A Major Project Final Report on

Smart Pharma Demand Forecasting

Submitted in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Engineering in **Software Engineering**
under Pokhara University

Submitted by:

Bishwajyoti Chaudhary, 181709

Prabin Kumar Shah, 181728

Deepak Yadav, 181711

Deekshya Maharjan, 181710

Under the supervision of

Rudra Nepal

Date:

08/07/2023

Department of Software Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Acknowledgement

The whole team of the project **"Smart Pharma Demand Forecasting"** is extremely thankful for the valuable guidelines and feedbacks from the respected teachers and supervisor. We are glad and feeling privileged to be able to present our process of transforming idea to a project that simulates the real-world problem solving product.

We are extremely thankful and deep gratitude towards our project coordinator respected **Dr. Roshan Chitrakar** for constantly guiding us towards the best path possible we could implement while doing this project.

We wish to record a deep sense of respect and gratitude to respected **Er. Rudra Nepal** Sir, project supervisor for providing us every kind of support feasible in the journey of idea to a minimal viable product. We would also like to thank him for his valuable guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

This project has a great memory within us in teammates from when we ideate and become more curious towards the implementation. The whole project is the output of efforts from each team members, supervisor, project coordinator and valuable feedbacks in each of the stage of the project presentation.

The journey from the idea to implantation analyzing the real scenario and majority of the issue and trying to draft a simple module to impact greatly was the greatest learning from whole team members. We followed the most efficient and standard approach of engineering the product from very beginning to the end. This project makes us the practically stronger to implement what we have learned from the whole engineering period. We learned things which could not be learned from the theoretical classes and lectures, but we got the chance to implement practically the critical knowledge in real field with aiming to solve the real-world problem.

We would like to thank Nepal college of Information Technology and Pokhara University for such a wonderful design of the syllabus which is impactful to learn and research the real scenario.

Finally, we are too much privileged to learn, implement and improve ourselves in each stage with the help of teachers, colleagues, classmates and friends have been directly or indirectly linked with the project and accomplishing this project.

Abstract

Smart Pharma Demand Forecasting is a web application designed to streamline the supply chain in the pharmaceutical industry. It empowers local pharmacies, stockists and pharmaceutical company to efficiently manage their inventory. This application facilitates collaboration between stockists and local pharmacies, enabling local pharmacies to conveniently order medicines directly from stockists. By eliminating the traditional process of phone calls and paper-based order notes, the application significantly improves efficiency and accuracy. Additionally, the application provides demand forecasting capabilities for pharmaceutical companies. By leveraging historical sales data, sales data patterns, and other relevant factors, the application employs advanced algorithms to generate reliable demand forecasts. This enables pharmaceutical companies to optimize their production schedules, inventory levels, and distribution strategies, ultimately improving overall supply chain management.As a web application, Smart Pharma Demand Forecasting ensures accessibility and flexibility. It can be accessed from any device with an internet connection, allowing local pharmacy to place orders and pharmaceutical companies for demand forecast. The web app is developed using modern web technologies, ensuring a seamless and secure user experience. In conclusion, Smart Pharma Demand Forecasting is a web application that revolutionizes the ordering process in the pharmaceutical industry and forecasts. The application enhances efficiency, reduces errors, and improves supply chain management.

Keywords: *Demand Forecasting, pharmaceutical companies, data patterns, stockist, local pharmacy, web application*

Table of Contents

List of Figures

List of Table:

1. Introduction

Smart Pharma Demand Forecasting is a web application that transforms the supply chain in the pharmaceutical industry. It enables seamless collaboration between local pharmacies and stockists, simplifying the ordering process. By eliminating the need for phone calls and paper-based notes, the application enhances efficiency and accuracy. Moreover, it offers advanced demand forecasting capabilities based on historical sales data and relevant factors, empowering pharmaceutical companies to optimize their production, inventory, and distribution strategies. The web app is easily accessible from any device with an internet connection and provides a secure and user-friendly experience. Overall, Smart Pharma Demand Forecasting revolutionizes pharmaceutical industry operations, improving inventory and streamlining the ordering process and forecasting the demand.

1.1 Problem statement

The traditional process of pharmacies ordering drugs through phone calls and suppliers manually recording orders on paper has resulted in increased chances of errors and a time-consuming process. This poses a significant problem in the pharmaceutical industry as companies struggle to deliver the correct quantity of products to stockists in a timely manner. The primary reason behind this problem is the failure of companies to identify rapid changes in demand quantity and remain unaware of competitor demand within the supply chain management process. This lack of real-time information and forecasting capabilities hampers effective supply chain management and leads to inefficiencies in inventory management and distribution. Therefore, there is a critical need for improved order management systems and accurate demand forecasting methods to address these challenges and enhance the overall performance of the pharmaceutical industry supply chain.

1.2 Project Objectives

To address the drawbacks mentioned in problem statement section, the project aims to evolve by achieving the following objectives:

1. To create a user-friendly web application that enables local pharmacies to place drug orders directly with stockists, eliminating the reliance on phone calls and paper-based order notes.

2. To implement advanced data analysis techniques and machine learning algorithms to achieve accurate demand forecasting, with the aim of minimizing overproduction.

3. To enable local pharmacies, stockists, and pharmaceutical company to monitor and manage inventory levels.

1.3 Significance of the Study

The Smart Pharma Demand Forecasting web app is significant for several reasons. The accurate demand forecasting enables pharmaceutical companies, stockists, and local pharmacies to manage their inventory level. The effective demand forecasting helps pharmaceutical companies, stockists, and local pharmacies to reduce costs associated with excess inventory and stockouts. With accurate demand forecasting, pharmaceutical companies, stockists, and local pharmacies can ensure timely availability of medicines, meeting the needs of customers promptly. The study of a Smart Pharma Demand Forecasting web app enables streamlining of the pharmaceutical supply chain. The data and insights provided by a Smart Pharma Demand Forecasting web app can aid in strategic decision-making for pharmaceutical companies. By accurately forecasting demand, the web app can help prevent out-of-stock situations at local pharmacies.

1.4 Scope and Limitation

1.4.1 Scope:

The Smart Pharma Demand Forecasting project aims to develop a web-based application to optimize demand forecasting and supply chain management by facilitating efficient collaboration among pharmaceutical companies, stockists, and local pharmacies. The scope of the project includes the following:

1. Demand forecasting: Pharmaceutical companies can utilize smart pharma demand forecasting to predict future demand for medicines based on various factor of historical sales data.

2. Inventory management: The system enables local pharmacies, stockists and pharmaceutical company to manage their inventory levels effectively.

3. Seamless ordering process: Local pharmacies can place orders directly with stockists, streamlining the ordering process and ensuring timely access to required medicines.

1.4.2 Limitation:

Here are some limitations of our system:

1. The accuracy of the app's forecasts will depend on the quality of the data that is used to train it. If the data is inaccurate or incomplete, the forecasts will be less accurate.

2. The app can be complex to use and maintain. This can be a challenge for smaller businesses or organizations that do not have the resources to invest in training and support.

2. Literature Study/Review

2.1 NepMeds:

In context of Nepal, NepMeds app provides a variety of health-related services such as connecting users with doctors, booking diagnostic tests, and ordering medicines and wellness products. The app also offers a wide range of categories for different types of medicines and healthcare products.

However, it seems that the app does not have the feature of demand forecasting, which is essential for pharmaceutical companies to optimize their inventory levels and improve supply chain management. Additionally, local pharmacies are unable to order medicines in bulk, which may limit their ability to save costs and improve their efficiency.

Therefore, there is still a need for a Pharmacy Management System with Demand Forecasting in Nepal that can address these limitations and provide pharmaceutical companies and retail pharmacies with the necessary tools to improve their operations. So we have come up with this idea to resolve the limitation of existing system.

2.2. Demand forecasting model for time-series pharmaceutical data using shallow and deep neural network model:

[6]The literature review explores the limitations of traditional linear forecasting models such as ARIMA, SARIMA, and ARMA in capturing non-linear behavior in time series data. This has led to the rise of machine learning methods, particularly Artificial Neural Networks (ANNs), for time series forecasting. Several studies have demonstrated the effectiveness of ANN-based time series forecasting compared to conventional methods. However, there is a lack of benchmark neural network parameter settings for the time series domain, as it heavily depends on the specific problem. In line with this, the paper focuses on developing ANN-based Demand Forecasting Models (DFMs) for pharmaceutical data. Shallow neural network models and a deep learning model called Long-Term Short Memory Neural Network (LSTM) are utilized. In this research paper, authors used weekly sales data for demand forecasting and sallow neural network, stacked LSTM. But in our app we will use daily sales data and Hybrid Model for data training and improve performance and limitation for above existing model.

3. Team members and Task Divided

Name	Roles	Responsibilities
Deekshya Maharjan	• Frontend Developer • Documentation	• Frontend Development. • End user documentation.
Bishwajyoti Chaudhary	• AI Developer • Project Leader • End user Documentation	• Model Training • Data analysis and visualization • Model Deployment • Review and approve all project deliverables (Initiation Plan, Detailed Plan, Testing etc.) • Develop required modules and integrate asper in design specification. • End user documentation.
Prabin Kumar Shah	• Backend Developer • System designer and Developer. • Database Administrator • Testing and Q/A. • End user Documentation.	• Designing and implementing the backend architecture • System analysis and development. • Design the various entity modeling and relationship between the databases. • End user Documentation.

Deepak Yadav	• Backend Developer • System Analysis and Design • End user Documentation.	• Designing and implementing the backend architecture • System analysis and Design of the feasible system. • End user Documentation.

Table 1: Team members and Task Divided

4. Methodology

4.1 System Work Flow

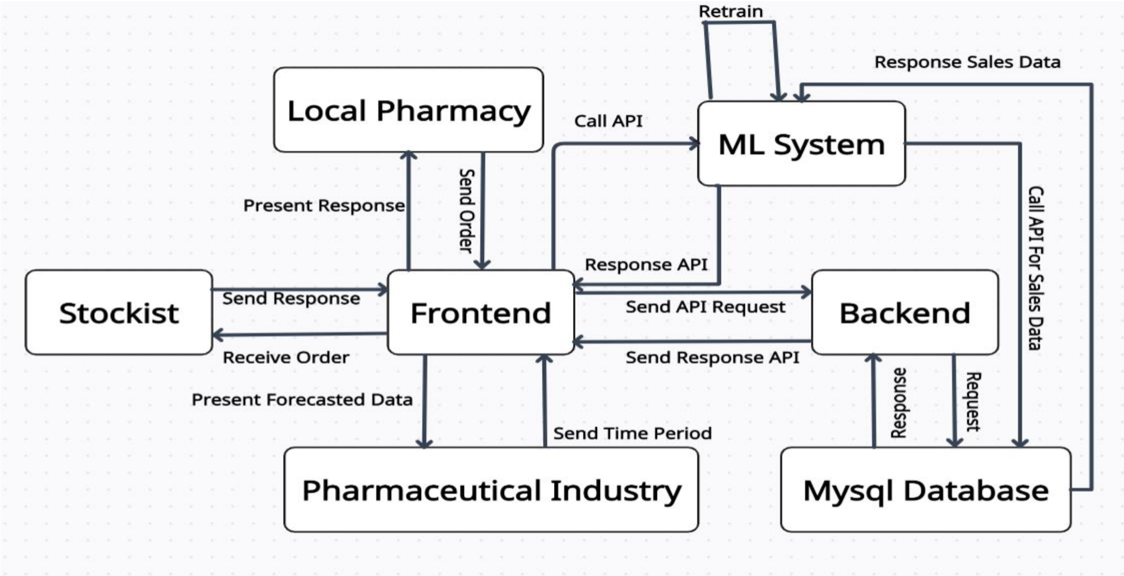


Figure 1: System Work Flow

4.2 Model Development:

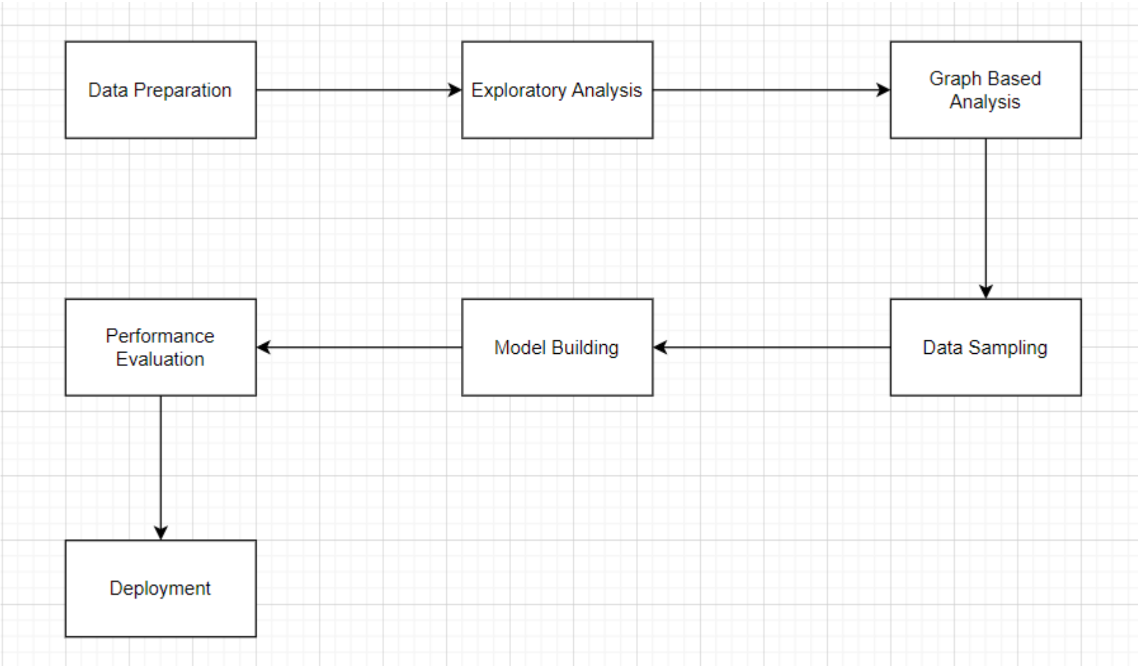


Figure 2: Model Development

Time Series Analysis:

A time series is a sequence of data points that occur in successive order over some period of time.

A time series can be yearly, quarterly, monthly, weekly, daily, hourly, minutes, or even seconds in length, depending on the frequency.

Forecasting is the next step in the process, and it involves predicting the series' future values. When you generate scientific forecasts based on historical time stamped data, you're doing time series forecasting. It entails developing models based on previous data and applying them to make observations and guide future strategic decisions. A key distinction in forecasting is that the future outcome is completely unknown at the time of the work and can only be anticipated by meticulous analysis and evidence-based priors.

Now forecasting a time series can be broadly divided into two types.

- Univariate Time Series Forecasting is when you utilize only the prior values of a time series to predict its future values.

-Multi Variate Time Series Forecasting is when you employ predictors other than the series (also known as exogenous variables) to forecast.

Let's discuss a few definitions related to time series first.

Definitions:

Level: Level is the average of the values of the series.

Trend: Trend shows a pattern in the data. For example, whether the stock prices are increasing with time(uptrend) or are they decreasing with time(downtrend) or time doesn't have that much effect on the prices(Horizontal trend).

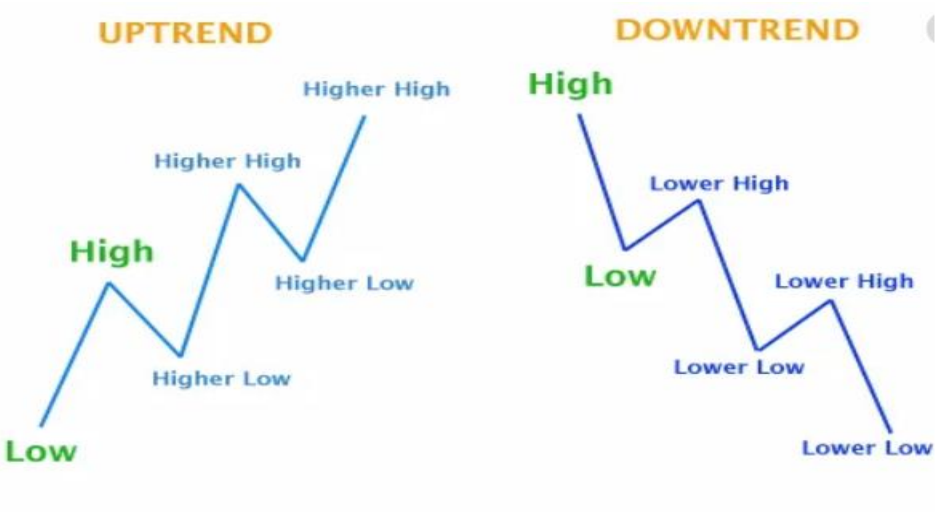


Figure 3: Trend

Seasonality: When the data shows a repetitive pattern for over a year, it can be termed as seasonal pattern.

Cyclic Patterns: These are the repetitive patterns shown over a longer period of time (more than one year).

Noise: The variations which do not show any pattern.

ARIMA models are a type of statistical model that can be used to analyse and forecast time series data. It gives a simple yet powerful way for creating time series forecasts by explicitly catering to a set of common structures in time series data.

ARIMA is an acronym for Autoregressive Integrated Moving Average. It's a more complex version of the Autoregressive Moving Average, with the addition of integration.

An ARIMA model is characterized by 3 terms: p, d, q where,

- p is the order of the AR term. The number of lag observations included in the model, also called the lag order.

- q is the size of the moving average window, also called the order of moving average.

- d is the number of differencing required to make the time series stationary.

There are a number of ways to find values of p, q and d:

- look at an autocorrelation graph of the data (will help if Moving Average (MA) model is appropriate)

- look at a partial autocorrelation graph of the data (will help if Autoregressive (AR) model is appropriate)

- look at extended autocorrelation chart of the data (will help if a combination of AR and MA are needed)

- try Akaike's Information Criterion (AIC) on a set of models and investigate the models with the lowest AIC values

- try the Schwartz Bayesian Information Criterion (BIC) and investigate the models with the lowest BIC values

Before working with non-stationary data, the Autoregressive Integrated Moving Average (ARIMA) Model converts it to stationary data. One of the most widely used models for predicting linear time series data is this one.

The ARIMA model has been widely utilized in banking and economics since it is recognized to be reliable, efficient, and capable of predicting short-term share market movements.

Convolutional Neural Network: Although traditionally developed for two-dimensional image data, CNNs can be used for 1D data. This allows CNN to be used in more general data type including texts and other time series data. Instead of extracting spatial information, you use 1D convolution to extract information along the time dimension. Conv1D: Convolving on time dimension

Architecture:

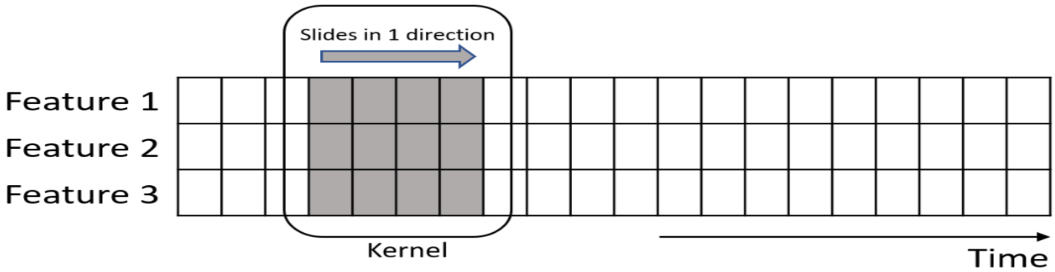


Figure 4: CNN Architecture

A one-dimensional CNN is a CNN with a convolutional hidden layer that operates on a one-dimensional sequence [1]. In some circumstances, such as with very lengthy input sequences, a second convolutional layer may be used, and then a pooling layer is used to reduce the output of the convolutional layer to the most important bits. A dense fully connected layer follows the convolutional and pooling layers, which interprets the features extracted by the convolutional part of the model. Between the convolutional layers and the dense layer, a flatten layer is employed to compress the feature mappings to a single one-dimensional vector.

Long Short Term Memory(LSTM): Long Short Term Memory(LSTM) is a special type of Recurrent Neural Network(RNN) which can retain important information over time using memory cells. This property of LSTMs makes it a wonderful algorithm to learn sequences that are interdependent and can help to build solutions like language translation, sales time series, chatbots, autocorrections, next word suggestions, etc.

Architecture:

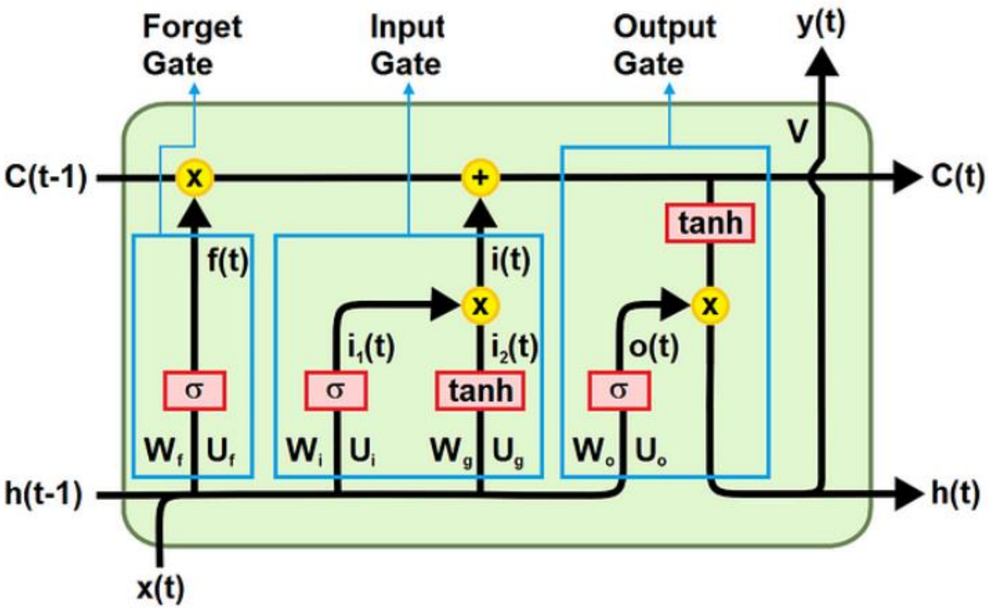


Figure 5: LSTM Architecture

Forget Gate - The first gate is the forget gate. This gate determines whether or not information should be preserved or destroyed. The sigmoid function passes the information from the previous hidden state as well as the information from the current input. If the output is close to 0, the information can be discarded, however if the output is close to 1, the information must be kept.

Input Gate - The input gate is the second gate. This is used to update the status of the cell. Initially, a sigmoid function is given the prior hidden state and the current input as inputs (the closer the output is to 1, the more important the information). In order to optimise network tuning, it additionally feeds the hidden state and current input through a tanh function to compress values between -1 and 1.

Cell State - The cell state may be computed once the input gate has been activated. First, the previous time step's cell state is element-wise multiplied by the forget gate's output. When values in the cell state are multiplied by values near to 0, this allows values in the cell state to be ignored. The input gate's output is then added to the cell state element by element. The new cell state is the output.

Output Gate - The third and final gate is the output gate that decides the value of the next hidden state, which contains information about previous inputs. First, the previous hidden state and current input are summed and passed to a sigmoid function. Then the new cell state is passed to the tanh function. At the end the tanh output with the sigmoid output are multiplied to decide what information the hidden state should contain. The output is the new hidden state. The new cell state and the new hidden state are then carried over to the next time step.

Gated Recurrent Unit (GRU): GRU or Gated recurrent unit is an advancement of the standard RNN. To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate. Basically, these are the two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction. If carefully trained, GRU can perform extremely well even in complex scenarios.

Architecture:

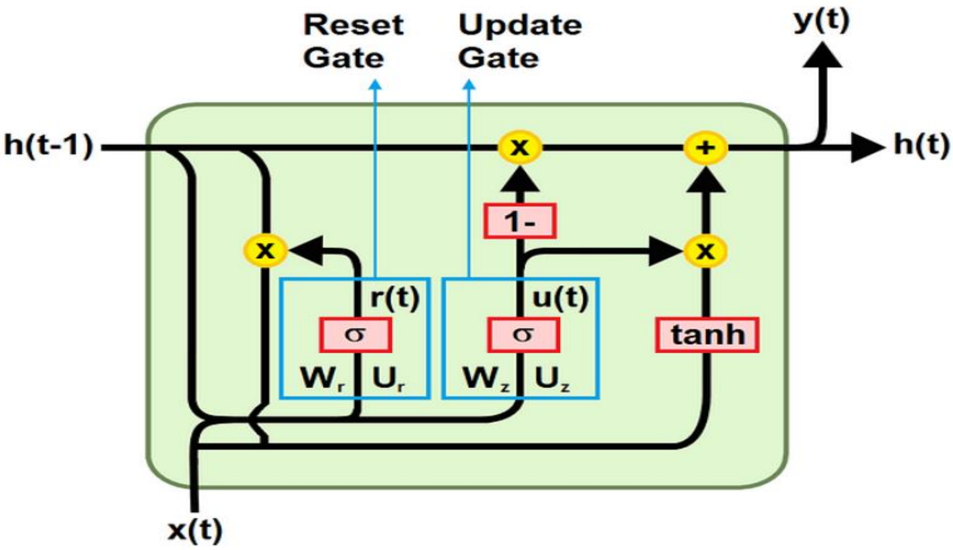


Figure 6: GRU Architecture

As shown in the figure above, a GRU unit is composed of:

Reset Gate -The first gate is the reset gate. It determines how to combine the new input with the previous memory, deciding how much of the information from previous time steps can be forgotten.

Update Gate -The second gate is the update gate. It helps the model to determine how much of the information from previous time steps needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.

Memory - It brings information along the entire sequence and represents the memory of the network.

4.3 Process model:

The framework we followed in developing this project is incremental model, which is a use of linear sequential model in an iterative manner. New functionalities will be added as each increment was developed. Linear sequential model will be applied to develop each increment. The phases of the linear sequential model are: Analysis, Design, Coding and Testing. The software repeatedly passes through these phases in iteration and an increment is delivered with progressive changes.

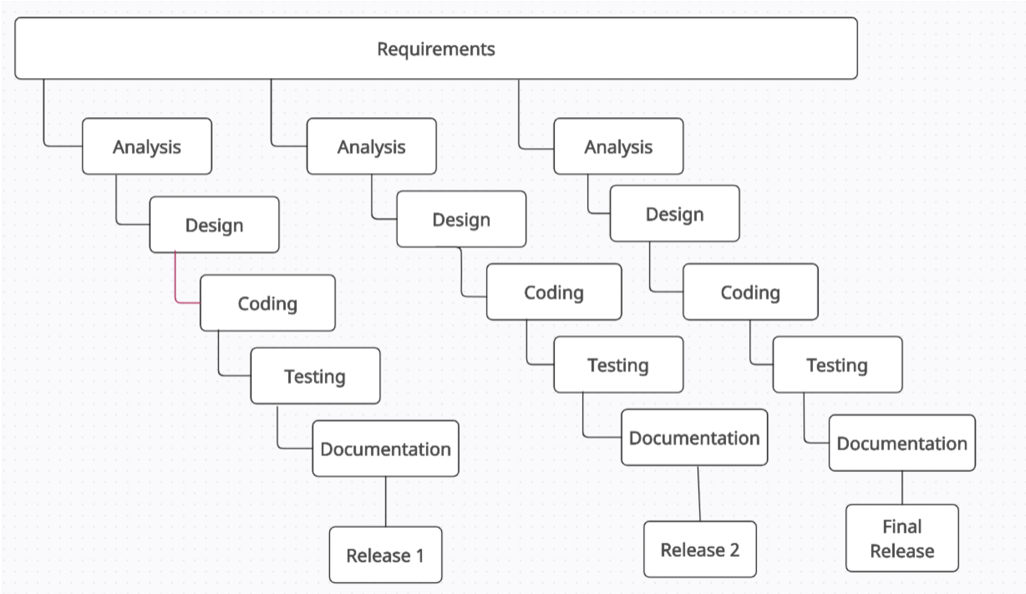


Figure 7: Incremental Order

4.3.1 Analysis Phase:

In this phase, analysis will be done in order to find out the requirements of the system. The outcome of this phase is an SRS which is an acronym for "System Requirement Specifications".

4.3.2 Design Phase:

In this phase the SRS will be translated into the system's design. Context Diagram, DFD, ER-Diagram, Use Case Diagram and Class Diagram will be developed.

4.3.3 Coding Phase:

In this phase coding will done according to the design and a working system will be developed by the end of this process.

4.3.4 Testing Phase:

In this phase, the system will be tested. With each testing a list of changes to the system will be developed, suggested and the changes will be applied to the software and the software will be delivered as a successive increment until a satisfying system will be achieved.

4.4 Programming Language and Other Tools

4.4.1 Programming Language

We have decided to use Java and Python as the programming language for the development of Smart Pharma Demand Forecasting.

Programming Language	Application
Frontend	HTML, CSS, Thymeleaf, Bootstrap, JavaScript
Backend	Java, Python

Table 2: Programming Languages

4.4.2 Database

MySQL is an open-source relational database management system that uses SQL (Structured Query Language) to manage and manipulate data stored in a database.

4.4.3 Framework to Be Used

Below are the major frameworks we have decided to use in the development of our project.

Framework	**Application**
Spring	Spring Framework is an open-source, lightweight, modular framework for building enterprise-grade Java applications.
Bootstrap	Bootstrap is a free and open-source front-end framework for building responsive web applications.
TesnsorFlow	TensorFlow is an open-source machine learning platform developed by Google.

Table 3: Framework

4.4.4 Tools to Be Used

Tools used in design, development and testing of software are mentioned in the table below:

Tools	**Application**

IDE (Visual Studio, STS)	To write code.
Git and GitHub	To manage the project’s source code and versions locally and remotely respectively.
EdrawMax	To design components.
MySQL Workbench	To manage databases.

Table 4: Tools

5. Requirement

Requirement analysis, in software engineering encompasses those tasks that go into determining the need and conditions to meet for a new or altered product, taking account of possibly conflicting requirements of the various stakeholders, such as beneficiaries and users. It is the early stage activity of requirement engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing system requirements.

5.1 Functional Requirements

1. User Authentication: The system supports user authentication and authorization for local pharmacies, stockists, and pharmaceutical companies.

2. Inventory Management: The system allows stockists and pharmaceutical companies to manage their inventory, including adding, updating, and removing medicines.

3. Demand Forecasting: The system utilizes machine learning algorithms to perform demand forecasting for medicines based on historical sales data, market trends, and other relevant factors.

4. Order Placement: Local pharmacies are able to browse and order medicines from stockists.

5.2 Input Requirements

1. Historical Sales Data: The system receives historical sales data from pharmaceutical companies to train the machine learning models for demand forecasting.

2. Inventory Data: Obtain inventory data from stockists and pharmaceutical companies, including product information, quantities, prices and expiration dates.

3. Real-Time Data: Continuously update the system with the latest data to improve the accuracy of demand forecasts and adapt to changing market conditions.

5.3 Output Requirements

1. Demand Forecast: The system should provide demand forecasts for each medicine, indicating the expected quantity required over a specific period.

2. **Reports and Analytics:** Pharmaceutical companies have access to reports and analytics that provide insights into demand patterns, inventory turnover, and forecasting accuracy.

5.4 Security Requirements

1. **Email OTP (One-Time Password):** Provides email-based OTP authentication to provide an additional layer of security during user login or sensitive operations.

2. **User Authentication:** The system is securely authenticated and authorizes users based on their roles and privileges.

3. **Role-Based Access Control:** The system enforces role-based access control to restrict access to certain functionalities and data based on user roles.

4. **Secure Communication:** All communication between the system components, including the web application, APIs, and database, are encrypted using secure protocols (e.g., HTTPS).

5. **Data Privacy:** The system complies with data privacy regulations and protects the personal information of users.

6. System Design

6.1 Use Case Diagram: A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The actors for our system are: Admin, Pharmaceutical Company, Stockist and Local Pharmacy. The simplified and graphical representation of what our system must actually do is represented below:

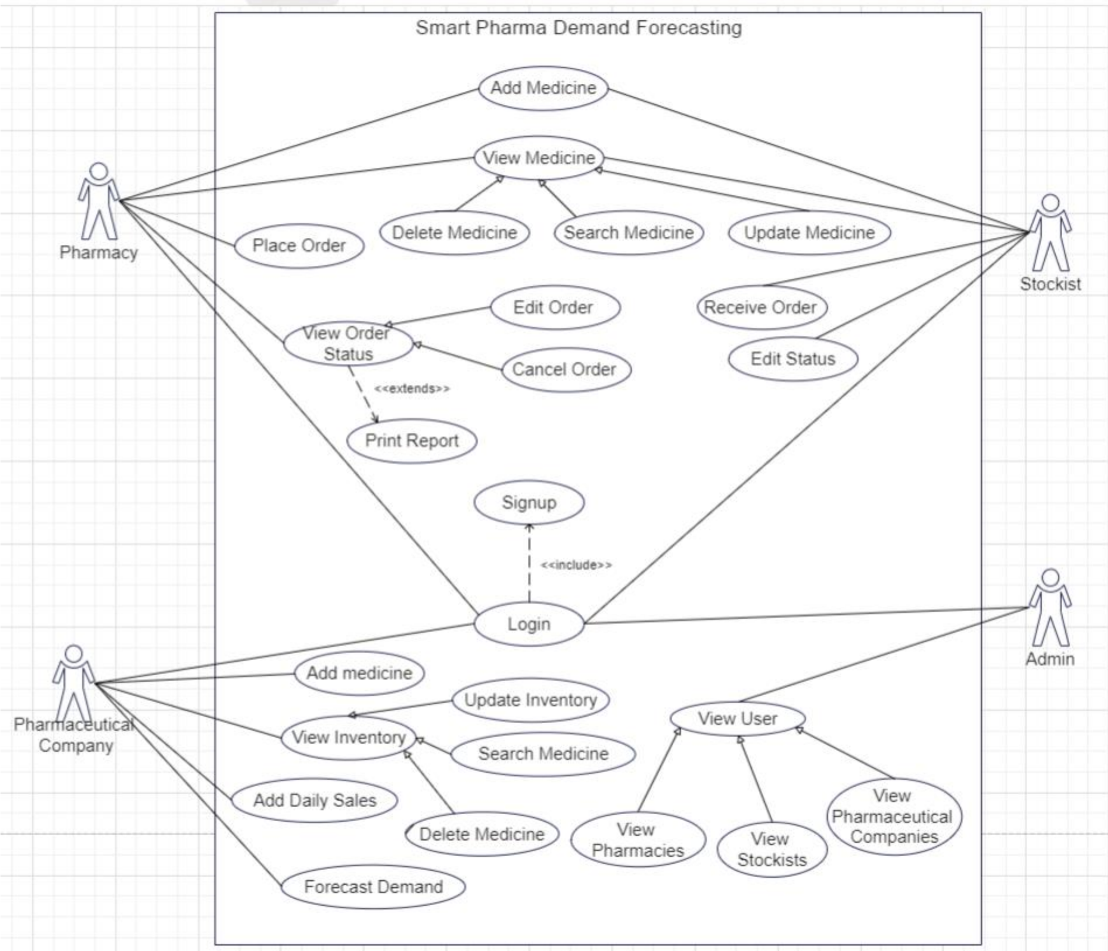


Figure 8: Use Case Diagram

6.2 Schema-Diagram: A schema diagram, also known as a database schema diagram or relational schema diagram, is a visual representation of the structure of a database system. It provides an overview of the tables/entities, their relationships, and the attributes/columns within each table/entity.

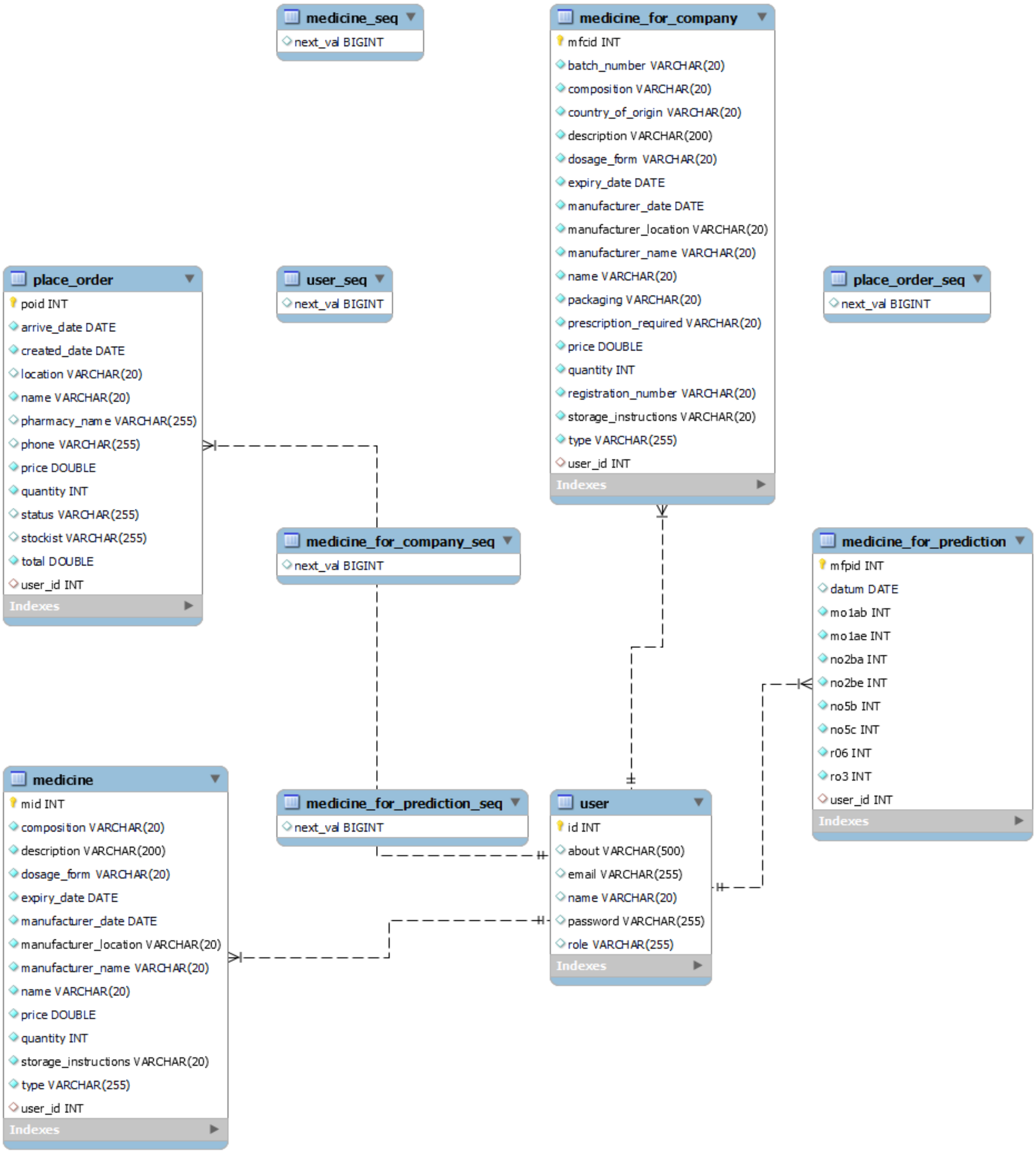


Figure 9: Schema-Diagram

6.3 Sequence Diagram

The sequence diagrams showing our various objects interactions arranged in time sequence is shown below.

6.3.1 Sequence diagram for Signup

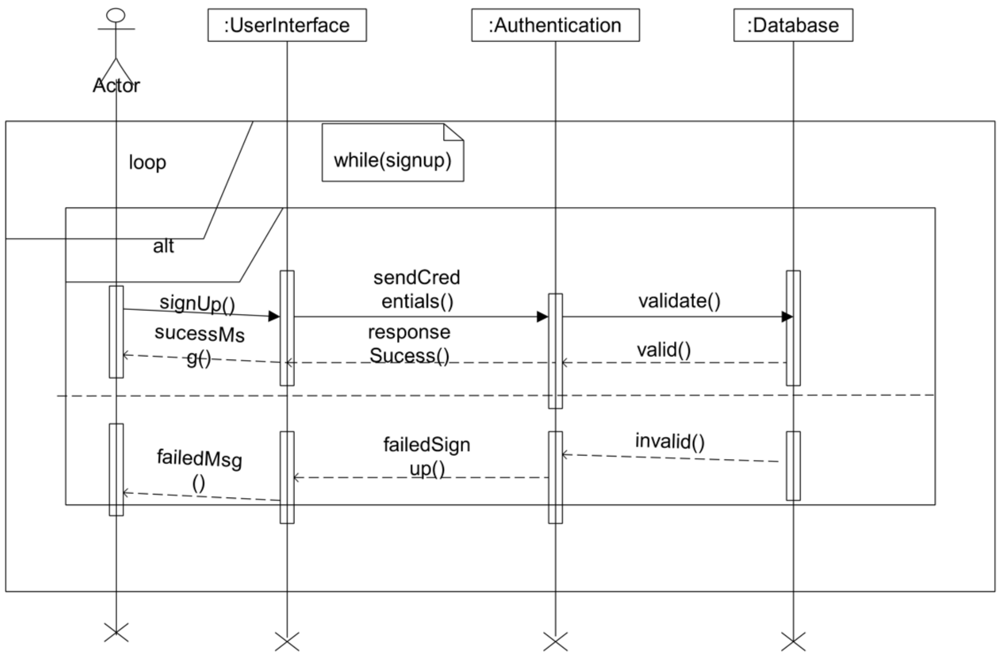


Figure 10: Sequence diagram for Signup

6.3.2 Sequence diagram for Login

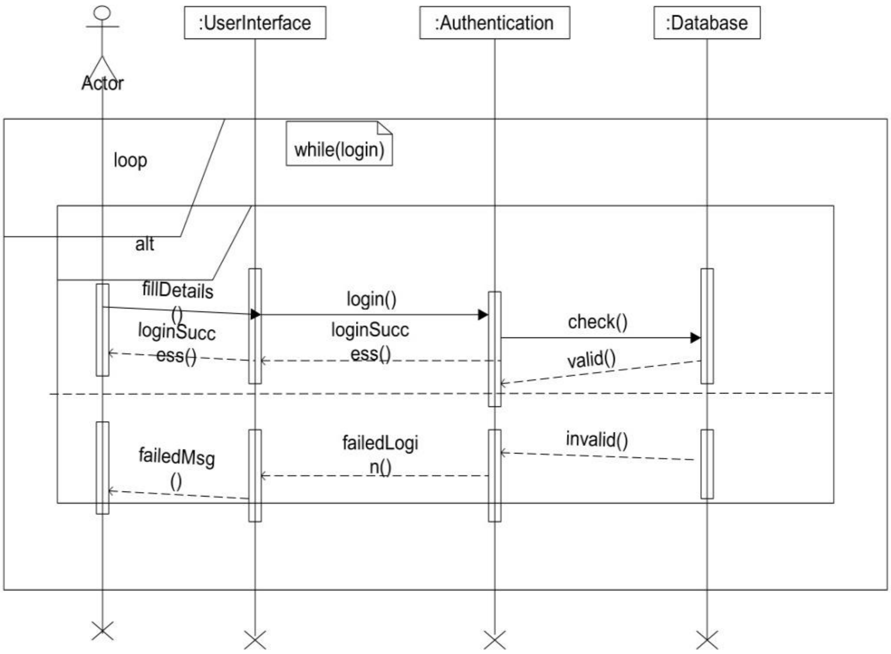


Figure 10: Sequence diagram for Login

6.3.3 Sequence diagram for Inventory

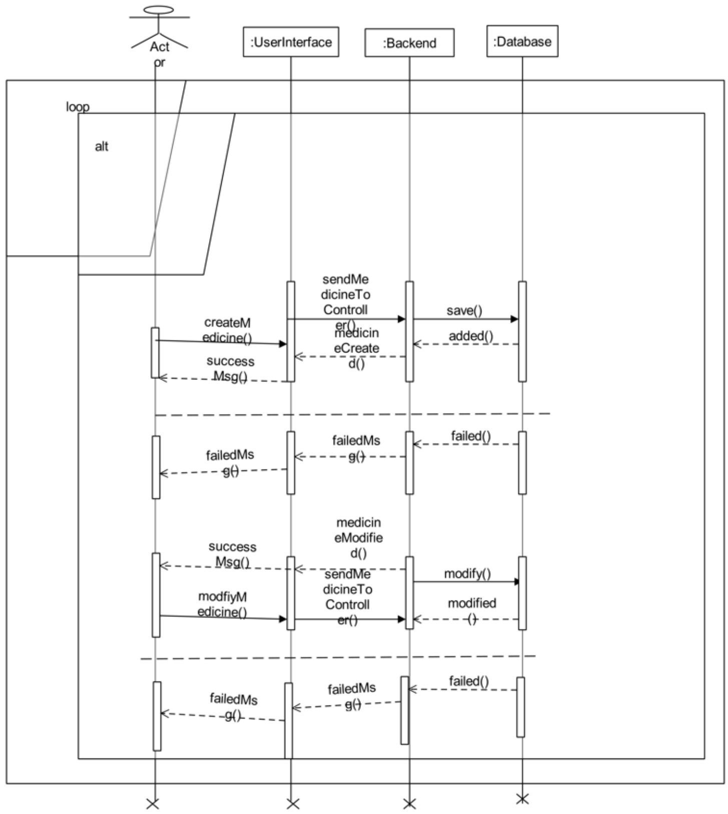


Figure 11: Sequence diagram for Inventory

6.3.4 Sequence diagram for Demand Forecasting

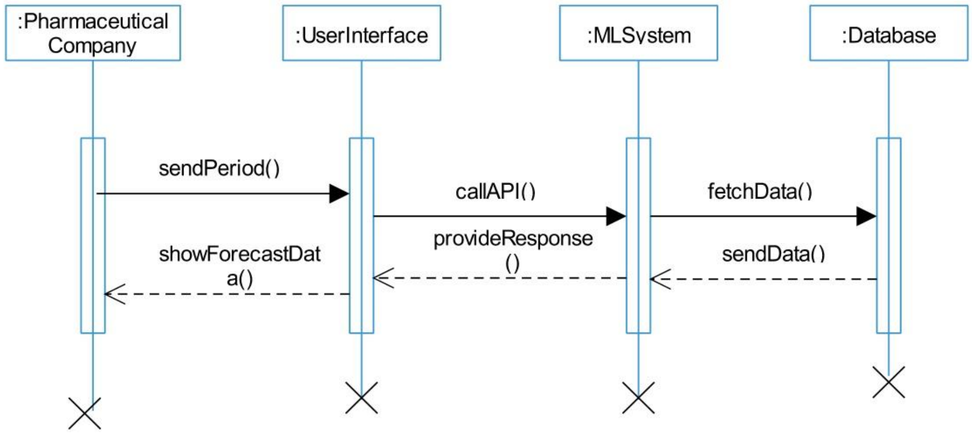


Figure 12: Sequence diagram for Demand Forecasting

6.4 Class Diagram: A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system. We designed the following class diagram to illustrate the system's classes, their attributes, operations (or methods), and the relationships among objects.

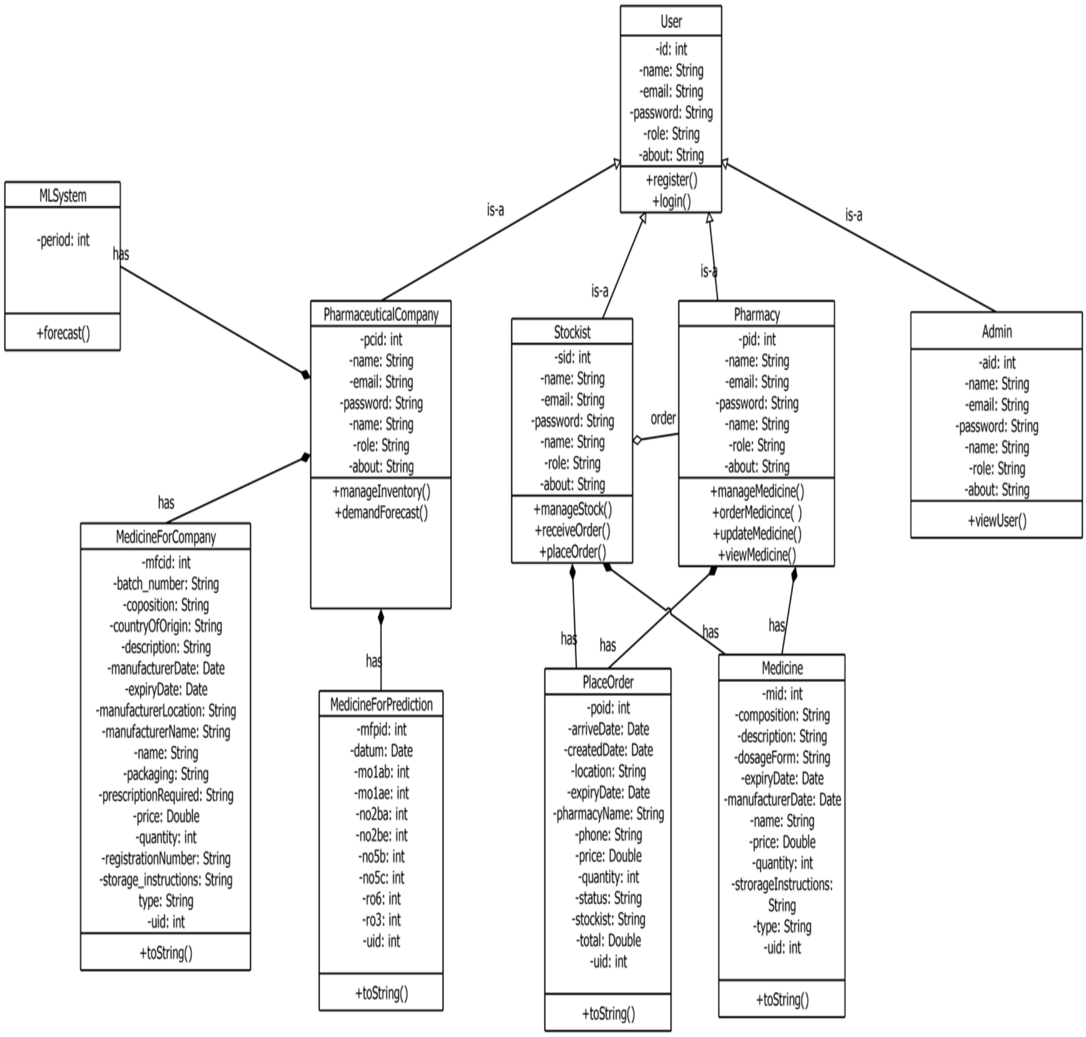


Figure 13: Class Diagram

6.5 Activity Diagram: The activity diagram for Smart Pharma Demand Forecasting based on their usage and consumption pattern is represented below:

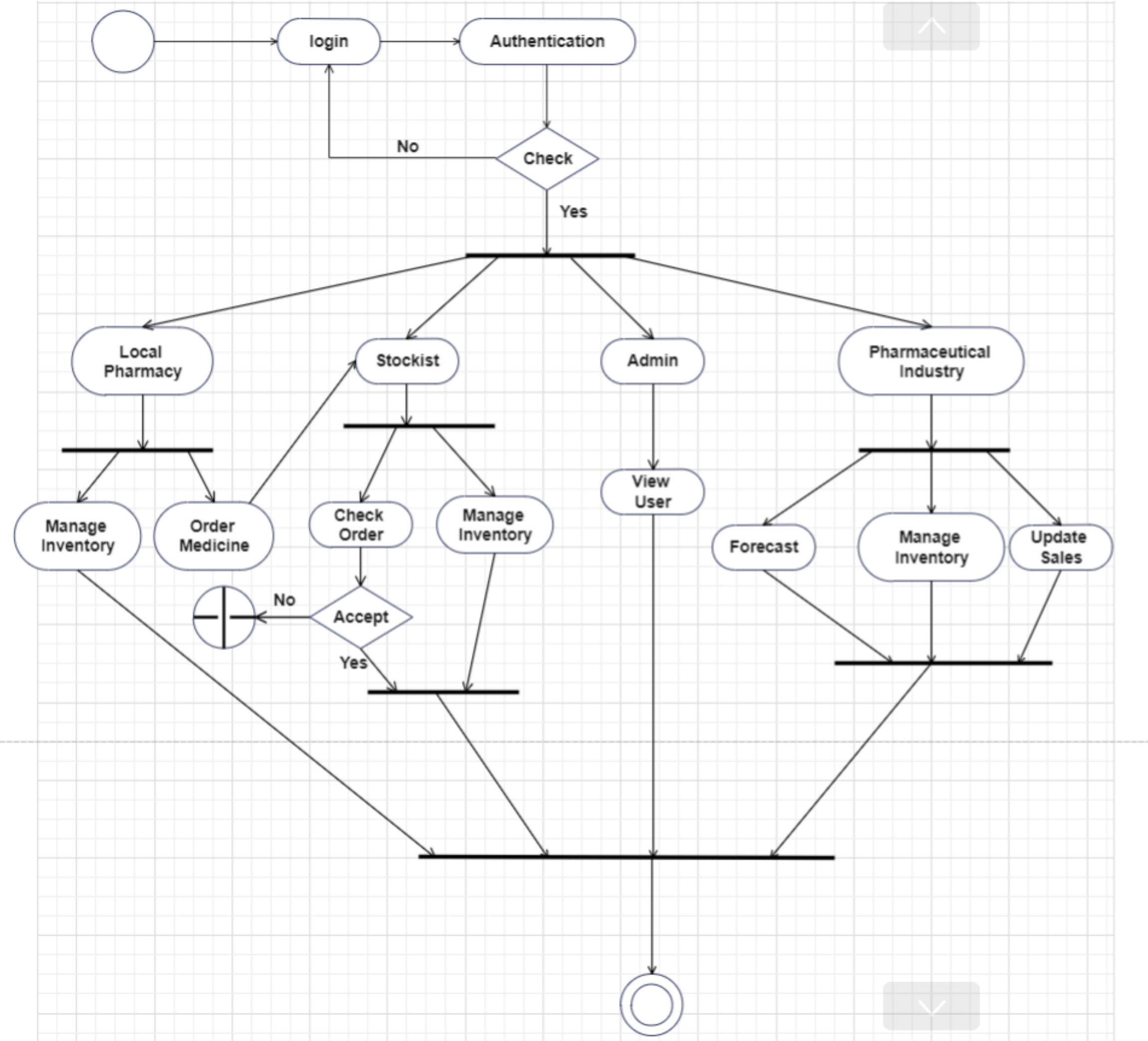


Figure 14: Activity Diagram

6.6 Data Flow Diagram

The data flow diagram of our system is shown below. It represents the flow of a data of a process.

6.6.1 0 Level Data Flow Diagram

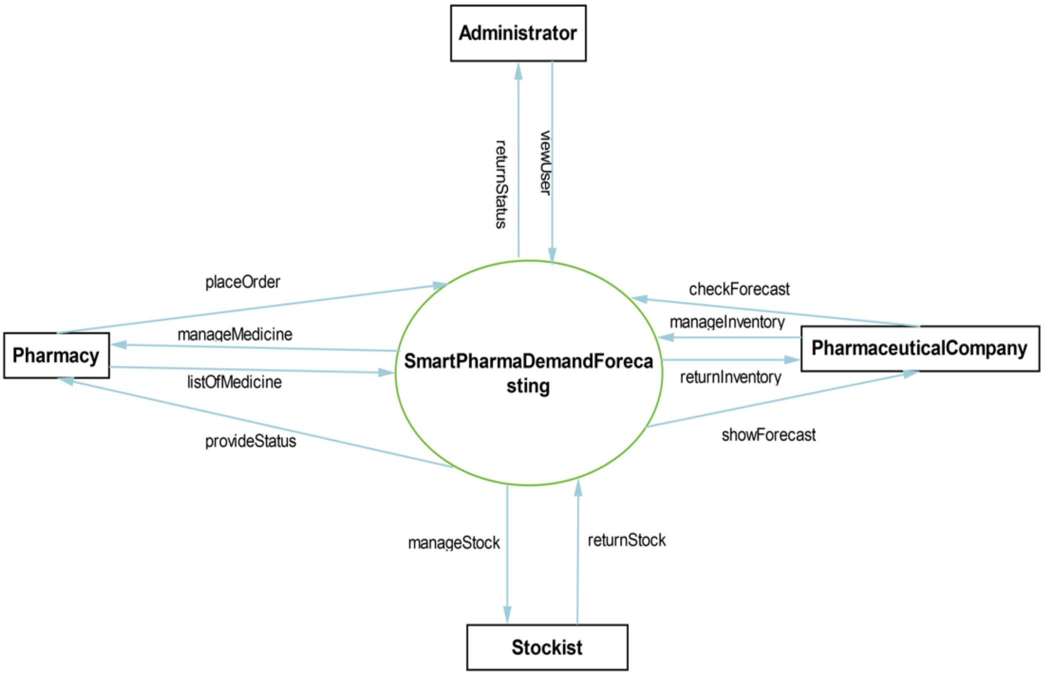


Figure 15: 0 Level Data Flow Diagram

6.6.2 1 Level Data Flow Diagram

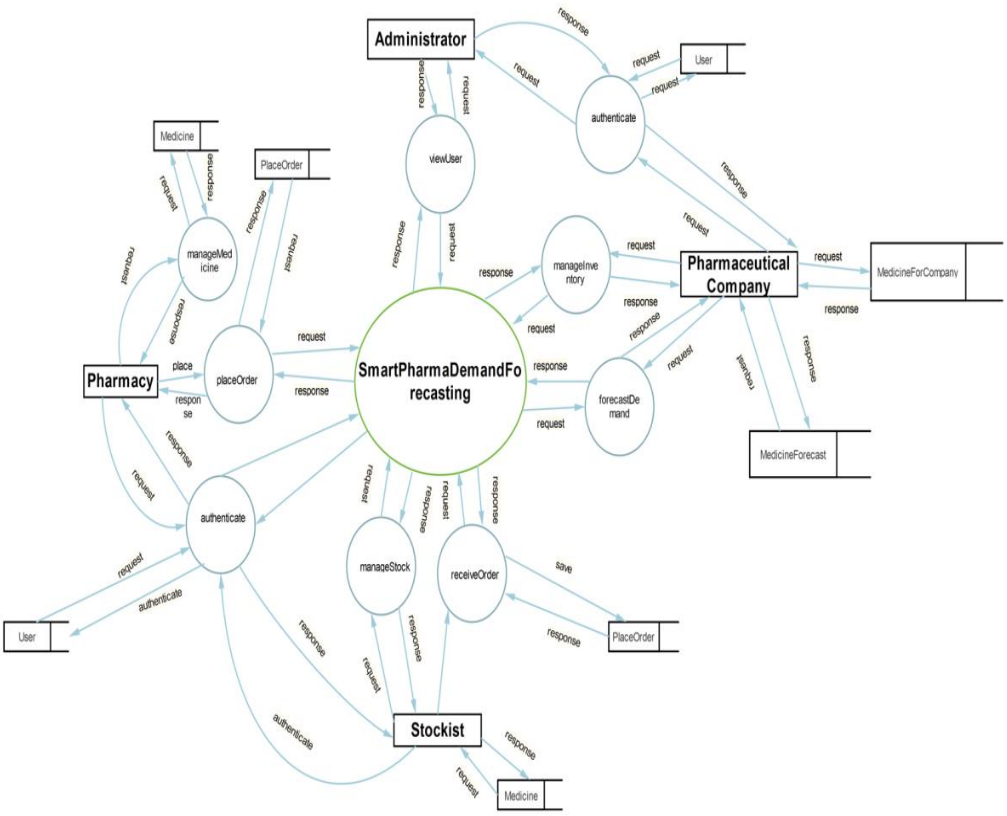


Figure 16: 1 Level Data Flow Diagram

7. Implementation Details:

7.1 Data Collection:

The data for this project was collected from Kaggle. We have used publicly available dataset for training the model. In this dataset, we divided the drugs into 8 classes like M01AB, M01AE, N02BA, N02BE, N05B, N05C, R03, R06 along with datum which contain date from 2014.

7.1.1. Dataset Description:

In this dataset, a selected group of drugs from the dataset has been classified based on the Anatomical Therapeutic Chemical (ATC) Classification System. The drugs in this dataset belong to the following ATC categories:

M01AB - Anti-inflammatory and antirheumatic products, non-steroids, Acetic acid derivatives and related substances

M01AE - Anti-inflammatory and antirheumatic products, non-steroids, Propionic acid derivatives

N02BA - Other analgesics and antipyretics, Salicylic acid and derivatives

N02BE/B - Other analgesics and antipyretics, Pyrazolones and Anilides

N05B - Psycholeptics drugs, Anxiolytic drugs

N05C - Psycholeptics drugs, Hypnotics and sedatives drugs

R03 - Drugs for obstructive airway diseases

R06 - Antihistamines for systemic use

The shape of the data has 2106 rows and 13 columns with no null values in this dataset. The notebook screenshot has been shown below.

```
In [3]: dataset.shape # checking shape of data set
Out[3]: (2106, 13)

In [4]: dataset.head()
Out[4]:
```

	datum	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Year	Month	Hour	Weekday Name
0	1/2/2014	0.0	3.67	3.4	32.40	7.0	0.0	0.0	2.0	2014	1	248	Thursday
1	1/3/2014	8.0	4.00	4.4	50.60	16.0	0.0	20.0	4.0	2014	1	276	Friday
2	1/4/2014	2.0	1.00	6.5	61.85	10.0	0.0	9.0	1.0	2014	1	276	Saturday
3	1/5/2014	4.0	3.00	7.0	41.10	8.0	0.0	3.0	0.0	2014	1	276	Sunday
4	1/6/2014	5.0	1.00	4.5	21.70	16.0	2.0	6.0	2.0	2014	1	276	Monday

```
In [5]: dataset.info() # show the information of  dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2106 entries, 0 to 2105
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   datum         2106 non-null   object 
 1   M01AB         2106 non-null   float64
 2   M01AE         2106 non-null   float64
 3   N02BA         2106 non-null   float64
 4   N02BE         2106 non-null   float64
 5   N05B          2106 non-null   float64
 6   N05C          2106 non-null   float64
 7   R03           2106 non-null   float64
 8   R06           2106 non-null   float64
 9   Year          2106 non-null   int64  
 10  Month         2106 non-null   int64  
 11  Hour          2106 non-null   int64  
 12  Weekday Name  2106 non-null   object 
dtypes: float64(8), int64(3), object(2)
memory usage: 214.0+ KB
```

Figure 17: Data Description

7.2 EDA and Graph Based Analysis:

In EDA, the shape of dataset is first checked. In this dataset, there are 2106 rows and 13 columns. The null value is also checked here. But in our data there are no null values as shown in figure below.

```
sns.heatmap(dataset.isnull())
<AxesSubplot:>
```

Figure 18: Heat Map

In below figure, the data distribution in histogram has been shown. The skewness of the data is shown in this plot so, it is concluded that there are outliers present in this dataset.

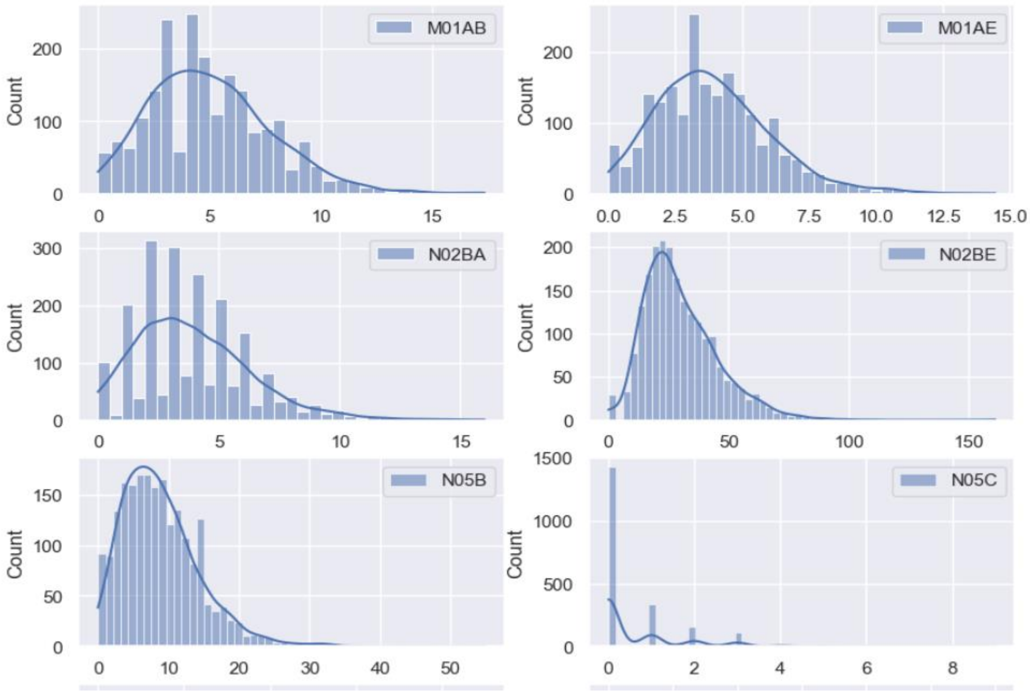


Figure 19: Histogram

In below box plot, outlier confirmation is shown. Here you can see that there is outlier for only upper bound in all classes of drugs.

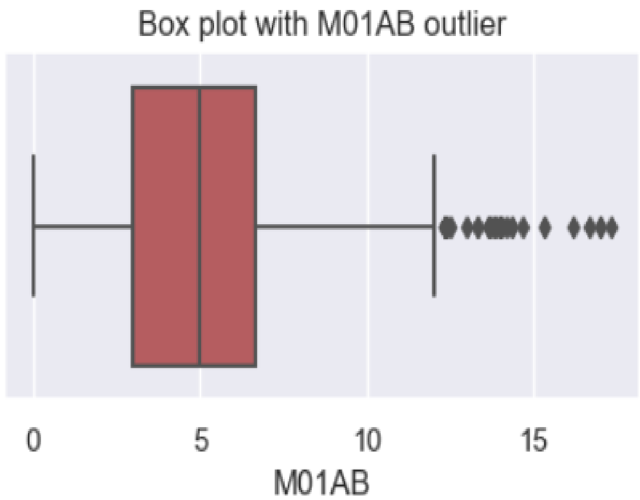


Figure 20: Box Plot M01AB

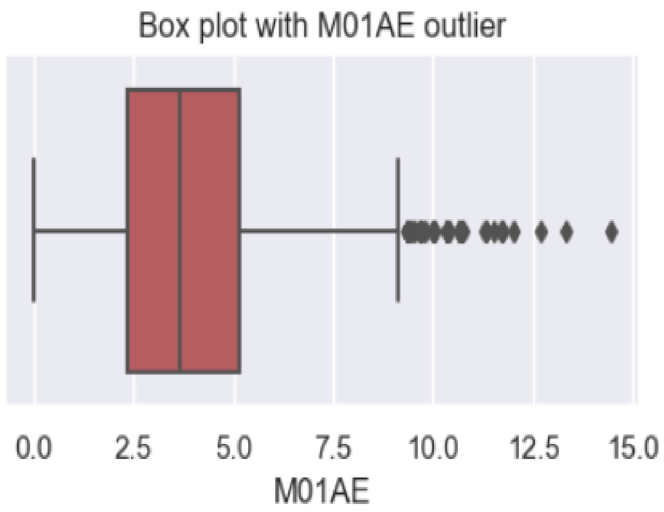


Figure 21: Box Plot M01AE

Here we have calculated upper bound and lower bound of outlier using inter quartile range. Here, upper bound is used to remove outlier from the dataset.

```
lower_bounds,upper_bounds = cal_upper_lowder_bound(dataset,targets)
```

```
Lower bound of M01AB:-2.505
Upper bound of M01AB:12.175
****************************************
Lower bound of M01AE:-1.8570000000000002
Upper bound of M01AE:9.335
****************************************
Lower bound of N02BA:-2.8000000000000007
Upper bound of N02BA:10.0
****************************************
Lower bound of N02BE:-9.949999999999996
Upper bound of N02BE:67.25
****************************************
Lower bound of N05B:-5.5
Upper bound of N05B:22.5
****************************************
Lower bound of N05C:-1.5
Upper bound of N05C:2.5
****************************************
Lower bound of R03:-9.5
Upper bound of R03:18.5
****************************************
Lower bound of R06:-3.5
Upper bound of R06:8.5
****************************************
```

Figure 22: Outliers Bound

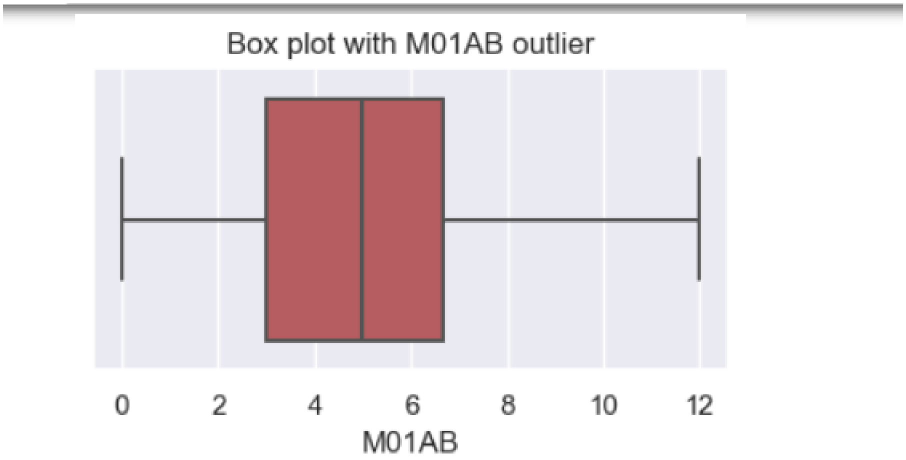


Figure 23: Box Plot M01AB after removing outliers

Time Series Analysis:

Now it's time for time series analysis because the dataset collected is on the regular interval of time. So univariant time series analysis is used because there is only one variable which does not depend on another.

The distribution of data based on time is shown in figure below. It does not indicate clearly that which trends it flows.

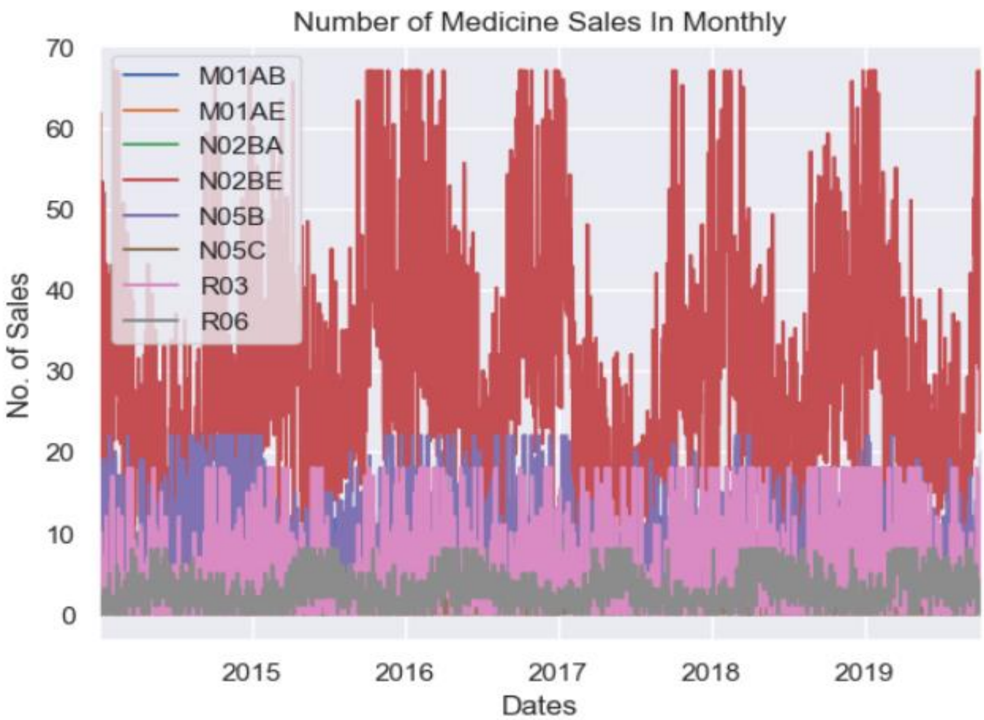


Figure 24: Number of Medicines sales

The time series data of each class separately which clearly show the trend is plotted in below figure. It is found that it looks like horizontal trend.

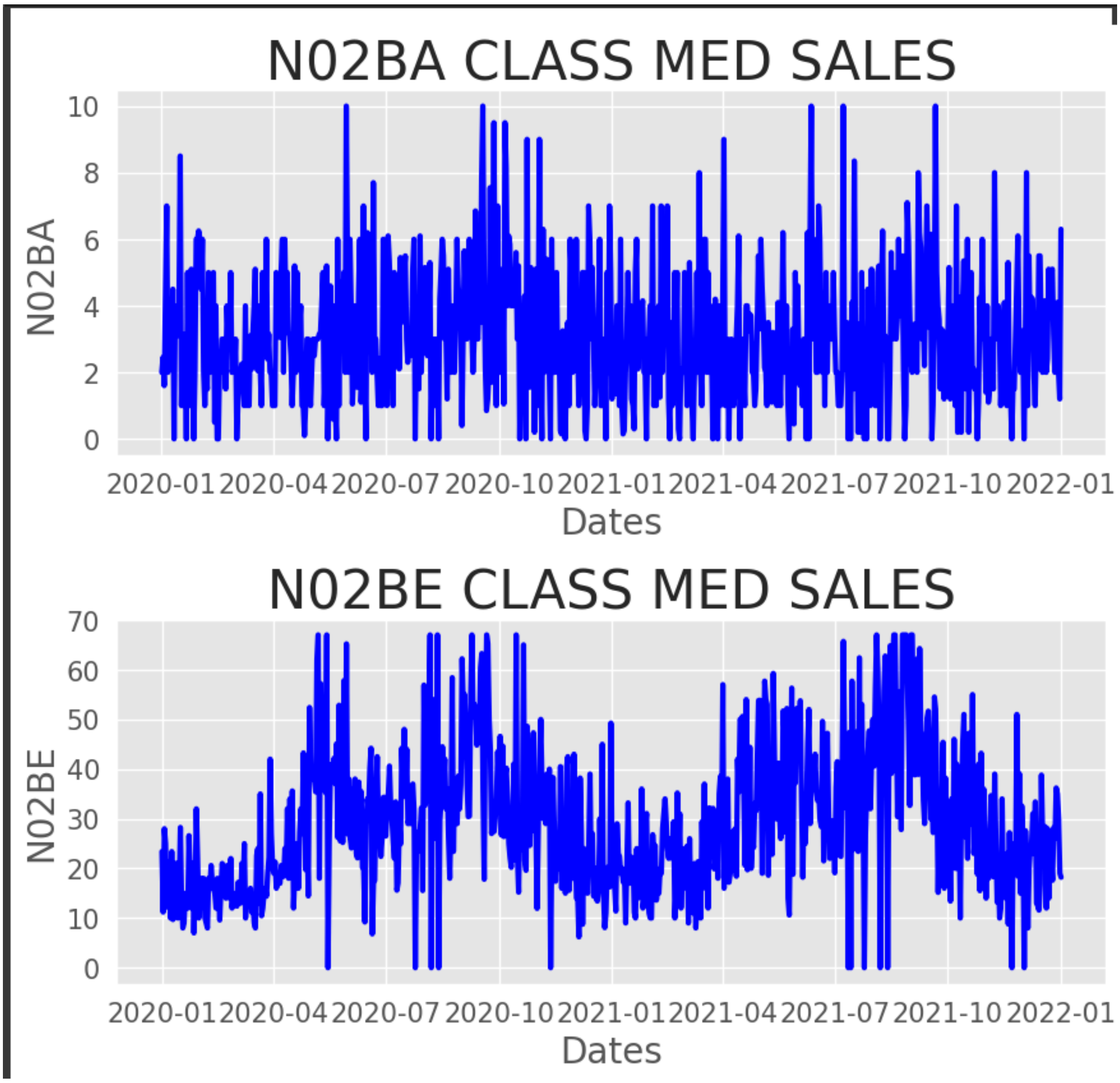


Figure 25: Time Distribution of N02BA and N02BE

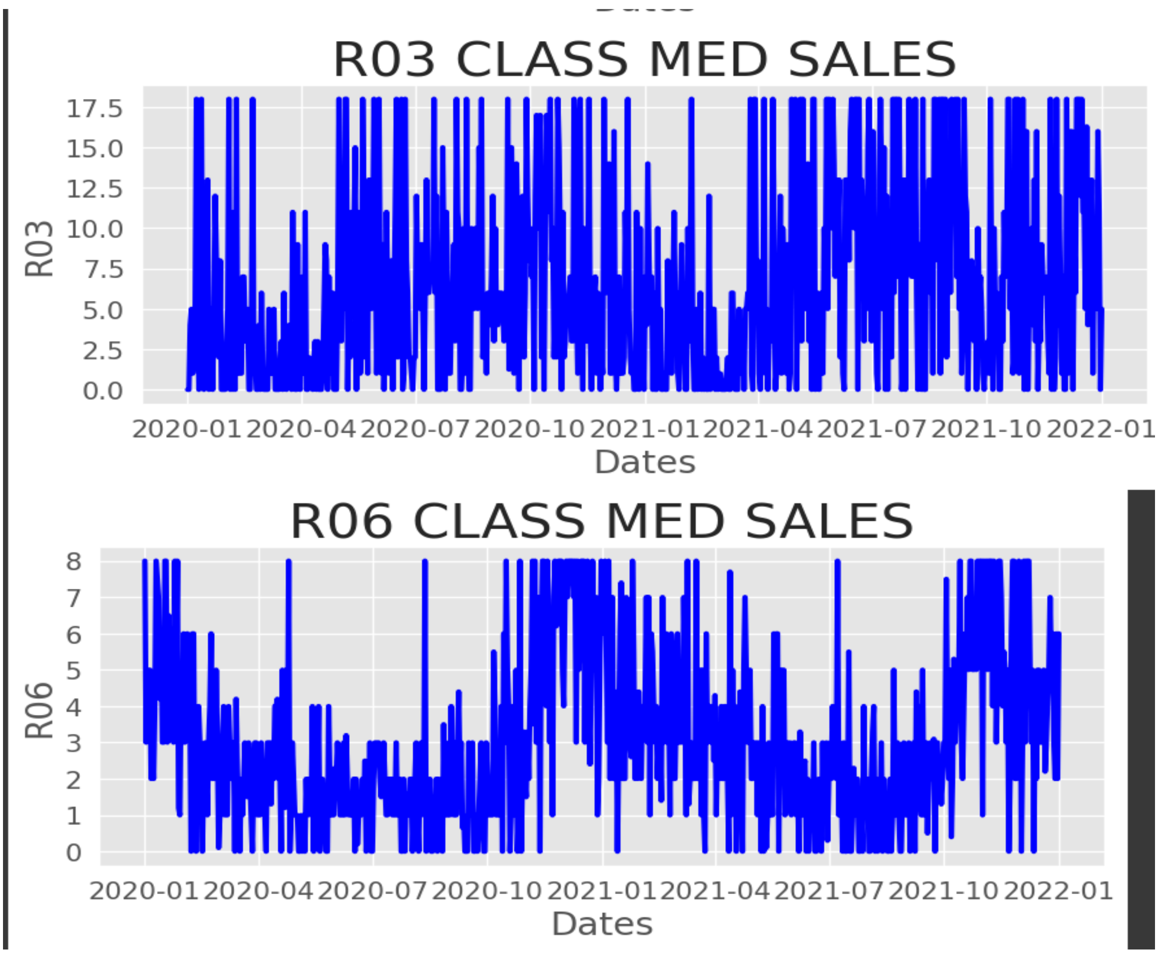


Figure 26: Time Distribution of R03 and R06

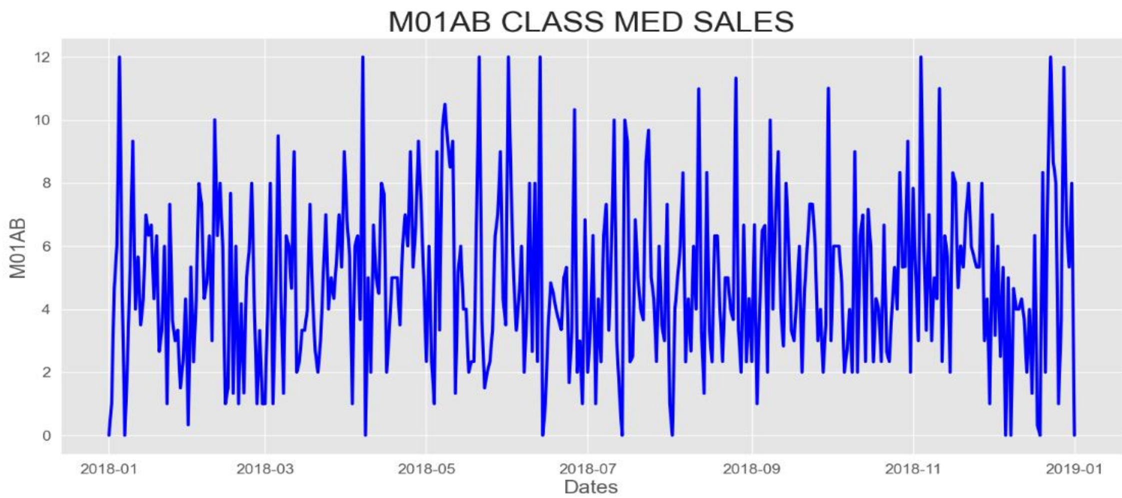


Figure 27: Time Distribution of M01AB

In this dataset, we have taken from 2018 to 2019 data sales to confirm the trend. It is found that there is horizontal trend in this data. It shows like it is stationery but we cannot confirm that it is stationary or not. We are using dickey fuller test for confirmation of the dataset to be stationary. If p-value < 0.05 then it is stationery.

If it is stationery then mean and standard deviation are constant. You can see the decomposition of time series data in below figure. Decomposition shows the three features like Trend, Seasonal and Residual. All of the features plotted below in the figure.

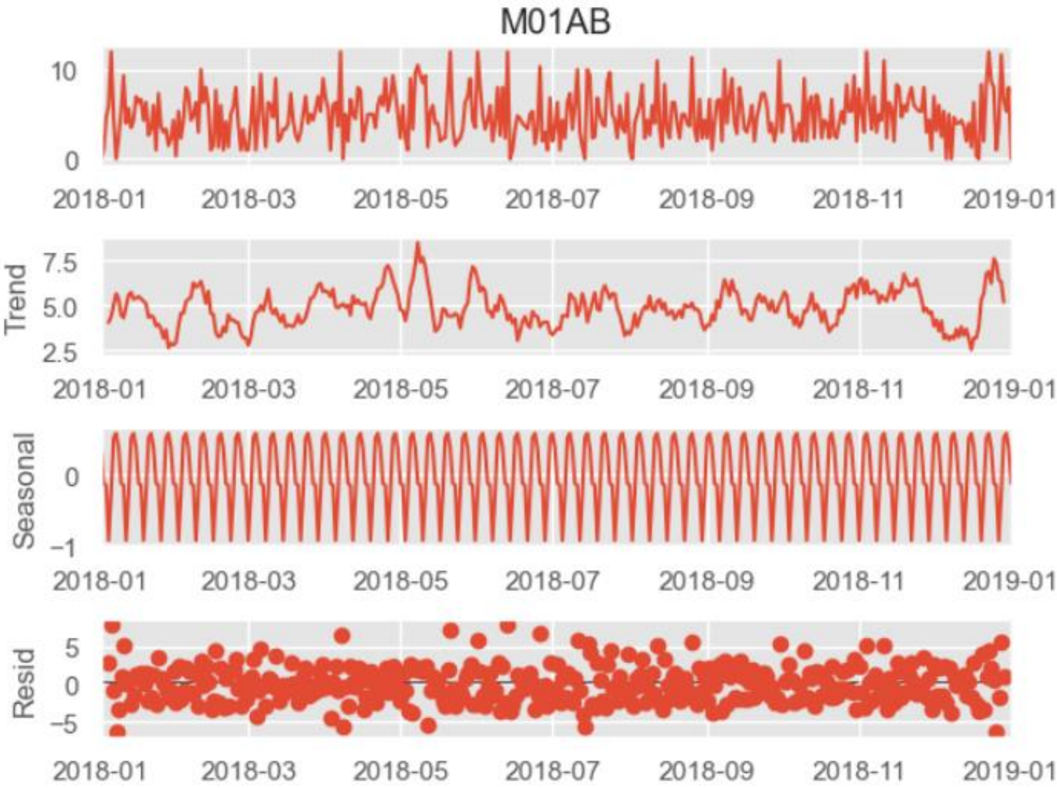


Figure 28: Decomposition of M01AB

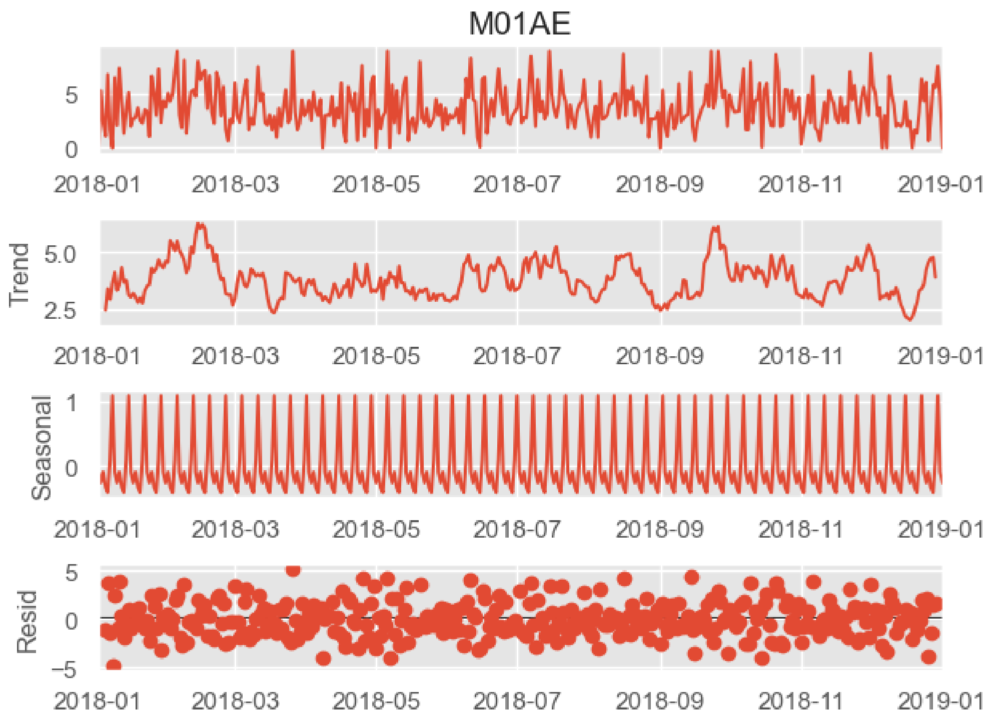


Figure 29: Decomposition of M01AE

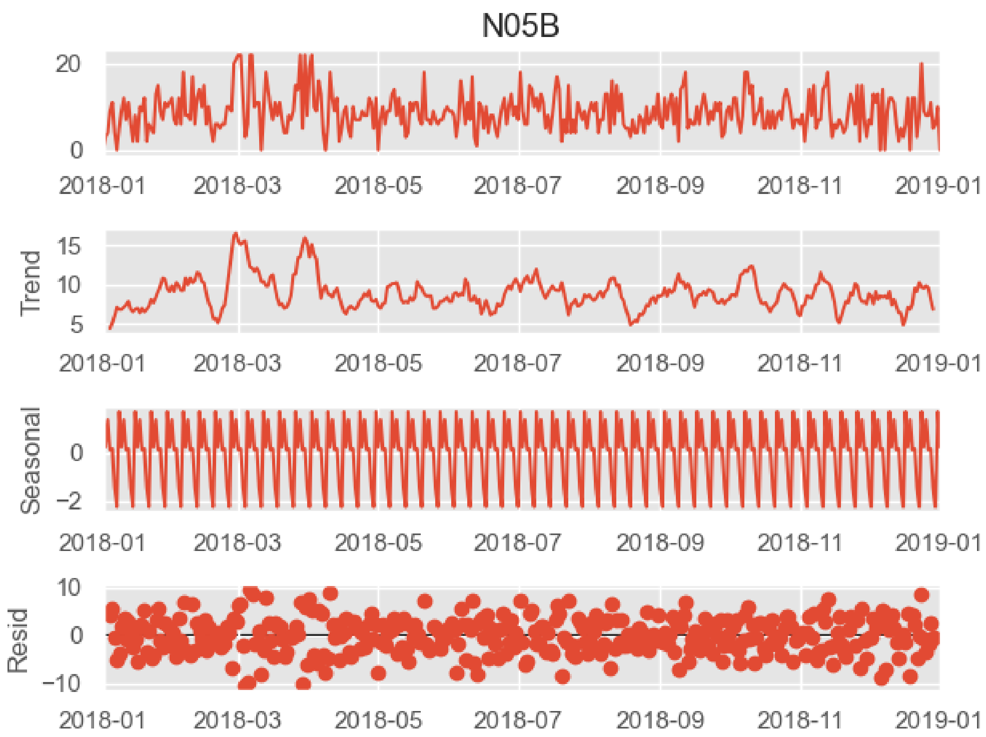


Figure 30: Decomposition of N05B

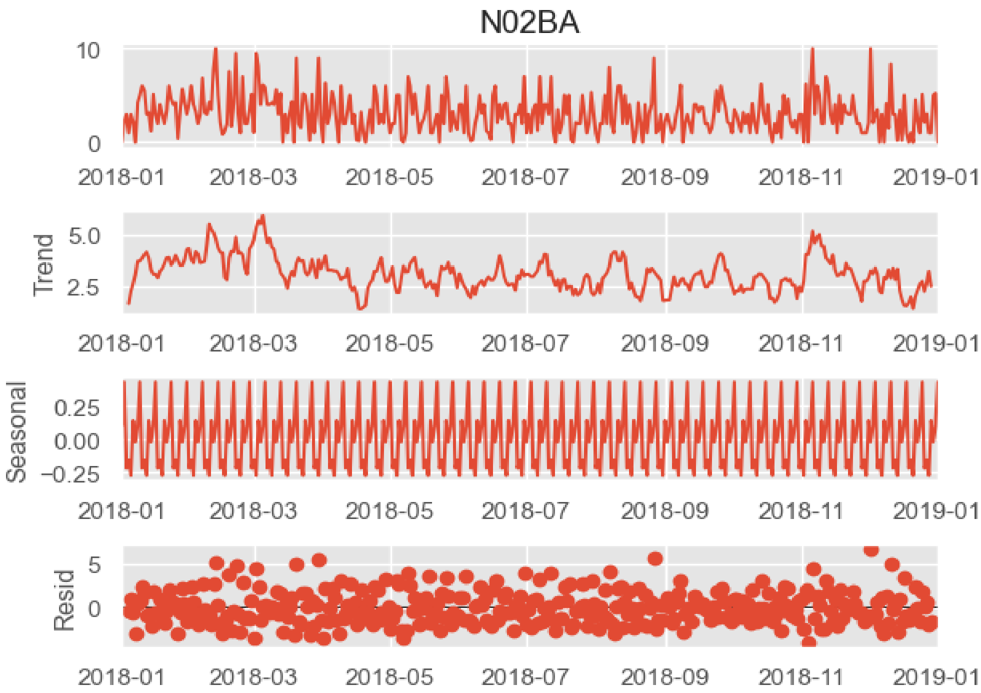


Figure 31: Decomposition of N02BA

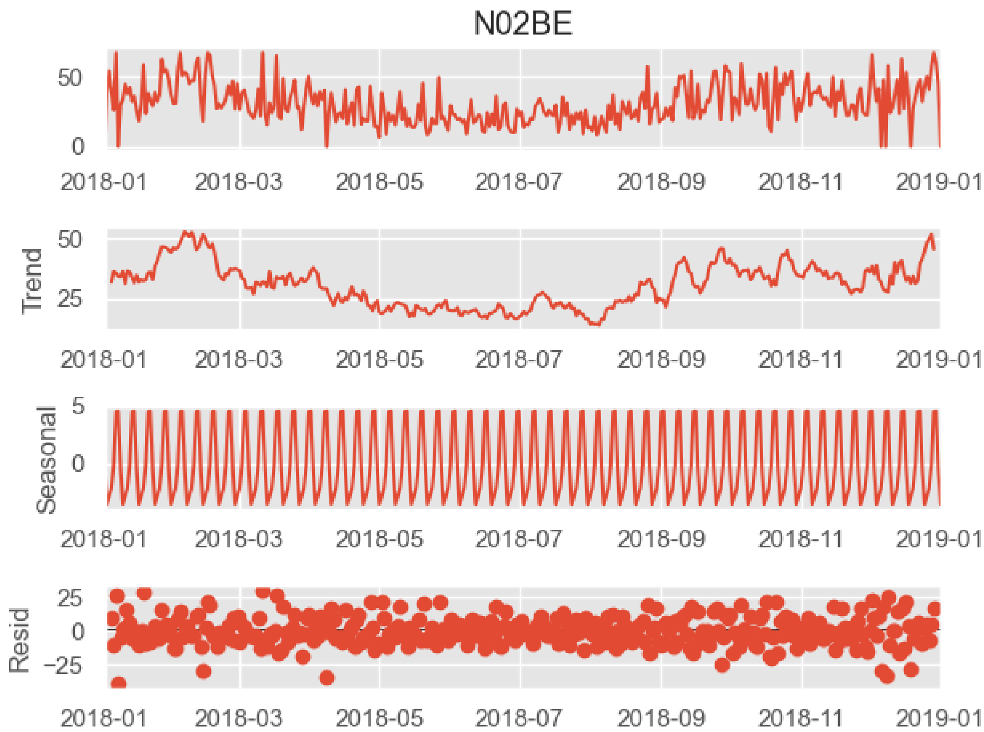


Figure 32: Decomposition of N02BE

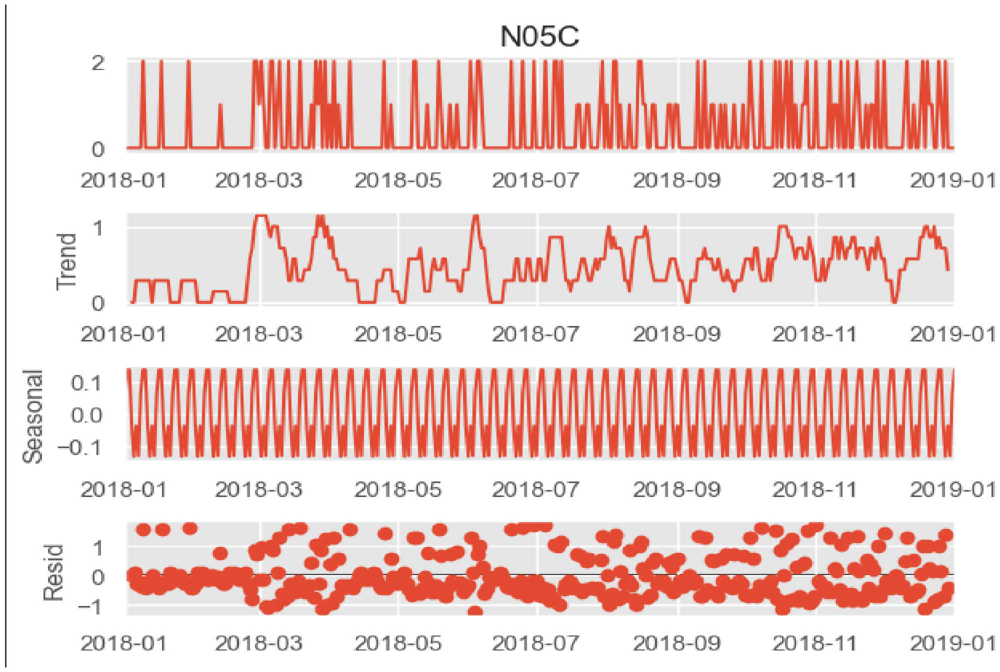
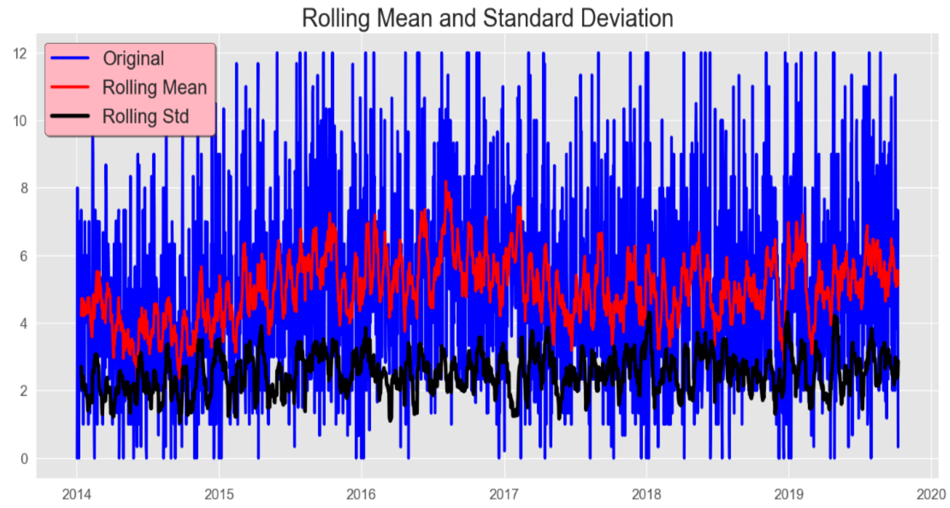


Figure 33: Decomposition of N05C




```
Results of dickey fuller test
Test Statistics                -8.159135e+00
p-value                         9.245049e-13
No. of lags used                1.400000e+01
Number of observations used     2.091000e+03
critical value (1%)            -3.433481e+00
critical value (5%)            -2.862923e+00
critical value (10%)           -2.567506e+00
dtype: float64
```

Figure 34: Rolling Mean and Standard Deviation

Here it can be seen that the p-value is 9.245049e-14 which is less than 0.05. Hence we can conclude that it is stationery.

The ACF and PACF plot can also be seen here.

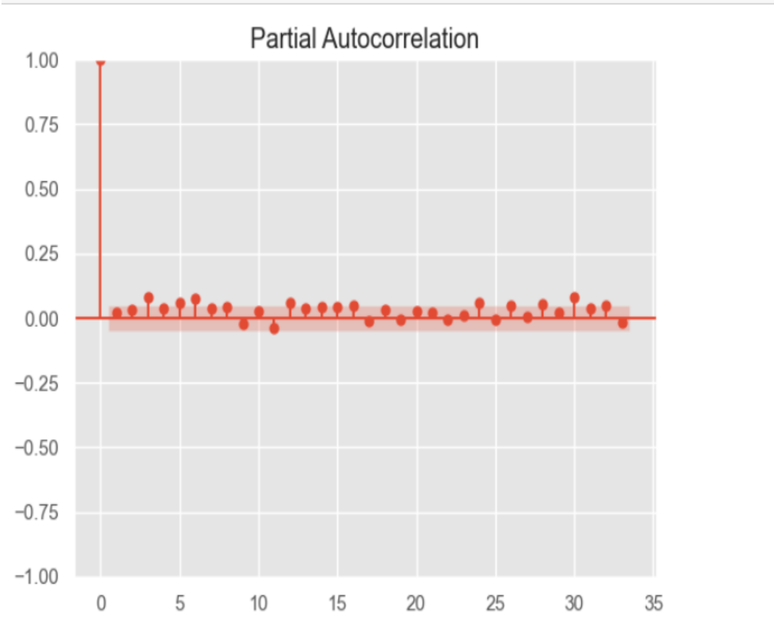


Figure 35: PACF Plot

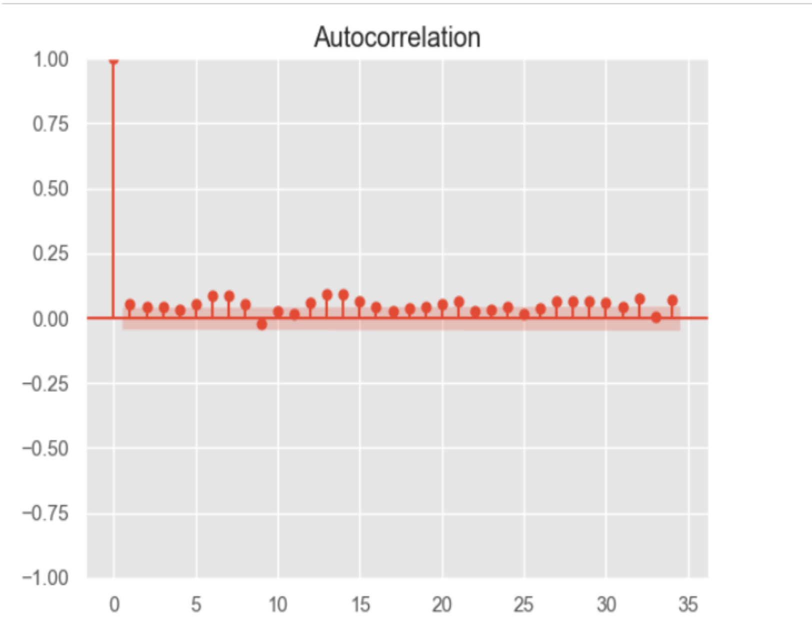


Figure 36: ACF Plot

7.3 Data Preparation: In Data preparation, 80% of data is used for training and 20% of data for testing.

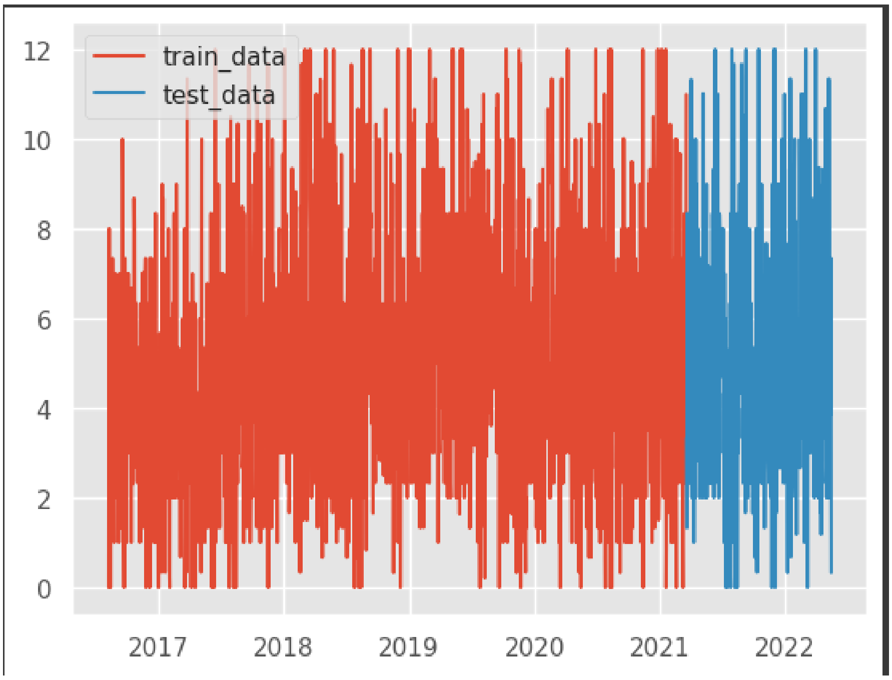


Figure 37: Splitting Data

The model ARIMA is trained here; in which root mean square error is more deviated from actual which can be seen in figure below that RMSE more deviated.

```
root mean square : 2.693842924328176 and order  (0, 0, 0)
root mean square : 2.6920022050235772 and order  (0, 1, 1)
```

Figure 38: RMSE ARIMA

So it is decided to use deep learning model forecasting forecasting.

First we the data is standardized, and then splitted in the way that window size = 60 for independent data is taken and after 60, 1 for dependent data as shown in figure below.

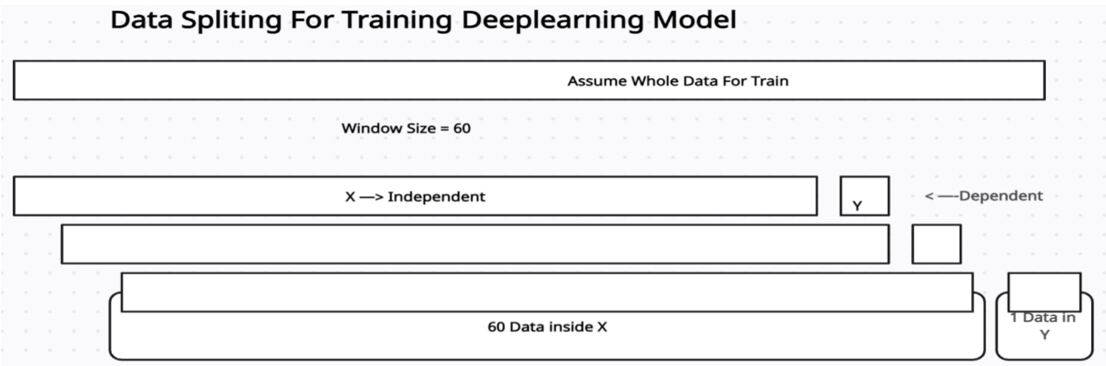


Figure 39: Data splitting for training deeplearning model

7.4. Model Training:

7.4.1. Params:

Optimizer: Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data.

Activation: The rectified linear activation function or ReLu for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

7.4.2: Model Layers:

GRU, RNN, CNN, LSTM, Bi-LSTM and Hybrid Model are used which is combination of the CNN, LSTM and Time Distribution for training the data. Except Hybrid model all model having single layers and params like 50 epoch, relu activation function, loss metrics rmse and adam optimizer for each epoch and batch size is 100.

Here, a new hybrid model is built in which first and second layers contain CNN, third layer Flatten, fourth layer Repeat Vector, fifth layer LSTM, sixth and seventh layer Time Distributed Dense Layers.

The model summary of each layer with numbers of param is shown below.

```
 conv1d_2 (Conv1D)              (None, 58, 256)

 conv1d_3 (Conv1D)              (None, 56, 256)

 flatten_1 (Flatten)            (None, 14336)

 repeat_vector_1 (RepeatVect     (None, 1, 14336)
 or)

 lstm_5 (LSTM)                  (None, 1, 128)

 time_distributed_2 (TimeDis     (None, 1, 100)
 tributed)

 time_distributed_3 (TimeDis     (None, 1, 1)
 tributed)

=================================================================
Total params: 7,616,969
Trainable params: 7,616,969
Non-trainable params: 0
```

Figure 40: Model Summary

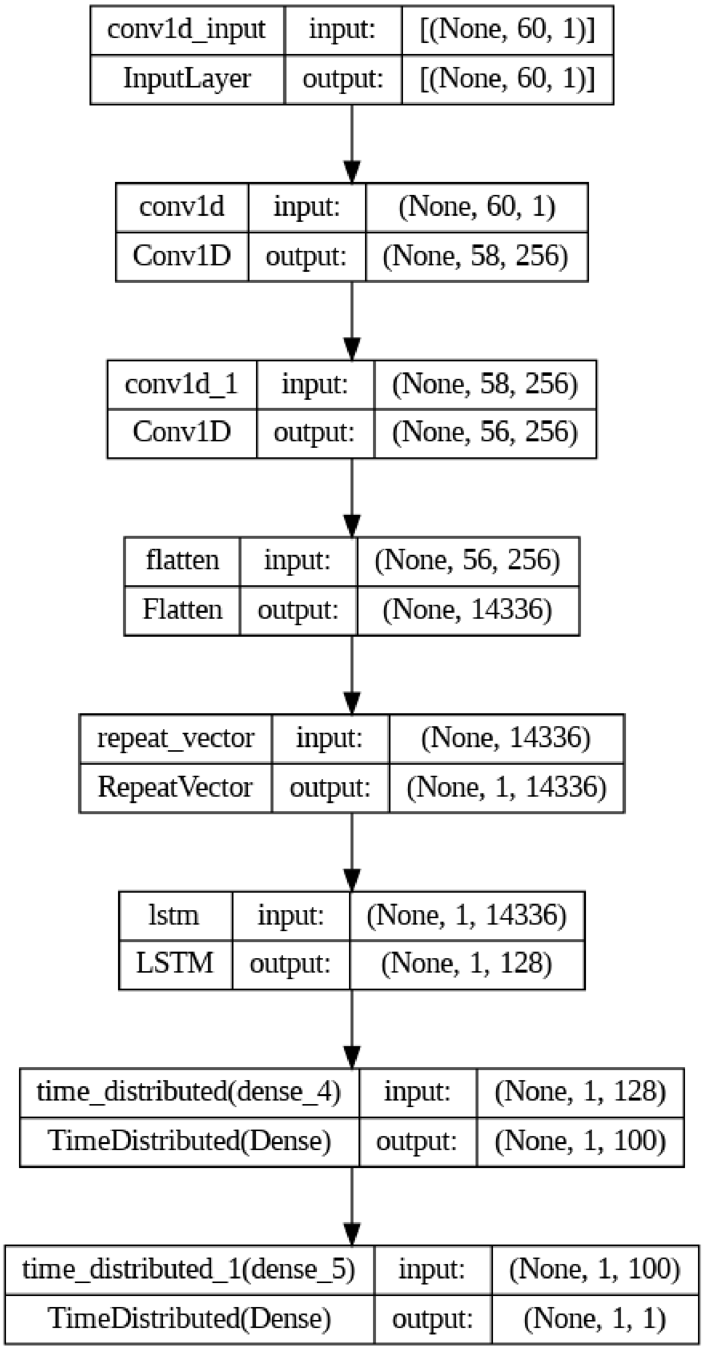


Figure 41: Model Architecture

7.5 Model Evaluation:

After Model Training, the next is Model Evaluation Phase, where RMSE, Mean Absolute Error and R squared are used.

Root Mean Square Error:

The root mean square error (RMSE) measures the average difference between a statistical model's predicted values and the actual values. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points.

Root mean squared error (RMSE) is the square root of the mean of the square of all of the error. The use of RMSE is very common, and it is considered an excellent general purpose error metric for numerical predictions.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} S_i - O_i{}^2}$$

Figure 42: RMSE Formula

Where O_i are the observations, S_i predicted values of a variable, and n the number of observations available for analysis. RMSE is a good measure of accuracy, but only to compare forecasting errors of different models or model configurations for a particular variable and not between variables, as it is scale-dependent.

R-squared:

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determinations for multiple regressions. The definition of R-squared is the percentage of the response variable variation that is explained by a linear model. R-squared = Explained variation / Total variation R-squared is always between 0 and 100%. 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. In general, the higher the R-squared, the better the model fits your data.

$$R^2 = 1 - \frac{\overset{\text{Sum Squared Regression Error}}{SS_{Regression}}}{\underset{\text{Sum Squared Total Error}}{SS_{Total}}}$$

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Figure 43: R-Squared formula

Here, it can be seen that performance of RNN, LSTM, Bi-LSTM and GRU are more deviated to error in below figure. The goodness of fits R-Squared is very low so, we have used another hybrid models.

	RMSE	MAE	R-Squared
RNN	1.288323	1.040863	-0.626728
LSTM	1.054313	0.842476	-0.089442
Bi-LSTM	1.016654	0.809617	-0.013005
GRU	1.024448	0.816055	-0.028596

Figure 44: Performance comparison

Here, it can be seen the hybrid model of each class are better than above models. So it is decided to forecast data on the hybrid model.

```
: Evaluation Metrics of M01AB]
 No.0 of RMSE : 0.1729812573821065, MAE: 0.044478520548772854, r2: 0.9700774845945054
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of M01AE]
 No.1 of RMSE : 0.22746234897838546, MAE: 0.07061078990499645, r2: 0.9482608797972352
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of N02BA]
 No.2 of RMSE : 0.2960777272525262, MAE: 0.08451347883882948, r2: 0.9123379794249787
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of N02BE]
 No.3 of RMSE : 0.22593793857404526, MAE: 0.05519871844058274, r2: 0.948952047912911
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of N05B]
 No.4 of RMSE : 0.21759841064253932, MAE: 0.07522848408149106, r2: 0.9526509316858408
```

Figure 45: Model Evaluation Metrics 1

```
: Evaluation Metrics of N05B]
 No.4 of RMSE : 0.21759841064253932, MAE: 0.07522848408149106, r2: 0.9526509316858408
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of N05C]
 No.5 of RMSE : 0.28477044216485237, MAE: 0.08863877110590061, r2: 0.9189057952692344
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of R03]
 No.6 of RMSE : 0.19066170186581524, MAE: 0.0527025206650951, r2: 0.963648115441631 ]
 (422, 1)]
 (1684, 60, 1)]
: Evaluation Metrics of R06]
 No.7 of RMSE : 0.14599103409532077, MAE: 0.03299557894724288, r2: 0.9786866179637789
```

Figure 46: Model Evaluation Metrics 2

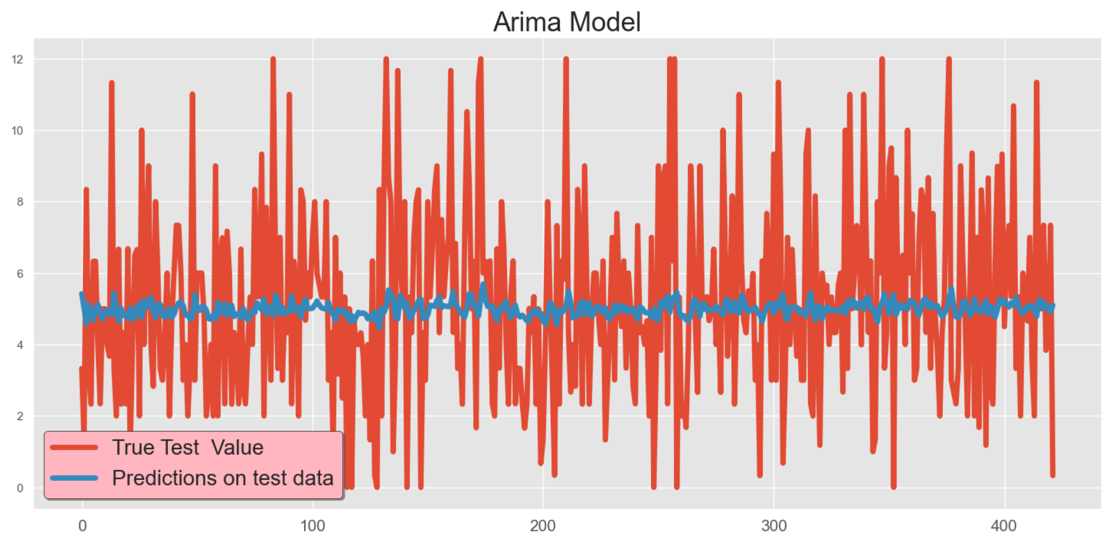


Figure 47: Testing Prediction result ARIMA

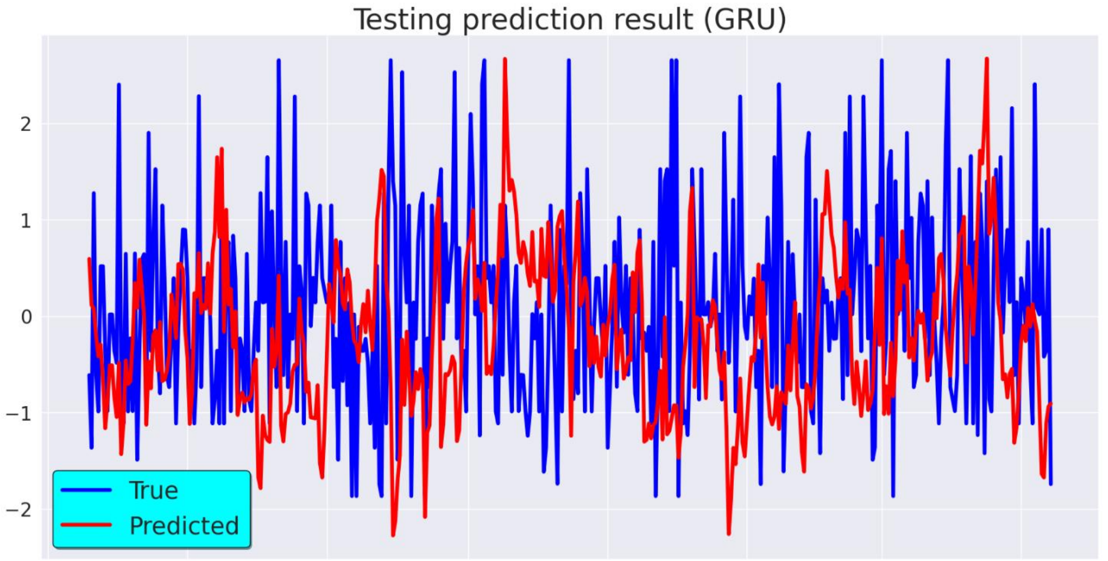


Figure 48: Testing Prediction result (GRU)

Here, in above figure, the testing prediction GRU model can be seen.

Now, the testing predicting result of hybrid model can be seen below.

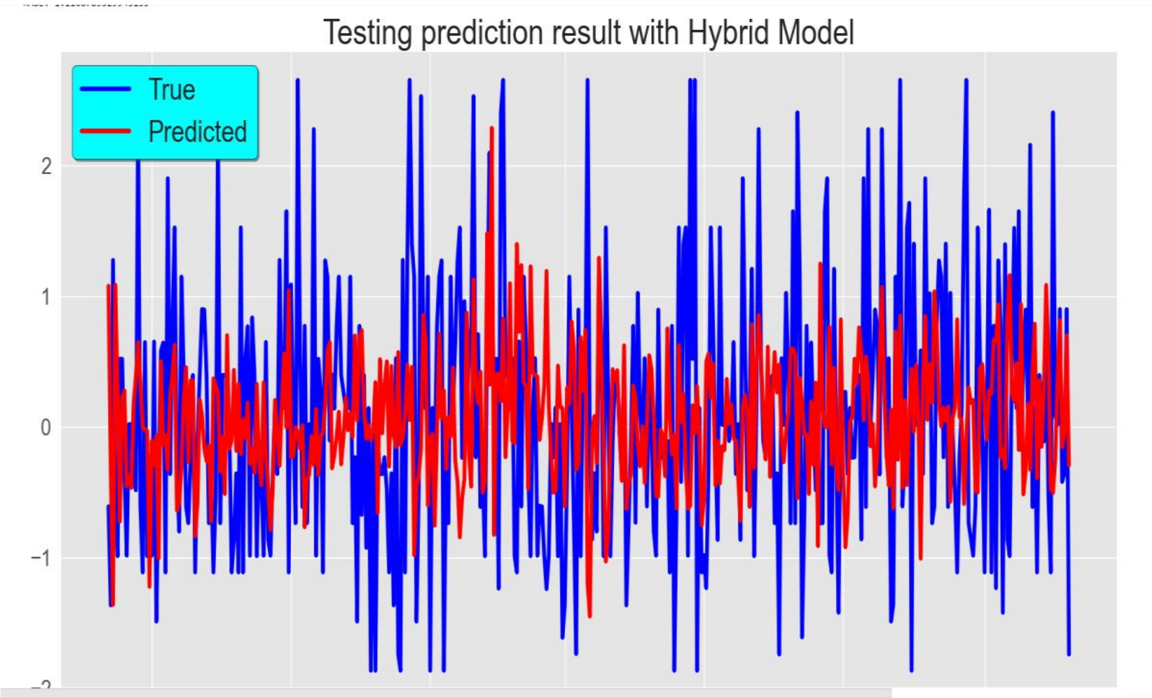


Figure 49: Testing Prediction result with Hybrid Model

In above figure, the fitting of the data is shown on actual, in hybrid model which shows little bit more accurate than other model.

7.6 Forecasting:

After Evaluation of the each model of each class, we are going to forecast the data on the basis of test data and also on our data.

The forecast on the test data is shown below.

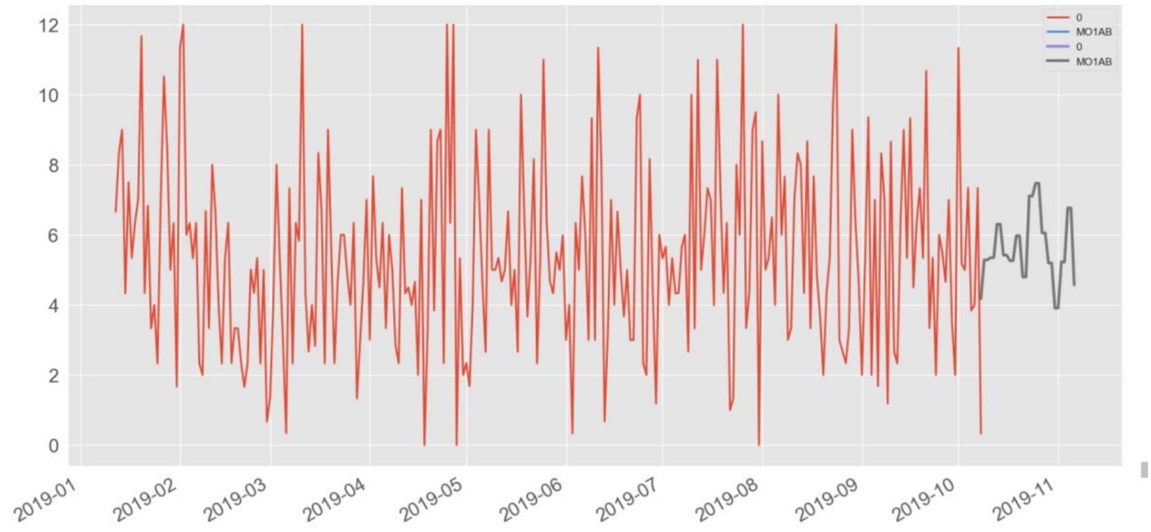


Figure 50: 30 days forecasting

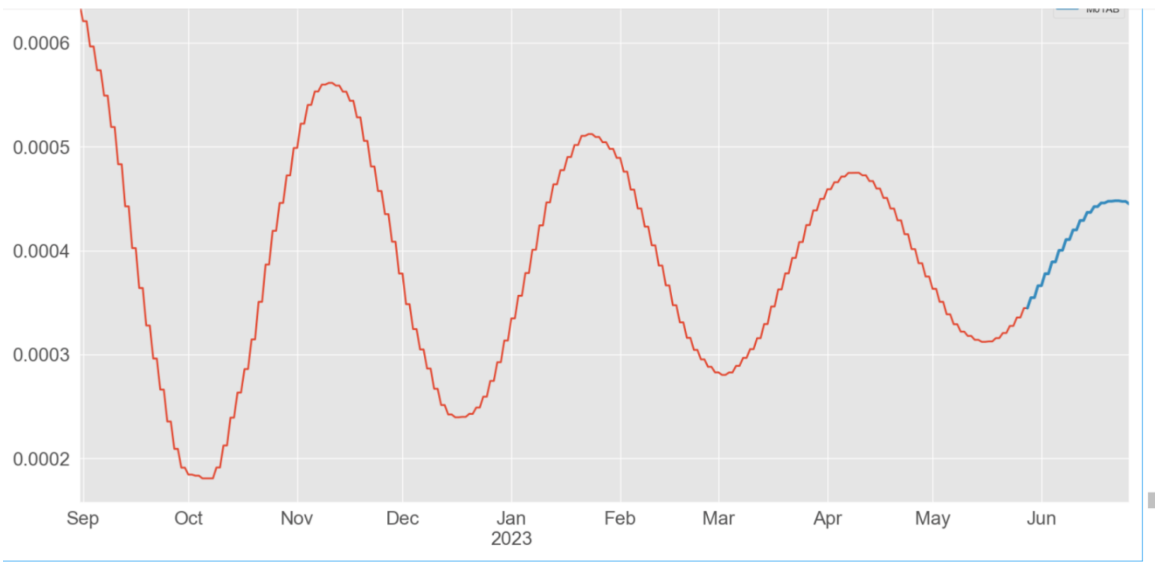


Figure 51: Forecasting till now

7.7 Deployment:

After checking forecasting on notebook, the project is deployed in local machine using modular programming. Also, API is made to connect frontend for forecasting. Then actual sales data are fetched from database and that data are used to retrain model itself on the interval of time, and we have fixed the interval of time seven days.

8. Testing

Testing is important phase to ensure that the system meets the requirements that guided its design and development, responds correctly to all kinds of inputs and achieves the general result its stakeholders desire. We evaluated to make sure that all the developed modules and packages worked properly and functions without any loss in preciseness. For this test plan of our work was created, in which elements such as validation, reliability and user acceptance were tested. The system was tested for normal condition, primarily. We used both unit testing and integration testing to assure the system is developed in right track as per the requirements analyzed and the product intended to work for the real people problems.

Test No.	Unit	Test	Expected Result	Actual Result	Evidence	Result
1.	Email Verification	Verify email through OTP	Email successfully verified	Invalid email	Test 1.1	Failed
2.	Email Verification	Verify email through OTP	Email successfully verified	Valid email	Test 1.2	Passed
2.	User Register	Check whether a new account can be created on filling up required details.	Account Successfully created	Account Successfully created	Test 2.1	Passed
3.	Login	Check Login credentials for valid email and password	User is successfully logged in	Invalid email and password	Test 3.1	Failed
4.	Login	Check Login credentials for valid email and password	User is successfully logged in	Valid email and password	Test 3.2	Passed

6.	Inventory Management	Add medicine to inventory	Medicine added to inventory	Medicine added to inventory	Test 4.1	Passed
7.	Inventory Management	Remove medicine from inventory	Medicine removed from inventory	Medicine removed from inventory	Test 4.2	Passed
8.	Place order	Place order for medicine	Order placed successfully	Order placed successfully	Test 5.1	Passed
9.	Demand Forecasting	Generate demand forecast	Demand forecast generated for the specified period	Demand forecast generated for the specified period	Test 6.1	Passed
10.	Demand Forecasting	View demand forecast	Demand forecast displayed	Demand forecast displayed	Test 7.1	Passed
11.	Reporting	Generate order report	Ordered report generated with relevant data	Ordered report generated with relevant data	Test 8.1	Passed

Table 5: Testing

9. Budget Estimates

9.1 Functional Project

Function point metric is used to collect direct measure of software engineering not only according to size but also according to functionality. FP is derived using an empirical relationship based on countable measures and assessment of s/w complexity.

Information DomainValues	Count	Weighted Value	Total Count (Weight * Count)
No of user inputs	42	4	168
No of user Outputs	9	5	45
No of User Inquires	5	4	20
No of Logical Files	12	11	132
No of External interfaces	2	7	14
Total Count			379

The value of Complexity Multiplier ranges from 0.65 to 1.35. Since, our project is average, the value of the Complexity Multiplier used is average. i.e. We have assumed an average value as 1.17.

Function Point (FP) = Total Count * Complexity Multiplier

$= 379 * [0.65 + 0.01 * \sum_{i=1}^{14} fi$

$= 379 * 1.17$

$= 443.43$

$= 444$

Average productivity = 10 FP / pm

Labor rate = Rs 8000 per month

Effort = Function Point (FP) / Average productivity

= 444/10

= 44.4

Total Project Cost = FP * (Labor Rate / Average Productivity)

= 444 * 8000/10

= Rs. 3,55,200

- Number of user inputs:

  Each user input that provides distinct application-oriented data to the software is counted.

- Number of user outputs:

  Each user output that provides application-oriented information to the user is counted. In this context "output" refers to reports, screens, error messages, etc. Individual data itemswithin a report are not counted separately.

- Number of user inquiries:

  An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

- Number of files:

  Each logical master file is counted.

- Number of external interfaces:

  All machine-readable interfaces that are used to transmit information to another system arecounted.

9.2 Line of Code

LOC (Lines of Code) is a simple and straight forward way of counting the productivity of a programmer in a given time period. Using Lines of Code metric, the project size is estimated by counting the number of source instructions in the developed program.

Estimated LOC = 4200

Average Productivity = 100 LOC/pm

Labor Rate = Rs 8000 per month

Now,

Estimated Project Cost = Estimated LOC * Cost per LOC

= 4200 * (Labor Rate / Average Productivity)

= 4200 * 8000 / 100

= Rs 3,36,000

10. Proposed Deliverable/Output

The project is delivered in the form of web application. The final project has following features:

1. A web application with a user-friendly interface.
2. Registration and login functionality for pharmaceutical companies, stockists, local pharmacies and admin.
3. Local pharmacies, stockists and pharmaceutical company can manage their inventory levels effectively.
4. Local Pharmacies can place orders for medicine to stockists through the app.
5. Stockists can receive orders from respective local pharmacies.
6. The system includes a demand forecasting feature for pharmaceutical companies.

11. Conclusion and Future Extensions

In conclusion, Smart Pharma Demand Forecasting has the potential to revolutionize the pharmaceutical industry by optimizing inventory management and streamlining the supply chain. By leveraging data analytics, machine learning, and artificial intelligence, local pharmacies, stockists, and pharmaceutical companies can collaborate effectively to ensure the availability of medicines while minimizing wastage and reducing costs.

However, there are several future extensions and considerations to enhance the capabilities of smart pharma demand forecasting:

1. Secure payment integration will be implemented.
2. Customers can also order medicine by uploading prescription.
3. Push notifications and reminders about orders and delivery status will be implemented.
4. Stockists can also order medicine from pharmaceutical company.
5. Real-time data sources such as point-of-sale systems, electronic health records, and weather conditions will be incorporated that can provide more accurate and up-to-date information for demand forecasting.

Bibliography/References

[1] "Software Engineering-The Incremental Model." [Online]. Available: [Online]. Available: https://www.geeksforgeeks.org/software-engineering-incremental-process-model/ Accessed on: June 1, 2023.

[2] Simeon Kostadinov, “Understanding GRU Networks.” [Online]. Available: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be . Accessed on: July 4, 2023

[3] "NepMeds: Nepali Online Pharmacy." [Online]. Available: https://www.nepmeds.com.np. Accessed on: 10 May, 2023.

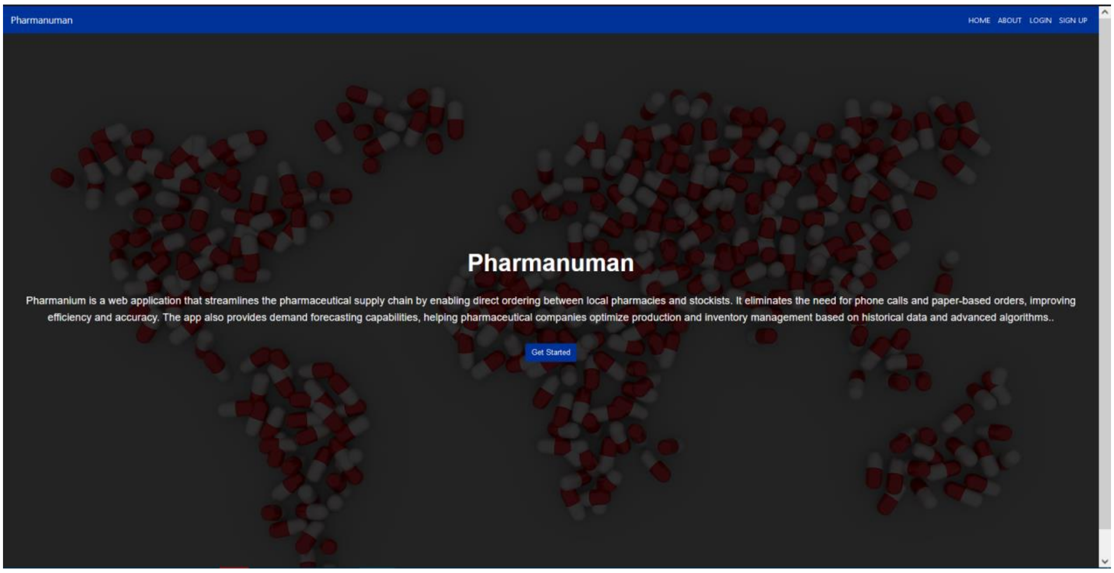
[4] Sumaiya Farzana, and N Prakash, “Machine Learning in Demand Forecasting.” Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3733548 Accessed on: June 10, 2023

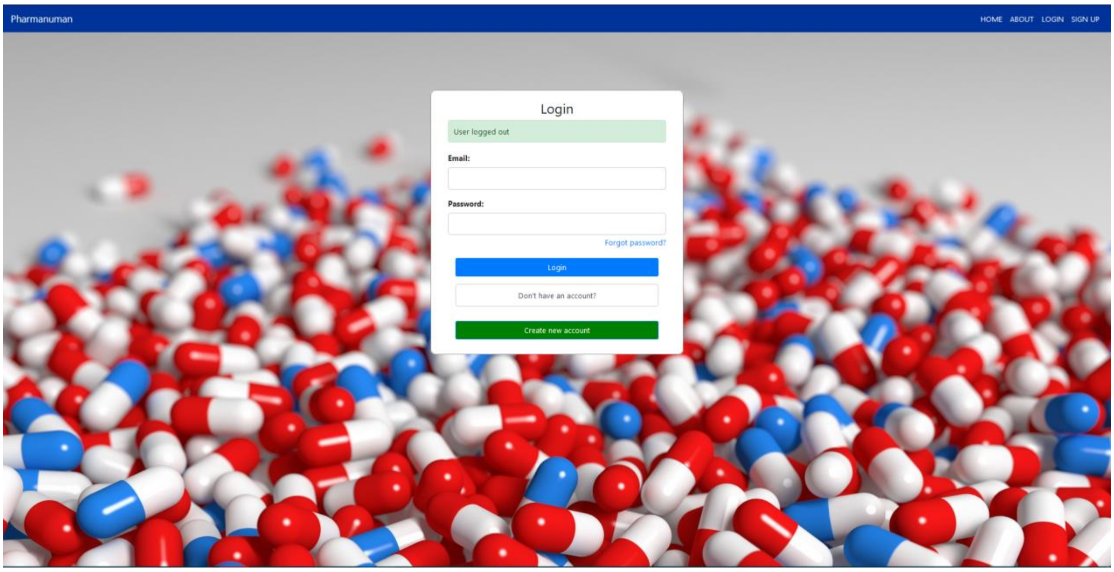
[5] Mariam Atwani, Mustapha Hlyal and, Jamila Elalami **“**A Review of Artificial Intelligence applications in Supply Chain

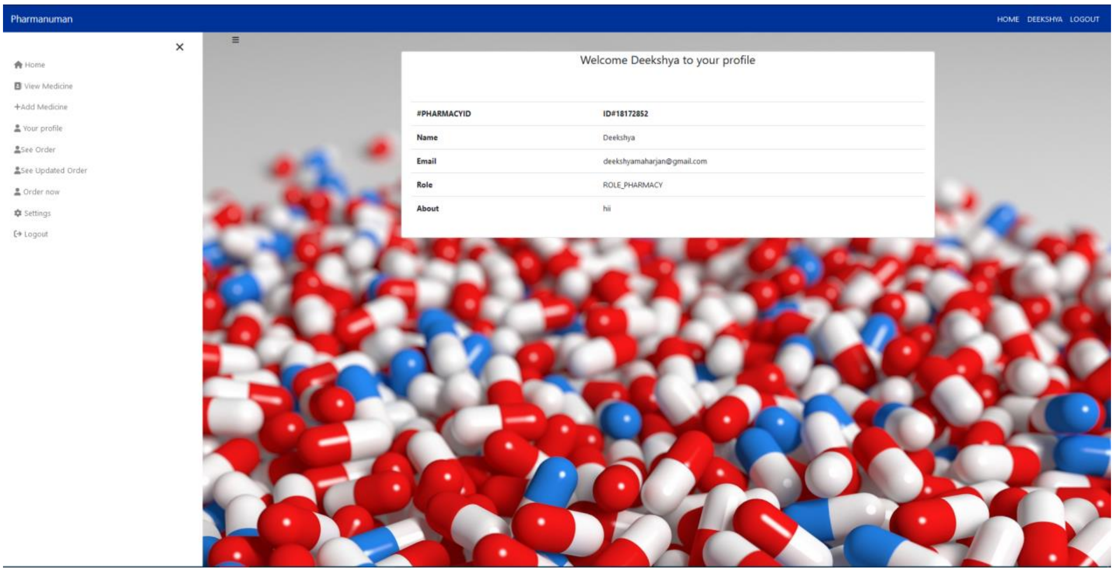
Available:https://www.researchgate.net/publication/361110972_A_Review_of_Artificial_Intelligence_applications_in_Supply_Chain Accessed on: June 10, 2023

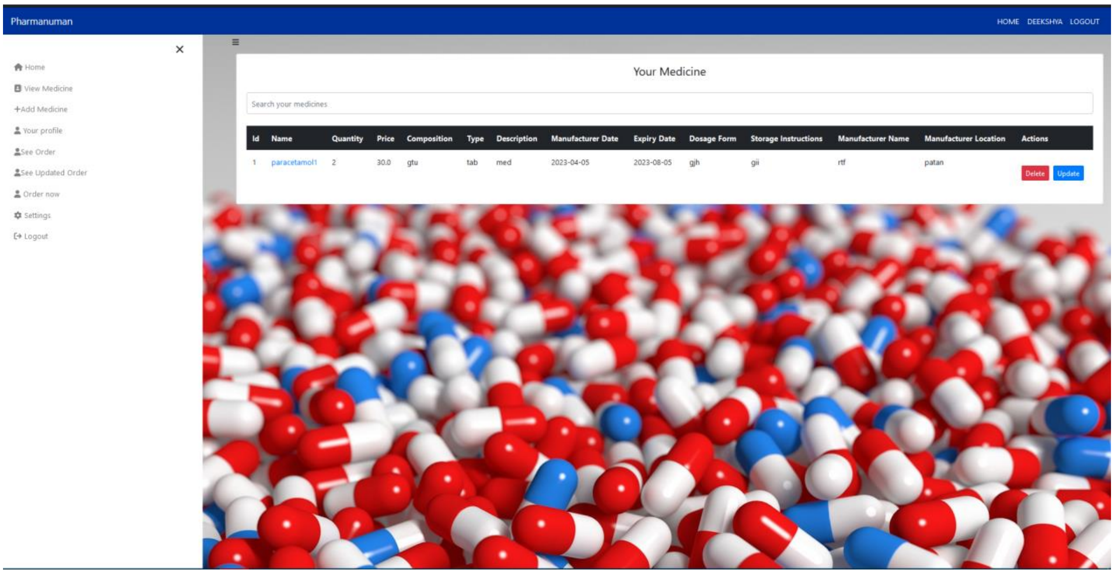
[6] R. Rathipriya, Abdul Aziz Abdul Rahman, S. Dhamodharavadhani, Abdelrhman Meero & G. Yoganandan **“**Demand forecasting model for time-series pharmaceutical data using shallow and deep neural network model” Available: https://link.springer.com/article/10.1007/s00521-022-07889-9 Accessed on: May 13, 2023

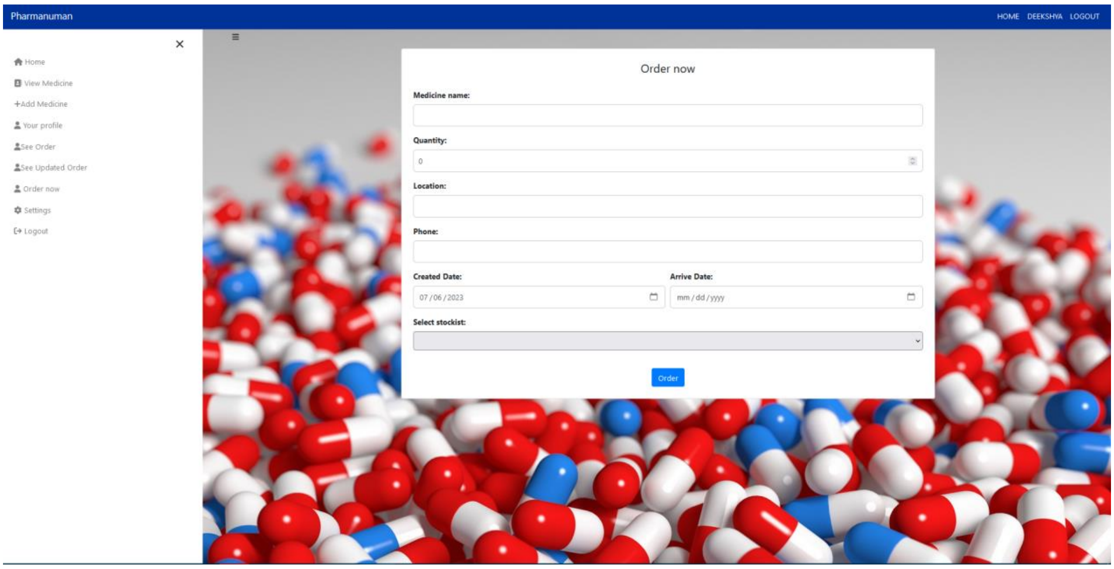
Appendix

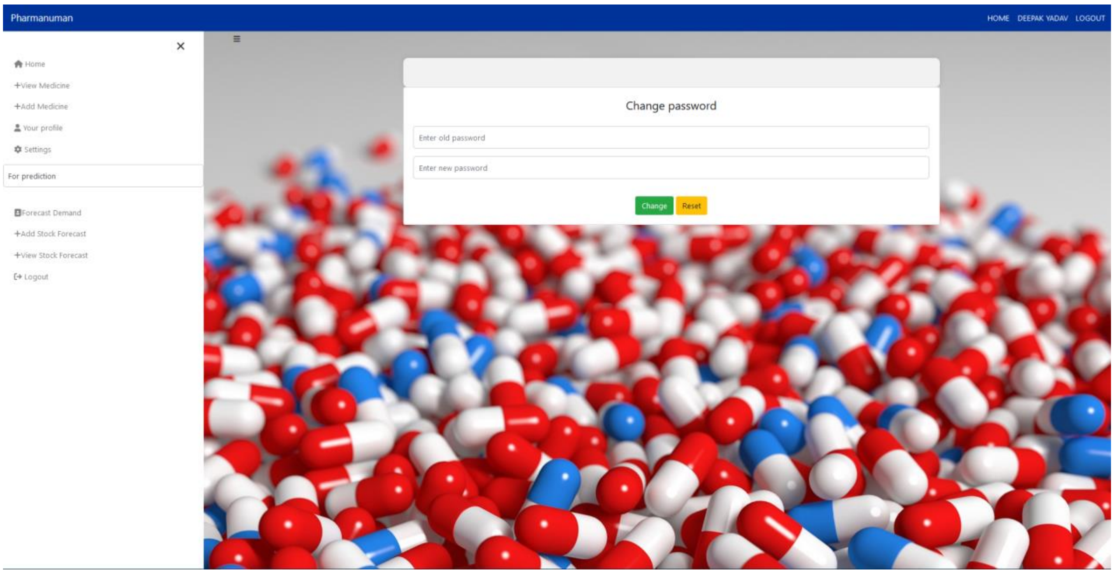

Pharmanuman
HOME ABOUT LOGIN SIGN UP
Pharmanuman
Pharmanium is a web application that streamlines the pharmaceutical supply chain by enabling direct ordering between local pharmacies and stockists. It eliminates the need for phone calls and paper-based orders, improving efficiency and accuracy. The app also provides demand forecasting capabilities, helping pharmaceutical companies optimize production and inventory management based on historical data and advanced algorithms..
Get Started

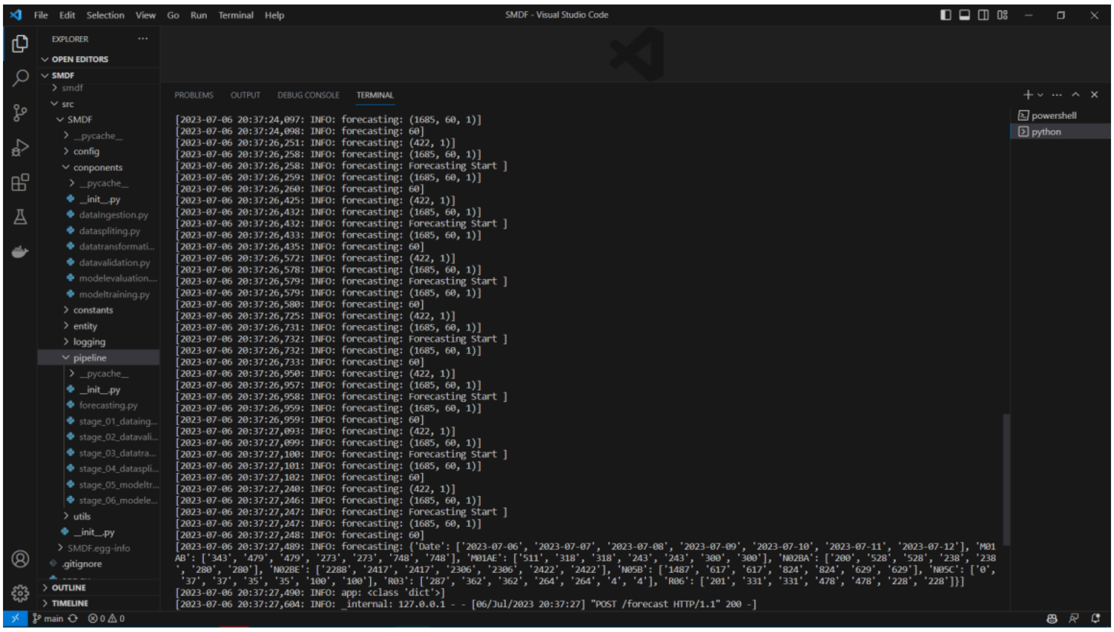

Pharmanuman
HOME ABOUT LOGIN SIGN UP
Login
User logged out
Email:
Password:
Forgot password?
Login
Don't have an account?
Create new account

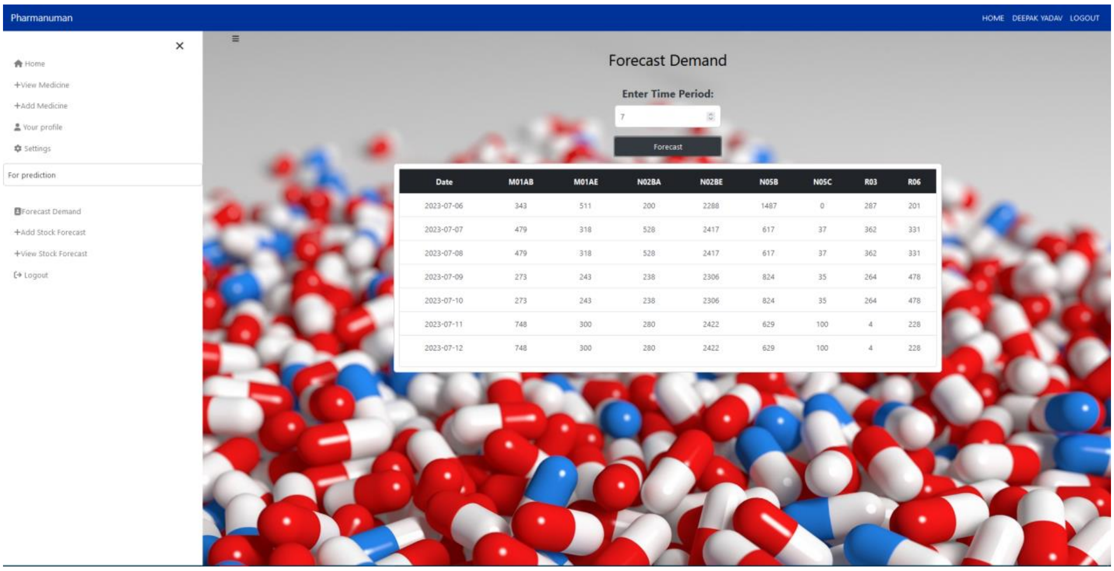

Pharmanuman
HOME
DEEKSHYA
LOGOUT
Home
View Medicine
+Add Medicine
Your profile
See Order
See Updated Order
Order now
Settings
Logout
Welcome Deekshya to your profile
#PHARMACYID
ID#18172852
Name
Deekshya
Email
deekshyamaharjan@gmail.com
Role
ROLE_PHARMACY
About
hii

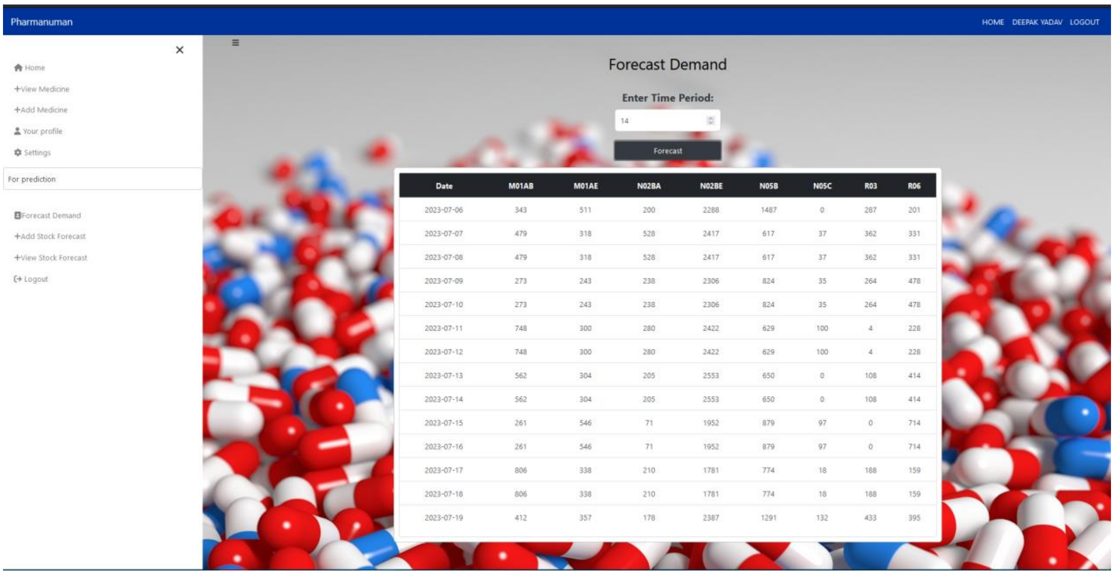

Pharmanuman
HOME
DEEKSHYA
LOGOUT
Home
View Medicine
+Add Medicine
Your profile
See Order
See Updated Order
Order now
Settings
Logout
Your Medicine
Search your medicines
Id
Name
Quantity
Price
Composition
Type
Description
Manufacturer Date
Expiry Date
Dosage Form
Storage Instructions
Manufacturer Name
Manufacturer Location
Actions
1
paracetamol1
2
30.0
gtu
tab
med
2023-04-05
2023-08-05
gjh
gii
rtf
patan
Delete
Update


Pharmanuman
HOME DEEKSHYA LOGOUT
Home
View Medicine
Add Medicine
Your profile
See Order
See Updated Order
Order now
Settings
Logout
Order now
Medicine name:
Quantity:
Location:
Phone:
Created Date:
Arrive Date:
Select stockist:
Order


Pharmanuman
HOME DEEPAK YADAV LOGOUT
Home
View Medicine
Add Medicine
Your profile
Settings
For prediction
Forecast Demand
Add Stock Forecast
View Stock Forecast
Logout
Change password
Enter old password
Enter new password
Change
Reset


File Edit Selection View Go Run Terminal Help
SMDF - Visual Studio Code
EXPLORER
OPEN EDITORS
SMDF
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
python
OUTLINE
TIMELINE
main


Pharmanuman
HOME DEEPAK YADAV LOGOUT
Home
View Medicine
Add Medicine
Your profile
Settings
For prediction
Forecast Demand
Add Stock Forecast
View Stock Forecast
Logout
Forecast Demand
Enter Time Period:
7
Forecast
Date M01AB M01AE N02BA N02BE N05B N05C R03 R06
2023-07-06 343 511 200 2288 1487 0 287 201
2023-07-07 479 318 528 2417 617 37 362 331
2023-07-08 479 318 528 2417 617 37 362 331
2023-07-09 273 243 238 2306 824 35 264 478
2023-07-10 273 243 238 2306 824 35 264 478
2023-07-11 748 300 280 2422 629 100 4 228
2023-07-12 748 300 280 2422 629 100 4 228



Date	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06
2023-07-06	343	511	200	2288	1487	0	287	201
2023-07-07	479	318	528	2417	617	37	362	331
2023-07-08	479	318	528	2417	617	37	362	331
2023-07-09	273	243	238	2306	824	35	264	478
2023-07-10	273	243	238	2306	824	35	264	478
2023-07-11	748	300	280	2422	629	100	4	228
2023-07-12	748	300	280	2422	629	100	4	228
2023-07-13	562	304	205	2553	650	0	108	414
2023-07-14	562	304	205	2553	650	0	108	414
2023-07-15	261	546	71	1952	879	97	0	714
2023-07-16	261	546	71	1952	879	97	0	714
2023-07-17	806	338	210	1781	774	18	188	159
2023-07-18	806	338	210	1781	774	18	188	159
2023-07-19	412	357	178	2387	1291	132	433	395