasting, failing to account for seasonal and peak times may easily knock your forecasts off track. These dates have a significant influence on maritime freight and should be prepared months in advance to avoid missing out on chances.

Chapter - 5 Company Introduction and Problem Description

5.1 Company Introduction

Deepak Spinners Ltd (DSL), located in Chandigarh, is a well-established name in the textile industry. The company has evolved into a leading manufacturer of dyed synthetic yarn. Other than the domestic market, they export to countries such as Syria, the Middle East, Turkey, Belgium, and the U.S.A. to name a few. High standards of trust, quality and satisfaction are endorsed by their customers' long-standing relations with them.

They manufacture yarn of counts (8 – 40) Ne in 100% Polyester, 100% Viscose, 100% Acrylic and Polyester Acrylic & Polyester Viscose blends. With over two decades of experience, the Company remains committed to the highest standards of quality and productivity.

Their product range sets a benchmark of superior quality and unmatched excellence. They aim to create and enhance customer value through a robust and sustainable business model that promotes our deep-rooted belief to synergize growth with responsibility.

5.1.1 History

- **1986**
  In 1986, the Company commenced operations with 12000 spindles and a dye house in Baddi, Himachal Pradesh.
- **1988**
  In 1988, the Company began manufacturing acrylic yarn in order to expand its product range.
- **1991**
  In 1991, the second unit was established in Guna, Madhya Pradesh with an initial capacity of 8000 spindles.
- **1993**
  In 1993, the Company received the prestigious status of an export house and expanded its production capacities.
- **2012**
  In 2012, the Company undertook modernization and de-bottlenecking of its unit in Baddi and added 13200 spindles at its unit in Guna.
- **2016**
  In 2016, the Company added another 14112 spindles at its unit in Guna.
- **2022**
  In 2022, the Company has begun the installation of a 3.7 MW solar plant at its unit in Guna.
- **Present**
  Presently, a total of 90,864 spindles are operational at both units along with a fibre dyeing plant.

5.1.2 Mission and Vision

Mission:

To deliver premium value to our stakeholders, continuously. Above all, we uphold our values to remain a responsible corporate citizen.

Vision:

To increase our core product offering in order to maintain a position of leadership.

5.2 Problem Description

The two branches of the company transport goods to their clients across the country. The textile company has been experiencing a transportation problem. The current policy followed is based on experience and intuitions. Deciding about where to transport in how much quantity does not follow any analytical method or computational technique. The problem is that current transportation system is not optimized for efficiency, resulting in increased costs and longer transportation times.

The company is interested in finding a solution of their problem so that the excess money spent in the transportation can be saved and utilized in a better way. They are planning to implement a new policy.

In addition, the textile industry is highly competitive, and effective inventory management plays a crucial role in maximizing profitability. To achieve this, accurate forecasting of future sales is essential. This will allow the company to make informed decisions about production levels, inventory management, and marketing strategies.

Also, the company wants to know that whether in the future they can expand its units in one of its branches i.e., Baddi, Himachal Pradesh and Guna, Madhya Pradesh.

To find the conclusion, we will be focusing on some of the aspects based on which a policy will be formed.

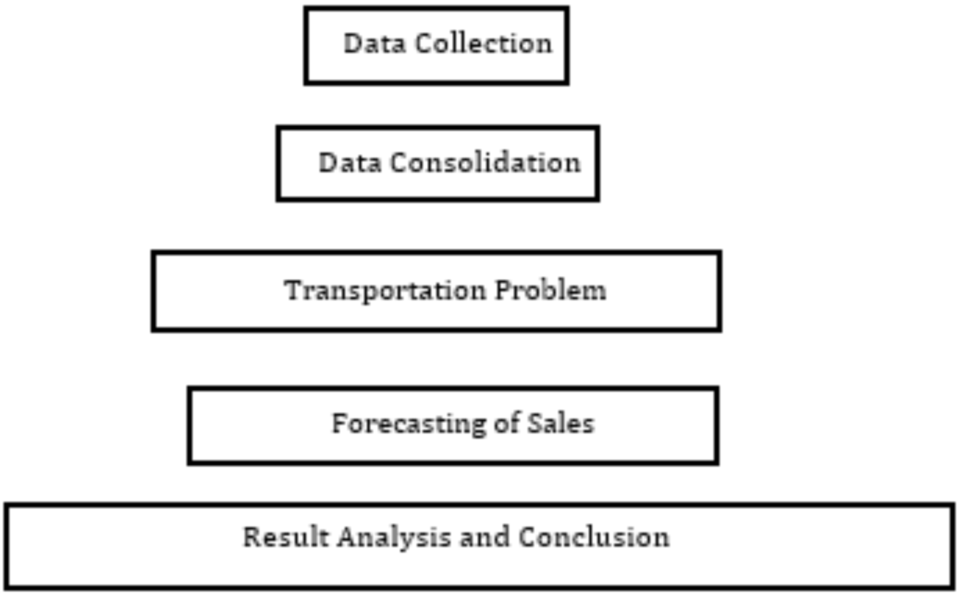
The following two issues will be addressed in this project:

- Finding optimal transportation of goods from the company’s two branches to all customers.
- Forecasting of the sales.

Chapter – 6 Solution Methodology

Primary Information was collected through the interaction with the executives. Secondary information and data was collected through the sales and freight database of the company.

The following diagram depicts the five steps of this research.



- Firstly, The data for the transportation problem collected on a monthly basis for a total of 29 destinations. The demand data of each destination corresponding to the supply of the two branches of the company and their relative cost were collected.
  With the help of Python programming, the optimal solution was obtained. Further to understand the impact of the input parameters on the optimal solutions, Sensitivity analysis was performed.
- Secondly, the monthly data for the supply forecasting was collected from April 2019 to March 2023 and it was consolidated into a single spreadsheet file. It consisted of month-wise distribution of the sales of the company's highest produced thread- NE 1/18 PO SG 54/55/56 (Polyester) in Guna along with their month-wise total.
  The forecasting was done with the help of Python Programming as it gives very detailed and enhanced results.
- Finally, the new policy was found based on the observations of the analysis for both transportation and sales. Then a comparison was made between the old and new policies on the basis of the total costs associated with them.

Chapter – 7 Data Analysis and Results

Transportation

We begin by importing few python libraries and loading our data for the project.

The two libraries imported are, pulp and pandas.

- The **pulp** library is a popular Python library for creating and solving optimization problems. It provides a high-level modelling language for defining decision variables, objectives, and constraints, and it can use a variety of solvers to find the optimal solution.
- The **pandas** library is a popular data analysis library in Python. It provides powerful data structures for working with tabular data, and it can read and write data in a variety of formats, including Excel.

An Excel file named **dp_final.xlsx** is read using the **pd.read_excel( )** function of the pandas module and stores the data in a pandas DataFrame called **df**.

We used the **df.head()** and **df.tail()** functions to confirm that the data was loaded correctly.

	S.no	Destination	Madhya Pradesh	Himachal	Demand
0	1	AHMEDABAD (VINAYAK TRANSPORT CO.)	2047.952048	3124.12	10
1	2	ALWAR (SHIV ROAD LINES)	2564.102564	2900.20	21
2	3	AMRITSAR (SHIV ROAD LINES)	3597.753974	3425.25	23
3	4	BADDI+LUDHIANA (NEW MAHAVEER TRANSPORT CO OF B...	3148.985985	2123.34	36
4	5	BHILWARA (SHIV ROAD LINES)	1698.304667	2635.34	141

Table 7.1: df.head()

	S.no	Destination	Madhya Pradesh	Himachal	Demand
25	26	SANTIPUR-KOLKATTA (NEW MAHAVEER TRANSPORT CO O...	6110.364289	6845.82	72
26	27	SANTIPUR-KOLKATTA (VINAYAK TRANSPORT CO.)	6439.850127	7259.34	35
27	28	SURAT (VINAYAK TRANSPORT CO.)	2339.244914	3035.45	9
28	29	SURAT (NEW MAHAVEER TRANSPORT CO OF BHARAT)	2259.615385	2837.16	10
29	30	SURAT (SHIV ROAD LINES)	2394.957983	3523.65	11

Table 7.2: df.tail()

Lists and dictionaries are created to represent the problem data.

Branch is a list of branches, **Warehouse** is a list of warehouses, **supply** is a dictionary that represents the supply available at each branch, **demand** is a dictionary that represents the demand at each warehouse, and **cost** is a dictionary of dictionaries that represents the transportation costs between branches and warehouses.

The PuLP problem object is created with the name "Transportation" and the objective of minimizing costs. The **LpProblem( )** function is used in the PuLP library to create a new optimization problem. It takes two arguments: the first argument is a string that represents the name of the problem, and the second argument is a constant that indicates the minimization problem (**LpMinimize**).

Further, a list of possible routes between branches and warehouses is created called **routes.**

A decision variable **x** using the **LpVariable( )** function from PuLP is created. The **dicts** method is used to create a dictionary of decision variables '**x'** with keys from the **Branch** and **Warehouse** lists, and an initial value of 0.

The objective function of the problem is defined using the **lpSum( )** function to sum the product of the decision variable '**x'** and the transportation cost '**cost'** for each possible route **(i,j)** in **routes.**

Constraints are created for each warehouse '**j'** and each branch '**i'** using the **lpSum( )** function to ensure that the total supply from all branches '**i'** to the warehouse '**j'** is equal to the demand for that warehouse and that the total supply from that branch to all warehouses is less than or equal to the supply available at the branch.

The **prob.solve( )** function starts the optimization process and find the optimal values of the decision variables that minimize or maximize the objective function, subject to the given constraints.

After calling solve( ), the **status** attribute of **prob** object will be updated through **LpStatus[ ]**, and the optimal solution can be accessed by calling the **value( )** function on the objective attribute of the prob object.

By performing the above-mentioned steps from the code, we get the following results:

```
Status:  Optimal

Optimal Solution:

Total Cost:  4522346.3159982655
```

Figure 7.1: Optimal status and solution

Hence the solution obtained is an optimal solution.

Now since the optimal solution has been derived, our next step is to gain the possible combination of allocations from branches to warehouses that give us the optimal solution along with their corresponding optimal cost.

We start by creating an empty Python list called **results**. This list will be used to store the results of the optimization problem.

Then we append the data for each variable to the list. We loop through each possible combination of **i** in **Branch** and **j** in **Warehouse**. For each combination, we append a dictionary to the **results** list. The dictionary contains the values for the following keys:

- '**From**': the value of **i**
- '**To**': the value of **j**
- '**Quantity**': the optimized value of the decision variable **x[i][j]** (using the **.varValue** attribute of the variable)
- '**Cost**': the cost of shipping from i to j as defined in the **cost** dictionary.

Essentially, we create a list of dictionaries that stores the optimal values for each decision variable, along with the associated cost. Then we convert the resultant list to a Pandas dataframe called **df_results** with the help of **pd.DataFrame( )** function.

pd.get_dummies( )creates dummy variables for the **From** column in the **df_results** dataframe. This is done so that each branch has its own column in the final output, which makes it easier to read and analyze the results. Then we concatenate the **df_results** dataframe with the **df_from** dataframe, which contains the dummy variables for the **From** column by the help of **pd.concat( )** function.

Next, we use **.drop( )** function to drop the original **From** column from the **df_results** dataframe, since it is no longer needed after the dummy variables have been created. A new column to the **df_results** dataframe is added called **Total Cost**, which is calculated as the product of the **Quantity** and **Cost** columns.

We filter out the rows by removing any rows from the **df_results** dataframe where the **Total Cost** is 0, since these routes were not actually used in the optimal solution. We then reset the index of the **df_results** dataframe, since some rows may have been removed in the previous step.

Ultimately, we print the final **df_results** dataframe, which contains the details of the optimal shipping routes, including the quantity shipped, the cost per unit, and the total cost for each route. The dataframe is organized by branch and warehouse, with dummy variables used to represent each branch, making it easy to read and analyze the results.

Following is the result obtained:

	To	Quantity	Cost	From_HP	From_MP	Total Cost
0	AHMEDABAD (VINAYAK TRANSPORT CO.)	10.0	2047.952048	0	1	20479.520480
1	ALWAR (SHIV ROAD LINES)	21.0	2564.102564	0	1	53846.153846
2	BHILWARA (SHIV ROAD LINES)	141.0	1698.304667	0	1	239460.958099
3	BHILWARA (VINAYAK TRANSPORT CO.)	135.0	1649.611589	0	1	222697.564575
4	BHIWANDI (VINAYAK TRANSPORT CO.)	11.0	3225.941598	0	1	35485.357577
5	COIMBATORE (SAFEXPRESS PVT.LTD.)	1.0	20063.211125	0	1	20063.211125
6	DELHI (VINAYAK TRANSPORT CO.)	225.0	2806.674505	0	1	631501.763645
7	DELHI (SHIV ROAD LINES)	161.0	2886.680280	0	1	464755.525156
8	HYDERABAD (SHIV ROAD LINES)	22.0	14790.833352	0	1	325398.333738
9	ICD MANDIDEEP (SHRIRAM LOGISTICS)	23.0	3204.778286	0	1	73709.900585
10	INDORE (SHIV ROAD LINES)	14.0	1330.345890	0	1	18624.842459
11	KANPUR (SHIV ROAD LINES)	34.0	2125.703780	0	1	72273.928530
12	KOLKATTA (SHIV ROAD LINES)	12.0	6278.586279	0	1	75343.035343
13	KOLKATTA (VINAYAK TRANSPORT CO.)	14.0	6160.653506	0	1	86249.149081
14	NHAVA SHEVA (VINAYAK TRANSPORT CO.)	9.0	4016.767429	0	1	36150.906862
15	PANIPAT (SHIV ROAD LINES)	2.0	3125.000000	0	1	6250.000000
16	PATNA (VINAYAK TRANSPORT CO.)	12.0	5201.831045	0	1	62421.972534
17	PURBA BARDDHAMAN (NEW MAHAVEER TRANSPORT CO OF...	17.0	6125.403720	0	1	104131.863236
18	SANTIPUR-KOLKATTA (SHIV ROAD LINES)	30.0	6333.333333	0	1	190000.000000
19	SANTIPUR-KOLKATTA (NEW MAHAVEER TRANSPORT CO O...	72.0	6110.364289	0	1	439946.228794
20	SANTIPUR-KOLKATTA (VINAYAK TRANSPORT CO.)	35.0	6439.850127	0	1	225394.754449
21	SURAT (VINAYAK TRANSPORT CO.)	9.0	2339.244914	0	1	21053.204222
22	SURAT (NEW MAHAVEER TRANSPORT CO OF BHARAT)	10.0	2259.615385	0	1	22596.153846
23	SURAT (SHIV ROAD LINES)	11.0	2394.957983	0	1	26344.537815
24	AMRITSAR (SHIV ROAD LINES)	23.0	3425.250000	1	0	78780.750000
25	BADDI+LUDHIANA (NEW MAHAVEER TRANSPORT CO OF B...	36.0	2123.340000	1	0	76440.240000
26	LUDHIANA (NEW MAHAVEER TRANSPORT CO OF BHARAT)	145.0	2456.560000	1	0	356201.200000
27	LUDHIANA (SHIV ROAD LINES)	68.0	2536.320000	1	0	172469.760000
28	LUDHIANA (VINAYAK TRANSPORT CO.)	115.0	2378.340000	1	0	273509.100000
29	LUDHIANA (SHANKAR TRANSPORT CO.)	32.0	2836.450000	1	0	90766.400000

Table 7.3: Optimal shipping routes with corresponding cost and quantity

Based on the given result, it can be concluded that the optimal transportation plan for the given scenario involves transporting goods from the Branches (HP and MP) to the Warehouses at different locations across India. The "Quantity" column shows the amount of goods to be transported from each Branch to each Warehouse, while the "Cost" column shows the cost associated with transporting that quantity of goods from each Branch to each Warehouse. The "Total Cost" column shows the overall cost of transporting all the goods from all the Branches to all the Warehouses as per the optimal plan. It can be observed that the total cost is minimized by using a combination of different transportation routes from different Branches to different Warehouses, as determined by the optimization algorithm used in the model.

Sensitivity Analysis

We first initialize an empty list called **sensitivity_results** to store the sensitivity analysis results.

It then loops through each constraint in the problem and checks if it's a dummy or internal constraint. If it's not, it creates a dictionary called **'s'** with the constraint name and its shadow price. If the constraint has slack, it adds the slack value to the dictionary and calculates the allowable increase or decrease in the objective function coefficient that would keep the current optimal solution feasible.

Next, the code loops through each variable in the problem and checks if it's a dummy or internal variable.

If it's not, it creates a dictionary called **'s'** with the variable name, its current value, and its reduced cost. If the variable has a value of zero and a negative reduced cost, it calculates the allowable increase in the objective function coefficient that would make the variable positive while keeping the current optimal solution feasible.

If the variable has a positive value and a positive reduced cost, it calculates the allowable decrease in the objective function coefficient that would make the variable zero while keeping the current optimal solution feasible.

The code then appends each sensitivity analysis result dictionary to the **sensitivity_results** list. It converts the list to a Pandas dataframe called **df_sensitivity** and prints the dataframe with the column names "**Constraint", "Shadow Price", "Value", and "Reduced Cost".**

By applying sensitivity analysis, we get the following results:

Sensitivity Analysis:

	Variable	Value	Reduced Cost
0	x_HP_AHMEDABAD_(VINAYAK_TRANSPORT_CO.)	0.0	1076.16800
1	x_HP_ALWAR_(SHIV_ROAD_LINES)	0.0	336.09744
2	x_HP_AMRITSAR_(SHIV_ROAD_LINES)	23.0	0.00000
3	x_HP_BADDI_LUDHIANA_(NEW_MAHAVEER_TRANSPORT_CO...	36.0	0.00000
4	x_HP_BHILWARA_(SHIV_ROAD_LINES)	0.0	937.03533
5	x_HP_BHILWARA_(VINAYAK_TRANSPORT_CO.)	0.0	726.72841
6	x_HP_BHIWANDI_(VINAYAK_TRANSPORT_CO.)	0.0	487.39840
7	x_HP_COIMBATORE__(SAFEXPRESS_PVT.LTD.)	0.0	5278.12890
8	x_HP_DELHI_(SHIV_ROAD_LINES)	0.0	675.54972
9	x_HP_DELHI_(VINAYAK_TRANSPORT_CO.)	0.0	316.88549
10	x_HP_HYDERABAD_(SHIV_ROAD_LINES)	0.0	2443.50660
11	x_HP_ICD_MANDIDEEP_(SHRIRAM_LOGISTICS)	0.0	9121.55170
12	x_HP_INDORE_(SHIV_ROAD_LINES)	0.0	1914.86410
13	x_HP_KANPUR_(SHIV_ROAD_LINES)	0.0	608.50622
14	x_HP_KOLKATTA_(SHIV_ROAD_LINES)	0.0	78974.41400
15	x_HP_KOLKATTA_(VINAYAK_TRANSPORT_CO.)	0.0	1774.34650
16	x_HP_LUDHIANA_(NEW_MAHAVEER_TRANSPORT_CO_OF_BH...	145.0	0.00000
17	x_HP_LUDHIANA_(SHANKAR_TRANSPORT_CO.)	32.0	0.00000
18	x_HP_LUDHIANA_(SHIV_ROAD_LINES)	68.0	0.00000
19	x_HP_LUDHIANA_(VINAYAK_TRANSPORT_CO.)	115.0	0.00000
20	x_HP_NHAVA_SHEVA_(VINAYAK_TRANSPORT_CO.)	0.0	662.15257
21	x_HP_PANIPAT__(SHIV_ROAD_LINES)	0.0	1128.23000
22	x_HP_PATNA_(VINAYAK_TRANSPORT_CO.)	0.0	236.05896
23	x_HP_PURBA_BARDDHAMAN_(NEW_MAHAVEER_TRANSPORT_...	0.0	799.43628
24	x_HP_SANTIPUR_KOLKATTA_(NEW_MAHAVEER_TRANSPORT...	0.0	735.45571
25	x_HP_SANTIPUR_KOLKATTA_(SHIV_ROAD_LINES)	0.0	804.49667
26	x_HP_SANTIPUR_KOLKATTA_(VINAYAK_TRANSPORT_CO.)	0.0	819.48987
27	x_HP_SURAT_(NEW_MAHAVEER_TRANSPORT_CO_OF_BHARAT)	0.0	577.54462
28	x_HP_SURAT_(SHIV_ROAD_LINES)	0.0	1128.69200
29	x_HP_SURAT_(VINAYAK_TRANSPORT_CO.)	0.0	696.20509
30	x_MP_AHMEDABAD_(VINAYAK_TRANSPORT_CO.)	10.0	0.00000
31	x_MP_ALWAR_(SHIV_ROAD_LINES)	21.0	0.00000
32	x_MP_AMRITSAR_(SHIV_ROAD_LINES)	0.0	172.50397
33	x_MP_BADDI_LUDHIANA_(NEW_MAHAVEER_TRANSPORT_CO...	0.0	1025.64600
34	x_MP_BHILWARA_(SHIV_ROAD_LINES)	141.0	0.00000
35	x_MP_BHILWARA_(VINAYAK_TRANSPORT_CO.)	135.0	0.00000
36	x_MP_BHIWANDI_(VINAYAK_TRANSPORT_CO.)	11.0	0.00000
37	x_MP_COIMBATORE__(SAFEXPRESS_PVT.LTD.)	0.0	0.00000
38	x_MP_DELHI_(SHIV_ROAD_LINES)	161.0	0.00000
39	x_MP_DELHI_(VINAYAK_TRANSPORT_CO.)	225.0	0.00000
40	x_MP_HYDERABAD_(SHIV_ROAD_LINES)	22.0	0.00000
41	x_MP_ICD_MANDIDEEP_(SHRIRAM_LOGISTICS)	23.0	0.00000
42	x_MP_INDORE_(SHIV_ROAD_LINES)	14.0	0.00000
43	x_MP_KANPUR_(SHIV_ROAD_LINES)	34.0	0.00000
44	x_MP_KOLKATTA_(SHIV_ROAD_LINES)	12.0	0.00000
45	x_MP_KOLKATTA_(VINAYAK_TRANSPORT_CO.)	14.0	0.00000
46	x_MP_LUDHIANA_(NEW_MAHAVEER_TRANSPORT_CO_OF_BH...	0.0	503.05230
47	x_MP_LUDHIANA_(SHANKAR_TRANSPORT_CO.)	0.0	161.42871
48	x_MP_LUDHIANA_(SHIV_ROAD_LINES)	0.0	462.21340
49	x_MP_LUDHIANA_(VINAYAK_TRANSPORT_CO.)	0.0	567.37420
50	x_MP_NHAVA_SHEVA_(VINAYAK_TRANSPORT_CO.)	9.0	0.00000
51	x_MP_PANIPAT__(SHIV_ROAD_LINES)	2.0	0.00000
52	x_MP_PATNA_(VINAYAK_TRANSPORT_CO.)	12.0	0.00000
53	x_MP_PURBA_BARDDHAMAN_(NEW_MAHAVEER_TRANSPORT_...	17.0	0.00000
54	x_MP_SANTIPUR_KOLKATTA_(NEW_MAHAVEER_TRANSPORT...	72.0	0.00000
55	x_MP_SANTIPUR_KOLKATTA_(SHIV_ROAD_LINES)	30.0	0.00000
56	x_MP_SANTIPUR_KOLKATTA_(VINAYAK_TRANSPORT_CO.)	35.0	0.00000
57	x_MP_SURAT_(NEW_MAHAVEER_TRANSPORT_CO_OF_BHARAT)	10.0	0.00000
58	x_MP_SURAT_(SHIV_ROAD_LINES)	11.0	0.00000
59	x_MP_SURAT_(VINAYAK_TRANSPORT_CO.)	9.0	0.00000

Table 7.4: Sensitivity analysis result

- The '**value**' column in the output of sensitivity analysis represents the current value of the decision variable in the optimal solution of the linear programming problem.

  In other words, it shows the quantity of the decision variable that should be produced or consumed to optimize the objective function, given the constraints and the coefficient values in the objective function.

  For instance, in the given output, the value of the decision variable x_HP_AMRITSAR_(SHIV_ROAD_LINES) is 23.0, which means that the optimal solution of the linear programming problem requires producing or consuming 23 units of this variable to maximize the objective function, while satisfying the given constraints.

  On the other hand, for some decision variables, the value is 0.0, which means that the optimal solution does not require any production or consumption of these variables to optimize the objective function.

- The '**reduced cost**' column refers to the amount by which the objective function coefficient of a non-basic variable would have to improve in order for that variable to enter the basis and become a basic variable.

In other words, the reduced cost of a variable is the difference between the value of the objective function if the variable is included in the solution (i.e., its coefficient in the objective function is positive) and the current value of the objective function. If the reduced cost of a variable is negative, then increasing the value of that variable will improve the objective function value and may lead to a better solution. If the reduced cost is zero or positive, then that variable is already at its optimal value or it cannot be included in the solution without worsening the objective function value.

Since the above obtained solution has a **positive** value for the reduced cost or is **zero**, this implies that the variable is already at its **optimal value** and it cannot be included to make the results better.

Overall, the sensitivity analysis results can provide valuable insights to decision-makers regarding the impact of changes in supply, demand, and costs on the optimal solution, and help them make more informed decisions.

Sales Forecast

We begin by importing the required libraries:

- The **pandas** library is used for data manipulation and analysis.
- The **numpy** library is used for numerical operations.
- The **matplotlib** libraries are used for data visualization.
- The “**SARIMAX”** model class from the "**statsmodels**" library, which is a Python library for time series analysis and modelling.
- The "**auto_arima**" function from the "**pmdarima**" library, which is a Python library for time series analysis and forecasting. The "auto_arima" function is used to automatically select the optimal parameters for an ARIMA model based on the input data.
- The "**mean_squared_error**" function from the "**sklearn**" library, which is a Python library for machine learning and data analysis. The "mean_squared_error" function is used to calculate the mean squared error between two arrays, which is a common evaluation metric for regression models.
- the “**ARIMA”** class from the **statsmodels”** library for time series analysis and forecasting.
- The “**ExponentialSmoothing”** class from the “**statsmodels** library for implementing the Holt-Winters forecasting method.
- The “**MinMaxScaler”** class from the “**sklearn.preprocessing”** module. It is often used to pre-process the input data before feeding it into the neural network model.
- the “**Sequential”** class from the ‘**tensorflow.keras.models’** module. The “Sequential” class allows the creation of a linear stack of neural network layers.
- the “**Dense”** and “**LSTM”** layer classes from the “**tensorflow.keras.layers”** module. The “Dense” layer represents a fully connected layer in a neural network, while the “LSTM” layer is a specialized layer for handling sequences and capturing long-term dependencies in the data.

The datasets are uploaded using the **pd.read_excel ()** function from the pandas library. The **'parse_dates'** parameter is used to convert the 'Date' column into a datetime format, and the **'index_col**' parameter is used to set the 'Date' column as the index of the DataFrame.

Then the dataset is split into training and testing sets. The **'train_size**' variable is calculated as 80% of the length of the DataFrame, and then converted to an integer using the 'int' function. The 'train' DataFrame is created by slicing the 'df' DataFrame from the beginning up to the 'train_size' index. The 'test' DataFrame is created by slicing the 'df' DataFrame from the 'train_size' index to the end of the DataFrame. We used the **train.head()** and **test.head()** functions to confirm that the data was loaded correctly.

Date	Sales
2012-04-01	846325.0
2012-05-01	936390.0
2012-06-01	955998.0
2012-07-01	927638.0
2012-08-01	1055278.0

Table 7.5: train.head()

Date	Sales
2021-01-01	1810619.52
2021-02-01	1374913.30
2021-03-01	1365767.70
2021-04-01	701655.70
2021-05-01	643814.00

Table 7.6: test.head()

Data visualisation is the most crucial step in forecasting as it presents us with the insights withing the data to better understand it and make sense out of it by observing patterns and trend.

Next we observe the characteristics of the data by plotting the time series of the original data using the **plt.plot( )** function of the matplotlib library (figure 7.2).

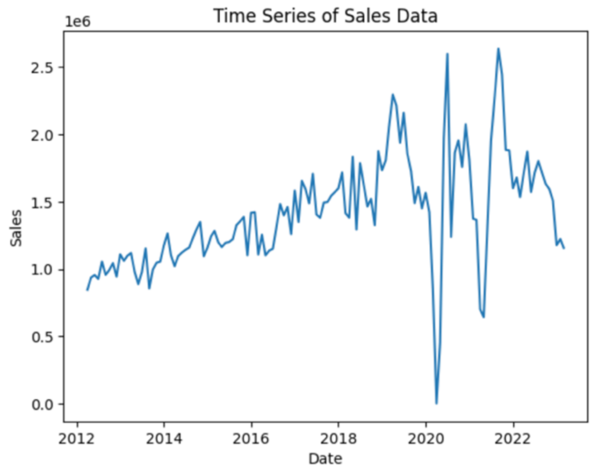
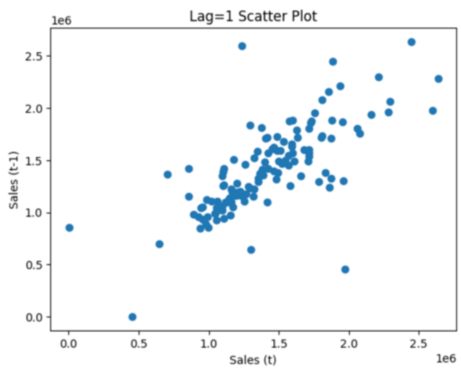


Figure 7.2: Time series of original data

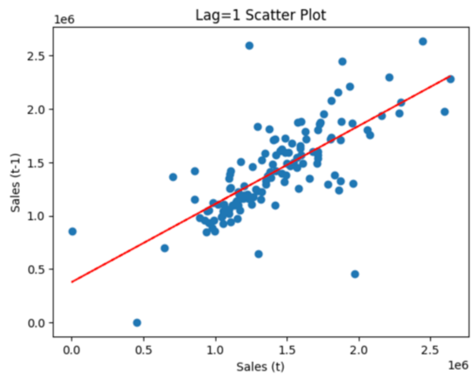
Two of the important data visualization operations are as follows:

(a) Time series modelling assumes a relationship between an observation and the previous observation. Previous observations in a time series are called lags. A useful type of plot to explore the relationship between each observation and a lag presented in (Figure 7.3). Here, Lag of 1 time period is plotted against the previous observations showing a relatively strong dependence between the two. Relationship between the current values and the lag values gave us the insight that using lag feature in our analysis would be practical.

(b) Another possibility for better capturing the shape of the trend is to add a trend line. By trying different trend lines we came up with linear trend line to be best fit, as it best approximates the data.



(a) Lag=1 scatter plot



(b)Fitting trend line

Figure 7.3: Data Visualisation

We try to forecast the sales using several models. The name of the models applied along with their corresponding RMSE vale is stated in the table given below:

Models	RMSE
Moving Average	504716.38
Exponential Smoothing	581789.99
ARIMA	516872.08
SARIMA	446817.24
LSTM	1672664.04

Table 7.7: Models applied along with their RMSE

Based on the above table we conclude that SARIMA is the best choice of model to be implemented for the sales forecasting.

We perform a grid search to determine the optimal hyperparameters for a SARIMA model using the **auto_arima** function. The auto_arima function is called with the following arguments:

- **train:** The training dataset used for model selection.
- **start_p, start_q:** The starting values for the order of the non-seasonal AR and MA components, which are set to be **1** and **1** respectively.
- **max_p, max_q:** The maximum values for the order of the non-seasonal AR and MA components ,which are set to be **3** and **3** respectively.
- **m:** The number of periods in each season, **12** in this case.
- **start_P:** The starting value for the order of the seasonal AR component,**0** in this case.
- **seasonal:** Specifies whether the model is seasonal or not, which is set to be **True**.
- **d, D:** The orders of differencing for the non-seasonal and seasonal components, respectively ,which are set to be **0** and **1** respectively.
- **trace:** Prints detailed debugging information during the grid search ,which is set to be **True**.
- **error_action:** Specifies how to handle errors encountered during the grid search, which is set to be ignored.
- **suppress_warnings:** Controls whether warning messages should be suppressed, which is set to be **True**.
- **stepwise:** Specifies whether to perform a stepwise search or an exhaustive grid search, which is set to be **True**.

The selected hyperparameters for the SARIMA model obtained through the grid search are printed. It reveals the optimal order values for the non-seasonal components (AR, I, MA) and the seasonal components (SAR, SI, SMA, m), highlighting the crucial parameter selection process for accurate time series forecasting.

```
Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,1,1)[12] intercept   : AIC=2626.812, Time=0.46 sec
 ARIMA(0,0,0)(0,1,0)[12] intercept   : AIC=2674.503, Time=0.04 sec
 ARIMA(1,0,0)(1,1,0)[12] intercept   : AIC=2626.872, Time=0.33 sec
 ARIMA(0,0,1)(0,1,1)[12] intercept   : AIC=2633.545, Time=0.19 sec
 ARIMA(0,0,0)(0,1,0)[12]             : AIC=2675.308, Time=0.04 sec
 ARIMA(1,0,1)(0,1,0)[12] intercept   : AIC=2640.532, Time=0.12 sec
 ARIMA(1,0,1)(1,1,1)[12] intercept   : AIC=2628.060, Time=1.10 sec
 ARIMA(1,0,1)(0,1,2)[12] intercept   : AIC=2627.295, Time=0.67 sec
 ARIMA(1,0,1)(1,1,0)[12] intercept   : AIC=2628.983, Time=0.37 sec
 ARIMA(1,0,1)(1,1,2)[12] intercept   : AIC=2627.688, Time=1.81 sec
 ARIMA(1,0,0)(0,1,1)[12] intercept   : AIC=2624.945, Time=0.43 sec
 ARIMA(1,0,0)(0,1,0)[12] intercept   : AIC=2639.762, Time=0.12 sec
 ARIMA(1,0,0)(1,1,1)[12] intercept   : AIC=2626.181, Time=1.65 sec
 ARIMA(1,0,0)(0,1,2)[12] intercept   : AIC=2625.337, Time=0.45 sec
 ARIMA(1,0,0)(1,1,2)[12] intercept   : AIC=2625.924, Time=1.03 sec
 ARIMA(0,0,0)(0,1,1)[12] intercept   : AIC=2667.722, Time=0.14 sec
 ARIMA(2,0,0)(0,1,1)[12] intercept   : AIC=2626.461, Time=0.27 sec
 ARIMA(2,0,1)(0,1,1)[12] intercept   : AIC=2627.509, Time=0.64 sec
 ARIMA(1,0,0)(0,1,1)[12]             : AIC=2627.983, Time=0.17 sec

Best model:  ARIMA(1,0,0)(0,1,1)[12] intercept
Total fit time: 10.127 seconds
(1, 0, 0)
(0, 1, 1, 12)
```

Figure 7.4: Selected hyperparameters for SARIMA

A SARIMAX model is instantiated with the training dataset, using the optimal order values for the non-seasonal components (AR, I, MA) and the seasonal components (SAR, SI, SMA, m) identified through the grid search. The model is then fitted to the training data, and the fitting results are printed as a summary. This summary provides detailed statistical information about the fitted SARIMA model:

```
                                      SARIMAX Results
==========================================================================================
Dep. Variable:                                Sales   No. Observations:                105
Model:             SARIMAX(1, 0, 0)x(0, 1, [1], 12)   Log Likelihood               -1310.992
Date:                              Mon, 15 May 2023   AIC                          2627.983
Time:                                      18:01:27   BIC                          2635.581
Sample:                                  04-01-2012   HQIC                         2631.051
                                       - 12-01-2020
Covariance Type:                                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.6791      0.049     13.836      0.000       0.583       0.775
ma.S.L12      -0.4830      0.134     -3.599      0.000      -0.746      -0.220
sigma2      1.171e+11    8.8e-13   1.33e+23      0.000    1.17e+11    1.17e+11
===================================================================================
Ljung-Box (L1) (Q):                   0.04   Jarque-Bera (JB):               133.70
Prob(Q):                              0.84   Prob(JB):                         0.00
Heteroskedasticity (H):               8.93   Skew:                            -0.56
Prob(H) (two-sided):                  0.00   Kurtosis:                         8.77
===================================================================================
```

Figure 7.5: Model Summary

The SARIMA model is used to obtain forecasts on the test set by using the **get_forecast** method with the **steps** parameter set to the length of the test set. The **forecast_mean_1** variable stores the predicted mean values of the forecast. Additionally, the **forecast_ci** variable holds the confidence intervals for the forecast. The predictions on the test data is as follows:

```
2021-01-01    1961550.521
2021-02-01    1789632.193
2021-03-01    1452358.217
2021-04-01    1012043.465
2021-05-01    1231912.339
2021-06-01    1873200.545
2021-07-01    2277150.406
2021-08-01    1472454.531
2021-09-01    1748107.164
2021-10-01    1736793.065
2021-11-01    1643462.027
2021-12-01    1821317.430
2022-01-01    1788482.640
2022-02-01    1672110.039
2022-03-01    1372554.521
2022-04-01     957852.577
2022-05-01    1195113.889
2022-06-01    1848212.470
2022-07-01    2260182.195
2022-08-01    1460932.227
2022-09-01    1740282.916
2022-10-01    1731479.991
2022-11-01    1639854.172
2022-12-01    1818867.508
2023-01-01    1786819.014
2023-02-01    1670980.350
2023-03-01    1371787.403
Freq: MS, Name: predicted_mean, dtype: object
```

Figure 7.6: Predictions on the test data

Next we calculate the root mean squared error (**RMSE**) between the actual values in the test set and the predicted values generated by the SARIMA model. The "**mean_squared_error**" function from the "**sklearn**" library is used to calculate the mean squared error, which is then passed to the "**np.sqrt**" function from the "**numpy**" library to calculate the square root and obtain the RMSE.

```
RMSE: 446817.235
```

Figure 7.7: RMSE calculated

The obtained value of RMSE is slightly higher than expected due to COVID which affected the sales of the company in the month of April-June drastically. This unexpected change caused a difference and therefore affected the value of RMSE.

Finally, we plot the actual, test and predicted values of the time series data.

- **plt.plot(train.index, train, label='Train')**: This plots the training data with the index values on the x-axis and the actual values on the y-axis. The label parameter is used to give a name to this plot in the legend.
- **plt.plot(test.index, test, label='Test')**: This plots the test data with the index values on the x-axis and the actual values on the y-axis. The label parameter is used to give a name to this plot in the legend.
- **plt.plot(test.index, test_pred, label='Predicted')**: This line plots the predicted values with the index values on the x-axis and the predicted values on the y-axis. The label parameter is used to give a name to this plot in the legend.
- **plt.fill_between(forecast_ci.index, forecast_ci.iloc[:, 0], forecast_ci.iloc[:, 1], alpha=0.2):** This line plots a shaded area between the lower and upper bounds of the confidence intervals.
- **plt.legend()**: This adds a legend to the plot, which shows the labels assigned to each plot using the label parameter.
- **plt.show()**: This displays the plot.

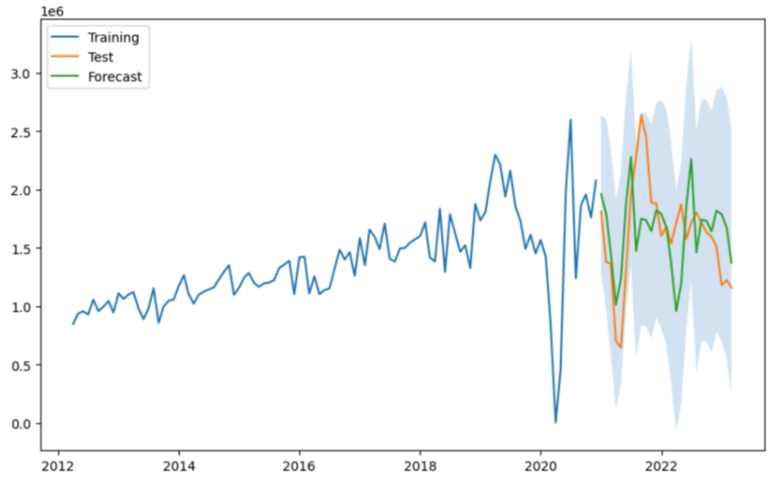


Figure 7.8: Actual vs Forecasted sales

The frequency of the index in the DataFrame (**df**) is explicitly set to 'MS', indicating that the data is monthly. A SARIMAX model is fitted to the entire dataset (**df**) using the selected hyperparameters obtained from the grid search. The model fitting results are stored in the **results** variable. Future dates are generated using the **pd.date_range** function, specifying a start date of '2023-04-01' and an end date of '2025-03-01', with a monthly frequency ('MS'). A DataFrame called **future** is created with the future dates as the index and the same columns as the original dataset.

Forecasts are generated by calling the **forecast** method on the **results** object, passing the number of steps equal to the length of the **future** DataFrame. The forecasted mean values are stored in the **forecast_mean_2** variable.

The confidence intervals for the forecasts are obtained using the **get_forecast** method on the **results** object, and the intervals are stored in the **forecast_ci** variable. Then we print the future forecast.

```
2023-04-01    1030738.293
2023-05-01    1193237.783
2023-06-01    1445836.159
2023-07-01    1795370.118
2023-08-01    1677997.263
2023-09-01    1818750.093
2023-10-01    1755218.638
2023-11-01    1604053.672
2023-12-01    1633704.638
2024-01-01    1454125.374
2024-02-01    1414883.799
2024-03-01    1310155.887
2024-04-01    1144387.808
2024-05-01    1277925.656
2024-06-01    1508942.771
2024-07-01    1842395.086
2024-08-01    1713038.720
2024-09-01    1844861.832
2024-10-01    1774676.245
2024-11-01    1618552.840
2024-12-01    1644508.940
2025-01-01    1462176.384
2025-02-01    1420883.146
2025-03-01    1314626.403
Freq: MS, Name: predicted_mean, dtype: object
```

Figure 7.9: Future Forecast

Next the training data is plotted using **plt.plot(df, label='Training data'),** where **df** represents the original dataset.The forecasted values are plotted using **plt.plot(forecast_mean_2, label='Forecast')**, where **forecast_mean_2** represents the forecasted mean values obtained from the SARIMA model.

A shaded area between the lower and upper bounds of the confidence intervals is filled using **plt.fill_between()**. This adds a visual representation of the uncertainty around the forecasted values. Then we add a legend to the plot using **plt.legend()** to provide clarity on the different elements represented in the plot.

Finally, the plot is displayed using **plt.show()**.

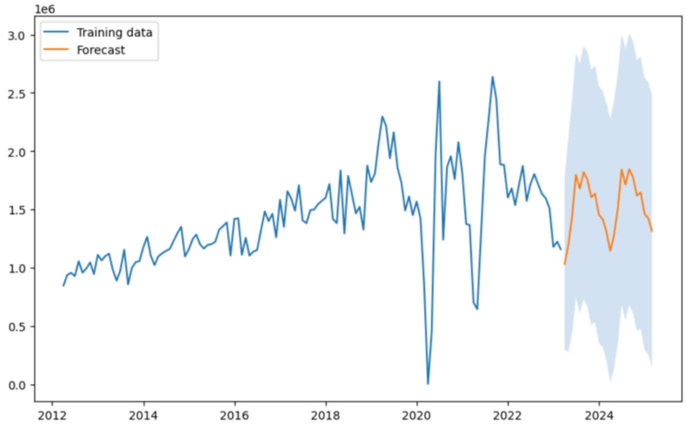


Figure 7.10: Forecast

Comparison between the transportation costs

Old Transportation Cost: Rs. 5141000.0 (As per given by the company)

Optimized Transportation Cost: Rs. 4502283.10

The implementation of optimized transportation routes has resulted in significant cost savings for the company. By switching to the optimized routes, the company can now save up to Rs. **638716.90** that were previously being spent on transportation.

Observed factors

From the optimal shipping routes, it can be observed that the preferrable transportation source is Guna, Madhya Pradesh due to its proximity to most of the destinations. Therefore, based on these results we try to determine whether the company can expand their unit in Guna by analysing their sales. By forecasting the future sales, we observe a consistent result in the future as well.

Recommended Policy

Therefore, considering the above concluded factors we formulate a policy to be recommended:

Through the optimization of shipping routes, significant cost savings can be achieved by the company, enabling the allocation of freed-up funds towards various other expenditures. Additionally, taking into consideration the boom in the future of the textile industry, the company can consider setting up their new units in Guna, Madhya Pradesh.

Chapter – 8 Conclusion

The transportation optimization has successfully reduced the total transportation cost to a significant level. This reduction in cost has been achieved by optimizing the transportation plan, considering the demand and supply constraints of the different routes and facilities.

The sensitivity analysis performed on the project has also provided insights into the impact of changes in the demand and supply constraints on the transportation cost. This information can be used to make future adjustments to the transportation plan based on changes in demand or supply.

Based on the results, the SARIMA model with parameters (1, 1, 0) (0, 1, 1)[12] was found to be the best fit for the time series data. The SARIMA model was applied to the time series data with the selected hyperparameters obtained through a grid search.

The model exhibited accurate forecasts on the test set, as demonstrated by the low root mean squared error (RMSE) value. The forecasted values, along with their confidence intervals, were plotted, visually depicting the model's performance in capturing the underlying patterns.

In conclusion, the SARIMA model was able to effectively model the time series data and generate accurate predictions. The model showcases its effectiveness in forecasting future values, providing valuable insights for decision-making and planning.

References

- Lazzeri, F. (2019). Time Series Forecasting: An Applied Machine Learning Approach.O'REILLY MEDIA.
- McKinney, W. (2017). Python for Data Analysis. O'REILLY MEDIA.
- Kanti Swarup, P. K. (1980). Operations Research. Sultan Chand & Sons.
- Taha, H. A. (2017). Operations research-an introduction (10th ed.). New Delhi: Pearson
- Machine Learning Using Python by Manaranjan Pradhan (Consultant, Indian Institute of Management, Bangalore) and U Dinesh Kumar (Professor, Indian Institute of Management, Bangalore)
- www.w3school.com
- www.geeksforgeeks.org
- www.stackoverflow.com

Appendix

Dataset:

Transportation Data

A	B	C	D	E
S.no	Destination	Madhya Pradesh	Himachal	Demand
1	AHMEDABAD (VINAYAK TRANSPORT CO.)	2047.95	3124.12	10
2	ALWAR (SHIV ROAD LINES)	2564.10	2900.20	21
3	AMRITSAR (SHIV ROAD LINES)	3597.75	3425.25	23
4	BADDI+LUDHIANA (NEW MAHAVEER TRANSPORT CO OF BHARAT)	3148.99	2123.34	36
5	BHILWARA (SHIV ROAD LINES)	1698.30	2635.34	141
6	BHILWARA (VINAYAK TRANSPORT CO.)	1649.61	2376.34	135
7	BHIWANDI (VINAYAK TRANSPORT CO.)	3225.94	3713.34	11
8	COIMBATORE (SAFEXPRESS PVT.LTD.)	20063.21	25341.34	0
9	DELHI (VINAYAK TRANSPORT CO.)	2806.67	3123.56	225
10	DELHI (SHIV ROAD LINES)	2886.68	3562.23	161
11	HYDERABAD (SHIV ROAD LINES)	14790.83	17234.34	22
12	ICD MANDIDEEP (SHRIRAM LOGISTICS)	3204.78	12326.33	23
13	INDORE (SHIV ROAD LINES)	1330.35	3245.21	14
14	KANPUR (SHIV ROAD LINES)	2125.70	2734.21	34
15	KOLKATTA (SHIV ROAD LINES)	6278.59	85253.00	12
16	KOLKATTA (VINAYAK TRANSPORT CO.)	6160.65	7935.00	14
17	LUDHIANA (NEW MAHAVEER TRANSPORT CO OF BHARAT)	2959.61	2456.56	145
18	LUDHIANA (SHIV ROAD LINES)	2998.53	2536.32	68
19	LUDHIANA (VINAYAK TRANSPORT CO.)	2945.71	2378.34	115
20	LUDHIANA (SHANKAR TRANSPORT CO.)	2997.88	2836.45	32
21	NHAVA SHEVA (VINAYAK TRANSPORT CO.)	4016.77	4678.92	9
22	PANIPAT (SHIV ROAD LINES)	3125.00	4253.23	2
23	PATNA (VINAYAK TRANSPORT CO.)	5201.83	5437.89	12
24	PURBA BARDDHAMAN (NEW MAHAVEER TRANSPORT CO OF BHARAT)	6125.40	6924.84	17
25	SANTIPUR-KOLKATTA (SHIV ROAD LINES)	6333.33	7137.83	30
26	SANTIPUR-KOLKATTA (NEW MAHAVEER TRANSPORT CO OF BHARAT)	6110.36	6845.82	72
27	SANTIPUR-KOLKATTA (VINAYAK TRANSPORT CO.)	6439.85	7259.34	35
28	SURAT (VINAYAK TRANSPORT CO.)	2339.24	3035.45	9
29	SURAT (NEW MAHAVEER TRANSPORT CO OF BHARAT)	2259.62	2837.16	10
30	SURAT (SHIV ROAD LINES)	2394.96	3523.65	11

Sales data

Date	Sales
Apr-12	846325
May-12	936390
Jun-12	955998
Jul-12	927638
Aug-12	1055278
Sep-12	957151
Oct-12	993252
Nov-12	1044274
Dec-12	944418
Jan-13	1109952
Feb-13	1061544
Mar-13	1098560
Apr-13	1119838
May-13	982868
Jun-13	888746
Jul-13	977721
Aug-13	1153513
Sep-13	856265
Oct-13	996664
Nov-13	1047395
Dec-13	1055540
Jan-14	1174551
Feb-14	1264892
Mar-14	1104827
Apr-14	1021670
May-14	1096429
Jun-14	1122998
Jul-14	1143295
Aug-14	1160004
Sep-14	1230093
Oct-14	1294944
Nov-14	1350396
Dec-14	1095936
Jan-15	1158935
Feb-15	1245454
Mar-15	1283862
Apr-15	1199826
May-15	1163972
Jun-15	1195077
Jul-15	1201353
Aug-15	1222484
Sep-15	1326430
Oct-15	1355102
Nov-15	1387702
Dec-15	1103119
Jan-16	1418497
Feb-16	1422959
Mar-16	1108458
Apr-16	1255929
May-16	1102615
Jun-16	1137807
Jul-16	1151106
Aug-16	1319142
Sep-16	1483012
Oct-16	1398953
Nov-16	1462012
Dec-16	1260294
Jan-17	1583030
Feb-17	1349930
Mar-17	1655772
Apr-17	1592314
May-17	1488224
Jun-17	1707920
Jul-17	1405586
Aug-17	1381310
Sep-17	1494006
Oct-17	1497881
Nov-17	1544113
Dec-17	1571588
Jan-18	1600184
Feb-18	1717515
Mar-18	1416401
Apr-18	1382049
May-18	1834329
Jun-18	1293637
Jul-18	1787203
Aug-18	1626601
Sep-18	1465166
Oct-18	1522015
Nov-18	1326442
Dec-18	1876095
Jan-19	1734118
Feb-19	1808304
Mar-19	2062186

Apr-19	2296991.8
May-19	2215064.1
Jun-19	1938253.5
Jul-19	2160797.4
Aug-19	1857578.4
Sep-19	1728021.6
Oct-19	1489754.9
Nov-19	1610644.4
Dec-19	1450602.3
Jan-20	1567331.1
Feb-20	1421785.3
Mar-20	855151.6
Apr-20	2040
May-20	453052.7
Jun-20	1974143.7
Jul-20	2598038.9
Aug-20	1239145.2
Sep-20	1864590.94
Oct-20	1956368.7
Nov-20	1759411.7
Dec-20	2076184.2
Jan-21	1810619.52
Feb-21	1374913.3
Mar-21	1365767.7
Apr-21	701655.7
May-21	643814
Jun-21	1303303.8
Jul-21	1958665.4
Aug-21	2283098.5
Sep-21	2637519.4
Oct-21	2445148.2
Nov-21	1885716.2
Dec-21	1881815.1
Jan-22	1600743.48
Feb-22	1680961.4
Mar-22	1535686
Apr-22	1713633.5
May-22	1872992.8
Jun-22	1572877.8
Jul-22	1715496.7
Aug-22	1803435.8
Sep-22	1715301.6
Oct-22	1634049.7
Nov-22	1594191.9
Dec-22	1510030.6
Jan-23	1177283.8
Feb-23	1223766.4
Mar-23	1157640.4

Code for transportation Problem

```
!pip install pulp
from pulp import *
import pandas as pd

# Load data
df = pd.read_excel("/content/dp final.xlsx")
df.head()
df.tail()
Branch = df.columns[2:(len(df.columns)-1)]
Warehouse = list(df["Destination"])
Warehouse=Warehouse[:len(Warehouse)-1]
supply={}
for i in Branch:
  supply[i]=df[i][(len(df[i]))-1]
demand = dict(zip(Warehouse, df['Demand']))
cost={}
for j in Branch:
  cost[j]=dict(zip(Warehouse, df[j][:(len(df[i]))-1]))


# Set problem variable
prob = LpProblem("Transportation", LpMinimize)
routes = [(i, j) for i in Branch for j in Warehouse]

# Decision variable
x = LpVariable.dicts("x", (Branch, Warehouse), 0)

# Objective function
prob += lpSum(x[i][j] * cost[i][j] for (i,j) in routes)
```

```
# Constraints
for j in Warehouse:
    prob += lpSum(x[i][j] for i in Branch) == demand[j]

for i in Branch:
    prob += lpSum(x[i][j] for j in Warehouse) <= supply[i]

# Solve the problem
prob.solve()

# Print the results
print("Status: ", LpStatus[prob.status])
print("\nOptimal Solution:")

print("\nTotal Cost: ", value(prob.objective))
# Create an empty list to hold the data
results = []

# Append the data for each variable to the list
for i in Branch:
    for j in Warehouse:
        results.append({
            'From': i,
            'To': j,
            'Quantity': x[i][j].varValue,
            'Cost': cost[i][j]
        })

# Convert the list to a Pandas dataframe
df_results = pd.DataFrame(results)
```

```
# Create dummy variables for the 'From' column
df_from = pd.get_dummies(df_results['From'], prefix='From')

# Concatenate the original dataframe with the dummy variables
df_results = pd.concat([df_results, df_from], axis=1)

# Drop the original 'From' column
df_results = df_results.drop('From', axis=1)

# Add a column for the total cost for each route
df_results['Total Cost'] = df_results['Quantity'] * df_results['Cost']

# Filter out the rows where the cost is 0
df_results = df_results[df_results['Total Cost'] != 0]

# Reset the index of the dataframe
df_results = df_results.reset_index(drop=True)

# Print the dataframe
df_results
# Sensitivity Analysis
sensitivity_results = []
for name, c in prob.constraints.items():
    sensitivity_results.append({'Variable': name, 'Shadow Price': c.pi})
    for v in prob.variables():
        if v.name in c:
            sensitivity_results.append({'Variable': (v.name, "=", v.varValue), 'Shadow Price': v.dj})

# Convert the sensitivity results to a Pandas dataframe
df_sensitivity = pd.DataFrame(sensitivity_results)
```

```
Destination = df_results['To'].sort_values(ascending=True).reset_index(drop=True)
df_sensitivity = pd.concat([df_sensitivity, Destination], axis =1 )
df_sensitivity = df_sensitivity[df_sensitivity['Shadow Price'] != 0]


# Print the dataframe
df_sensitivity
```

Code for Sales Forecasting

```
!pip install pmdarima
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima.arima import auto_arima
from sklearn.metrics import mean_squared_error
# load the data
df = pd.read_excel('/content/Book2.xlsx', parse_dates=['Date'], index_col='Date')
# plot the time series of the original data
plt.plot(df.index, df['Sales'])
plt.title('Time Series of Sales Data')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.show()
# split data into train and test sets
train_size = int(len(df) * 0.8)
train = df[:train_size]
```

```
test = df[train_size:]
train.head()
test.head()
# create a lagged version of the time series
lagged_data = pd.concat([df['Sales'], df['Sales'].shift(1)], axis=1)
lagged_data.columns = ['Sales', 'Sales_Lag1']

# remove missing values (first row)
lagged_data.dropna(inplace=True)

# plot the scatter plot
plt.scatter(lagged_data['Sales'], lagged_data['Sales_Lag1'])
plt.title('Lag=1 Scatter Plot')
plt.xlabel('Sales (t)')
plt.ylabel('Sales (t-1)')
# create a lagged version of the time series
lagged_data = pd.concat([df['Sales'], df['Sales'].shift(1)], axis=1)
lagged_data.columns = ['Sales', 'Sales_Lag1']

# remove missing values (first row)
lagged_data.dropna(inplace=True)

# plot the scatter plot
plt.scatter(lagged_data['Sales'], lagged_data['Sales_Lag1'])
plt.title('Lag=1 Scatter Plot')
plt.xlabel('Sales (t)')
plt.ylabel('Sales (t-1)')

# add trend line
z = np.polyfit(lagged_data['Sales'], lagged_data['Sales_Lag1'], 1)
p = np.poly1d(z)
```

```
plt.plot(lagged_data['Sales'], p(lagged_data['Sales']), "r--")


plt.show()
# Fit MA model
model = ARIMA(train, order=(0, 0, 1))
results = model.fit()
# Evaluate model on test set
forecast = results.predict(start=len(train), end=len(df)-1, typ='levels')
forecast_mean_1 = pd.Series(forecast, index=test.index)
forecast_mean_1 = forecast_mean_1.apply(lambda x:'%.3f'%x)
# Calculate and print RMSE
mse = mean_squared_error(test, forecast_mean_1)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
from statsmodels.tsa.holtwinters import ExponentialSmoothing
# Fit Exponential Smoothing model
model = ExponentialSmoothing(train, trend='add', seasonal='add', seasonal_periods=12)
results = model.fit()
# Evaluate model on test set
forecast = results.forecast(steps=len(test))
forecast_mean_1 = forecast
forecast_ci = None
forecast_mean_1.apply(lambda x:'%.3f'%x)
# Calculate and print RMSE
mse = mean_squared_error(test, forecast_mean_1)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
# Fit ARIMA model
model = ARIMA(train, order=(1, 1, 1))
results = model.fit()
# Evaluate model on test set
```

```
forecast = results.forecast(steps=len(test))
forecast_mean_1 = forecast[0]
forecast_ci = results.conf_int()
forecast_mean_1 = round(forecast_mean_1, 3)
# Get forecast values
forecast_values = pd.Series(forecast[0], index=test.index)
# Calculate RMSE
mse = mean_squared_error(test, forecast_values)
rmse = np.sqrt(mse)
print('RMSE:', round(rmse, 3))


#fit LSTM model
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from sklearn.metrics import mean_squared_error
# Scale the data
scaler = MinMaxScaler()
train_scaled = scaler.fit_transform(train)
test_scaled = scaler.transform(test)
# Define the number of time steps and features
n_steps = 10
n_features = 1
# Create input and output data for the model
def create_sequences(data, n_steps):
   X = []
   y = []
   for i in range(n_steps, len(data)):
      X.append(data[i-n_steps:i, 0])
      y.append(data[i, 0])
   return np.array(X), np.array(y)
```

```
X_train, y_train = create_sequences(train_scaled, n_steps)
X_test, y_test = create_sequences(test_scaled, n_steps)

# Reshape input data for LSTM
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], n_features))
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], n_features))

# Define the model architecture
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(n_steps, n_features)))
model.add(Dense(1))

# Compile the model
model.compile(optimizer='adam', loss='mse')

# Fit the model to the training data
model.fit(X_train, y_train, epochs=50, batch_size=32)

# Generate predictions on the test data
test_predictions = model.predict(X_test)
test_predictions = scaler.inverse_transform(test_predictions)

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(y_test, test_predictions))
print('RMSE: %.3f' % rmse)

# Perform grid search for SARIMA hyperparameters
stepwise_model = auto_arima(train, start_p=1, start_q=1,
                    max_p=3, max_q=3, m=12,
                    start_P=0, seasonal=True,
```

```
                    d=None, D=1, trace=True,
                    error_action='ignore',
                    suppress_warnings=True,
                    stepwise=True)

# Print the selected hyperparameters
print(stepwise_model.order)
print(stepwise_model.seasonal_order)
# Fit SARIMA model with the selected hyperparameters
model = SARIMAX(train, order=stepwise_model.order,
seasonal_order=stepwise_model.seasonal_order)
results = model.fit()
print(results.summary())
# Evaluate model on test set
forecast = results.get_forecast(steps=len(test))
forecast_mean_1 = forecast.predicted_mean
forecast_ci = forecast.conf_int()
forecast_mean_1.apply(lambda x: '%.3f' % x)
# Calculate RMSE on test set
mse = mean_squared_error(test, forecast_mean_1)
rmse = np.sqrt(mse)
print('RMSE: %.3f' % rmse)
# Plot forecast and actual values
plt.figure(figsize=(10, 6))
plt.plot(train, label='Training')
plt.plot(test, label='Test')
plt.plot(forecast_mean_1, label='Forecast')
plt.fill_between(forecast_ci.index, forecast_ci.iloc[:, 0], forecast_ci.iloc[:, 1], alpha=0.2)
plt.legend(loc='upper left')
plt.show()
# Set the frequency of the index explicitly
df.index.freq = 'MS'
```

```
# Fit SARIMA model using the entire dataset with the selected hyperparameters
model = SARIMAX(df, order=stepwise_model.order,
seasonal_order=stepwise_model.seasonal_order)
results = model.fit()


future_dates = pd.date_range(start='2023-04-01', end='2025-03-01', freq='MS')
future = pd.DataFrame(index=future_dates, columns=df.columns)
forecast = results.forecast(steps=len(future))
forecast_mean_2 = forecast
forecast_ci = results.get_forecast(steps=len(future)).conf_int()


# Print the predicted mean values
print(forecast_mean_2.apply(lambda x: '%.3f' % x))
# Plot forecast and actual values
plt.figure(figsize=(10, 6))
plt.plot(df, label='Training data')
plt.plot(forecast_mean_2, label='Forecast')
plt.fill_between(forecast_ci.index, forecast_ci.iloc[:, 0], forecast_ci.iloc[:, 1], alpha=0.2)
plt.legend(loc='upper left')
plt.show()
```