ORDER DEMAND FORECASTING THROUGH CUSTOMER BEHAVIOR AND SEASONAL PATTERN

PROJECT REPORT

Submitted by

KARTHICK M	**(412419104049)**
TARUN H	**(412419104137)**
SYED ABUTHAHIR A	**(412419104136)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI SAI RAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2023

SRI SAI RAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)

ANNA UNIVERSITY, CHENNAI -600025

BONAFIDE CERTIFICATE

Certified that this project report "**ORDER DEMAND FORECASTING THROUGH CUSTOMER BEHAVIOR AND SEASONAL PATTERN"** is the bonafide work of **KARTHICK M (412419104049), TARUN H (412419104137), SYED ABUTHAHIR (412419104136)**" who carried out the project work under my supervision.

SIGNATURE
Dr. B. SREEDEVI M.Tech., Ph. D
HEAD OF THE DEPARTMENT
Department of Computer Science
and Engineering
Sri Sai Ram Institute of Technology
West Tambaram,
Chennai-600044.

SIGNATURE
SUPERVISOR
Mr. T. PRABAHAR GODWIN JAMES M.Tech.,
ASSISTANT PROFESSOR
Department of Computer Science
and Engineering
Sri Sai Ram Institute of Technology
West Tambaram,
Chennai-600044.

Submitted for University Project Examination held on …………………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A successful man is one who can lay a firm foundation with the bricks others have thrown at him. —*David Brinkley*

Such a successful personality is our beloved Founder Chairman, **Thiru.MJF.Ln. LEO MUTHU.** At first, we express our sincere gratitude to our beloved chairman through prayers, who in the form of a guiding star has spread his wings of external support with immortal blessings.

We express our gratitude to our Chairman and CEO **Mr. J.SAI PRAKASH LEOMUTHU** and our Trustee **Mrs. J. SHARMILA RAJA** for their constant encouragement for completing this project.

We express our sincere thanks to our beloved Principal, **Dr. K. PALANIKUMAR** for having given us spontaneous and whole hearted encouragement for completing this project.

We are indebted to our HEAD OF THE DEPARTMENT **Dr. B. SREEDEVI** for her support during the entire course of this project work.

We express our gratitude and sincere thanks to our guide **Mr. T. PRABAHAR GODWIN JAMES** for his valuable suggestions and constant encouragement for successful completion of this project.

Our sincere thanks to our project coordinator **Ms. P. SUGANTHI** for her kind support in bringing out this project.

We thank all the teaching and non-teaching staff members of the Department of Computer Science and Engineering and all others who contributed directly or indirectly for the successful completion of the project.

TABLE OF CONTENTS

ABSTRACT

The Bullwhip Effect is a significant problem that many supply chains face, as even minor fluctuations in customer demand can cause a cascade of effects that can result in production and inventory imbalances, ultimately leading to a decline in the supply chain's effectiveness and profitability. One of the primary drivers of the Bullwhip Effect is the lack of communication and information-sharing among different supply chain participants, which can result in misunderstandings and incorrect readings of demand signals. To combat the Bullwhip Effect and improve supply chain efficiency and profitability, it is essential to employ efficient forecasting techniques that can accurately predict future demand. The deep learning model proposed in this study uses data from the previous seven years to forecast product demand at the retailer level by examining a range of variables. Specifically, the model utilizes the RNN-LSTM model, which is well-suited to analyzing sequential data, such as historical sales figures. By utilizing this deep learning model, supply chain participants can anticipate changes in demand and adjust their production and inventory levels accordingly, helping to prevent imbalances and reduce the impact of the Bullwhip Effect. Furthermore, accurate forecasting can help retailers reduce the amount of excess inventory they hold, minimizing the costs associated with storing and maintaining surplus goods. This can have a significant positive impact on a company's bottom line and overall profitability. In conclusion, the Bullwhip Effect is a major challenge that can severely impact the effectiveness and profitability of supply chains. To address this issue, accurate demand forecasting is crucial, and the deep learning model proposed in this study represents a promising solution. By leveraging advanced technologies such as RNN-LSTM models, supply chain participants can gain valuable insights into future demand patterns and take proactive measures to optimize their operations and increase profitability.

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATION
LSTM	Long-Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
NumPy	Numerical Python
Pandas	Pannel Data
TF	TensorFlow
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SUPPLY CHAIN MANAGEMENT

Supply chain management is the coordination and management of all activities involved in the creation and delivery of products and services, from the procurement of raw materials to the delivery of finished goods to customers. It is a crucial aspect of modern business operations, as it enables companies to optimize their resources, reduce costs, and improve customer satisfaction.

A typical supply chain includes various stages, such as procurement, production, logistics, warehousing, and distribution. Effective supply chain management requires companies to have a clear understanding of each stage of the process and to optimize them for maximum efficiency and effectiveness.

The key components of supply chain management include planning, sourcing, making, delivering, and returning. Planning involves forecasting demand and creating a strategy for meeting that demand. Sourcing involves selecting suppliers and negotiating contracts. Making involves manufacturing the product or providing the service. Delivering involves logistics and transportation of the product to the customer. Returning involves managing returns and recycling or disposing of products.

Supply chain management is also impacted by various external factors, such as global market trends, regulations, and consumer preferences. As a result, companies must be flexible and adaptable in their supply chain management practices to remain competitive in today's fast-paced business environment.

Overall, effective supply chain management is critical for companies to remain competitive and meet the needs of their customers. It requires a coordinated effort across all stages of the supply chain, as well as a commitment to continuous improvement and innovation.

1.1.1 CHALLENGES IN SCM

The following are the different kind of challenges that is faced by the supply chain network which can disrupt its efficiency and effectiveness.

- Lack of visibility
- Inventory management

- Supplier Management
- Transportation and Logistics
- Demand visibility
- Risk management
- Data management

The above-mentioned challenges can eventually lead to a great impact on an organization's performance and that will be resulting in a consequence that an organization can suffer such as increased inventory cost, decreased customer satisfaction, and decreased efficiency such as resulting in delays, forecast error and other inefficiencies, reduced revenue because of out of stock in the inventory which cannot be able to meet the consumer demand and increased risk due to natural disasters the product could not be delivered on time to the consumer.

1.1.2 WHAT IS DEMAND FORECASTING

Demand forecasting is a crucial aspect of business planning that helps organizations prepare for the future by predicting the demand for their products or services. The process involves analyzing various factors, including historical data, market trends, and external factors that may impact demand. The objective of demand forecasting is to provide businesses with the information they need to make informed decisions about production, inventory management, pricing, and marketing.

Demand forecasting is essential for businesses to operate efficiently and effectively. Accurate forecasts enable companies to optimize their operations and minimize costs by ensuring they have the right products, in the right place, at the right time, and at the right price to meet customer demand. Businesses that fail to forecast demand correctly may suffer from overstocking or understocking, which can lead to financial losses, missed opportunities, and damage to their reputation.

There are several methods used in demand forecasting, including quantitative and qualitative methods. Quantitative methods rely on statistical data to generate forecasts, while qualitative methods involve human judgment and experience to create predictions. Some of the popular quantitative methods include time-series analysis, regression analysis, and econometric modeling. These methods analyze historical data to identify patterns and trends that can be used to predict future demand. Qualitative methods may include expert opinion, surveys, and focus groups, among others. The choice of method often depends on the type of data available and the level of accuracy required for the forecast.

Time-series analysis is a quantitative method that involves analyzing historical data to identify patterns and trends in demand. This method assumes that the past behavior of demand is a good indicator of future demand. The analysis can be done using different statistical techniques such as moving averages, exponential smoothing, and trend analysis. Moving averages involve calculating the average demand for a specific period, and this average is used to forecast future demand. Exponential smoothing uses a weighted average of past demand, with more weight given to recent data. Trend analysis involves identifying and extrapolating trends in historical data to make forecasts.

Regression analysis is another quantitative method used in demand forecasting. This method identifies the relationship between demand and other factors such as price, advertising, and seasonality. Regression analysis involves fitting a mathematical equation to the historical data to create a model that can be used to predict future demand. The accuracy of the forecast depends on the quality of the data and the appropriateness of the model.

Econometric modeling is a complex quantitative method that combines statistical techniques and economic theory to forecast demand. This method considers the impact of various external factors such as economic indicators, weather patterns, and political events on demand. Econometric models can be customized to fit specific industries and product categories, and they are particularly useful for long-term forecasting.

Qualitative methods involve using expert opinion, surveys, and focus groups to create predictions. These methods are particularly useful when historical data is unavailable or when there are significant changes in market conditions that may affect demand. Expert opinion involves gathering information from industry experts to make predictions about future demand. Surveys and focus groups involve gathering feedback from customers and stakeholders about their preferences and behaviors.

In conclusion, demand forecasting is a critical process that enables businesses to plan their operations effectively and efficiently. Accurate forecasting can help companies optimize their operations, minimize costs, and remain competitive in their markets. Businesses that rely on inaccurate or unreliable demand forecasting may suffer from overstocking or understocking, leading to financial losses and missed opportunities. The choice of method depends on the type of data available and the level of accuracy required for the forecast. Therefore, businesses need to choose the most appropriate method to generate reliable forecasts and make informed decisions.

1.1.3 APPLICATIONS OF DEMAND FORECASTING

Demand forecasting is an essential tool for businesses to predict future consumer demand and optimize their operations accordingly. By analyzing historical data and identifying trends, businesses can estimate the level of demand for their products or services and plan their production, inventory, and marketing strategies accordingly. This information can be used to make informed decisions about pricing, promotions, and resource allocation, which can ultimately lead to increased profitability and customer satisfaction. Additionally, demand forecasting can help businesses identify potential bottlenecks in their supply chain and make adjustments to prevent stockouts or overstocking. Overall, demand forecasting is a crucial aspect of business planning and can provide valuable insights to help businesses stay competitive and meet the needs of their customers.

1.1.4 CUSTOMER BEHAVIOUR AND SEASONAL PATTERN

Customer behavior refers to the actions and decisions made by customers when purchasing products or services. It includes factors such as preferences, needs, habits, and purchasing patterns. Customer behavior is a crucial aspect of demand forecasting, as it helps businesses understand how customers interact with their products or services, and how this interaction may change over time.

Seasonality pattern, on the other hand, refers to the cyclic variations in demand for a product or service that occur at regular intervals throughout the year. Seasonality patterns are common in industries such as retail, hospitality, and tourism, where demand fluctuates based on the time of year, holidays, and other seasonal events. Seasonality patterns are important to consider in demand forecasting, as they can significantly impact sales and inventory management.

In demand forecasting, understanding customer behavior and seasonality patterns is critical to developing accurate and effective forecasts. Businesses can use data on customer behavior and historical sales patterns to identify trends and patterns, which can be used to predict future demand. For example, by analyzing customer behavior and seasonality patterns, a retailer may be able to anticipate increased demand for summer clothing in the months leading up to summer, and adjust their inventory and marketing strategies accordingly.

Overall, customer behavior and seasonality patterns are two important factors that should be considered in any demand forecasting process, as they can have a significant impact on the accuracy and effectiveness of the forecast.

1.2 OVERVIEW OF THE PROJECT

Demand forecasting is the process of predicting the future demand for a product or service. It is a critical element of any business strategy as it helps companies to optimize their production, inventory, pricing, and marketing decisions. Accurate demand forecasting enables businesses to avoid stockouts, minimize waste, improve customer satisfaction, and increase profitability.

Demand forecasting can be done using qualitative or quantitative methods. Qualitative methods involve collecting data through surveys, expert opinion, and market research to gain insight into consumer behavior and preferences. Quantitative methods use statistical models to analyze historical data on sales, pricing, promotions, and other relevant factors to forecast future demand.

There are several techniques and tools used for demand forecasting, including regression analysis, time series analysis, neural networks, and artificial intelligence. The choice of method depends on the complexity of the market, the type and availability of data, and the level of accuracy required.

Overall, demand forecasting is a crucial process that helps businesses make informed decisions and stay competitive in their respective markets.

1.3 ARCHITECTURE OF DEMAND FORECASTING

The general architecture diagram basically shows how the model was trained under the given dataset to predict the future demand for the products so that the manufacturing firm or any other supply chain partners can be able to place the order only the customer demands this will further give many other benefits like optimizing the inventory to avoid extra storage of the inventory items so that the unwanted spaces can be utilized by the some other products which are currently demanded by the customers. The following will describe the detailed information about each and every step in the architecture diagram.

- **Input Dataset:** The input dataset consists of fields like product code, warehouse name, product category, date of demand for the product, and order demand which says the count of orders for the specific product on that date.
- **Data Preprocessing:** This step is done to prepare the raw data into further analysis which involves cleaning of data, transforming and formatting of raw data into more usable and understandable form. Data preprocessing is specifically done to ensure that the raw data is accurate, consistent and relavent to the analysis.
- **Analyzing the data based on multiple factors:** The factors will include analysis based on the warehouse which shows detailed information about the demand for the product at the specific warehouse on each or monthly basis if we need detailed information about the demand of the product, analysis based on product category which basically separseparatesproducts into a category which comes under that based on that the analysis will be performed, analysis on the monthly and yearly basis for the given dataset.
- **LSTM Model:** LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that is commonly used in deep learning for time-series data analysis and prediction. It is particularly useful when dealing with

sequences of data that have long-term dependencies, such as speech recognition or natural language processing.

- **Visualization of data:** After optimizing the model for better performance the next step is to visualize the data for the product demand on a monthly basis or yearly basis this will give a clear picture of the demand graph for each and every product.

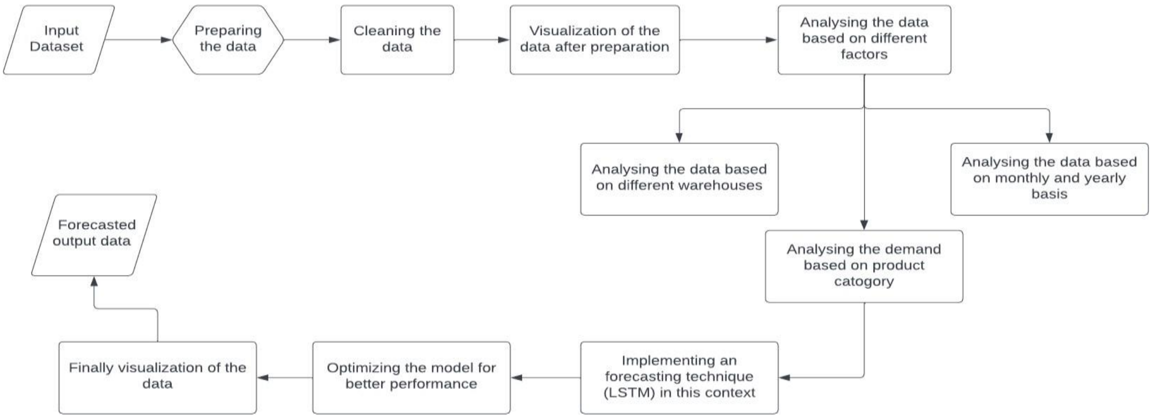


Fig 1.1 Architecture Of Demand Forecasting

1.4 SCOPE OF THE PROJECT

The scope of demand forecasting refers to the range of applications and areas where demand forecasting is useful and relevant. Demand forecasting is an important tool for businesses and organizations to plan their operations and make informed decisions about production, inventory, marketing, and resource allocation. The scope of the demand forecasting includes the following areas:

❖ **Production Planning:** Demand forecasting helps businesses to estimate the future demand for their products and plan their production accordingly.

❖ **Inventory Management:** Demand forecasting helps businesses to determine the optimal level of inventory to maintain based on expected future demand.

❖ **Sales And Marketing:** Demand forecasting helps businesses to plan their sales and marketing strategies, such as pricing, promotions, and advertising, based on expected future demand.

❖ **Resource Allocation:** Demand forecasting helps businesses to allocate their resources, such as labor and capital, in a more efficient and effective manner based on expected future demand.

❖ **Financial Planning:** Demand forecasting helps businesses to forecast their future revenue and cash flows, which is important for financial planning and budgeting.

❖ **Supply Chain Management:** Demand forecasting helps businesses to manage their supply chain, including procurement, transportation, and logistics, based onexpected future demand.

Overall, the scope of demand forecasting is quite broad and includes a range of applications across different industries and sectors.

1.5 OBJECTIVE

The objective of demand forecasting is to estimate the future demand for a product or service. The primary goal is to provide decision-makers with the information they need to make informed decisions about production, inventory, marketing, and resource allocation. The specific objectives of demand forecasting include:

✓ **To estimate the future demand for a product or service:** Demand forecasting aims to provide accurate estimates of the future demand for a product or service so that businesses can plan their operations accordingly.

✓ **To improve production planning:** By forecasting demand, businesses can plan their production schedules to ensure that they have enough inventory to meet customer demand.

✓ **To optimize inventory management:** Demand forecasting helps businesses to determine the optimal level of inventory to maintain based on expected future demand, which can help to reduce costs and minimize the risk of stockouts.

✓ **To support sales and marketing decisions:** Demand forecasting provides valuable information that can help businesses to plan their sales and marketing strategies, such as pricing, promotions, and advertising.

✓ **To facilitate resource allocation:** By forecasting demand, businesses can allocate their resources, such as labor and capital, in a more efficient and effective manner based on expected future demand.

- ✓ **To support financial planning and budgeting:** Demand forecasting helps businesses to forecast their future revenue and cash flows, which is important for financial planning and budgeting.

Overall, the objective of demand forecasting is to provide decision-makers with the information they need to make informed decisions about the future of their business.

CHAPTER 2

LITERATURE SURVEY

2.1 Solar Radiation Forecasting Based on the Hybrid CNN-CatBoost Model.

In 2023, Hyojeoung Kim, Sujin Park, Hee-Jun Park, Heung-Gu Son, and Sahn Kim presented Solar Radiation Forecasting using the Hybrid CNN-CatBoost Model. This paper compared the CatBoost machine learning and CNN deep learning model and presented it as a single model CNN-CatBoost hybrid model prediction method that gives better performance. They also noticed that the accuracy changed when adding wind speed and precipitation to the hybrid model. Hyojeoung Kim [2023] proposed a solution for predicting solar radiation which will resolve the issues in solar energy due to climate change.

2.2 Customer Order Behavior Classification Via Convolutional Neural Network in the Semi-Conductor Industry.

In 2022, Marc Ratusny, Maximilian Schiffer, and Hans Ehm presented Customer Order Behavior Classification via Convolutional Neural Network in the Semi-Conductor Industry. This paper discusses the development of a framework where they utilize data enrichment via synthetical training samples, Integrating synthetically generated data into the training phase allowed them to strengthen the inclusion of rare pattern variants that were identified during the initial analysis. Actual customer data is used to benchmark the performance of the framework and it shows that the baseline CNN approach outperforms all available state-of-the-art benchmark models.

2.3 Profit Prediction Using ARIMA, SARIMA, and LSTM Models in Time Series Forecasting: A Comparison.

In 2022, Uppala Meena Sirisha, Manjula C. Belavagi, and Girija Attigeri presented Profit Prediction Using ARIMA, SARIMA, and LSTM Models in Time Series Forecasting: A Comparison. In this paper they studied the statistical methods-Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) models, as well as the deep learning method, Long Short-Term Memory (LSTM) Neural Network model. The models were fitted and used to predict profit on test data, resulting in accuracies of approximately 93.84% (ARIMA), 94.378% (SARIMA), and 97.01% (LSTM). Forecasts for the next 5 years were made, and the results show that the LSTM method outperforms both statistical models in creating the best model.

2.4 Retail Demand Forecasting using CNN- LSTM Model

In 2022, Nithin Soundar S J presented Retail Demand Forecasting Using CNN-LSMT Model. In this paper, they proposed a solution using a CNN-LSTM model to forecast retail demand. Equipped with the Swish Activation Function it works better than the traditional ReLU (Rectified Linear Unit). Data from 10 stores each consisting of 50 items are taken as input. The experiment results suggest using CNN- LSTM Model as it has considerably lower RMSE (Root Mean-Squared Error).

2.5 Sales Forecasting of Retail Stores using Machine Learning Techniques.

In 2018, Akshay Krishna, Akhilesh V, Animikh Aich, Chetana Hegde presented Sales Forecasting of Retail Stores using Machine Learning Techniques. In this paper, they have implemented normal regression techniques and as well as boosting techniques in our approach and have found that the boosting algorithms have better results than the regular regression algorithms. They observed that the AdaBoost algorithm has the highest RMSE value of 1350.72 and the algorithm with the least RMSE value is GradientBoost having 1088.64. It is concluded that without proper hyperparameter tuning the AdaBoost algorithm won't be able to perform as expected and the performance deteriorates.

2.6 Information Sharing in the Supply Chains of Products With Seasonal Demand.

In 2017, Yeu-Shiang Huang, Chia-Hsien Ho, and Chih-Chiang Fang presented Information Sharing in the Supply Chains of Products With Seasonal Demand. In this paper, they considers a two-echelon supply chain with seasonal consumer demand, in which the impacts of the degree of information sharing on the supplier’s profits are investigated. Since the variance in the supplier’s inventory would marginally decrease as the degree of information sharing increases, the benefits gained by the supplier due to information sharing are thus a convex function. This can be used to obtain the optimal degree of information sharing with the aim of maximizing profits,the seasonal demand is described by a SARMA time series model. The results of sensitivity analyses show that the correlations of demand for successive periods and estimation errors would both have great effects on the benefits gained by information sharing.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROBLEM DEFINITION

The goal of demand forecasting is to predict the amount of a product or service that customers will buy during a specific period, usually ranging from weeks to months or years in advance. This prediction is essential for businesses to make informed decisions about production planning, inventory management, pricing, and marketing strategies. To address this problem, businesses need to collect and analyze historical sales data, as well as external data sources such as market research, industry reports, and social media trends. They must also use statistical and machine learning techniques to identify patterns and trends in the data, and build models that can forecast future demand with reasonable accuracy. The success of demand forecasting depends on the quality of the data, the accuracy of the models, and the ability of businesses to adapt to changes in market conditions. Effective demand forecasting can help businesses optimize their operations, reduce costs, improve customer satisfaction, and gain a competitive advantage in the marketplace.

3.2 EXISTING SYSTEM

The existing system proposed uses the CNN-LSTM model with an activation function as relu to predict the future demand which is placed by the customers which takes the past sales history of the product and outputs the demand raised by the customers in the future if the changes happened dynamically it will fail to predict the output in those scenarios.

3.2.1 LIMITATIONS OF THE EXISTING SYSTEM

- Inaccurate forecasted data
- Eventually leads to great loss for the firm or organization.
- Only work for passed input data.
- This technique will fail if the demand changes dynamically.

3.3 PROPOSED SYSTEM

In today's dynamic business environment, accurate forecasting of product demand is crucial for organizations to remain competitive and meet customer needs.

Implementing machine learning techniques such as the RNN-LSTM model can significantly improve the accuracy of product demand forecasts, providing businesses with valuable insights to optimize their supply chain operations and reduce costs.

The RNN-LSTM model is particularly useful for forecasting demand in time-series data because it can capture long-term dependencies and patterns in the data. By training the model on historical sales data, businesses can predict future demand with higher accuracy, allowing them to make better decisions about inventory management, production planning, and distribution.

Reducing the bullwhip effect in the supply chain is another critical aspect of effective demand forecasting. The bullwhip effect refers to the amplification of demand variability as it moves up the supply chain, leading to inefficiencies and increased costs. By implementing accurate demand forecasting techniques and sharing this information with supply chain partners, businesses can reduce the bullwhip effect and improve overall supply chain performance.

Encouraging cross-border data flow is another essential factor in effective demand forecasting. With the increasing globalization of business, data sharing across borders is essential to ensure effective communication and collaboration between supply chain partners. By promoting cross-border data flow, businesses can access valuable data sources and insights, helping to improve the accuracy of demand forecasts and optimize supply chain operations.

In conclusion, implementing machine learning techniques such as the RNN-LSTM model can significantly improve the accuracy of product demand forecasting, helping businesses to make better decisions about inventory management, production planning, and distribution. By reducing the bullwhip effect in the supply chain and encouraging cross-border data flow, organizations can further improve the efficiency and effectiveness of their supply chain operations, leading to increased competitiveness and customer satisfaction.

3.3.1 ADVANTAGES OF PROPOSED SYSTEM

- ✓ **Increased accuracy:** Machine learning models such as RNN-LSTM can significantly improve the accuracy of demand forecasting by analyzing historical data and identifying patterns and trends that are difficult for humans to detect.
- ✓ **Real-time forecasting:** By continuously updating the model with new data, businesses can get real-time demand forecasts, allowing them to respond quickly to changes in market conditions.
- ✓ **Reduced costs:** Accurate demand forecasting can help businesses optimize their inventory levels, production planning, and distribution, reducing costs associated with excess inventory and stockouts.
- ✓ **Improved customer satisfaction**: Accurate demand forecasting can help businesses meet customer demand more effectively, leading to improved customer satisfaction and loyalty.
- ✓ **Innovations in data sharing:** Encouraging cross-border data flow can lead to innovations in data sharing and collaboration, allowing businesses to access valuable data sources and insights from across the globe.
- ✓ **Improved supply chain performance:** By reducing the bullwhip effect in the supply chain, businesses can improve overall supply chain performance, leading to increased efficiency and reduced costs.
- ✓ **Strategic decision making:** Accurate demand forecasting can help businesses make better strategic decisions about production, inventory management, and distribution, leading to improved competitiveness and long-term growth.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

Environment	:Google Colab
Operating System	:Windows 7 or later
Python Runtime	:version 3.8 or above
Libraries	: Numpy, pandas, Matplotlib, Keras, seaborn, sklearn.

4.2 HARDWARE REQUIREMENTS

Processor	: Intel(R)core(TM)i5-2410M CPU@2.30GHz Processor
Speed	:2.30 GHz
Operating System	: 64-bit operating system
RAM	:4 GB RAM

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 GOOGLE COLAB

Google Colab, short for Google Collaboratory, is an online development environment that enables users to create and run Python notebooks in their web browser. This cloud-based platform provides a range of computing resources, including CPUs, GPUs, and TPUs, that allow users to run complex and resource-intensive code with ease. With Google Colab, users can write, execute, and collaborate on Python code in real-time, and share their work with others through a simple link.

One of the key features of Google Colab is its integration with Google Drive, which allows users to store and access their notebooks and data files in the cloud. This makes it easy to work on projects from any device, anywhere, and collaborate with others in real-time. Additionally, Google Colab offers a range of pre-installed libraries and tools, such as TensorFlow and Scikit-Learn, making it an ideal platform for machine learning and data analysis projects.

Another significant advantage of Google Colab is its free access to powerful computing resources. Users can take advantage of GPUs and TPUs, which provide significant speedups in training and running machine learning models. This enables users to run complex computations that would otherwise require expensive hardware and software.

In summary, Google Colab is an accessible and versatile tool that offers a wide range of features and capabilities for Python developers. Whether you are a beginner or an experienced developer, Google Colab provides a powerful and flexible environment to develop, collaborate, and experiment with Python code.

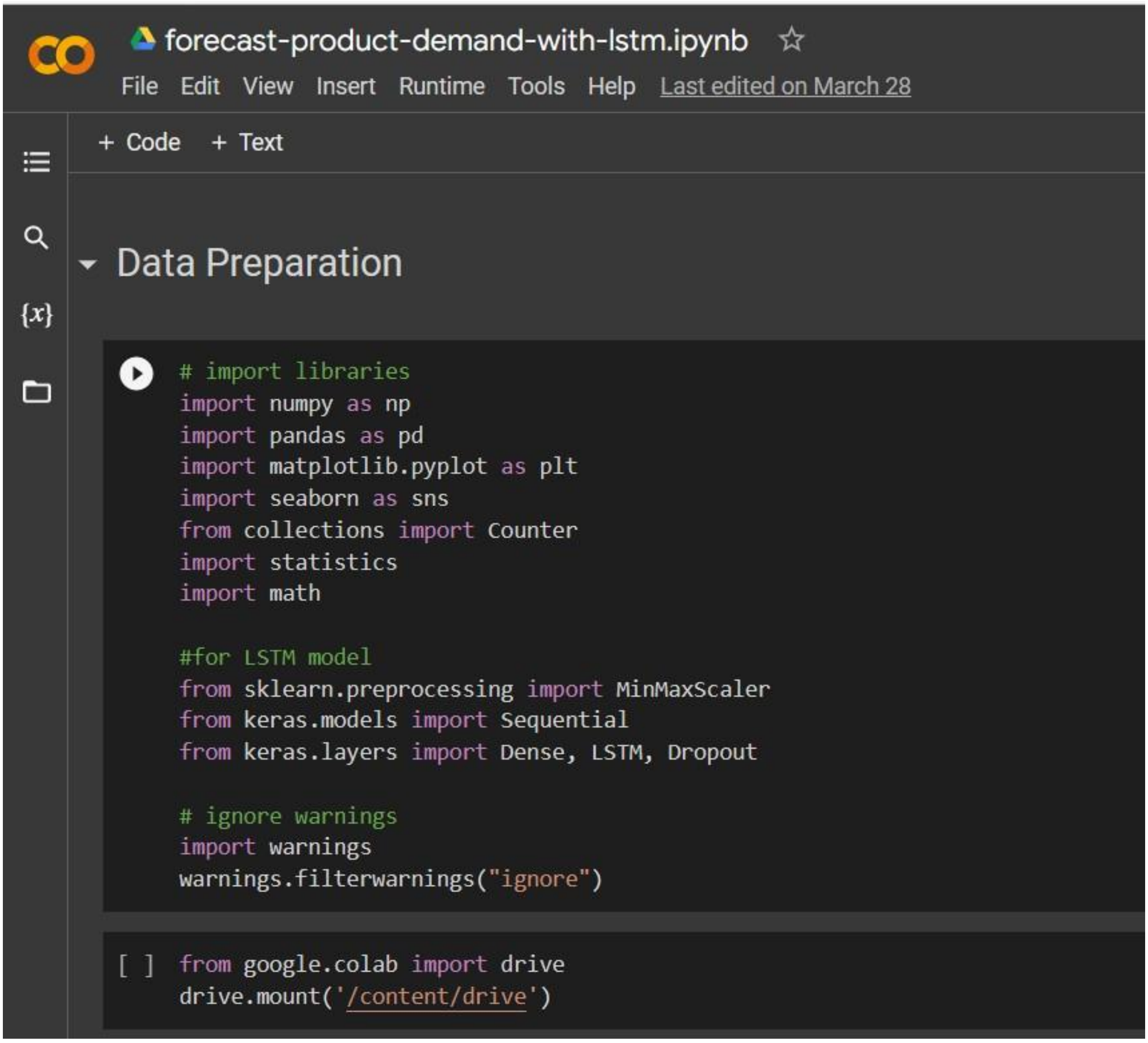


Fig 5.1 Google Colab User Interface

5.2 PYTHON RUNTIME ENVIRONMENT

Python runtime environment refers to the environment in which Python code is executed. It includes the Python interpreter, standard libraries, and any additional libraries or modules that may be required for the code to run. The runtime environment is responsible for executing the code and providing access to system resources such as the file system, network, and user interfaces.

The Python interpreter is the core component of the runtime environment. It reads the Python code and executes it, translating it into machine-readable instructions that can be executed by the computer's processor. Standard libraries are included with the Python installation and provide commonly used functionality such as file input/output, regular expressions, and network communication.

Additional libraries and modules can be installed to extend the functionality of the runtime environment. These may include third-party libraries, custom modules, or packages. The Python runtime environment can be customized to suit the needs of the developer or the specific application.

CHAPTER 6

SYSTEM DESIGN

6.1 ARCHITECTURE

The UI for the application is made using the *tkinter* package for python. Various modules are created to undergo individual tasks. Firstly, there is a timer module which countdowns the time in decrements and notifies when the Shift has ended. Once the Shift has ended, an automatic email is sent to respective authorities. The email consists of an excel file which contains the Shift details. The Automatic Mail sending module is made possible using the *smtplib* and *email* package available in python.

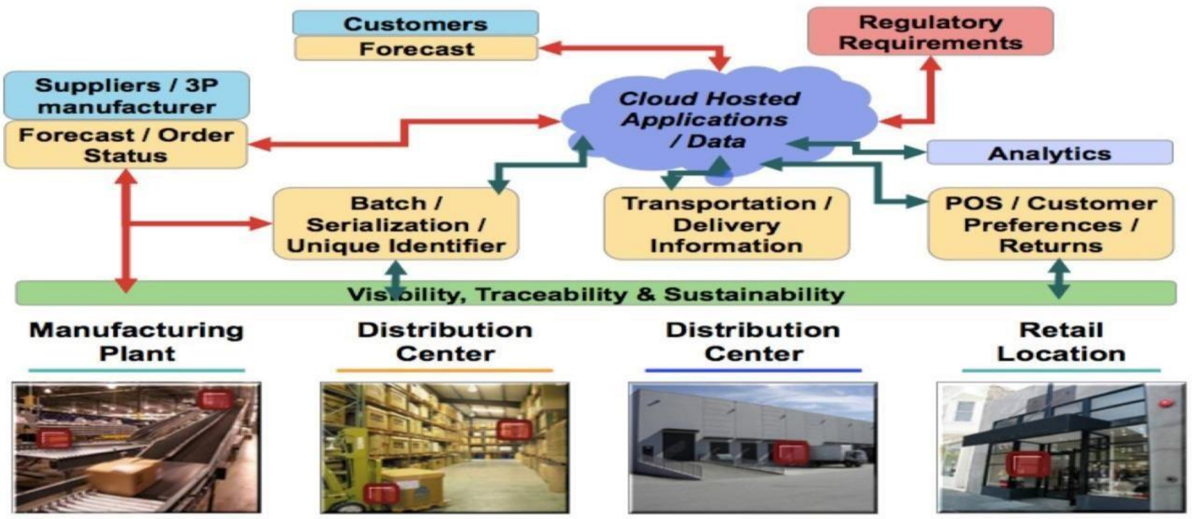


Fig 6.1 System Architecture

In case, the report of the shift is required in between the shift/before the shift ends, there is a Generate Report button. This button calls the module which is responsible for generating the report at the instance when it is clicked. The shift details are stored in an excel file using the pandas package. The excel file is named as the date and time at which the report was generated.

In order to get the live updates of the good and bad count from the Manufacturing Unit to the Assembly Unit, the values are published to a MQTT server online. In this case, we have used HiveMQ MQTT server. Using the paho-mqtt package in python, we would be able to call the necessary functions and publish the required values to the server. Simultaneously, the Assembly Unit would be receiving the values by subscribing to the respective topic in which it is getting published. Again, these values are displayed in the Assembly Unit application with tkinter being its base UI.

6.2 MODULES

- Data Preparation
- Exploratory Data Analysis and Visualization
- Forecasting Order demand with LSTM Model
- Build LSTM Model
- Optimize the model for better accuracy

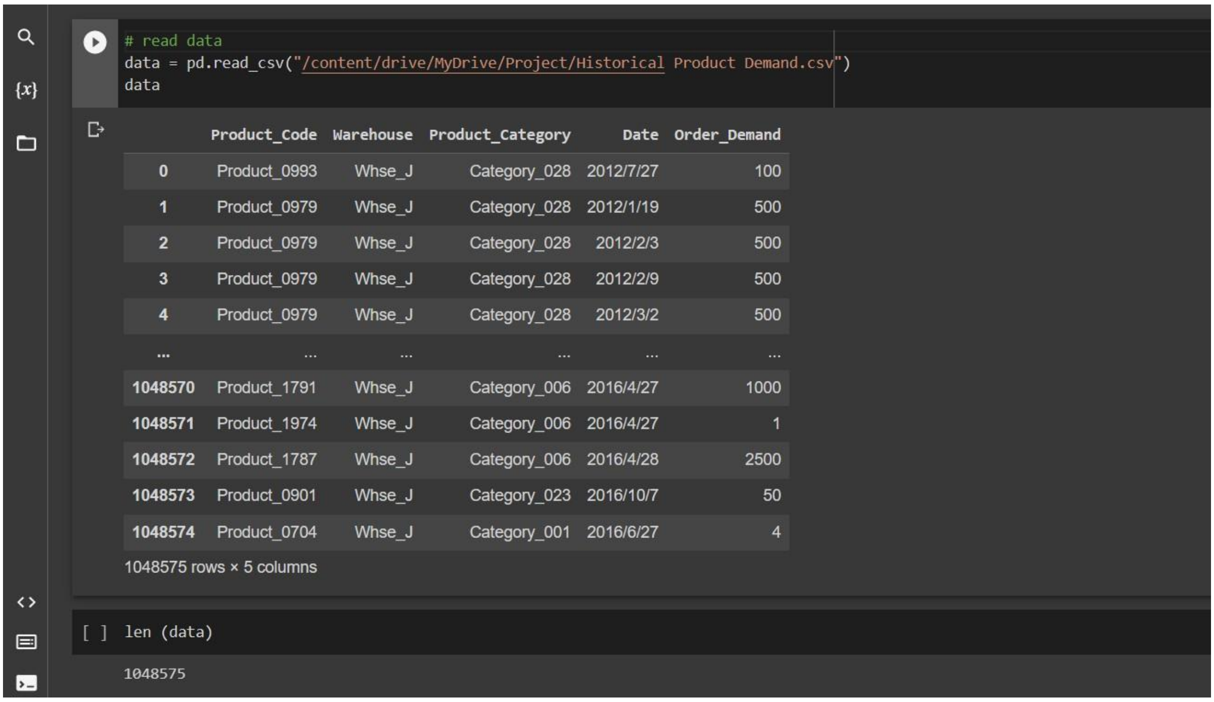
6.2.1 DATA PREPARATION

Data preparation is the process of collecting, cleaning, and organizing data to make it suitable for analysis. In the context of forecasting product demand using machine learning techniques such as the RNN-LSTM model, data preparation is a crucial step in ensuring the accuracy and reliability of the model's predictions.

Data preparation involves several steps, including data collection, data cleaning, data transformation, and data integration. In the context of demand forecasting, data collection involves gathering historical sales data, customer data, and market data from various sources. Data cleaning involves identifying and correcting errors in the data, such as missing values, outliers, and inconsistencies.

Data transformation involves converting the data into a suitable format for analysis, such as aggregating sales data by time period, and normalizing data to eliminate differences in scale. Data integration involves combining data from multiple sources to create a unified dataset for analysis.

Data preparation is a time-consuming process that requires careful attention to detail to ensure the accuracy and completeness of the data. However, it is a critical step in developing accurate and reliable forecasting models, as the quality of the data can have a significant impact on the accuracy of the model's predictions.



```
# read data
data = pd.read_csv("/content/drive/MyDrive/Project/Historical Product Demand.csv")
data
```

	Product_Code	Warehouse	Product_Category	Date	Order_Demand
0	Product_0993	Whse_J	Category_028	2012/7/27	100
1	Product_0979	Whse_J	Category_028	2012/1/19	500
2	Product_0979	Whse_J	Category_028	2012/2/3	500
3	Product_0979	Whse_J	Category_028	2012/2/9	500
4	Product_0979	Whse_J	Category_028	2012/3/2	500
...	...	...	...	...	...
1048570	Product_1791	Whse_J	Category_006	2016/4/27	1000
1048571	Product_1974	Whse_J	Category_006	2016/4/27	1
1048572	Product_1787	Whse_J	Category_006	2016/4/28	2500
1048573	Product_0901	Whse_J	Category_023	2016/10/7	50
1048574	Product_0704	Whse_J	Category_001	2016/6/27	4

1048575 rows × 5 columns

```
[ ] len (data)
1048575
```

Fig 6.2. Data Preparation

6.2.2 Exploratory Data Analysis And Visualization

Exploratory data analysis (EDA) and visualization are important steps in the data preparation process. EDA involves examining and summarizing data to gain insights into its characteristics, including its distribution, central tendency, variability, and relationships between variables.

Visualization is a powerful tool for EDA that enables analysts to explore data visually and identify patterns, trends, and relationships that may not be apparent in the raw data. Common visualization techniques used in EDA include scatter plots, histograms, box plots, and heat maps.

EDA and visualization are critical for developing accurate and reliable forecasting models, as they provide insights into the underlying structure of the data and highlight potential issues that may affect the accuracy of the model's predictions. For example, EDA may reveal seasonal trends or anomalies in the data that need to be taken into account in the forecasting model.

Visualization can also help stakeholders understand the data and its implications, facilitating communication and collaboration between different teams and departments within an organization.

In summary, EDA and visualization are important steps in the data preparation process, enabling analysts to gain insights into data characteristics and identify potential issues that may affect the accuracy of forecasting models. Visualization is a powerful tool for communicating insights and facilitating collaboration, making it an essential part of the data analysis process.

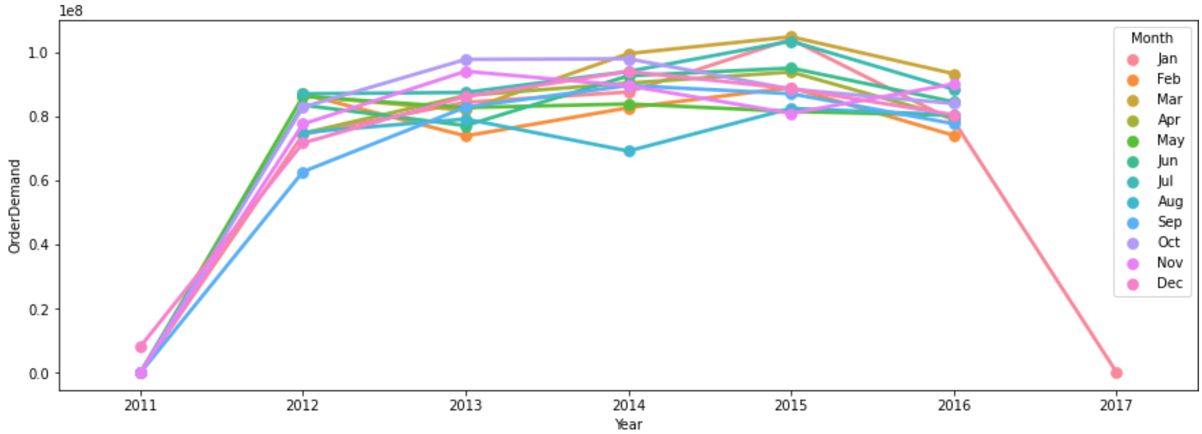


Fig 6.3 Monthly Analysis of Demand

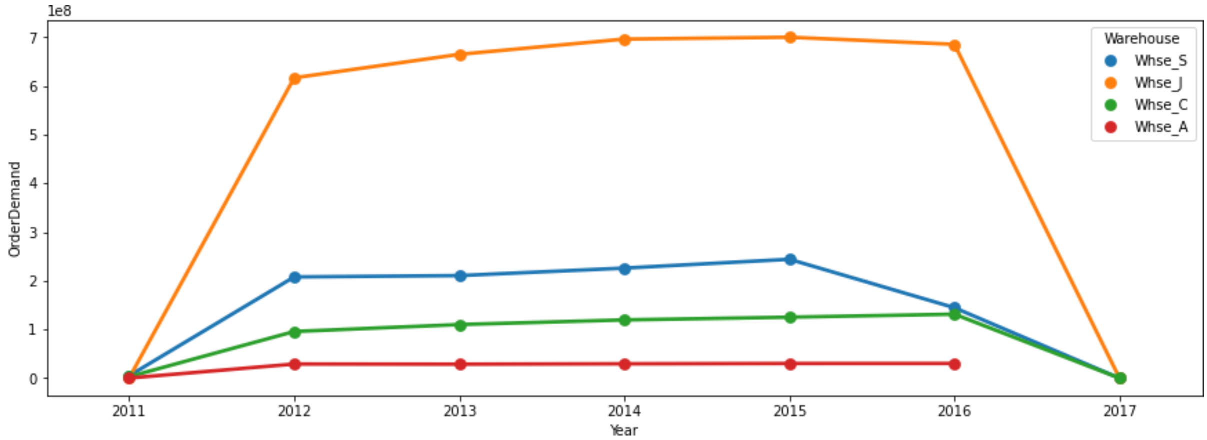


Fig 6.4. Warehouse Based Analysis of demand

6.2.3 FORECASTING ORDER DEMAND WITH LSTM MODEL

To implement LSTM for order demand forecasting, several steps need to be followed. These include:

✓ **Normalize the data:** The first step is to normalize the data to ensure that the LSTM model can process it effectively. Normalization involves scaling the data to a range of 0 to 1, which can be achieved using techniques such as min-max normalization or standardization.

✓ **Create the training dataset:** The next step is to create a training dataset that the LSTM model can learn from. This involves selecting a historical time period and using it as the basis for training the model. The dataset should include features such as order quantity, date, and any other relevant information that may impact order demand.

✓ **Create X_train and y_train:** The training dataset is then split into input and output sequences. X_train represents the input sequence, which includes the historical order data, and y_train represents the output sequence, which includes the order demand for the next time period.

✓ **Reshape the data:** Before training the LSTM model, the input sequence (X_train) needs to be reshaped into a 3D array to be compatible with the LSTM architecture. The reshaping involves specifying the number of samples, time steps, and features.

✓ **Create the testing dataset:** Once the LSTM model has been trained, it needs to be evaluated using a testing dataset. The testing dataset includes historical data that the model has not seen before and is used to test the model's ability to make accurate predictions.

✓ **Create X_test and y_test**: The testing dataset is also split into input and output sequences, X_test and y_test, respectively, using the same methodology as for the training dataset.

By following these steps, the LSTM model can be trained and evaluated for order demand forecasting, enabling businesses to make more accurate predictions and optimize their production and inventory management processes.object based segmentation. Minimum spanning tree method is used as a proposed work for image segmentation.

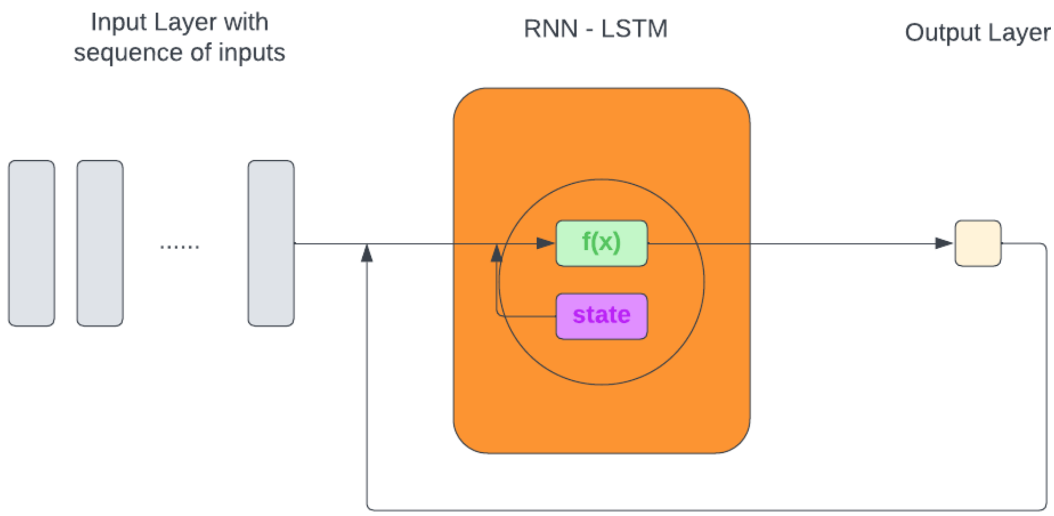


Fig 6.5. LSTM MODEL WORKING

6.2.4 BUILD LSTM MODEL

Building the LSTM model for order demand forecasting involves several steps. Once the data has been preprocessed and split into training and testing sets, the following steps can be followed:

- ✓ **Build the LSTM model:** The first step is to build the LSTM model using a deep learning framework such as TensorFlow or Keras. The model architecture should include one or more LSTM layers, as well as any additional layers such as dense layers, dropout layers, or batch normalization layers.
- ✓ **Compile the LSTM model:** Once the model architecture has been defined, it needs to be compiled using an optimizer such as Adam and a loss function such as mean squared error (MSE). Additional metrics such as mean absolute error (MAE) can also be included to evaluate the model's performance.
- ✓ **Train the LSTM model:** The next step is to train the LSTM model using the training dataset. This involves specifying the number of epochs and batch size, and monitoring the model's performance on both the training and validation datasets.
- ✓ **Predict with the LSTM model:** Once the model has been trained, it can be used to make predictions on the testing dataset. The predicted values can then be compared to the actual values to evaluate the model's accuracy.

- ✓ **Plot the data:** Finally, the predicted and actual values can be plotted to visualize the model's performance and identify any patterns or trends in the data.

By following these steps, the LSTM model can be built and trained for order demand forecasting, providing businesses with valuable insights into future demand trends and enabling them to optimize their production and inventory management processes.

```
# Build the LSTM Model
model = Sequential()
model.add(LSTM(units=512, return_sequences=True, activation='relu', input_shape=(X_train.shape[1], 1)))
model.add(LSTM(units=256, activation='relu', return_sequences=False))
model.add(Dense(units=1))
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 60, 512)           1052672

 lstm_5 (LSTM)               (None, 256)               787456

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 1,840,385
Trainable params: 1,840,385
Non-trainable params: 0
_________________________________________________________________
```

```
# compile the LSTM model
model.compile(optimizer="Adam", loss="mean_squared_error", metrics=['mae'])
```

Fig 6.6. Build LSTM Model

6.2.5 OPTIMIZE THE MODEL FOR BETTER ACCURACY

Building an optimized LSTM model for order demand forecasting involves experimenting with different model parameters and hyperparameters to achieve better performance. Once the initial LSTM model has been built and evaluated, the following steps can be followed to build an optimized LSTM model:

- ✓ **Change the parameters of the first LSTM model**: The first step is to experiment with different model parameters such as the number of LSTM layers, the number of units in each layer, and the activation function used in each layer. By changing these parameters, the model's performance can be optimized.
- ✓ **Compile the model:** Once the optimized model architecture has been defined, it needs to be compiled using an appropriate optimizer and loss function. This may involve experimenting with different optimizers such as Adam, RMSprop, or SGD, and different loss functions such as mean squared error (MSE), mean absolute error (MAE), or root mean squared error (RMSE).

- ✓ **Train the optimized model over many epochs:** The optimized model should be trained over many epochs to improve its accuracy. This involves specifying the number of epochs and batch size, and monitoring the model's performance on both the training and validation datasets.
- ✓ **Predict with optimized LSTM model:** Once the model has been trained, it can be used to make predictions on the testing dataset. The predicted values can then be compared to the actual values to evaluate the model's accuracy.
- ✓ **Plot the data:** Finally, the predicted and actual values can be plotted to visualize the model's performance and identify any patterns or trends in the data.

By following these steps, an optimized LSTM model can be built for order demand forecasting, providing businesses with more accurate and reliable predictions of future demand trends. This can help them to optimize their production and inventory management processes, reduce waste and costs, and improve customer satisfaction.

```
[ ] # change the parameters of first LSTM model and build the Optimized LSTM Model
    optimized_model = Sequential()

    optimized_model.add(LSTM(512, activation='relu', return_sequences=True, input_shape=(X_train.shape[1], 1)))

    optimized_model.add(LSTM(256, activation='relu', return_sequences=False))

    optimized_model.add(Dense(128))

    optimized_model.add(Dense(64))

    optimized_model.add(Dense(32))

    optimized_model.add(Dense(1))

    optimized_model.summary()

    Model: "sequential_3"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     lstm_6 (LSTM)               (None, 60, 512)           1052672

     lstm_7 (LSTM)               (None, 256)               787456

     dense_6 (Dense)             (None, 128)               32896

     dense_7 (Dense)             (None, 64)                8256

     dense_8 (Dense)             (None, 32)                2080

     dense_9 (Dense)             (None, 1)                 33
```

Fig 6.7. Optimized Model Summary

CHAPTER 7
CONCLUSION

In conclusion, the methodology for demand forecasting using the LSTM model involves several steps such as data preparation, exploratory data analysis, building the LSTM model, and optimizing it for better performance. The LSTM model is a powerful tool for time series forecasting, and has been shown to outperform traditional statistical models in many applications.

By preprocessing the data and splitting it into training and testing sets, businesses can build and train an LSTM model to forecast future demand trends with greater accuracy. The model can be optimized by experimenting with different model parameters and hyperparameters, and trained over many epochs to improve its performance. Predicted values can be compared to actual values to evaluate the model's accuracy, and the data can be plotted to visualize the model's performance and identify any patterns or trends in the data.

The demand forecasting architecture using the LSTM model is a valuable tool for businesses in many industries, as it enables them to make more informed decisions about production and inventory management. By accurately predicting future demand trends, businesses can reduce waste, optimize their supply chain, and improve customer satisfaction. Furthermore, by encouraging cross-border data flow and implementing machine learning techniques such as the LSTM model, businesses can stay ahead of the curve in an increasingly competitive global market.

The demand forecasting methodology outlined above is not only a valuable tool for businesses, but also solves real-world problems that businesses face on a daily basis. For instance, accurate demand forecasting can help businesses optimize inventory levels, reducing costs and minimizing waste. In addition, demand forecasting can help businesses plan production schedules more efficiently, ensuring that they have enough products on hand to meet customer demand without overproducing.

One example of how demand forecasting can help solve real-world business problems is in the retail industry. Retailers face the challenge of stocking the right products in the right quantities to meet customer demand while minimizing inventory costs. By accurately forecasting demand, retailers can adjust their inventory levels to ensure that they always have the right products on hand, reducing the risk of stockouts and lost sales.

Another example is in the food and beverage industry, where accurate demand forecasting is critical for managing perishable inventory. By forecasting demand, restaurants and food distributors can order the right amount of food and ingredients, minimizing waste and ensuring that fresh products are always available for customers.

In the manufacturing industry, demand forecasting can help businesses plan production schedules more efficiently. By accurately forecasting demand, manufacturers can schedule production runs for specific products, reducing the risk of overproduction and underproduction. This can help businesses minimize costs and improve customer satisfaction by ensuring that products are available when customers need them.

In summary, accurate demand forecasting is a valuable tool for businesses across many industries. By providing insights into future demand trends, businesses can optimize their inventory levels, plan production schedules more efficiently, and reduce costs. This can lead to improved profitability, increased customer satisfaction, and a competitive advantage in the marketplace.

Accurate demand forecasting can play a key role in reducing the bullwhip effect in the supply chain. The bullwhip effect refers to the phenomenon where small changes in consumer demand can result in amplified fluctuations in upstream supply chain activities. This can lead to inefficient inventory management, increased costs, and decreased customer satisfaction.

By accurately forecasting demand, businesses can reduce the bullwhip effect by providing suppliers with more accurate information about future demand trends. This can help suppliers better plan their production schedules, reducing the risk of overproduction and underproduction. By avoiding stockouts and backorders, suppliers can improve lead times, reduce the need for expediting, and improve customer satisfaction.

Accurate demand forecasting can also help businesses better manage their inventory levels. By forecasting demand with greater accuracy, businesses can avoid stockouts and minimize inventory holding costs. This can help to reduce the cost of carrying excess inventory and avoid stockouts, which can be costly to a business in terms of lost sales and customer dissatisfaction.

Finally, accurate demand forecasting can help businesses improve their supply chain performance by enabling more effective collaboration between supply chain partners. By sharing accurate demand forecasts with suppliers, businesses can improve communication and coordination, reducing the risk of misaligned production schedules and improving overall supply chain efficiency.

In summary, accurate demand forecasting can help reduce the bullwhip effect in the supply chain by providing suppliers with more accurate information about future demand trends, improving inventory management, and facilitating better collaboration between supply chain partners.

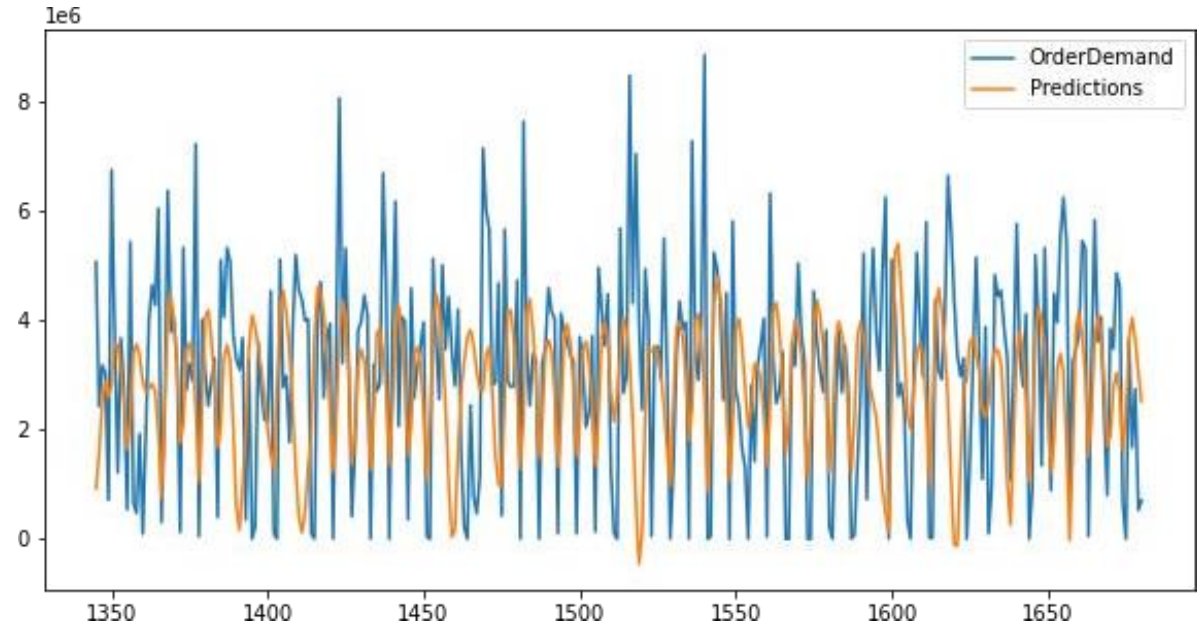


Fig 7.1 Predict Output

Model Name	MSE (Mean Squared Error)	MAE (Mean Absolute Error)
ETS	1.3192	0.149
XG-Boost	0.8219	0.059
LSTM	0.0387	0.1579

Fig 7.2 Model output Comparison

APPENDIX-1

SAMPLE CODINGS

```
# -*- coding: utf-8 -*-
"""Copy of forecast-product-demand-with-lstm.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/10L8YIiXT0nI_8fcKGnzdw2fA851vYJca

## Data Preparation
"""

# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import statistics
import math

#for LSTM model
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout

# ignore warnings
import warnings
warnings.filterwarnings("ignore")

from google.colab import drive
drive.mount('/content/drive')

# read data
data = pd.read_csv("https://raw.githubusercontent.com/abuthahir-19/Demand-
Forecasting/master/Code/Historical%20Product%20Demand.csv/Historical%20Prod
uct%20Demand.csv?token=GHSAT0AAAAAAB75RKEUZAO2G47B7E6PE5QIZBAGXBQ")

# rename the columns
data.rename(columns = {'Product_Code': 'ProductCode',
                       'Product_Category': 'ProductCategory',
                       'Order_Demand': 'OrderDemand'}, inplace = True)
data.head()

# check the null data
data.isnull().sum()

# drop the missing values, we can not fill the date so best way drop
missing samples
data.dropna(inplace=True)

# check the null data again
data.isnull().sum()
```

```
# sort the data according yo date column
data.sort_values('Date', ignore_index=True, inplace=True)
data.head()

# there are () int the OrderDemand column and we need to remove them
data['OrderDemand'] = data['OrderDemand'].astype(str).str.replace("(","")
data['OrderDemand'] = data['OrderDemand'].astype(str).str.replace(')',"")

# change the dtype as int64
data['OrderDemand'] = data['OrderDemand'].astype('int64')

# creating Year, Month, Day field for further analysis

# first way
# data[["Year", "Month", "Day"]] = data["Date"].str.split("/", expand =
True)
# data

# second way change date columns dtype as datetime
from datetime import datetime as dt

# convert the 'Date' column to datetime format
data['Date']= pd.to_datetime(data['Date'])

# create Year, Month, Day columns
data['Year'] = data["Date"].dt.year
data['Month'] = data["Date"].dt.month
data['Day'] = data["Date"].dt.day
# i used second way because i think it is more usable for dtypes

"""## Exploratory Data Analysis and Data Visualization"""

# information about data
data.info()

# statistical information about OrderDemand
data["OrderDemand"].describe()

# information about categorical variables
data[["ProductCode", "Warehouse", "ProductCategory"]].describe()

# Number of samples according to ProductCategory
plt.figure()
plt.barh(data["ProductCategory"].value_counts().index,
data["ProductCategory"].value_counts(), color = "b")
plt.xlabel("Frequency")
plt.ylabel("Product Category")
plt.title("Product Category - Data Frequency")
plt.show()
print(f"Number of ProductCategory
\n{data['ProductCategory'].value_counts()}")

# Number of samples according to Warehouse
sns.countplot(x="Warehouse", data=data)
plt.xticks(rotation = 0)
plt.show()
```

```
print(f"Number of samples according to Warehouse
\n{data['Warehouse'].value_counts()}")

sns.catplot(x="Month", y="OrderDemand", hue="Year", col="Warehouse",
            data=data, kind="bar", height=4)
plt.show()

sns.violinplot(x="Year", y="OrderDemand", data=data)
plt.show()

"""#### Yearly Analysis"""

df = data[['OrderDemand',
'Year']].groupby(["Year"]).sum().reset_index().sort_values(by='Year',
ascending=False)
f, ax=plt.subplots(figsize=(15, 5))
sns.pointplot(x='Year', y='OrderDemand', data=df)
plt.show()

# plot a pie chart and see percentages of Order Demand according to the
years
plt.pie(df['OrderDemand'], labels=df['Year'].unique(), autopct='%1.2f%%')
plt.show()

"""#### Monthly Analysis"""

temp_data = data.copy()
temp_data.Month.replace([1,2,3,4,5,6,7,8,9,10,11,12], ['Jan', 'Feb', 'Mar',
'Apr', 'May',
                                                        'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec'], inplace=True)
df = temp_data[['OrderDemand',
                'Month', 'Year',]].groupby(["Year",

"Month"]).sum().reset_index().sort_values(by=['Year',

'Month'], ascending=False)
f, ax=plt.subplots(figsize=(15, 5))
sns.pointplot(x='Year', y='OrderDemand', data=df, hue='Month',
hue_order=['Jan', 'Feb', 'Mar', 'Apr', 'May',

'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()

# Monthly pivot table
df =  (df.pivot(index='Year', columns='Month', values='OrderDemand'))
df = df.loc[:, ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec']]
df

custom_dict = {'Jan':0, 'Feb':1, 'Mar':2, 'Apr':3, 'May':4, 'Jun':5,
               'Jul':6, 'Aug':7, 'Sep':8, 'Oct':9, 'Nov':10, 'Dec':11}
temp_data = data.copy()
temp_data.Month.replace([1,2,3,4,5,6,7,8,9,10,11,12], ['Jan', 'Feb', 'Mar',
'Apr', 'May',
                                                                'Jun',
'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], inplace=True)
```

```
df = temp_data[["OrderDemand", 'Month', 'Year']].groupby(["Year",

"Month"]).sum().reset_index().sort_values(by=['Year',

'Month'], ascending=True)
df = df.iloc[df['Month'].map(custom_dict).argsort()]
f, ax=plt.subplots(figsize=(15, 5))
sns.pointplot(x='Month', y="OrderDemand", data=df, hue="Year")
plt.show()

# Statistical information about monthly data
df.describe()

"""#### Warehouse Based Analysis"""

df = data[["OrderDemand", 'Year', 'Warehouse']].groupby(["Year",

"Warehouse"]).sum().reset_index().sort_values(by=['Warehouse','Year'],
ascending=False)
f, ax=plt.subplots(figsize=(15, 5))
sns.pointplot(x='Year', y="OrderDemand", data=df, hue="Warehouse")
plt.show()

df = (df.pivot(index='Year', columns='Warehouse', values='OrderDemand'))
df

# Statistical information about according to warehouse data
df.describe()

"""#### Product Category Based Analysis"""

df = data[["OrderDemand",
           'ProductCategory', 'Warehouse']].groupby(["ProductCategory",

"Warehouse"]).sum().reset_index().sort_values(by=['OrderDemand'],

ascending=False)
df = df.pivot(index='ProductCategory', columns='Warehouse',
values='OrderDemand')
df

"""## Forecast the Order Demand with LSTM Model"""

# for better results use the data between 2012-01-01 and 2016-12-31
df = data[(data['Date']>='2012-01-01') & (data['Date']<='2016-12-
31')].sort_values('Date', ascending=True)
df = df.groupby('Date')['OrderDemand'].sum().reset_index()
df

# Visualize the order demand as time series
plt.figure(figsize=(16, 8))
plt.title("Order Demand Graph")
plt.plot(df["Date"], df["OrderDemand"])
plt.xlabel("Time", fontsize=14,)
plt.ylabel("Order Demand", fontsize=14)
plt.show()
```

```
# Create new data with only the "OrderDemand" column
orderD  =  df.filter(["OrderDemand"])
# Convert the dataframe to a np array
orderD_array = orderD.values
# See the train data len
train_close_len = math.ceil(len(orderD_array) * 0.8)
train_close_len

# Normalize the data
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(orderD_array)
scaled_data

# Create the training dataset
train_data = scaled_data[0 : train_close_len, :]
# Create X_train and y_train
X_train = []
y_train = []
for i in range(60, len(train_data)):
    X_train.append(train_data[i - 60 : i, 0])
    y_train.append(train_data[i, 0])
    if i <= 60:
        print(X_train)
        print(y_train)

#  make X_train and y_train np array
X_train, y_train = np.array(X_train), np.array(y_train)

# reshape the data
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_train.shape

# create the testing dataset
test_data = scaled_data[train_close_len - 60 : , :]
# create X_test and y_test
X_test = []
y_test = df.iloc[train_close_len : , :]
for i in range(60, len(test_data)):
    X_test.append(test_data[i - 60 : i, 0])

# convert the test data to a np array and reshape the test data
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

"""## Build a LSTM Model"""

# Build the LSTM Model
model = Sequential()

model.add(LSTM(units=512, return_sequences=True, activation='relu',
input_shape=(X_train.shape[1], 1)))


model.add(LSTM(units=256, activation='relu', return_sequences=False))


model.add(Dense(units=1))
```

```
# compile the LSTM model
model.compile(optimizer="Adam", loss="mean_squared_error", metrics=['mae'])

# train the LSTM model
model.fit(X_train, y_train,
          epochs=3,
          batch_size=100,
          verbose=1)

# predict with LSTM model
predictions = model.predict(X_test)
predictions = scaler.inverse_transform(predictions)

# plot the data
train = orderD[:train_close_len]
valid = orderD[train_close_len:]
valid["Predictions"] = predictions
#visualize the data
plt.figure(figsize=(16, 8))
plt.title("Forecast with LSTM Model")
plt.xlabel("Time", fontsize=14)
plt.ylabel("Order Demand", fontsize=14)
plt.plot(df["Date"][:train_close_len], train["OrderDemand"])
plt.plot(df["Date"][train_close_len:], valid[["OrderDemand",
"Predictions"]])
plt.legend(["Train", "Validation", "Predictions"], loc="lower right")
plt.show()

"""## Build a Optimized LSTM Model"""

# change the parameters of first LSTM model and build the Optimized LSTM
Model
optimized_model = Sequential()

optimized_model.add(LSTM(512, activation='relu', return_sequences=True,
input_shape=(X_train.shape[1], 1)))

optimized_model.add(LSTM(256, activation='relu', return_sequences=False))

optimized_model.add(Dense(128))

optimized_model.add(Dense(64))

optimized_model.add(Dense(32))

optimized_model.add(Dense(1))

# compile the model
optimized_model.compile(optimizer="Adam", loss="mean_squared_error",
metrics=['mae'])

# train the optimized model
optimized_model.fit(X_train, y_train,
          batch_size=32,
          epochs=20,
          verbose=1)
```

```
# Predict with optimized LSTM model
o_predictions = optimized_model.predict(X_test)
o_predictions = scaler.inverse_transform(o_predictions)

# plot the data
train = orderD[:train_close_len]
valid = orderD[train_close_len:]
valid["Predictions"] = o_predictions
#visualize the data
plt.figure(figsize=(16, 8))
plt.title("Forecast with Optimized LSTM Model")
plt.xlabel("Time", fontsize=14)
plt.ylabel("Order Demand", fontsize=14)
plt.plot(df["Date"][:train_close_len], train["OrderDemand"])
plt.plot(df["Date"][train_close_len:], valid[["OrderDemand",
"Predictions"]])
plt.legend(["Train", "Validation", "Predictions"], loc="upper right")
plt.show()
```

APPENDIX-2

SCREEN SHOT

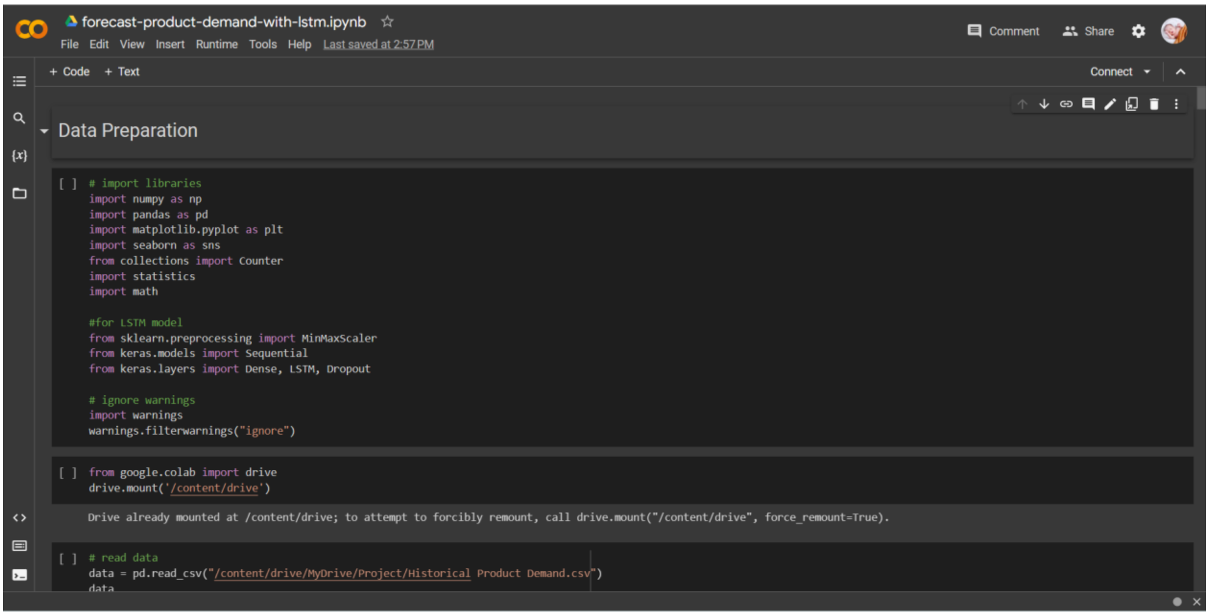


Fig 8.1 Workspace Of GOOGLE COLAB

	Product_Code	Warehouse	Product_Category	Date	Order_Demand
1	Product_Code	Warehouse	Product_Category	Date	Order_Demand
2	Product_0993	Whse_J	Category_028	27-07-2012	100
3	Product_0979	Whse_J	Category_028	19-01-2012	500
4	Product_0979	Whse_J	Category_028	03-02-2012	500
5	Product_0979	Whse_J	Category_028	09-02-2012	500
6	Product_0979	Whse_J	Category_028	02-03-2012	500
7	Product_0979	Whse_J	Category_028	19-04-2012	500
8	Product_0979	Whse_J	Category_028	05-06-2012	500
9	Product_0979	Whse_J	Category_028	27-06-2012	500
10	Product_0979	Whse_J	Category_028	23-07-2012	500
11	Product_0979	Whse_J	Category_028	29-08-2012	500
12	Product_0979	Whse_J	Category_028	29-08-2012	500
13	Product_0979	Whse_J	Category_028	18-09-2012	500
14	Product_0979	Whse_J	Category_028	11-10-2012	500
15	Product_0979	Whse_J	Category_028	01-11-2012	500
16	Product_0979	Whse_J	Category_028	29-11-2012	500
17	Product_0979	Whse_J	Category_028	26-12-2012	500
18	Product_1159	Whse_J	Category_006	06-01-2012	50000
19	Product_1159	Whse_J	Category_006	18-01-2012	100000
20	Product_1159	Whse_J	Category_006	02-02-2012	50000
21	Product_1159	Whse_J	Category_006	22-02-2012	50000
22	Product_1159	Whse_J	Category_006	02-03-2012	50000
23	Product_1159	Whse_J	Category_006	09-03-2012	50000
24	Product_1159	Whse_J	Category_006	23-03-2012	50000
25	Product_1159	Whse_J	Category_006	06-04-2012	50000
26	Product_1159	Whse_J	Category_006	16-04-2012	50000
27	Product_1159	Whse_J	Category_006	07-05-2012	50000

Fig 8.2 Sample Training Dataset

REFERENCES

1) H. Kim, S. Park, H. -J. Park, H. -G. Son and S. Kim, "Solar Radiation Forecasting Based on the Hybrid CNN-CatBoost Model," in IEEE Access, vol. 11, pp. 13492-13500, 2023.
2) M. Ratusny, M. Schiffer and H. Ehm, "Customer Order Behavior Classification Via Convolutional Neural Networks in the Semiconductor Industry," in IEEE Transactions on Semiconductor Manufacturing, vol. 35, no. 3, pp. 470-477, Aug. 2022, doi: 10.1109/TSM.2022.3187100.J.
3) U. M. Sirisha, M. C. Belavagi and G. Attigeri, "Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison," in IEEE Access, vol. 10, pp. 124715-124727, 2022, doi: 10.1109/ACCESS.2022.3224938.
4) S. S. J. Nithin, T. Rajasekar, S. Jayanthy, K. Karthik and R. R. Rithick, "Retail Demand Forecasting using CNN-LSTM Model," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 1751-1756, doi: 10.1109/ICEARS53579.2022.9752283.
5) A. Krishna, A. V, A. Aich and C. Hegde, "Sales-forecasting of Retail Stores using Machine Learning Techniques," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 160-166, doi: 10.1109/CSITSS.2018.8768765.
6) Y. -S. Huang, C. -H. Ho and C. -C. Fang, "Information Sharing in the Supply Chains of Products With Seasonal Demand," in IEEE Transactions on Engineering Management, vol. 64, no. 1, pp. 57-69, Feb. 2017, doi: 10.1109/TEM.2016.2623327.