20IT928 PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY & ENTREPRENEURSHIP

RETAIL STORE STOCK INVENTORY ANALYTICS

A PROJECT REPORT

Submitted by

Narra Jayani	**111720102083**
Nivedha V.	**111720102090**
Nivethaa M.	**111720102091**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

R.M.K. ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

December 2023

R.M.K. ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

BONAFIDE CERTIFICATE

Certified that this project report **"Retail Store Stock Inventory Analytics"** is the bonafide work of **Narra Jayani (111720102083), Nivedha V. (111720102090), Nivethaa M. (111720102091)** who carried out the **20IT928 Professional Readiness for Innovation, Employability and Entrepreneurship** work under my supervision**.**

SIGNATURE	**SIGNATURE**
Dr. T. Sethukarasi, M.E., M.S. Ph.D., **Professor and Head** Department of Computer Science and Engineering R.M.K. Engineering College R.S.M. Nagar, Kavaraipettai, Tiruvallur District– 601206.	**Mr. D. Kirubakaran, M.E.,** **Supervisor** **Assistant Professor** Department of Computer Science and Engineering R.M.K. Engineering College R.S.M. Nagar, Kavaraipettai, Tiruvallur District–601206.

Submitted for the Project Viva –Voce held on ……………………… at **R.M.K. Engineering College**, Kavaraipettai, Tiruvallur District– 601206.

INTERNAL EXAMINER

ACKNOWLEDGEMENT

We earnestly portray our sincere gratitude and regard to our beloved **Chairman Shri. R. S. Munirathinam, our Vice Chairman, Shri. R. M. Kishore** and **our Director, Shri. R. Jyothi Naidu,** for the interest and affection shown towards us throughout the course.

We convey our sincere thanks to our **Principal**, **Dr. K. A. Mohamed Junaid,** for being the source of inspiration in this college.

We reveal our sincere thanks to our **Professor and Head of the Department, Computer Science and Engineering, Dr. T. Sethukarasi,** for her commendable support and encouragement for the completion of our project.

We would like to express our sincere gratitude for our Project Guide **Mr. D. Kirubakaran, M.E., Assistant Professor** for their valuable suggestions towards the successful completion for this project in a global manner.

We take this opportunity to extend our thanks to all faculty members of the Department of Computer Science and Engineering, parents and friends for all that they meant to us during the crucial times of the completion of our project.

ABSTRACT

Retail sector handles stock in bulk quantities which brings in the need for a digitized system that can analyze and help managing the inventory effectively. Analytics in this sector has become a mandatory supporting component as the business actions and decisions are taken potently and quickly based on it. Here the proposed work uses SARIMA (Seasonal Autoregressive Integrated Moving Average) statistical analysis model for performing the analysis on the retail store data. Data sets used in this model are as the sales history of the store, inventory and demand history of the store. Unlike other data analytics model, SARIMA has seasonal component ans uses parameters as p(auto-regressive order), d(degree of differencing), q(moving-average order), P(seasonal AR order), D(seasonal differencing) and Q(seasonal MA order) that help in handling seasonal time series data. Also, SARIMA model has smaller MAD (Mean Absolute Deviation) value when compared to widely used Holt Winter's Exponential Smoothing model which makes it outperform other time series analytics model. Overall, the SARIMA model efficiently performs the retail store stock inventory analytics and calculates the inventory needed to fulfill the customer requirements over a period of time benefiting the retail store with maximum profit.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1	ARIMA	Autoregressive Integrated Moving Average
2	SARIMA	Seasonal Autoregressive Integrated Moving Average
3	ACF	Autocorellation Function
4	PACF	Partial Autocorellation Function
5	MAPE	Mean Absolute Percentage Error
6	MAD	Mean Absolute Deviation
7	MSD	Mean Squared Deviation
8	ARMA	Autoregressive Moving Average
9	MA	Moving Average
10	DPUT	Demand per unit time
11	DLT	Demand lead time
12	TRI	Triangular distribution
13	ANN	Artificial neural network
14	ESM	Exponential smoothing method
15	GA	Genetic algorithm
16	DE	Differenntial evaluation

17	RMSE	Root mean square error
18	LSTM	Long short term memory
19	IDE	Integrated development environment
20	NV	NewsVendor

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Retail store handles stock in large scale on daily basis that makes the monitoring and managing of the inventory more tedious. The traditional retail store inventory management is a cubersome methodology which has inefficient monitoring, tracking and management. This bring in the need for a robust digitized inventory management system that performs retail store stock analytics seamlessly to achieve less inventory on hand and more stocks on sale with less manual labor.

Challenges in traditional retail store stock inventory management can be listed as,

1. Manual Labor Intensiveness
2. Optimization of Inventory on-hand
3. Stock analytics for seasonal data

Objective of the proposed work in managing the stock inventory of the retail store stock is given as,

1. Develop a user-friendly digitized inventory management system
2. Implement the system to enable timely decision making
3. Optimize the inventory levels to achieve the balance between the customer demand and excess stock
4. Calculate the optimal inventory level, reorder level of the inventory, optimal restock quantity and total estimated cost for maintaing the optimal inventory

In conclusion, the development of an optimal digitized inventory management system is imperative for retail store to address the shortcoming in traditional retail store stock inventory management and thrive the need in today's retail store domain.

1.2 LITERATURE SURVEY

1.2.1 DOUBLE EXPONENTIAL SMOOTHING METHOD

Retno Mumpuni, Sugiarto and Rais Alhakim explains inventory forecasting system using double exponential smoothing model in the paper 'Design and Implementation of Inventory Forecasting System using Double Exponential Smoothing Method'. This paper uses double exponential smoothing to perform forecasting task using level and trend components for calculation in each period. The algorithm deals with two weights also know as the smoothing parameters to update granules each time. MAPE-Mean Absolute Percentage Error and MAD-Mean Absolute Deviation are used to evalute the model. MAPE indicates the level of accuracy of the adjusted time series based values. They also express the level of accuracy as a percentage. MAD provides the error percentage. Another component MSD-Mean Squared Deviation provides a sensitive measurement of errors. Overall, MAPE is used as the indicator to measure the forecasting value. The model proposed here mainly try to address the planning and implementation of an inventory forecasting information system. At the end the accuracy difference between original and forecast is around 2.65% that gives MAPE value around 33.18%.[1]

1.2.2 ARIMA MODEL WITH RESIDUAL OPTIMIZATION

Wen Hu and Xinchao Zhang in the paper 'Commodity sales forecast based on ARIMA model residual optimization' explains about how ARIMA model is widely used in many forecasting studies. The main consideration they consider in ARIMA model is that there is no auto correlation information worth extracting the residual sequence. Thus they implement an algorithm that split the residual data as intervals of small columns of data input and the input mode is normalized and linear transformed to preprocess. The outputs are processed through various layers number of times until a pattern class is determined to which they belong. The performance of this model shows better prediction performance compared to traditional forecasting method. This is achieved by the ART2 residual error overall making advantages in sales forecasting. Further optimization can be done of ART2 parameters for better modeling models. [2]

1.2.3 ANALYSIS OF TREND USING ARIMA MODEL

Mohankumari C, Vishukumar M and Nagaraja Rao Chillale in the paper 'ANALYSIS OF DAILY STOCK TREND PREDICTION USING ARIMA MODEL' expalins how forecasting is necessity in human life and have many adopted statistical techniques like Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA), Auto Regrssive Conditional Heteroscedaticity (ARCH),,Generalized Auto Regressive Conditional Heteroscadasticity (GARCH), ARMA-EGARCH and few soft computing and evolutionary computing methods. Among all ARIA models have shown efficient capability to generate short term prediciton. They expresses how ARIMA model predicts future value of a variable with linear combination of past values, past errors. Thus with the results obtained, this paper concludes that the ARIMA odels with emerging forecasting techniques can compete reasonably well in short term predictions.[3]

1.2.4 ARIMA TRIANGULAR DISTRIBUTION

Fernando Rojas in the paper 'A methodology for stochastic inventory modelling with ARMA triangular distribution for new products' explains the model of a stochastic inventory policy of continuous review with random deman described with temporal dependence through an autoregressive moving average (ARMA) model. Here the model is built with triangular demand distriution exhibited with cumulative distribution function and qunatile function. Thus the parameterization of the mean and standard deviation of DPUT with TRI distribution under a temporal structure of a condition autoregressive model to information passed as ARMA, achieves a dynamic characterization of a policy involving DPUT and DLT. This paper has limitation of it could be extended to have multiple optimization targets on TC.[4]

1.2.5 EXPONENTIAL WEIGHED MOVING AVERAGES

Charles C. Holt in the paper 'Forecasting seasonals and trends by exponentially weighed moving averages' provide a systematic development of the forecasting expressions for exponential weighed moving averages. Here properties of the model is exhibited as (1) declining weight is put on older data, (2) it is extremely easy to compute, and (3) minimum data is required. Stochastic process is approached where the weighed average of all past observations are used to forecast the present mean of the distribution. This provides flexibility with forecasts of seasonals and trends. It also gives a optimal filter design for stationary time series using the criteria of minimizing the sum of the square of the errors. Further exploration is needed in the underlying stochastic theory.[5]

1.2.6 STOCK TREND PREDICTION USING ATIMaA MODEL

Uma Devi, Sundar, and Dr. Ali in the paper 'An Effective Time Series Analysis for Stock Trend Prediciton using ATIMaA Model for NIfty Midcap-50' explains the seasonal trend and flow is the highlight of the stock market. Eventually investors as well as the stock broking company will also observe and capture the variations, as constant growth of the index. This will help new investor as well as existing ones to make a strategic decision. It can be achieved by experience and the constant observation by the investors. In order to overcome the above said issues, ARIMA algorithm has been suggested in three steps, Step 1: Model identification , Step 2: Model estimation and Step 3: Forecasting.[6]

1.2.7 STOCK INDEX FORECASTING

Wang, J.J., Wang J.Z., Zhang Z.G., and Guo S.P. in the paper 'Stock index forecasting based on a hybrid model' defined that in recent time, to improve stock price predictive models, unique strength of the hybrid approaches are being exploted and engaged in different forecasting model. ANNs from artificial intelligence is also used for a artifical intelligence perspective. From statistical models perspective ARIMA models have been derived. Generally, from two perspectives: statistical and artificial intelligence techniques the prediction can be done. Here a hybrid approach is proposed by combining ESM (Exponential smoothing model), ARIMA (Autoregressive integrated moving average), and BPNN (back propogation neural network) to be the most advantageous of all three models. The weigh of the proposed hybrid model in this paper is determined by genetic algorithm (GA). Performance evalution of this model outperforms the traditional models. [7]

1.2.8 COMPARISON BETWEEN HYBRID MODEL AND ANN

Sabhyata Lamichhane, Bin Mei, Jacek Siry in the paper 'Comparison between a time series hybrid model and an artificial neural network' analyses the predicitive ability of classical econometric models and artifical neural networks (ANN). This paper exhibits that although ANN models have gained popularity and have been proven to be superior in various fields, there is limited research on the reliability and applicability of ANN models in forecasting future values. A study is made in this paper which makes a significant contribution by laying the groundwork for forecasting the prices and future values in the market using ANN models. Additionally it expands by implementing and analyzing two forecasting models as a hybridization of traditional econometric models and a widely used artificial neural network.[8]

1.2.9 INVENTORY MANAGEMENT WITH DEMAND

Rojas F., and Leiva V. explains about stochastic programming for forecasting model in the paper '.Inventory management in food companies with demand statistically dependent'. As its name suggests, stochastic programming is a mathematical programming problem (linear, nonlinear, integer, etc.) which contains in its formulation some stochastic element that is unknown, but that can be estimated from its probability distribution. The expected value of the objective function based on the TC of inventory must be optimized in a continuous review policy of an assortment of products. Stochastic programming can be used to solve this optimization problem by the differential evolution (DE) algorithm, which belongs to the family of genetic algorithms, imitating the natural process of choice in evolutionary fashion.[9]

1.2.10 COMPARISON BETWEEN HYBRID MODEL AND ANN

Nitin Merh, Vinod Prakash Saxena and Kamal Raj Pardasan in the paper 'A comparison between Hybrid Approaches of ANN and ARIMA for Indian Stock Trend Forecasting' defines that ARIMA models are known to be robust and efficient, especially for short-term prediction than the popular ANNs techniques. The paper tries an attempt to develop hybrid models of three layer feed forward back propogation artificial neural network and auto regressive integrated moving average (ARIMA) for forecasting future value and trend. Performance of the models have been evalutaded by calculating the RMSE, MAPE and MSPE. Results show that hybrid of ANN and ARIMA is better.[10]

1.3 SYSTEM REQUIREMENTS

1.3.1 HARDWARE REQUIREMENTS

To run the model hardware with Windows, mac OS or Linux operating system is preferred. The hardware must be inbuilt with Intel Pentium 4 or later versions of processor. 2GB minimum memory is required, 4GB is recommended. Screen resolution of 1280x1024 or larger and application window of 1024680 is required. Internet connection is recommended.

Components	Requirements
CPU	Intel Core i7 or AMD Ryzen 7 or higher
GPU	Nvidia GeForce RTX 3060 or AMD Radeon RX 6700 XT or higher
RAM	32GB or more
Storage	1TB or more of SSD storage
Deployment	Server hardware
Server Hosting	Hosting Services

Table 1.1 : Hardware requirements

1.3.2 SOFTWARE REQUIREMENTS

Initially a web browser is needed to run the software. Any of the popular browsers as Chrome, Firefox, Opera, Safari, edge can be used. To develop the software Pycharm IDE or Anaconda environment is used. Other online environment like Google Colaboratory is also recommended for the development of the machine learning algorithm. For working with the Flask framework, text editors like Atom, VS Code can be used for flexible environment.

1.3.3 FEASIBILITY STUDY

1.3.3.1 SCOPE OF THE PROJECT

The project scope focuses to address the problem using the SARIMA time series model that performs data analysis on the retail store datasets. Few examples of the datasets used by this model are as the sales history of the store, inventory and demand history of the product in the store etc

To calculate and forecast optimal inventory parameters as the stock quanity, reorder level, restock quantity, total estimate cost, Newsvendor formula from the Newsvendor model is used.

1.3.3.2 OBJECTIVE OF THE PROJECT

The objective of the work is to handle and manage the bulk stock in the retail store in an efficient digitized way involving less manual labor. Thus SARIMA (Seasonal Autoregressive Integrated Moving Average) time series forecasting model is used to create a forecasting model with Newsvendor model to forecast optimal inventory needed to satisfy the customer over a period of time.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Different time series forecasting models are available to forecast the future need based on the previous time series data. Some of these models used in retail sector to forecast inventory are as the LSTM (Long Short-Term memory) model, Simple Exponential Smoothing model, Holt Winter's Exponential Smoothing model, ARIMA (Autoregressive Integrated Moving Average) model etc.

2.1.1 DISADVANTAGES OF EXISTING SYSTEM

Though these different models as Long Short Term Memory, Simple Exponential Smoothing, Holt Winter’s Exponential Smoothing, ARIMA forecast and predict the future inventory need, the accuracy of the result obtained is not very efficient. Thus an efficient retail store stock inventory analytics model is needed to forecast the optimal inventory needed to fulfill the customer orders.

2.2 PROPOSED SYSTEM

The proposed solution addresses the problem in managing the retail store inventory using the time series forecasting algorithm. In this proposed solution the SARIMA - Seasonal Autoregressive Integrated Moving Average model is used that is regression model with seasonal component which is efficient to make inventory forecasting and thus managing it efficiently. Newsvendor formula from the Newsvendor model is used which is widely used in supply chain domains and inventory in industry.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

Out of the different models available in this algorithm, SARIMA (Seasonal Autoregressive Integrated Moving Average) model is preferred as it outperforms the others with its ability to forecast future need with past seasonal time series data. SARIMA when compared with the other time series analytics model produces smaller MAD (Mean Absolute Deviation) value. This makes SARIMA model feasible to be used for forecasting. SARIMA has three non-seasonal parameters as p(AR order), d(degree of differencing), q(MA order) and three seasonal parameters as P(seasonal AR order), D(seasonal differencing), Q(seasonal MA order) that are calculated using ACF (Autocorrelation function) and PACF (Partial autocorrelation function). Newsvendor formula is used in the proposed SARIMA model to calculate the optimal order quantity for the inventory. Overall the proposed SARIMA model predicts the optimal inventory needed.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

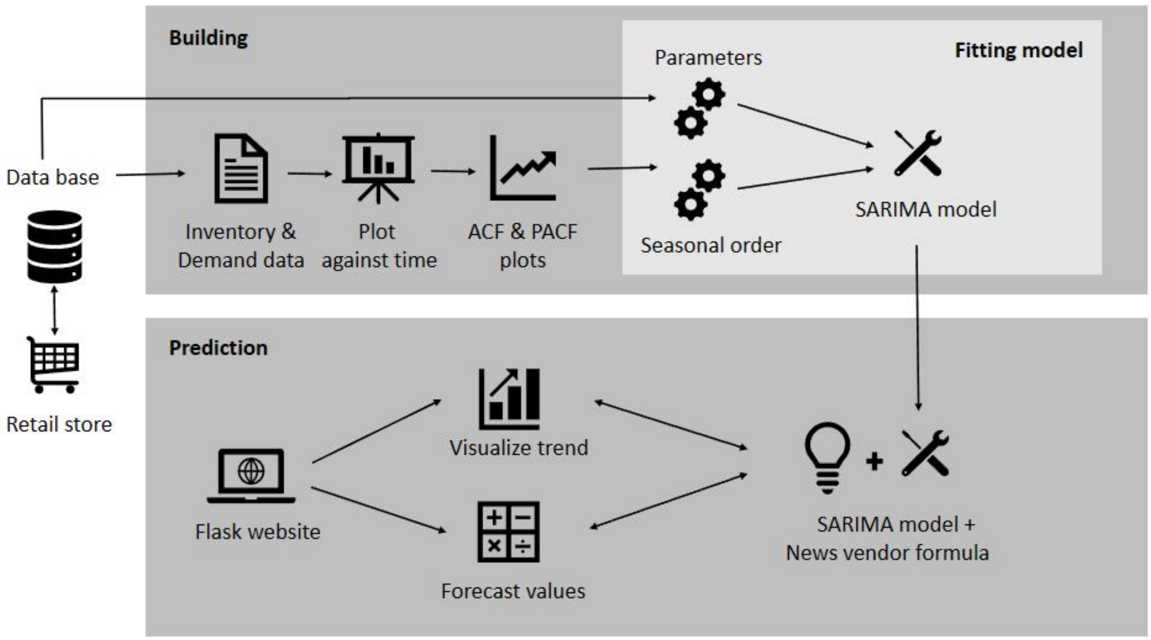

Fig 3.1 : System Architecture diagram

The system architecture represents the overview of the proposed work. Here the retail store stock inventory analytics is achieved with the time series forecasting algorithm which is built as the SARIMA model. Thus SARIMA model is the core of the inventory management system in the retail store. The architecture has two blocks, one is the building block that involves building the machine learning model for the analytics. Another block is the predicition block that gives the overview of the internal processing that happens within the model built for making the prediction for the reatil store stock inventory management.

The architecture starts from the retail store where retailer operating the system interfaces with the SARIMA model implemented via the Flask website developed for evaluationg and forecasting the optimal inventory. This SARIMA model used in the website is built with the database from the retail store. The history of sales, demand over time, inventory over time, seasonal trend data and many such time series data set from the retail store is maintained as datasets that is used by the system at time of visualizing trend, forecasting demand and calculating inventory.

Thus the building phase involves data collection where the inventory and demand data from the retail store database is retrieved followed by plotting the data against time along with ACF (Autocorellation Function) and PACF (Partial Autocorrelation function) to obtain the parameters for fitting the SARIMA model. The paramters involved in SARIMA model fitting are seasonal order, initial inventory, load time and service level. Further the model is combined with News vendor formula for calculating the optimal inventory and thus forecasts values as, optimal order quantity, reorder point, safety stock and estimated total cost. Thus the overview of the system architecture of the retail store stock inventory analytics model built with the SARIMA time series algorithm.

3.2 UML DIAGRAMS

3.2.1 USE CASE DIAGRAM

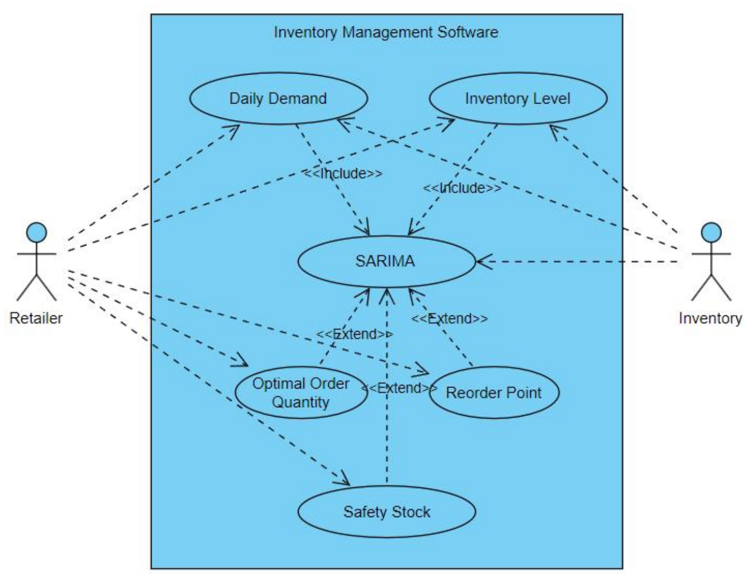

Fig 3.2 : Use case diagram

The use case diagram depicts the primary actors and the system for managing and forecasting the retail store stock inventory. The primary actors involved in this use case is the retailer and the system is the inventory management software. The core of the system is the SARIMA model that performs the demand forecasting and inventory optimization. The system in back end stores the data in the database which in here is the inventory of the retail store.

The use case diagram depicts different use cases and operations of the actors. The retailer can view the demand and inventory trend in his retail store via a graphical visualization for individual stock. Then they can forecast the future inventory stock quanitty by giving generate from the flask website.

This website runs the SARIMA model developed with the Newsvendor formula for the fed dataset to forecast the optimal inventory values as order quanity, reorder level, safety stock and toatal estimate cost to maintain it. The inventory system at the backend of the Flask website manages the whole machine learning model and the dataset.

3.2.2 CLASS DIAGRAM

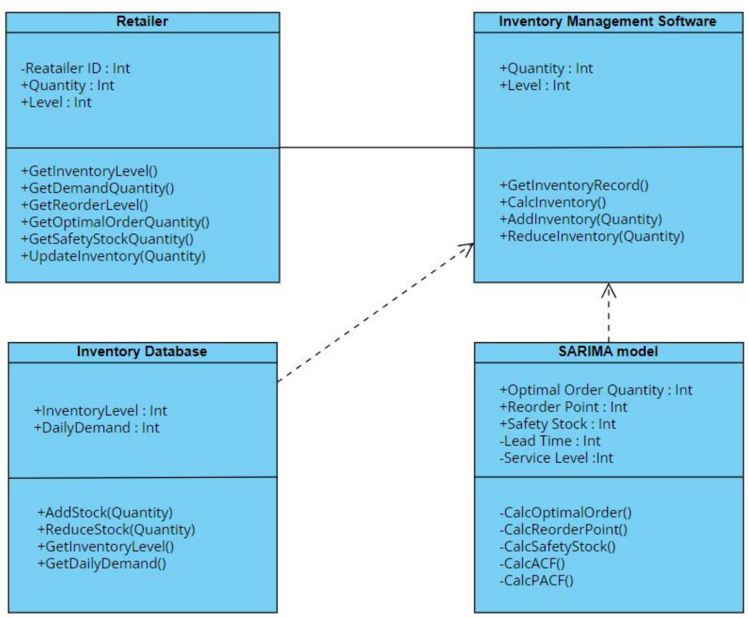

Fig 3.3 : Class diagram

The class diagram depicts the static structure and the operations of the system. There are four classes involved in the system they are as, retailer class, inventory management software class, SARIM model class and the inventory database class. SARIMA model class and the inventory class are in association with the inventory management class as any changes in any of the classes are affected and reflected everywhere.

Reatiler performs operations as updating inventory by adding stock or removing stock then can view the restock quantity and reorder point for the stock inventory using the software. The software performs the functions for the user using the SARIMA model and the data sets from the retail store is managed in the inventory database.

3.2.3 SEQUENCE DIAGRAM

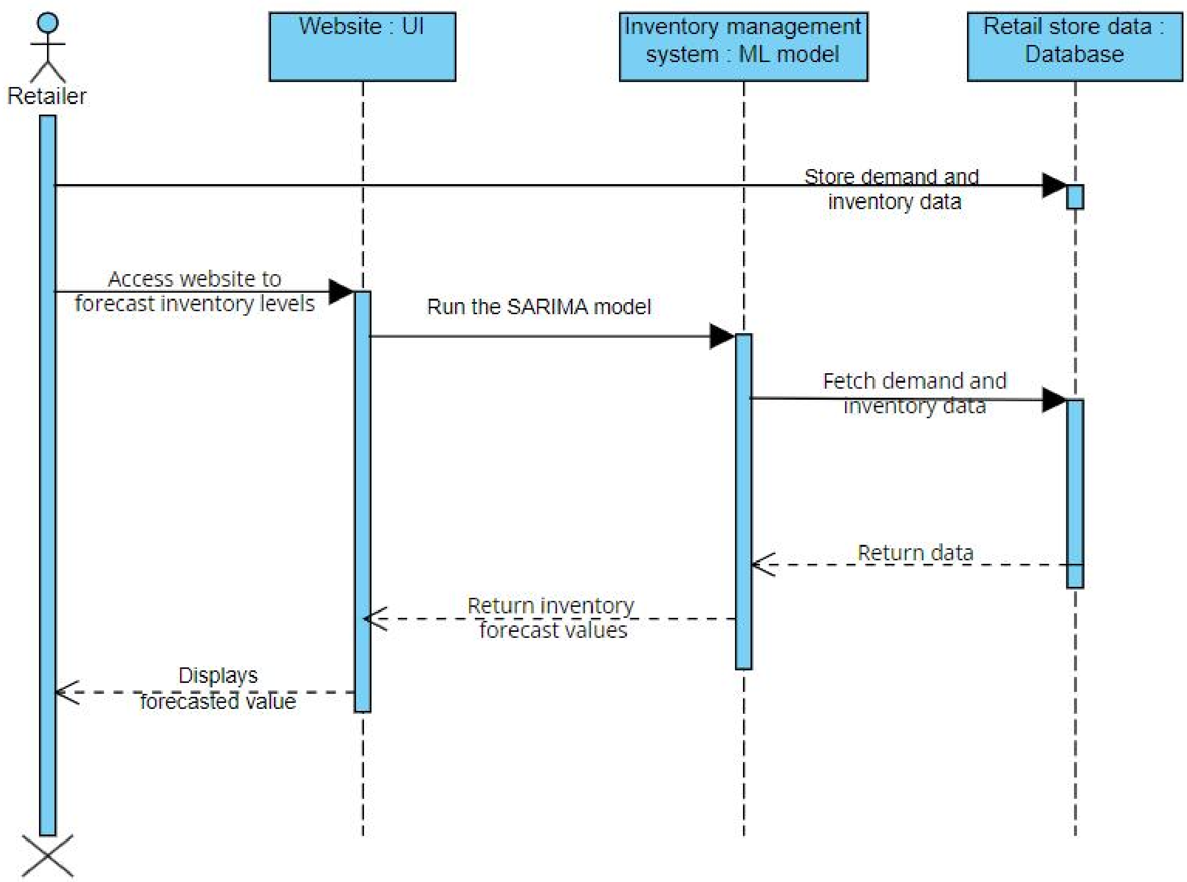

Fig 3.4 : Sequence diagram

Sequence diagrams represents the interaction within the system that describes how operations are carried out. The interaction that takes place in a collaboration tha either realizes a use case or an operation. Sequence diagram also captures the high-level interactions between the user of the system and the system, between the system and other systems or between subsystems.

The sequence diagram of the proposed system has different components as the user who is the retailer of the retail store, website that retailer uses to forecast the inventory parameters to maintain optimal inventory level, the machine learning model built with SARIMA algorithm and Newsvendor formula that evaluates the optimal inventory parameters as inventory quantity, reorder level, safety stock and the total estimated cost. All these operations of the system is based on the dataset where the demand and inventory over time of the particular stock is maintained by the shop.

The operation starts with the retailer accessing the website to view the demand over time and inventory over time of the stock in his shop and then clicks on the generate button available in the website to evaluate and optimal inventory parameters which is displayed in the website on click.

The optimal inventory parameters are evaluated by invoking the machine learning model built at the backend which uses SARIMA algorithm and Newsvendor formula. SARIM algoirthm is a time series forecasting algorithm that uses time sequence data which in case is the demand and inventory over time to forecast the future values. Now to forecast the values optimally, the Newsvemdor formula is used. Newsvendor formula is based on the Newsvendor model that is designed specially for managing the inventory at optimal level.

Thus overall the sequence diagram displays the high-level object interactions in the system that realizes a use case or operation of the system.

3.2.4 ACTIVITY DIAGRAM

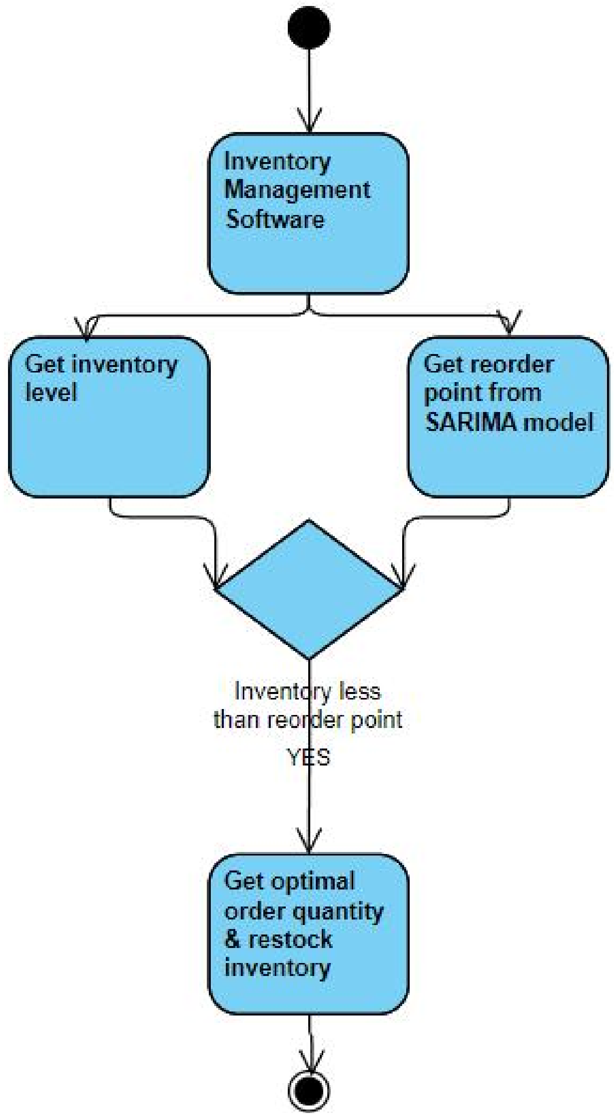

Fig 3.5 : Activity diagram

The activity diagram depicts the flow of activity of the system. Here the activity starts with interacting with the system where the user can forecast various values from the SARIMA model developed. Thus the inventory can be viewed for the level and if it’s found to be below the reorder level then the optimal inventory quantity is restocked. Thus the inventory can be managed efficiently.

3.2.5 DATA FLOW DIAGRAM

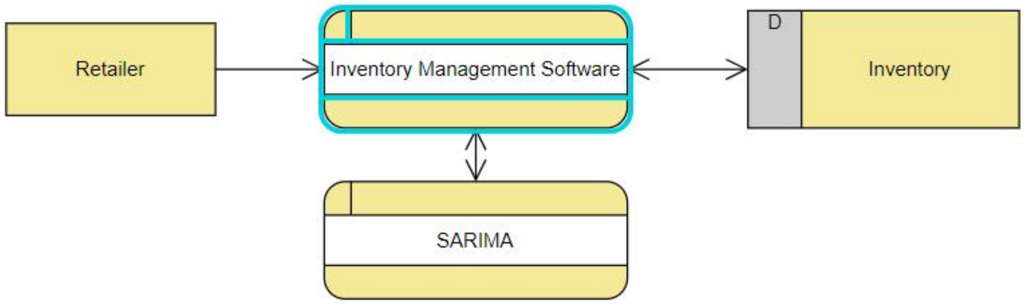

Fig 3.6 : Data flow diagram

The data flow diagram depicts the flow of process and data in the system. Here the system has external entities, process and data store. The external entity is the actors or sinks outside the system and in the proposed work the retailer is the external entity who comes in use with the system. The process here is the inventory management system that uses another process SARIMA for its working and the data store of this process is the inventory where the time series data set of the retail store is maintained.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULES

The modules in the developed retail store stock inventory analytics model is as,

Module	**Function of module**
Visualize demand	Shows visualization of the demand over time
Visualize Inventory	Shows visualization of the inventory level over time
Demand Forecasting	Processes the time series data set for forecasting the demand and calculating parameters
Inventory optimizing	Calculates the optimal inventory quantity for the next 10 day
Website implementation	Flask website for providing retailers a simple user interface to generate the forecast values

Table 4.1 : Modules

4.2 MODULES DESCRIPTION

4.2.1 VISUALIZE DEMAND

The first module of the work is visualizing the demand over time. The retail store has demand of stocks every day which is recorded and visualized as a graph. From this we could figure out the trend in demand of the stock over time.

Here in the proposed work the previous demand history of a specific stock from the retail store data is used as the input to visualize the demand over time. The visualization achieved in the proposed system is as,

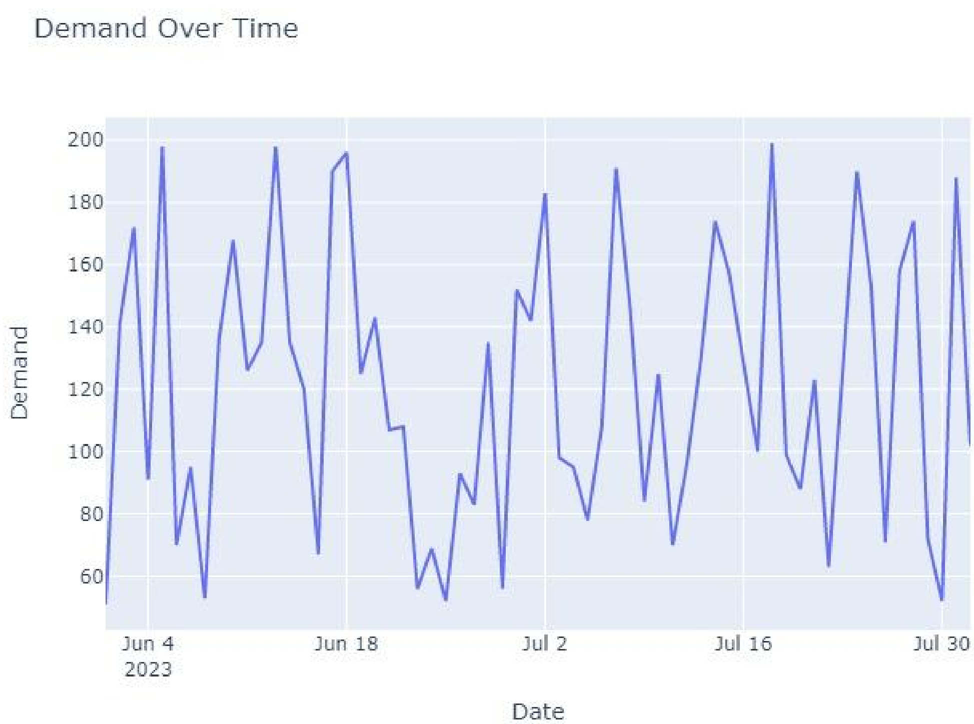

Fig 4.1 : Demand visulaization

4.2.2 VISUALIZE INVENTORY

The second module of the work is visualizing the inventory level over time. As the demand of the stock increases and decreases day wise, the inventory level changes and gets reduced reciprocally to it. Thus this trend of inventory level is visualized over time to view the stock flow in the retail store.

Here in the proposed work the inventory datasedt from the retail store is used as the input to visualized the inventory level over time. The seasonal trend and the stock moving of the retail show can be easily understood by this graphical visualization. The visualization of the inventory over time used in the proposed system dataset is as,

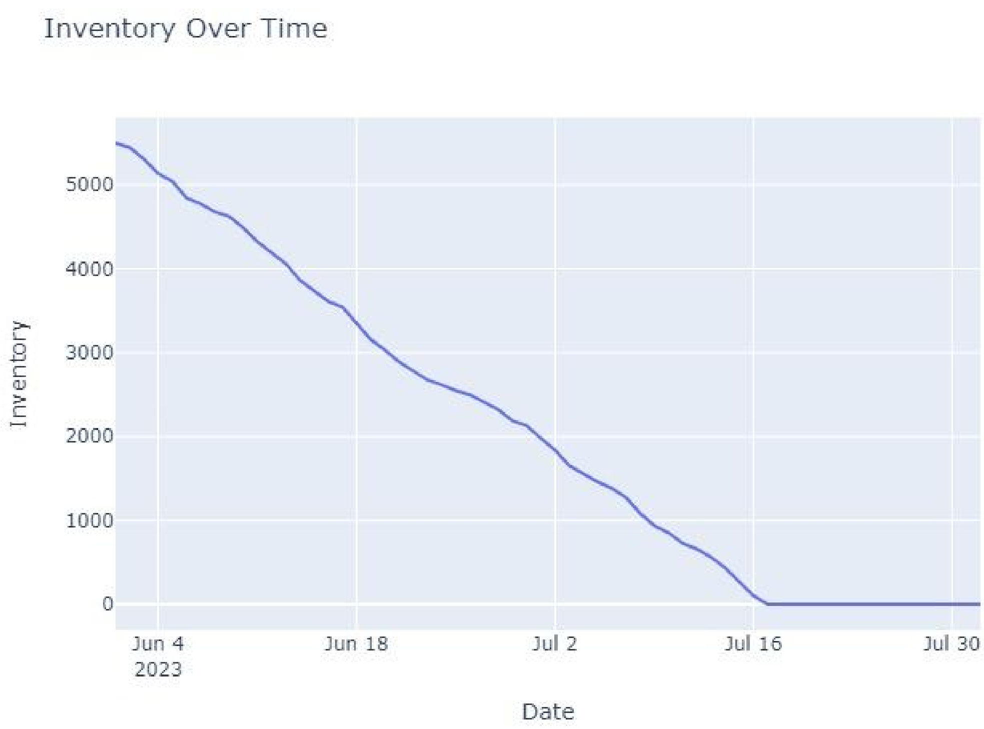

Fig 4.2 : Inventory visulaization

4.2.3 DEMAND FORECASTING

The third module of the work is demand forecasting. Demand forecasting is the concept where the demand of the particular stock is evaluated. Here the proposed work uses SARIMA model which is time series forecasting algorithm used in forecasting the demand in different time series scenarios.

The SARIMA model to forecast the demand needs the parameters p(auto-regressive order), d(degree of differencing), q(moving-average order), P(seasonal AR order), D(seasonal differencing) and Q(seasonal MA order). To calculate this the ACF (Autocorrelation function) and PACF (Partial Autocorrelation function) plots are laid. Thus the order and seasonal order are found at the end of this module to fit with the SARIMA model.

The ACF (Autocorrelation function) and PACF (Partial Autocorrelation function) plots laid for the proposed work is as,

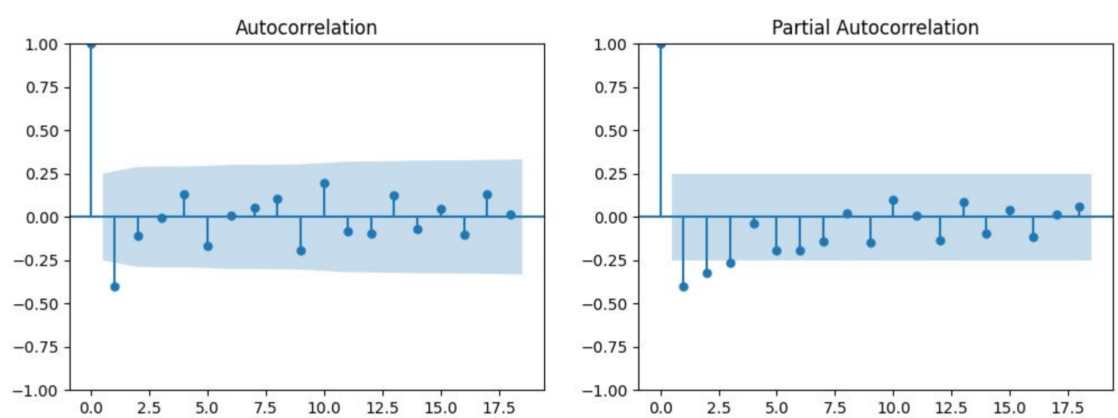

Fig 4.3 : ACF & PACF plot

4.2.4 INVENTORY OPTIMIZING

The fourth module of the work is the inventory optimizing. This is the important module as maintaining the optimal inventory is based on the model used in here. In this module the inventory parameters to maintain optimal inventory level and other values are calculated by fitting the parameters evaluated from demand forecasting to the SARIMA model and using the Newsvendor formula within.

Newsvendor formula is from the Newsvendor model which is used here to calculate the optimal inventory quantity that is to be maintained initially by the retailer in the store, restock point at which the inventory is restocked, safety stock to cater the customers at time of uncertainity.

The Newsvendor formula used in the proposed work is given as,

```
# Calculate the optimal order quantity
z = np.abs(np.percentile(forecasted_demand, 100 * (1 - service_level)))
order_quantity = np.ceil(forecasted_demand.mean() + z).astype(int)

# Calculate the reorder point
reorder_point = forecasted_demand.mean() * lead_time + z

# Calculate the optimal safety stock
safety_stock = reorder_point - forecasted_demand.mean() * lead_time
```

Fig 4.4 : Newsvendor formula

Here the forecasted demand is the Pandas series created with the predicted values and date indices. Sevice level is the probability of the stock not stocking out in the store. With these values from the demand forecast module the optimal inventory parameters are calucluated in this module.

4.2.5 MODULE 5 - WEBSITE IMPLEMENTATION

A flask website is developed using VScode IDE and Spyder IDE. Here the flask website provides the user interface for the retailer where the retailer can give generate values to view the forecasted values for optimal inventory i.e optimal order quantity, reorder point to refill inventory, safety stock to maintain in inventory for catering the need of customers in uncertainity and the total cost to mainatain this.

The flask website is developed with a simple user interface that has a button ‘Generate’ which on click directs the user to another page where the forecasted value from module 4 is displayed in the website. With this forecasted value the user can maintain optimal inventory thus achieving an optimal inventory management in the retail store.

4.3 ALGORITHMS

4.3.1 TIME SERIES ALGORITHM

Time series is a machine learning technique that forecasts target value based solely on a known history of target values. It is a specialized form of regression, known in the literature as auto-regressive modeling. There is not a single best technique to solve time-series forecasting problems [11]. In order to deal with time-series forecasting, each problem might be solved with a different approach. Moving Average (MA) is one of the simplest prediction techniques for making projections about time-series without a noticeable seasonal pattern [12]. In several papers, a more advanced version of MA called Autoregressive Integrated Moving Average (ARIMA) has been used. For instance, Ramos, Santos, and Rebelo (2015) used ARIMA and exponential smoothing method to compare the performances of these methods on forecasting the retail sales of women's footwear. [13]. The purpose of forecasting data points is to provide a basis for production control and production planning and to optimize industrial processes and economic planning.[14]

Seasonal ARIMA (SARIMA) is another classical forecasting method. This technique has been used successfully in various applications. SARIMA have different components as Seasonal (S), Aitoregressive (AR), Integrated (I), Moving Average (MA).

In an autoregression model, we forecast the variable of interest using a linear combination of past values of that variable. The term autoregression indicates that it is a regression of the variable against itself. That is, we use lagged values of the target variable as our input variables to forecast values for the future. An autoregression model of order p will look like:

$$mt = 0 + 1mt-1 + 2mt-2 + 3mt-3+\ldots+ pmt-p$$

In the above equation, the currently observed value of m is a linear function of its past p values. The autocorrelation function (ACF) is the correlation between the current and the past values of the same variable. Partial Autocorrelation (PACF) on the other hand measures only the direct correlation between past values and current values.

Integrated represents any differencing that has to be applied in order to make the data stationary. Moving average models uses past forecast errors rather than past values in a regression-like model to forecast future values. A moving average model can be denoted by the following equation:

$$mt = 0 + 1et-1 + 2et-2 + 3et-3+\ldots+ qet-q$$

In the above equation, e is called an error and it represents the random residual deviations between the model and the target variable. SARIMA stands for Seasonal-ARIMA and thus the seasonality contribution is important to the forecast. The importance of seasonality is quite evident and ARIMA fails to encapsulate information implicitly. Here the SARIMA model is used to forecast the inventory in the retail store where the seasonal order is caluclated from the ACF and PACF plots.

4.3.2 NEWS VENDOR MODEL

The newsvendor model is a core concept in supply chain and inventory management. The premise is simple. Imagine a vendor, selling newspapers on the street. Each morning, they have one chance to buy newspapers in bulk from the printer. How many copies of today's paper should the vendor stock, knowing that unsold copies end up worthless?

All things considered, the newspaper vendor's dilemma is easy to solve, but the method used to solve it can be applied to a wide variety of more complicated problems. Here in retail store this news vendor formula can be used to calculate the optimal inventory values.

One core concept of the newsvendor is that you should favor outcomes that are less painful. That is to say, if you are going to be wrong, would you prefer to be overstocked or understocked? And just how far are you willing to go? The newsvendor model provides a structured way to think through such decisions and choose a stocking point in the face of uncertainty.

Here in retail store it is prefered to keep safety stock instead to be understocked at the time of uncertainity. With this the optimal safety stock to maintain can be calculated to cater the customers at the time of uncertainity.

There are few challenges with this News vendor model as calculating the overage or underage coost could be complex especially for products with shell life. Overage costs are generally more straightforward, but a common trap is to assume all inventory will get used eventually.

4.4 TESTING

Software testing techniques are methods used to design and execute tests to evaluate software applications. There are different testing methods available to test the working and the functioning of the application. Here the developed SARIMA model to forecast and manage the retail store stock inventory is evaluated for the working using different testing methods.

4.4.1 TESTING METHODS

4.4.1.1 UNIT TESTING

Unit testing involves testing of the software as individual units such that functioning of each of these units is verified and validated. This testing is typically done and is used to ensure that the individula units of the software are working as intended. Unit testing helps identifying bugs in early stage of development before it becomes difficult and expensive to fix.

Here in the proposed work, unit testing is carried out by initially diving the systen into smaller units. The proposed system can be divided into different units to check for the working. as, plotting the demand over time, plotting the inventory over time, plotting ACF & PACF, fitting the model, forecasting the result using NewsVendor formula and website unit.

Here each of the unit is tested individually as plotting is working as expected with the imported libraries, SARIMA model fits with the seasonal order evaluated with the plots and parameters from the dataset, the NewsVendor formula forecasts the resul for the optimal inventory management and finally the website unit runs successfully displaying the forecasted value.

4.4.1.2 INTEGRATION TESTING

Integration testing involves testing of the different components of the software such that these components work propoerly when working together and yields the expected results. It is used to identify and resolve any issues that may arise with different units of the software are combined. Integration testing is done after testing the unit testing to verify that the different units of the software work together as intended.

Here the different modules of the software can be integrated to work together to build the SARIMA model for retail store stock inventory analysis to maintain optimal inventory in the retail store.

The different modules integrated and verified are, plotting the ACF & PACF to obtain seasonal order and using this seasonal order to fit the SARIMA model, integrating the fitted SARIMA model with the NewsVendor model where NewsVendor formul is used to forecast the optimal inventory parameters as optimal order quantity, reorder point of the inventory, safety stock to cater the customer at times of uncertainity and estimated total cost to manage the inventory.

Then finally the integration functioning of the whole SARIMA model with the flask website is checked by starting the website from the IDE and redirecting to the port it runs. There the forecasted values are generated at the backend and viewed by the retailers.

Overall the integrated working of the different units works as intended and the errors and bugs created as the time of developemnt is addressed for the seamless finally output.

4.4.1.3 REGRESSION TESTING

Regression testing is the method of testing that is used to ensure that the changes made to the software do not introduce new bugs or cause existing functionality to break. This regression testong is done after making changes and updates to the code or after creating new features, fixing bugs in the code. This is thus used to verify that the software still works as intended.

Regression testing is perfromed in different ways as,

1. Retesting : Involves testing the entire application or specific functionality that was affected by the changes. Here in the proposed system retesting was done at different points of time such as testing the website each time the UI is updated and new page is added. Retesting is done with the SARIMA model at times when the object is created and the values are generated.

2. Re-execution : Involves running a previously executed test suite to ensure that the changes did not break any exisitng functionality. Re-execution is done by starting the website each time the changes when features where added in the machine learning code and erros were fixed in the machine learning model.

3. Comparison : Involves comparing the current version of the software with a previous version to ensure that the changes did not break any existing functionaloty. Here in the proposed work comparison testing is carried out by checking the forecast values by changing the parameters in the model and making sure the working of the other functionality on changes made with the machine learning model in the backend.

CHAPTER 5

RESULTS & DICUSSIONS

Using the time series algorithm and News vendor model the retail store stock inventory analytics is done. Among the different time series algorthm the SARIMA algorithm is used. This is by analysing the different time series algorithm like ARIMA, exponential smoothing, LSTM and other.

SARIMA with the seasonal component provides flexibility with the time series dataset. It is more efficient and provides less error value compared to other value. The accuracy is more optimal compared with other efficient model.

News vendor model is the efficient model used as core in inventory management. This is a powerful model that combining this with the SARIMA model forecasts the optimal inventory parameter as optimal order quantity, reorder point, safety stock and total cost in maintaining the optimal quantity.

CHAPTER 6

CONCLUSION

The proposed SARIMA model digitized the retail store stock inventory management in an efficient way using the Newsvendor formula to calculate the optimal inventory needed fulfill the customers while benefiting the retailers with maximum profit.

The parameters forecasted to maintain optimal inventory for the reatil store stock are, optimal order quanity that gives the order quanitity that the retailer can make when the stock is low, reorder point that gives the point of inventory at which the reorder can be made so the reatiler won't run out of stock, safety stock that is to be maintained apart from the inventory to cater the customers at the time of uncertainity and finally the estimated cost for maintaing this inventory suggested so that user can plan his finances accordingly.

Thus the retail store stock inventory analytics for optimal inventory management using SARIMA time series algorithm and NewVendor model.

REFERENCES

1. Retno Mumpuni, Sugiarto & Rais Alhakim (2020). *Design and Implementation of Inventory Forecasting System using Double Exponential Smoothing Method*. 2020 Information Technology International Seminar (ITIS)

2. Wen Hu & Xinchao Zhang (2020). *Commodity sales forecast based on ARIMA model residual optimization.* 2020 5th International Conference on Communication, Image and Signal Processing (CCISP)

3. Mohankumari C., Vishukumar M., & Nagaraja Rao Chillale (2019). *Analysis of daily stock trend using ARIMA model.* International Journal of Mechanical Engineering and Technology (IJMET)

4. Fernando Rojas (2016). *A methodology for stochastic inventory modelling with ARMIA triangular distribution for new products.* 2017 Cogent Business & Management

5. Charles C. Holt (2004). *Forecasting seasonals and trends by exponentially weighted moving averages.* 2004 International Journal of Forecasting.

6. Uma Devi, B., Sundar, D., & Dr. Ali, P. (2013). *An Effective Time Series Analysis for Stock Trend Prediction Using ARIMaA Model for NIfty Midcap-50*. International Journal of Data Mining & Knowledge Management Process.

7. Wang, J.J., Wang, J.Z., Zhang, Z.G., & Guo, S.P. (2012). *Stock index forecasting based on a hybrid model.* 2012 Omega Volume 40.

8. Sabhyata Lamichhane, Bin Mei & Jacek Siry (2023). *Comparison between a time series hybrid model and an artificial neural network*. Forest Policy and Economics Volume 154.

9. Rojas F., and Leiva V. (2016). *Inventory management in food companies with demand statistically dependent*. Academia Revista Vol 29.

10. Nitin Merh, Vinod Prakash Saxena & Kamal Raj Pardasani (2010). *A comparison between Hybrid Approaches of ANN and ARIMA for Indian Stock Trend Forecasting*. Business Intelligence Journal.

11. G.P. Zhang, D.M. Kline (2007). *Quarterly time-series forecasting with neural networks*. IEEE Transactions on Neural Networks.

12. S. Chopra, P. Meindl (2016). Supply chain management: Strategy, planning, and operation. Das Summa Summarum des Management.

13. Yasaman Ensafi, Saman Hassanzadeh Amin, Guoqing Zhang, Bharat Shah (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. International Journal of Information Management Data Insights, Volume 2, Issue 1.

14. Hussan Al-Chalabi, Yamur K. Al-Douri and Jan Lundberg (2018). Time Series Forecasting using ARIMA Model A Case Study of Mining Face Drilling Rig. The Twelfth International Conference on Advanced Engineering Computing and Applications in Sciences.

15. E.V. Gijo (2014). Demand forecasting of tea by seasonal ARIMA model. International Journal of Business Excellence, 2011 Vol.4 No.1.

16. Subhash Arun Dwivedi, Amit Attry, Darshan Parekh, Kanika Singla (2021). Analysis and forecasting of Time-Series data using S-ARIMA, CNN and LSTM. 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).

17. Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin (2018). *A Comparison of ARIMA and LSTM in Forecasting Time Series*. 17th IEEE International Conference on Machine Learning and Applications.

18. Contreras, Javier, et al. (2003). *ARIMA models to predict next-day electricity prices*. IEEE transactions on power systems

19. Elsworth, Steven, and Stefan Güttel (2020). *Time series forecasting using LSTM networks: A symbolic approach.*

20. Yu, Yong, et al (2019) *A review of recurrent neural networks: LSTM cells and network architectures.*

21. Hyndman, Rob J.and George Athanasopoulos (2018) *Forecasting: principles and practice*. Texts.

22. T.H. McCoy, A.M. Pellegrini, R.H. Perlis (2018). *Assessment of time-series machine learning methods for forecasting hospital discharge volume*. *JAMA Netw Open.*

23. T. Vantuch and I. Zelinka (2014). *"Evolutionary based ARIMA models for stock price forecasting"*. Interdisciplinary Symposium on Complex Systems, vol. 14.

24. G.E. Box, G.M. Jenkins, G.C. Reinsel and G.M. Ljung, (2016). *Time series analysis: forecasting and control*, 4th ed, John Wiley & Sons.

25. G.P. Zhang (2003). *"Time series forecasting using a hybrid ARIMA and neural network model"*. Neurocomputing, vol. 50.

APPENDIX I - SOURCE CODE

Source Code of the SARIMA model - App.py

```
import pandas as pd
import numpy as np
import plotly.express as px
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX

data = pd.read_csv("demand_inventory.csv")
print(data.head())
data = data.drop(columns=['Unnamed: 0'])
fig_demand = px.line(data, x='Date',
                    y='Demand',
                    title='Demand Over Time')
fig_demand.show()
fig_inventory = px.line(data, x='Date',
                       y='Inventory',
                       title='Inventory Over Time')
fig_inventory.show()
data['Date'] = pd.to_datetime(data['Date'],
                                   format='%Y/%m/%d')
time_series = data.set_index('Date')['Demand']
differenced_series = time_series.diff().dropna()


# Plot ACF and PACF of differenced time series
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
plot_acf(differenced_series, ax=axes[0])
plot_pacf(differenced_series, ax=axes[1])
plt.show()
order = (1, 1, 1)
seasonal_order = (1, 1, 1, 2)
model = SARIMAX(time_series, order=order, seasonal_order=seasonal_order)
model_fit = model.fit(disp=False)

future_steps = 10
predictions = model_fit.predict(len(time_series), len(time_series) +
future_steps - 1)
predictions = predictions.astype(int)
print(predictions)
```

```python
# Create date indices for the future predictions
future_dates = pd.date_range(start=time_series.index[-1] + pd.DateOffset(days=1),
periods=future_steps, freq='D')

# Create a pandas Series with the predicted values and date indices
forecasted_demand = pd.Series(predictions, index=future_dates)

# Initial inventory level
initial_inventory = 5500

# Lead time (number of days it takes to replenish inventory)
lead_time = 1

# Service level (probability of not stocking out)
service_level = 0.95


# Calculate the optimal order quantity using the Newsvendor formula
z = np.abs(np.percentile(forecasted_demand, 100 * (1 - service_level)))
order_quantity = np.ceil(forecasted_demand.mean() + z).astype(int)


# Calculate the reorder point
reorder_point = forecasted_demand.mean() * lead_time + z


# Calculate the optimal safety stock
safety_stock = reorder_point - forecasted_demand.mean() * lead_time


# Calculate the total cost (holding cost + stockout cost)
holding_cost = 0.1  # it's different for every business, 0.1 is an example
stockout_cost = 10  # # it's different for every business, 10 is an example
total_holding_cost = holding_cost * (initial_inventory + 0.5 * order_quantity)
total_stockout_cost = stockout_cost * np.maximum(0, forecasted_demand.mean() *
lead_time - initial_inventory)


# Calculate the total cost
total_cost = total_holding_cost + total_stockout_cost
print("Optimal Order Quantity:", order_quantity)
print("Reorder Point:", reorder_point)
print("Safety Stock:", safety_stock)
print("Total Cost:", total_cost)
```

Source Code of the Flask website - Web.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov  2 18:36:01 2023

@author: LENOVO
"""
from flask import Flask, render_template
import pickle
import plotly

app=Flask(__name__)
order_quantity=pickle.load(open('models/quantity.pkl','rb'))
reorder_point=pickle.load(open('models/reorder.pkl','rb'))
safety_stock=pickle.load(open('models/safety.pkl','rb'))
demand_plot=pickle.load(open('models/demandplt.pkl','rb'))

@app.route('/')
def home():
   return render_template('base.html')

@app.route('/generate')
def generate():
   return
render_template('generate.html',o1=order_quantity,o2=reorder_point,o3=safety_sto
ck)

@app.route('/plot')
def plot():
    return render_template('plot.html')
   #return plotly.offline.plot(demand_plot,
filename="D:/Documents/Activities/PRIEE/Spyder_Code/templates/plot.html")

if __name__ == "__main__":
    app.run()
```

Source Code of the Base webpage - base.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>The retail store stock inventory analytics</title>
</head>
<body style="background-color: #F7EFE5; margin: auto; width: 50%; text-align:
center;">
    <h1>Forecast Demand and Optimize inventory</h1>
    <div style="margin: auto; width: 50%;">
        <!--
        <form action="/plot">
            <p>View demand over time plot</p>
            <button>View</button>
        </form>
        <br>
        -->
        <form action="/generate">
            <h4>Generate reorder point and optimal order quantity</h4>
            <button>Generate</button>
        </form>
    </div>
</body>
</html>
```

Source Code of the Generate webpage - generate.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>The retail store stock inventory analytics</title>
</head>
<body style="background-color: #F7EFE5; margin: auto; width: 50%; text-align: center;">
    <h1>The forecasted </h1><br><br>
    <div style="margin: auto; width: 80%;">
        <h3>Reorder level : {{ o2 }}</h3>
        <p>When the inventory level falls to this <b>{{ o2 }}</b> point restock the inventory</p><br>
        <h3>Optimal inventory quantity : {{ o1 }}</h3>
        <p>Restock the inventory with the <b>{{ o1 }}</b> quantity when the inventory falls to the reorder level calculated</p><br>
        <h3>Safety stock : {{ o3 }}</h3>
        <p>Safety stock of <b>{{ o3 }}</b> is maintained to cater at time of uncertainity</p>
    </div>
</body>
</html>
```

APPENDIX II - SCREENSHOTS

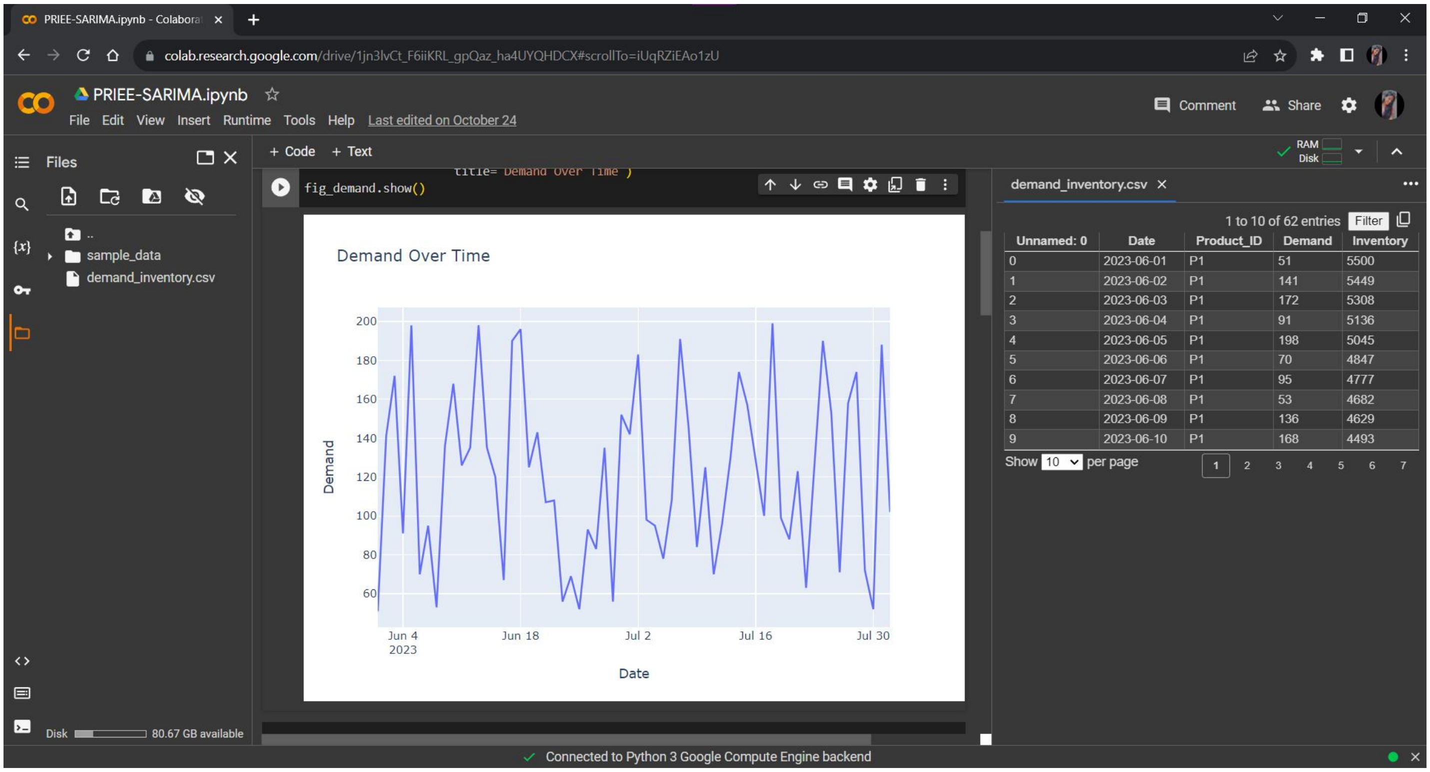

Fig 1 : Visualizing the demand of the stock over time using plotly

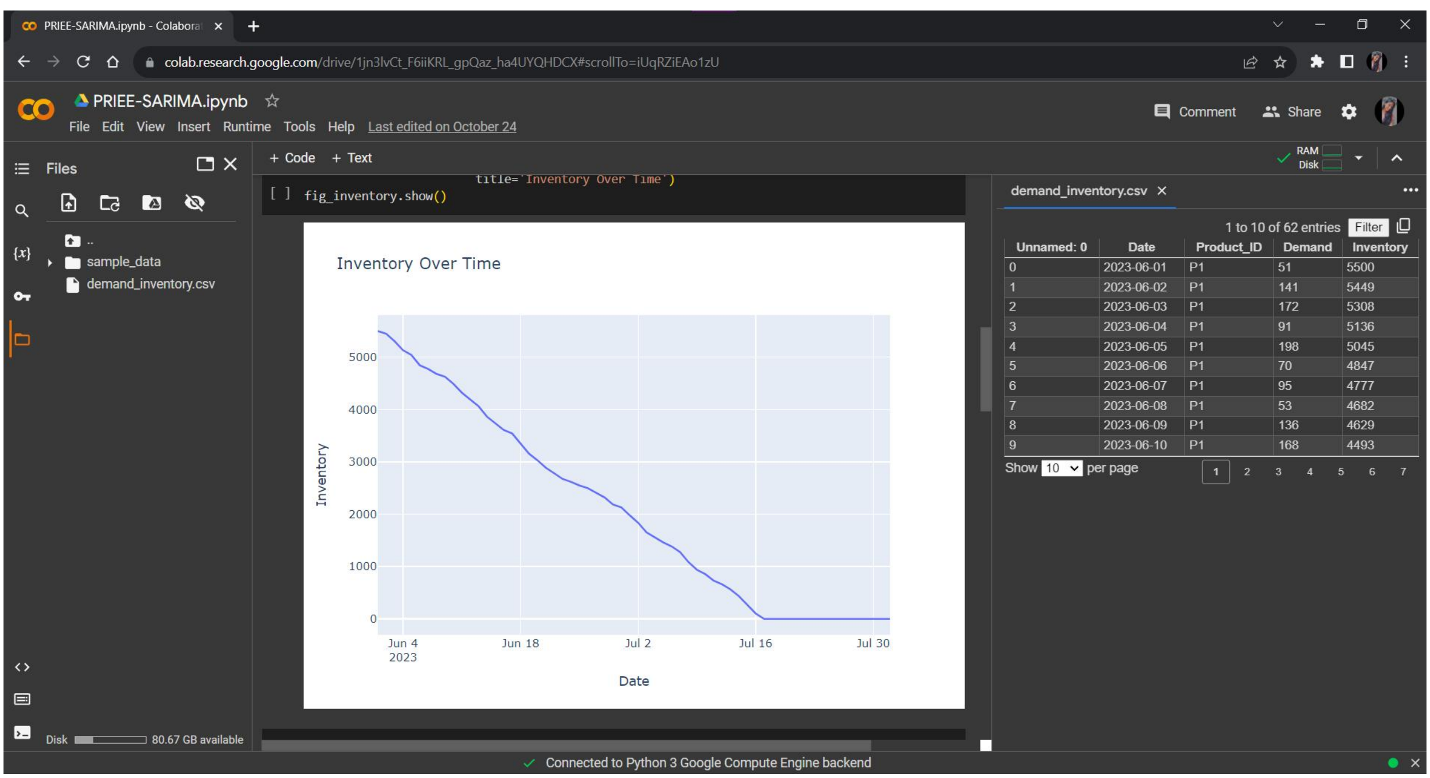

Fig 2 : Visualizing the inventory level of the stock over time using plotly

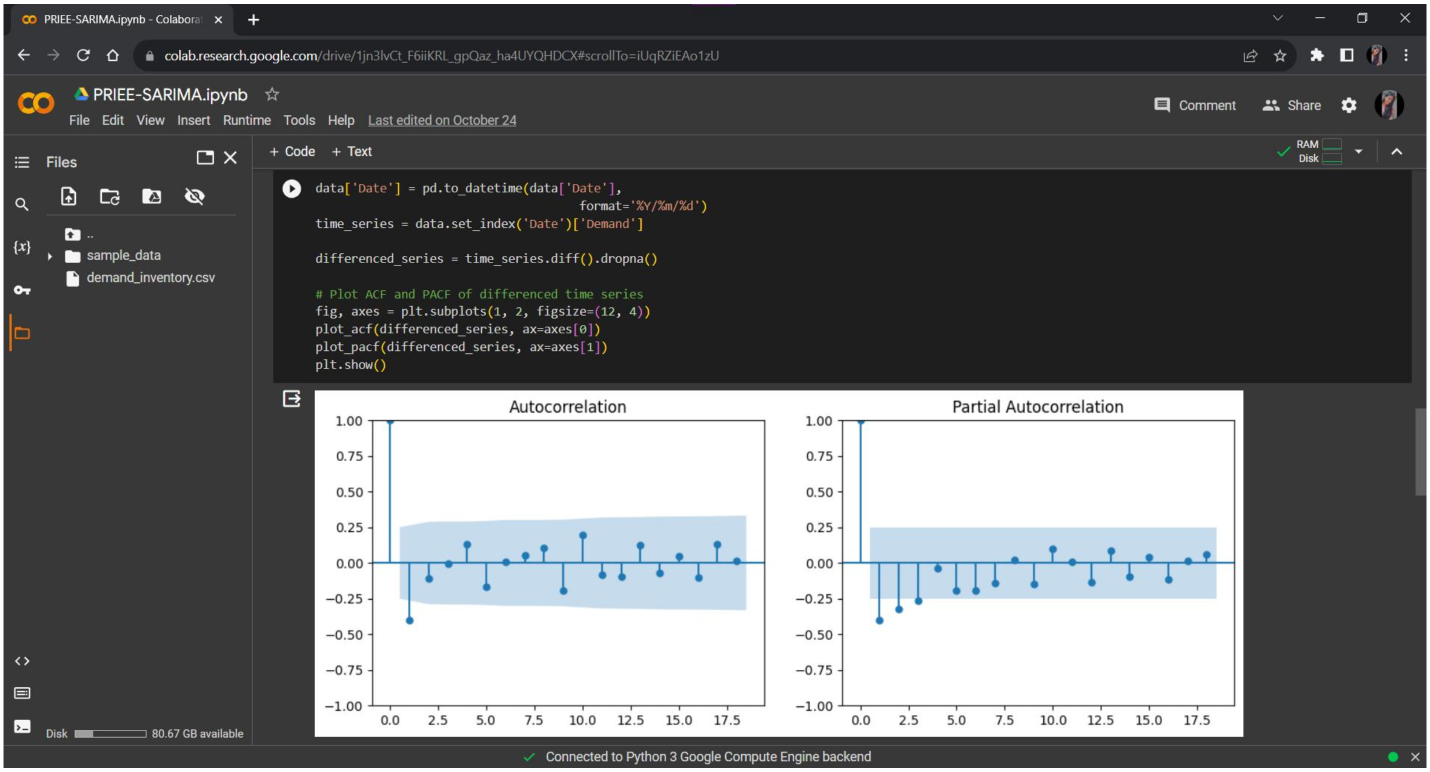

Fig 3 : Plotting the ACF and PACF graphs of the data for calculating the order for the SARIMA model

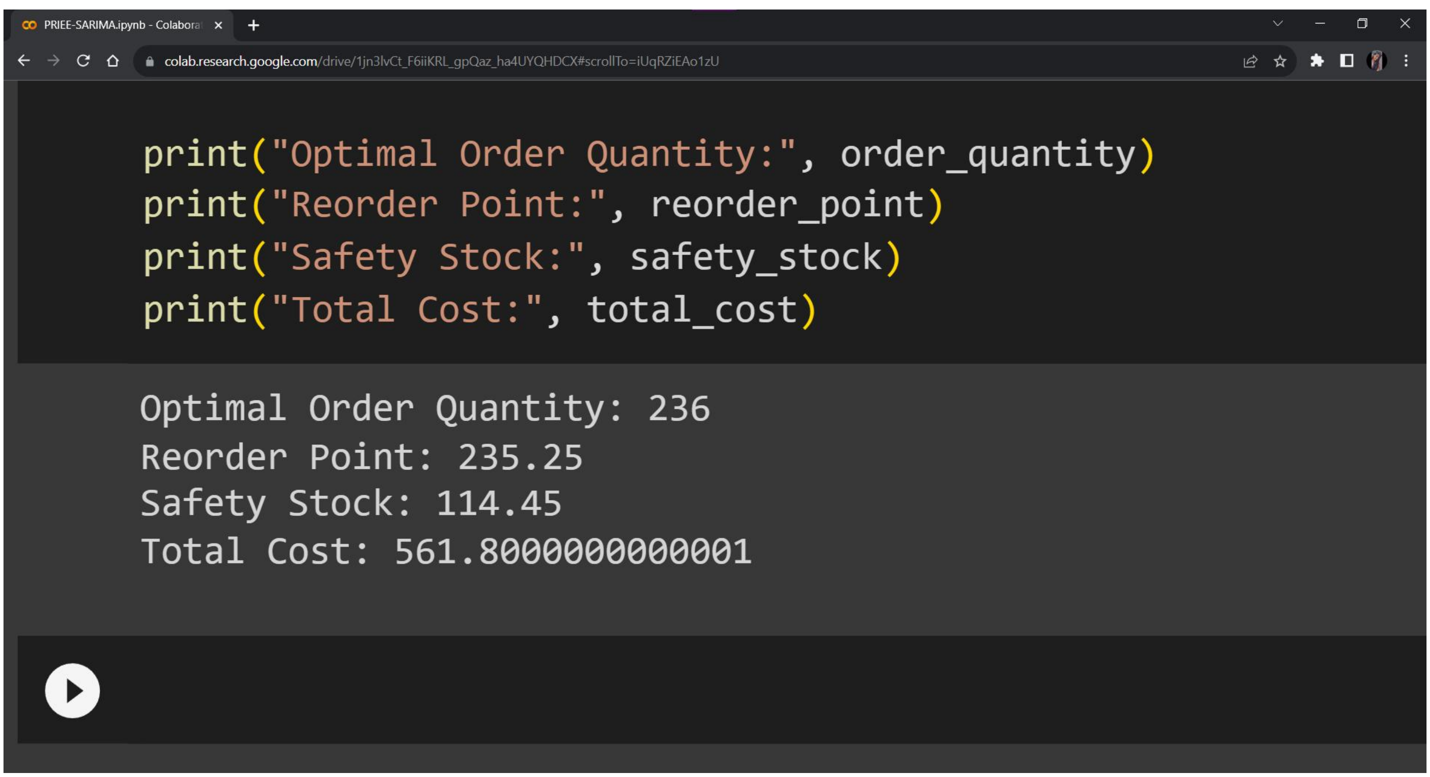

Fig 4 : Results of the forecast producing the optimal order quantity, reorder point and safety stock for the next 10 days for the retail store

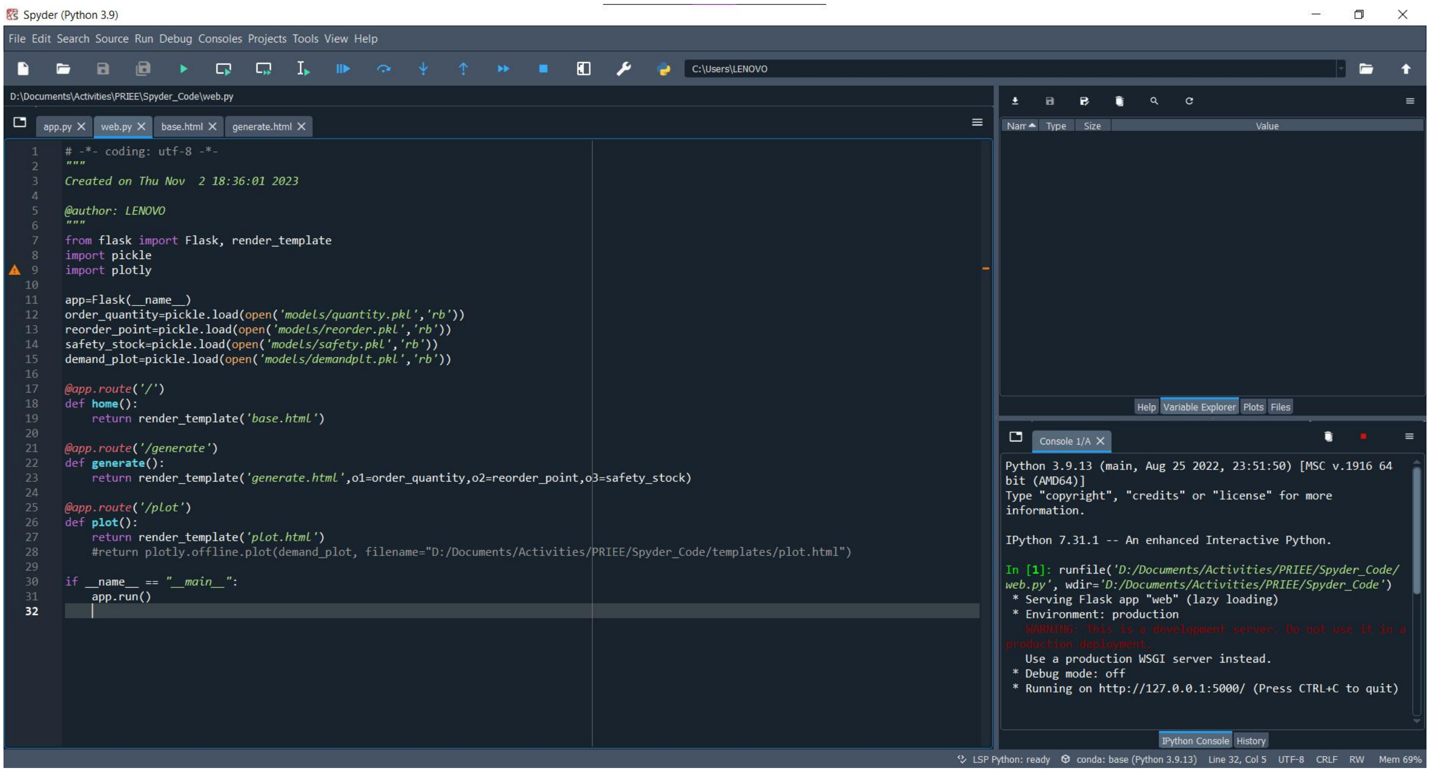

Fig 5 : SARIMA model is implemented as a web application with Flask framework

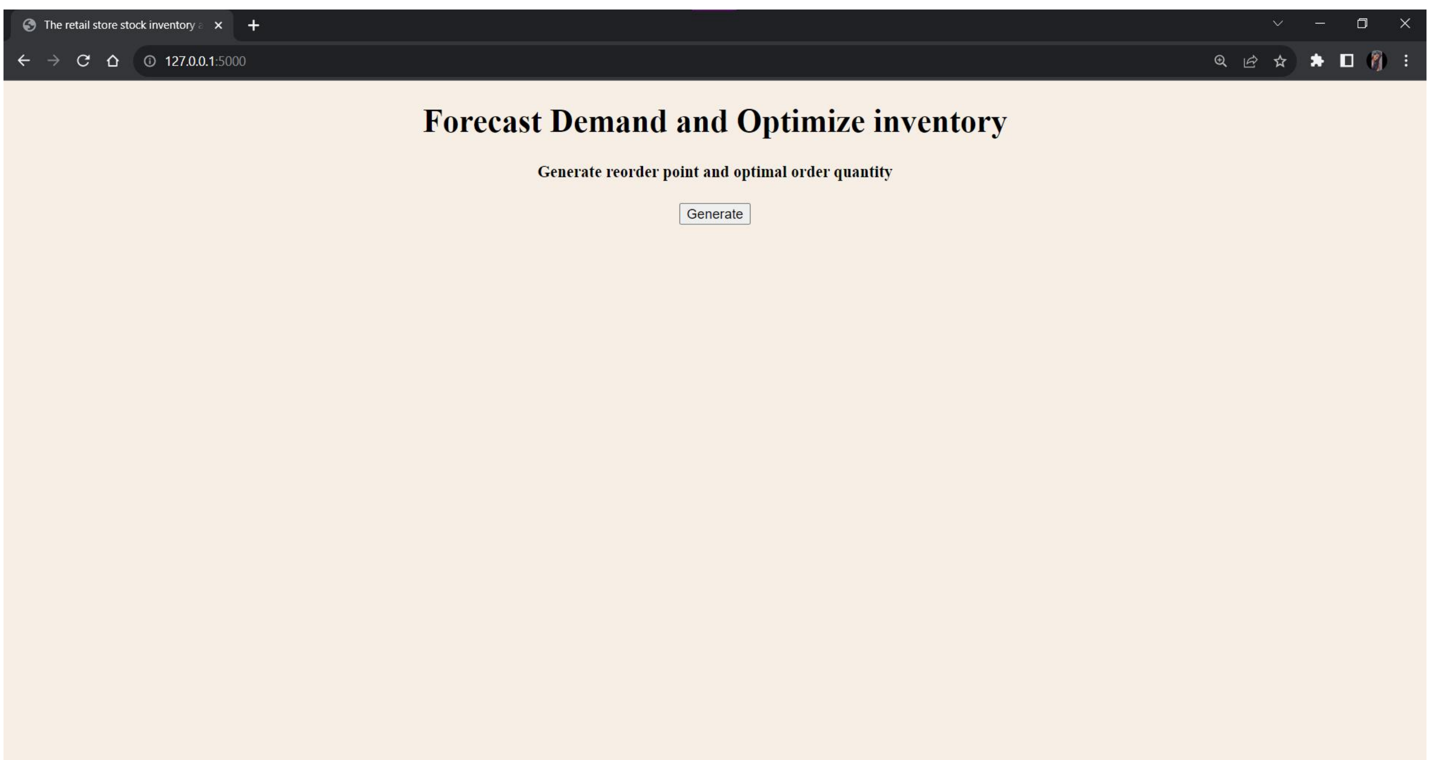

Fig 6 : A simple overview of the web application GUI where the retailer can generate the forecast value for the next 10 days for his store

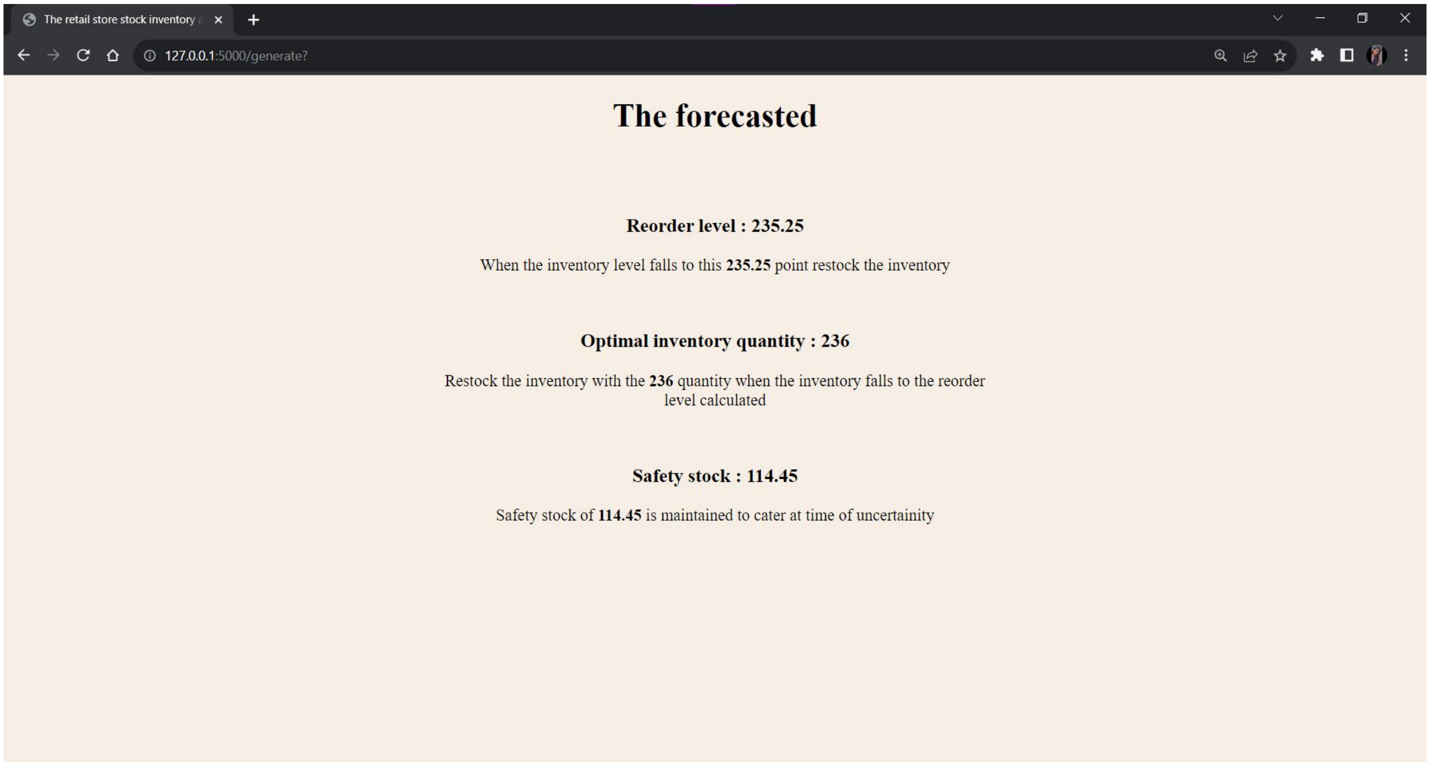

Fig 7 : The website displaying the forecasted value to the retailer