A Internship Report
on

Forecasting Application

Submitted for partial fulfillment of the requirements for the award of the degree
of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Mr. Thimirishetty Karthikeya (2451-20-733-152)

Maturi Venkata Subba Rao Engineering College
(An Autonomous Institution)
Department of Computer Science and Engineering
(Affiliated to Osmania University & Recognized by AICTE)
Nadergul(V), Balapur(M), RR Dist. Hyderabad – 501 510

Academic Year: 2023-24.

GHX.

June 30, 2023
Ref/GHX/INT/07-01

To Whosoever It May Concern

This is to certify that **Mr. Thimirishetty Karthikeya,** Implementation intern with our organization for the period of 1 Month from May 29, 2023, to June 29, 2023, has successfully completed the Internship Program, to our satisfaction.

Project: Hospital Procedural Demand Forecast.
(using Time Series forecasting methods which empowers OR and Supply chain leaders in making business decisions.)

Issued for the record of the concerned.

Yours Sincerely,

For GHX India Pvt. Ltd.

GHX India Pvt. Ltd. INDIA

Saritha Gorthy
Associate Director, HR India

GHX India Private Limited (Formerly Known as Management Health Solutions India Pvt. Ltd.)
CIN No: U74900TG2014FTC096967 infohyd@ghx.com www.ghx.com 040 6926 7777
802, 8th Floor, Bldg. No. 9, Raheja Mindspace IT Park, HITEC City, Hyderabad, Telangana – 500081, India

Maturi Venkata Subba Rao Engineering College
(An Autonomous Institution)

(Affiliated to Osmania University, Hyderabad)
Nadergul(V), Hyderabad-501510

Certificate

This is to certify that the Summer Internship Report entitled "Forecasting Application", is the bonafide record of the summer internship carried out by **Mr. Thimirishetty Karthikeya (2451-20-733-152)** in partial fulfilment of the requirements for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering** from **Maturi Venkata Subba Rao (MVSR) Engineering College**, affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2023 - 24 under our guidance and supervision.

Coordinator **Internal Evaluator** **Head of the Department**

DECLARATION

This is to certify that the work reported in the present project entitled "**Forecasting Application"** is a record of bonafide work done by me in the Department of Computer Science and Engineering, Maturi Venkata Subba Rao Engineering College, Osmania University. The report is based on the project work done entirely by me and not copied from any other source.

Mr. Thimirishetty Karthikeya
2451-20-733-152

ACKNOWLEDEGEMENTS

I would like to express my sincere gratitude and indebtedness to my Summer Internship Coordinator N.Sabitha, Assistant Professor for her valuable suggestions and interest throughout the Internship.

I am also thankful to our principal**, Dr. G. Kanaka Durga** and **J. Prasanna Kumar**, Head, Department of Computer Science and Engineering, Maturi Venkata Subba Rao Engineering College, Hyderabad for providing excellent infrastructure for completing this Summer Internship successfully as a part of our B.E. Degree (CSE).

I convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed. Finally, I would like to take this opportunity to thank my family for their support through the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

Mr. Thimirishetty Karthikeya

2451-20-733-152

Course Name: Summer Internship

Course Code: SI 671 CS

AY: 2023-2024

Credits: 2

Course Objectives :

To prepare the students

- To give an experience to the students in solving real life practical problems with all its constraints.
- To give an opportunity to integrate different aspects of learning with reference to real life problems.
- To enhance the confidence of the students while communicating with industry engineers and give an opportunity for useful interaction with them and familiarize with work culture and ethics of the industry.

Course Outcomes:

On successful completion of this course student will be

- Formulate a problem to map the requirements of real world scenario.
- Design/develop a small and suitable product in hardware or software.
- Exhibit the skills to use contemporary technologies used by the industry.
- Evaluate the solution against pre-existing alternatives with reference to pre specified criteria.
- Demonstrate an understanding of work culture and ethics of the industry.
- Display effective technical communication skills both orally and written in the form of a report.

ABSTRACT

Transforming the healthcare sector is at the core of this internship project, as it aims to revolutionize healthcare operations through the implementation of advanced demand forecasting. To achieve this, the project focuses on creating a specialized application specifically designed for the healthcare industry. By combining four sophisticated machine learning models, namely ARIMA, Holt's Winter, Exponential Smoothing, and Moving Averages, the application equips healthcare professionals with a versatile set of tools to accurately predict procedural demands. This breakthrough is vital in tackling the complex obstacles of inventory management and securing uninterrupted patient care.

In the intricate landscape of healthcare supply chains, a critical need for accurate inventory management arises. To meet this demand, our project offers a cutting-edge forecasting system. Grounded in machine learning techniques known for their proficiency in identifying trends, seasonality, and distinct patterns in healthcare demand, our methodology stands out. Our user-friendly application makes it effortless for healthcare providers to input historical data, enabling them to make well-informed choices about resource allocation and inventory levels with ease.

In a practical setting, like a surgical ward in a hospital, this application is absolutely essential. Its precise predictions enable proactive decisions about resource allocation, thus minimizing the chances of running out of supplies, having too much inventory, or experiencing delays in procedures. This not only improves operational effectiveness and cuts costs, but most importantly, it enhances the level of patient care provided.

To summarize, this internship project offers a revolutionary remedy for the difficulties encountered in the healthcare industry. By incorporating advanced machine learning techniques, the application effectively caters to the practical requirements of healthcare providers, underscoring the significance of data-informed choices. This breakthrough not only improves healthcare processes but also showcases the tangible influence of technology in tackling real-world obstacles.

TABLE OF CONTENTS

CONTENTS

LIST OF FIGURES

CHAPTER I
INTRODUCTION

Global Healthcare Exchange (GHX) is a leading healthcare technology company that provides cloud-based supply chain solutions to hospitals and other healthcare providers. Founded in 2000, GHX has a mission to reduce the cost of doing business in healthcare by automating supply chain processes and improving visibility into the products used in patient care. The company's network connects tens of thousands of healthcare organizations across the globe, including hospitals, health systems, group purchasing organizations, and suppliers. GHX's products and services help healthcare providers to:

- Reduce costs by automating supply chain processes and eliminating unnecessary spending.
- Improve efficiency by streamlining supply chain processes and reducing the time it takes to order and receive supplies.
- Enhance quality by providing visibility into the products used in patient care and ensuring that the right supplies are available at the right time.
- GHX is committed to helping healthcare providers to reduce costs, improve efficiency, and enhance quality.

During my internship at GHX, I had the opportunity to work on a project to develop a forecasting app that would help hospitals better manage their inventory demand. The healthcare industry is a complex and dynamic environment where the demand for supplies can fluctuate rapidly. This makes it challenging for hospital supply chains to manage inventory demand effectively, which can lead to shortages or overstocking of critical items.

1.1 Problem Statement

The project aimed to address the pressing need for accurate procedure demand forecasting in hospital supply chains. By developing precision forecasts that encompass patient volumes, treatment protocols, and potential emergency situations, hospitals can optimize their inventory levels, ensuring the availability of crucial supplies for specific procedures.

1.2 Proposed System

The objective of the project was to develop a forecasting app that would help hospitals to:
Reduce costs by more accurately predicting demand, avoiding stockouts, and overstocking.
Improve patient care by having the right supplies on hand to provide timely and effective care.
Increase efficiency by streamlining the forecasting process and reducing the time spent on manual forecasting tasks.

The approach that we took to develop the forecasting app was as follows:

1. We collected data on historical procedure demand, patient volumes, treatment protocols, and other relevant factors.
2. We used the data to train a machine learning model to predict future procedure demand.
3. We developed a user-friendly web-based app that hospitals can use to access the forecasts.

1.3 Scope of the Project

The scope of the project to develop a forecasting app at GHX included the following:

- Data Collection: Gathering historical procedure demand data, patient volumes, treatment protocols, and other relevant factors from various healthcare organizations.
- Machine Learning Model Development: Employing machine learning techniques to analyze the collected data and train a predictive model for future procedure demand.

- User-Friendly App Development: Designing and implementing a user-friendly web-based application for hospitals to access and utilize the forecasting insights.
- App Deployment and Implementation: Integrating the forecasting app into GHX's existing supply chain management platform and assisting hospitals in adopting and utilizing the app effectively.
- Continuous Monitoring and Improvement: Monitoring the performance of the forecasting app, gathering feedback from hospitals, and implementing enhancements to improve its accuracy and usability.

The project's scope was carefully defined to ensure that the developed forecasting app would address the specific needs of hospitals in managing inventory demand, while also being user-friendly and scalable to support a wide range of healthcare organizations.

CHAPTER II
TOOLS AND TECHNOLOGIES

2.1 Literature Survey

Efficiently managing the flow of goods and services is essential for optimal patient care in hospitals. A thorough examination of existing research highlights the difficulties of this field, including the importance of accurate demand forecasting, inventory maintenance, and the intricacies of healthcare logistics. The significance of effective supply chain management in hospitals cannot be underestimated, as it directly affects healthcare costs and patient outcomes. By identifying best practices and strategies, we can enhance proficient hospital supply chain management.

Accurate forecasting plays a crucial role in maximizing the efficiency of hospitals in the healthcare industry. It is widely recognized in research that precise demand forecasting has a direct impact on inventory management, cost reduction, and ultimately, patient care. Real-world case studies provide valuable insights into the practical benefits of forecasting in the healthcare setting, demonstrating successful implementations and underscoring the integral role it plays in improving overall operations.

The merging of machine learning and healthcare prediction is a rapidly growing field of study. Extensive research indicates a rising fascination with utilizing machine learning techniques to forecast demands within healthcare supply chains. Notable successes demonstrate improved accuracy in forecasting, resulting in lower costs and improved patient well-being. However, the potential challenges in implementing machine learning, such as safeguarding sensitive data and understanding model outcomes, are also addressed. Despite these obstacles, the literature strongly suggests that incorporating machine learning into healthcare prediction has immense potential to streamline operations and decrease reliance on manual forecasting methods.

2.2 Hardware Requirements

The project requires the following hardware components:

- Processor: A multi-core processor, such as Intel Core i5 or higher, is recommended to handle the computational requirements of the machine learning algorithms and data processing tasks efficiently.

- Memory: Sufficient RAM, preferably 8 GB or higher, is necessary to accommodate large datasets and perform complex computations without memory constraints.
- Storage: Adequate storage space is required to store the project files, datasets, trained models, and any additional resources utilized during the development and deployment phases.

2.3 Software Requirements

The project relies on the following software tools and technologies:

- Python: Python programming language provides a versatile and extensive ecosystem for machine learning, data analysis, and web development. It serves as the primary language for implementing the project components.

- Scikit-learn: Scikit-learn is a popular machine learning library in Python, offering a wide range of algorithms for classification, data preprocessing, and model evaluation.

- Flask: Flask is a lightweight web framework in Python that enables the development of web applications. It is employed to create the user interface and integrate the trained models.

- Pandas: Pandas is a powerful library for data manipulation and analysis in Python. It is used for handling and processing the dataset, performing exploratory data analysis, and preparing the data for model training.

- NumPy: NumPy is a fundamental library for numerical computations in Python. It provides efficient array operations and mathematical functions, which are utilized in various data processing and model training tasks.

- HTML, CSS, JavaScript: These web technologies are utilized for designing and developing the user interface. HTML is used for structuring the web pages, CSS is employed for styling and layout, and JavaScript is utilized for adding interactivity and dynamic behavior to the application.

- Visual Studio Code: Visual Studio Code is a versatile code editor with a rich set of features and extensions. It provides an integrated development environment for writing, debugging, and managing the project codebase.

- Git: Git is a distributed version control system that facilitates collaboration, code management, and tracking of changes in the project. It is used to maintain the project code repository, manage branches, and track the project's evolution over time.

These tools and technologies were chosen based on their effectiveness, compatibility with the project requirements, and the availability of extensive community support and resources. They provide a robust and efficient foundation for implementing the forecasting application, enabling effective data processing, machine learning, and web application development.

CHAPTER III
SYSTEM DESIGN

3.1 Flow Charts

Flow charts will be used to visually represent the sequential process of Forecasting Application. The flow charts will illustrate the step-by-step workflow, starting from the input of data to the final output of the forecasting. Each stage of the process, such as data collection, feature extraction, classification, and result presentation, will be depicted using symbols and arrows to showcase the logical flow and decision-making points. The flow charts will help in understanding the system's operation, identifying potential bottlenecks or areas for improvement, and facilitating communication among project stakeholders.

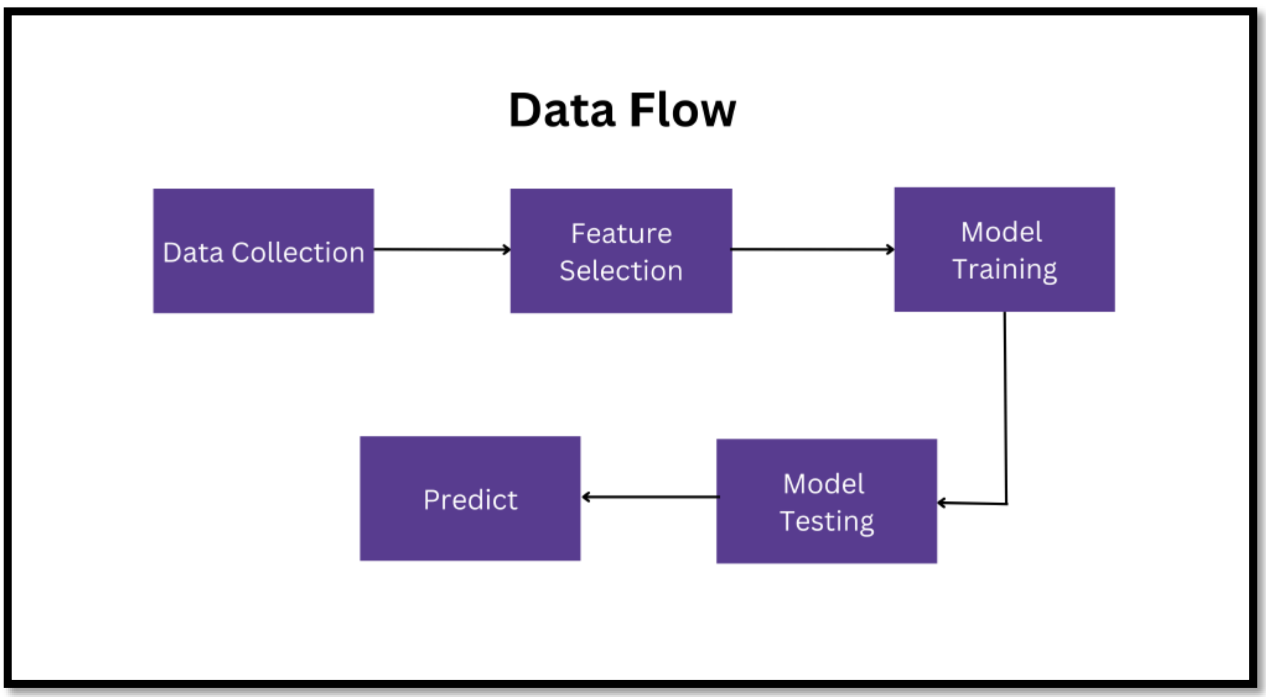

Fig 3.1 Data Flow

3.2 System Architecture

The system architecture for app will comprise various distinct stages for data preprocessing, model training and prediction, and evaluation. This separation promotes code maintainability and scalability.

Components:

Data Input:

- Accepts time series data in various formats (e.g., CSV, JSON).
- Provides options for manual upload or integration with external data sources.

Data Preprocessing:

- Handles missing values imputation using techniques like mean/median filling, interpolation, etc.
- Detects and removes outliers.
- Performs feature engineering like lag creation, rolling statistics, etc.
- Standardizes data scaling if necessary.

Model Training:

- Offers a selection of pre-built time series forecasting models, including:
  - ARIMA
  - Simple Exponential Smoothing (SES)
  - Holt-Winters
  - Moving Average (MA)
- Allows parameter tuning for each model through optimization algorithms.
- Trains and stores multiple models for comparison and ensemble forecasting.

Forecast:

- Uses the trained model(s) to forecast future values for the chosen time horizon.
- Generates forecasts for individual data points or aggregated intervals.

Evaluation:

Calculates various metrics to assess the prediction accuracy, such as:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Mean Absolute Percentage Error (MAPE)
- R-squared score

Output & Reporting:

- Presents the generated forecasts in a user-friendly format (e.g., tables, charts).
- Visualizes the predicted values vs. actual values on charts.
- Allows downloading forecast data and evaluation metrics for further analysis.
- Generates reports summarizing the prediction process and performance.

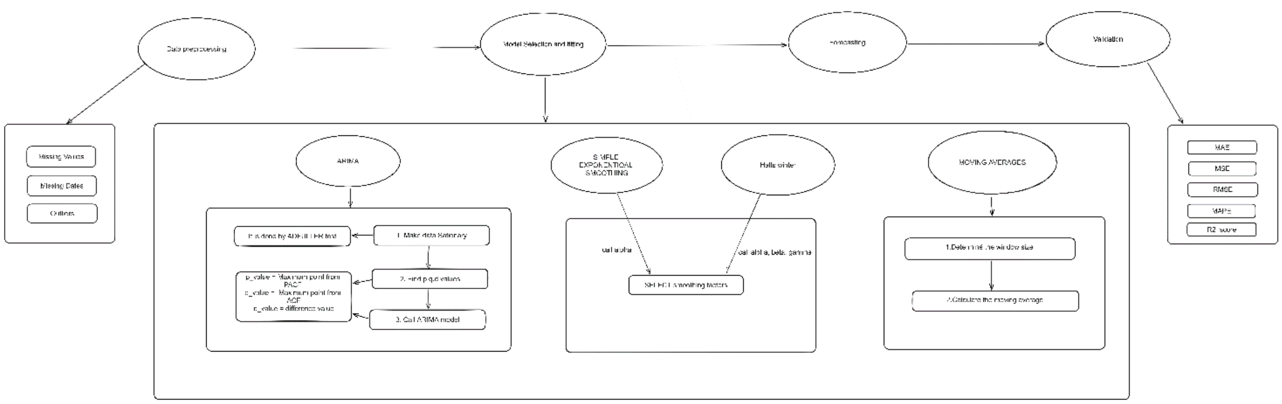

Fig 3.2 System Architecture

3.3 UML Diagrams

Use case diagram:

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of the use case.

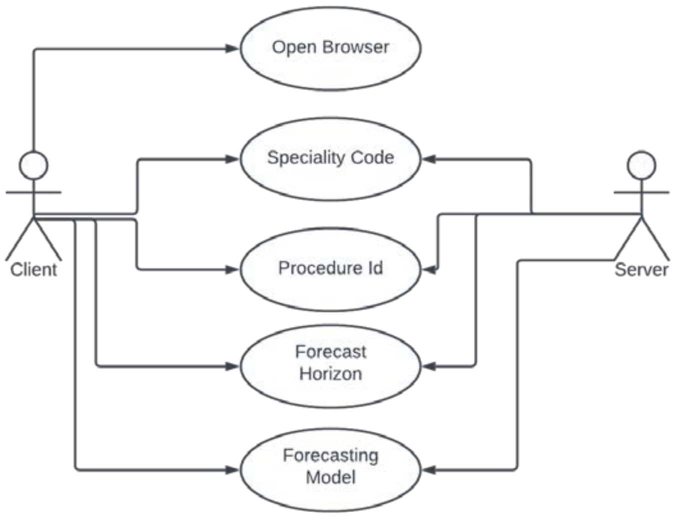

Fig 3.3 Use Case Diagram

Class diagram:

In software Engineering, A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's methods), and the relationships among objects translating the models into programming code.

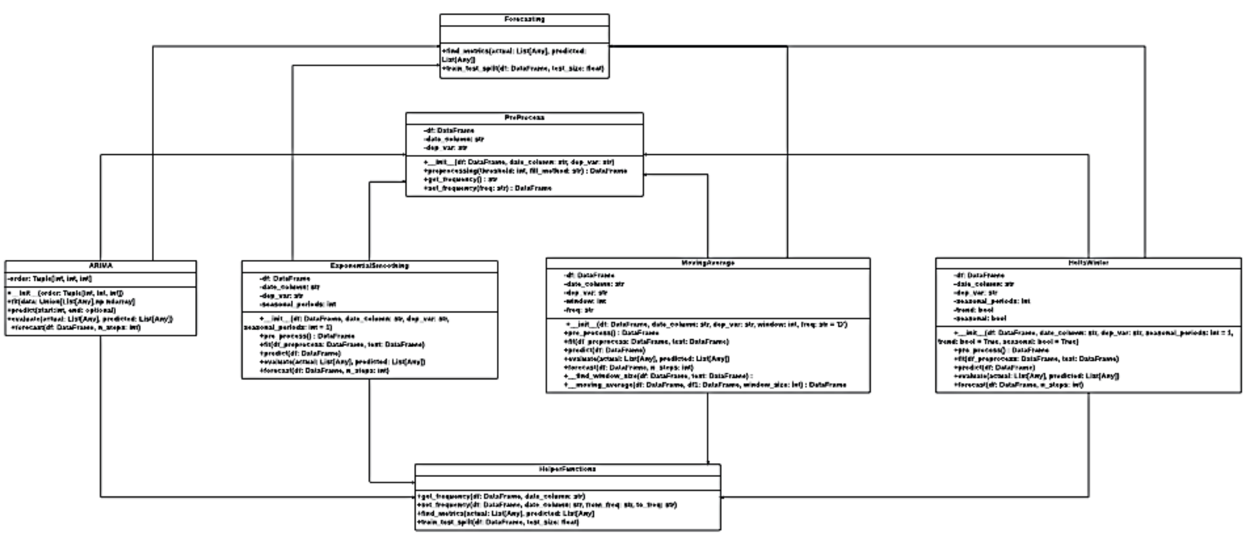

Fig 3.4 Class Diagram

State diagram:

A state diagram is an illustration of all the possible behavioral states a software system component may exhibit and the various state changes it's predicted to undergo over the course of its operations.

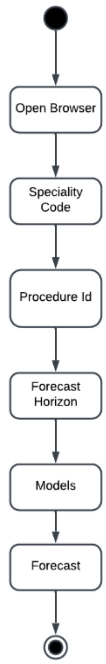

Fig 3.5 State Diagram

CHAPTER IV
SYSTEM IMPLEMENTATION & METHODOLOGIES

4.1 System Implementation

The system implementation involves the utilization of various algorithms and methodologies. The following algorithm provides an overview of the steps involved in the process:

Data Preparation:

- The first step in developing the forecasting app was to collect historical procedure demand data.
- Performed data cleaning and preprocessing to remove irrelevant or noisy information.
- Handled missing values and ensure data consistency.

Feature Extraction:

- Identified relevant features for forecasting, such as treatment protocols, and other relevant factors from various healthcare organizations.
- This data provided a rich source of information for training the machine learning models.

Model Selection and Training:

- Developed appropriate machine learning algorithms for forecasting, such as Arima, Simple Exponential Smoothing, Moving Averages, Holt-Winters.

  The core of the forecasting app consisted of four machine learning models: ARIMA, Simple Exponential Smoothing, Moving Averages, Holt-Winters.

Each model was selected based on its strengths and suitability for different use cases:

- ARIMA: ARIMA is particularly effective in handling time series data with seasonality, making it suitable for forecasting procedure demand that exhibits cyclical patterns.
- Simple Exponential Smoothing: SES is a simple and computationally efficient model that excels at short-term forecasting. It is well-suited for hospitals that prioritize quick and accurate forecasts for immediate decision-making.
- MA: MA models are known for their ability to capture short-term trends and smooth out fluctuations, which is crucial for accurately predicting demand in the near future.
- Holt-Winters: Holt-Winters is a more sophisticated variant of exponential smoothing that incorporates trend and seasonal components, making it suitable for forecasting data with both trend and seasonality.

- We split the dataset into training and testing sets to evaluate model performance.
- Trained the selected models using the training data, tuning hyperparameters as needed.
- Then evaluated model performance using metrics like accuracy, precision, recall, and F1 score.

Model Evaluation and Refinement:

- Assessed the trained models' performance on the testing set and analyze the results.
- Identified areas of improvement, such as addressing false positives or false negatives.
- Iterated and refined the models by adjusting parameters, applying feature engineering techniques, or considering ensemble methods.

4.2 Algorithms

1.ARIMA (Autoregressive Integrated Moving Average)

ARIMA, or Autoregressive Integrated Moving Average, is a statistical forecasting method that captures the relationship between past and future values of a time series. It is particularly well-suited for handling time series data with seasonality, making it a popular choice for forecasting demand in healthcare, retail, and other industries that experience cyclical fluctuations.

ARIMA models are characterized by three parameters:

- p (Autoregressive Order): This parameter determines the number of lagged values of the time series that are used to predict the current value. A higher value of p indicates a stronger dependence on past values.

- d (Degree of Integration): This parameter indicates the number of times the time series needs to be differenced to make it stationary, meaning that the mean and variance are constant over time. Differencing removes non-stationary patterns, such as trends or seasonality.

- q (Moving Average Order): This parameter specifies the number of lagged forecast errors that are used to improve the forecast accuracy. A higher value of q indicates that more weight is placed on recent forecast errors.

ARIMA models are typically represented by the notation ARIMA(p, d, q), where the values of p, d, and q are determined through an iterative process of model selection and evaluation.

Application in Time Series Forecasting:

ARIMA is particularly effective for forecasting time series data that exhibits seasonality and trend. It can handle complex patterns and relationships between past values, making it a versatile forecasting tool.

2.Moving Averages

Moving averages are a simple and effective class of forecasting models that rely on averaging historical values to predict future values. They are particularly useful for smoothing out fluctuations and capturing short-term trends in time series data.
There are two main types of moving averages: simple moving averages (SMA) and weighted moving averages (WMA).

Simple Moving Average (SMA): An SMA is the average of the most recent n observations, where n is the window size. It assigns equal weights to all observations within the window.

Weighted Moving Average (WMA): A WMA assigns different weights to different observations, typically giving more weight to recent observations and less weight to older observations. This allows for more emphasis on recent trends.

Application in Time Series Forecasting:

Moving averages are suitable for short-term forecasting and smoothing out fluctuations in time series data. They are easy to understand and implement, making them a popular choice for quick and intuitive forecasts.

3.Simple Exponential Smoothing (SES)

Simple exponential smoothing (SES) is a forecasting method that uses an exponential weighting scheme to combine historical data with the current forecast. It is a simple and computationally efficient model that is well-suited for short-term forecasting.

SES is characterized by a smoothing parameter α (alpha), which determines the weight given to the most recent observation. A higher value of α places more weight on recent observations, while a lower value places more weight on historical data.

The SES formula is:

$$\mathbf{Ft = \alpha * Yt + (1 - \alpha) * Ft\text{-}1}$$

where:

- F_t is the forecast for period t
- Y_t is the actual value in period t
- F_{t-1} is the forecast for period t-1
- α is the smoothing parameter

Application in Time Series Forecasting:

SES is particularly useful for short-term forecasting and tracking rapidly changing trends in time series data. It is simple to implement and computationally efficient, making it a practical choice for real-time forecasting applications.

4.Holt-Winters

Holt-Winters is an exponential smoothing method that extends SES by incorporating trend and seasonality into the forecasting process. It is a powerful and versatile model that is well-suited for forecasting time series data with both trend and seasonal patterns.

Holt-Winters is characterized by three parameters:

- α (alpha): The smoothing parameter for the level component, similar to SES.
- β (beta): The smoothing parameter for the trend component.
- γ (gamma): The smoothing parameter for the seasonal component.

The Holt-Winters formulas are:

$Level_t = \alpha * Y_t + (1 - \alpha) * (Level_{t-1} + Trend_{t-1})$

$Trend_t = \beta * (Level_t - Level_{t-1}) + (1 - \beta) * Trend_{t-1}$

$Seasonal_t = \gamma * (Level_t - Level_{t-L}) + (1 - \gamma) * Seasonal_{t-L}$

where:

- $Seasonal_t$ is the estimated seasonal component of the time series at period t
- $Level_t$ is the estimated level of the time series at period t
- $Level_{t-L}$ is the estimated level of the time series at period t-L
- L is the length of the seasonal pattern

The γ parameter determines the weight given to the most recent observation within the seasonal period. A higher value of γ indicates a stronger emphasis on recent seasonal patterns.

Application in Time Series Forecasting:

Holt-Winters is particularly useful for forecasting time series data that exhibits strong seasonality. It can handle both trend and seasonality simultaneously, making it a versatile tool for accurately predicting future values.

4.3 Workflow of the project

1. ARIMA (AutoRegressive Integrated Moving Average):

Step 1: Preprocess the data

Check if the time series is stationary (constant mean and variance over time) using techniques like the Augmented Dickey-Fuller (ADF) test. If it's not stationary, apply differencing until you achieve stationarity.

Step 2: Identify the order

Determine the order of the ARIMA model by analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. These plots help identify the optimal values for the p (autoregressive), d (integration), and q (moving average) parameters.

Step 3: Build the ARIMA model

Fit the ARIMA model to the preprocessed data using the determined order.

Step 4: Validate the model

Evaluate the model's performance using metrics like mean absolute error (MAE), mean squared error (MSE), or root mean squared error (RMSE) on a validation dataset or through cross-validation.

Step 5: Forecast future values

Use the fitted ARIMA model to generate forecasts for future time points.

2. Simple Exponential Smoothing:

Step 1: Preprocess the data

Ensure the time series is relatively stationary (no trend or seasonality).

Step 2: Select the smoothing factor

Determine the smoothing factor, denoted as α (alpha), which controls the weight given to the most recent observations. This can be done through techniques like grid search or optimization algorithms to minimize the error.

Step 3: Build the model

Apply the simple exponential smoothing model using the chosen smoothing factor to the preprocessed data.

Step 4: Validate the model

Evaluate the model's performance using metrics like MAE, MSE, or RMSE on a validation dataset or through cross-validation.

Step 5: Forecast future values

Use the fitted simple exponential smoothing model to generate forecasts for future time points.

3.Holt's Winter (Triple Exponential Smoothing):

Step 1: Preprocess the data

Ensure the time series is stationary or transform it accordingly.

Step 2: Select the smoothing factors

Determine the smoothing factors for level (α), trend (β), and seasonality (γ) components. These factors control the weights given to different components when making forecasts. They can be chosen using techniques like grid search or optimization algorithms.

Step 3: Build the model

Apply the Holt's Winter model using the chosen smoothing factors to the preprocessed data.

Step 4: Validate the model

Evaluate the model's performance using metrics like MAE, MSE, or RMSE on a validation dataset or through cross-validation.

Step 5: Forecast future values

Use the fitted Holt's Winter model to generate forecasts for future time points.

\

4.Moving Averages:

Step 1: Determine the window size

Decide on the size of the moving window, which represents the number of past observations to consider for calculating the moving average.

Step 2: Calculate the moving average

Compute the moving average by taking the average of the values within the defined window at each time point.

Step 3: Validate the model

Evaluate the model's performance using metrics like MAE, MSE, or RMSE on a validation dataset or through cross-validation.

Step 4: Forecast future values

Use the calculated moving average to generate forecasts for future time points.

4.4 Validation

1.Mean Absolute Error (MAE):

The average of the absolute differences between the forecasted and actual values.

2.Mean Squared Error (MSE):

The average of the squared differences between the forecasted and actual values.

3.Root Mean Squared Error (RMSE):

The square root of the MSE, providing a measure in the same unit as the original data.

4.Mean Absolute Percentage Error (MAPE):

The average percentage difference between the forecasted and actual values.

5.Forecast Error Variance (FEV):

The variance of the forecast errors, indicating the dispersion of the errors.

6.R-squared (R^2):

A measure of how well the model fits the data, indicating the proportion of the variance in the data explained by the model.

CHAPTER V
APPLICATION

Login Page

The login page acts as a safe and secure gateway to the forecasting app, with strict user authentication required. By entering their exclusive username and password, individuals gain access to the system, safeguarding the confidentiality and integrity of data. With this straightforward and efficacious authentication process, privacy is optimized and access is limited to authorized healthcare experts, ultimately ensuring the utmost security for sensitive forecasting information.

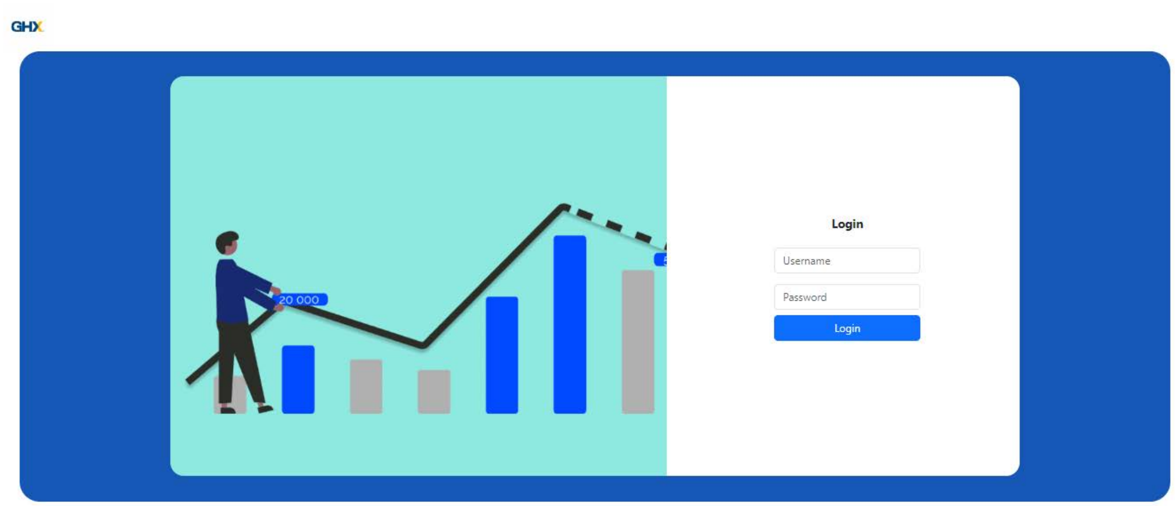

Fig 5.1 Login page

User Interface

The intuitive and user-friendly web application provides healthcare professionals with a guided experience. The interface features four selection lists, allowing users to easily navigate the procedure demand forecasting process. The first list allows for a choice between Orthopedics and Cardiology to define the desired medical specialty. The second list offers a variety of specific procedures, such as "Orknarth" and "CTACBG", for customization. The third list provides flexibility in selecting the time interval for forecasting, with options ranging from daily to yearly. As for the fourth list, users can select from advanced forecasting models including ARIMA, SES, MA, and Holt-Winters. Once all selections have been made, clicking the "Go" button will generate a visual representation of the data using High Charts. This dynamic interface seamlessly integrates user inputs, providing actionable insights for healthcare professionals in managing inventory and optimizing resource allocation.

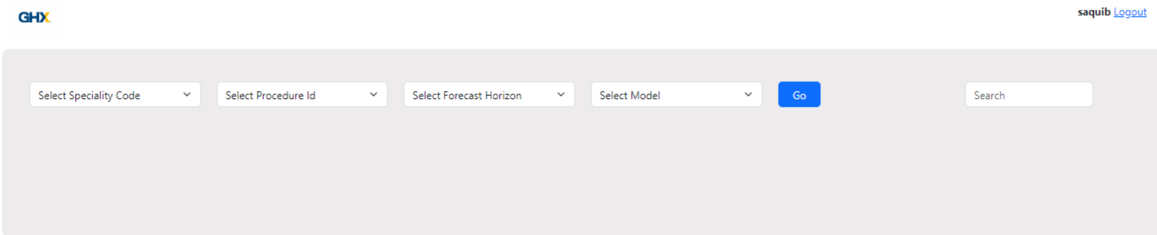

Fig 5.2 Interface

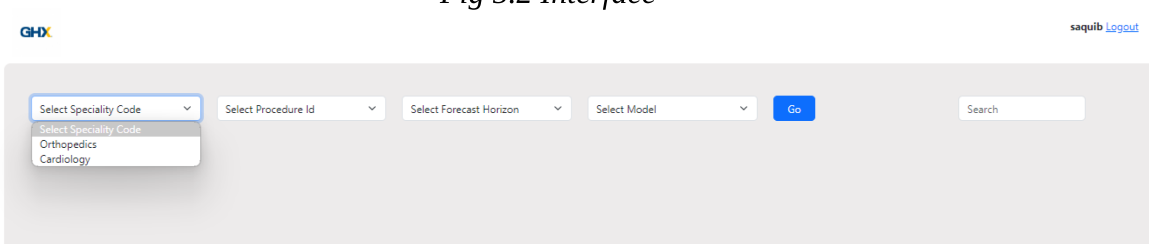

Fig 5.3 Specialty Code

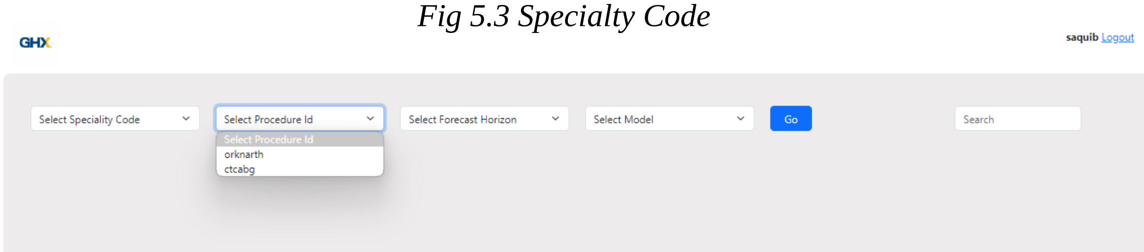

Fig 5.4 Procedure Id

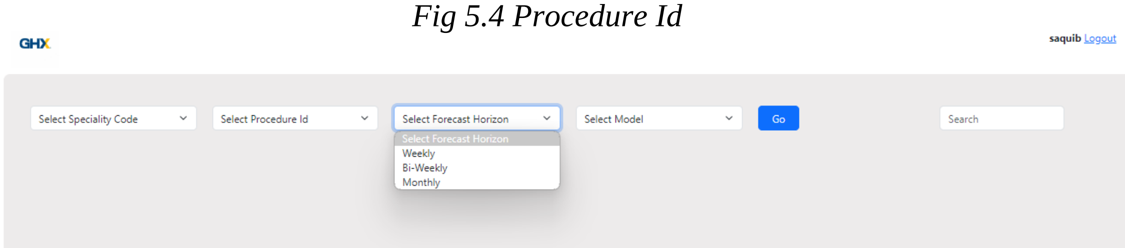

Fig 5.5 Horizon

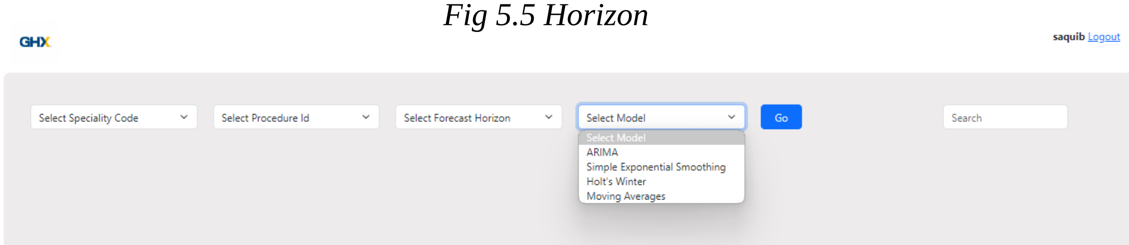

Fig 5.6 Models

ARIMA Model:

This image depicts a visually insightful graph generated by the ARIMA model for weekly forecasting within the web app interface. The x-axis represents the time dimension, showcasing weeks, while the y-axis signifies the forecasted demand for a chosen medical specialty and procedure. The graph exhibits a smooth representation of predicted values, capturing both short-term trends and recurring patterns inherent in weekly demand fluctuations.

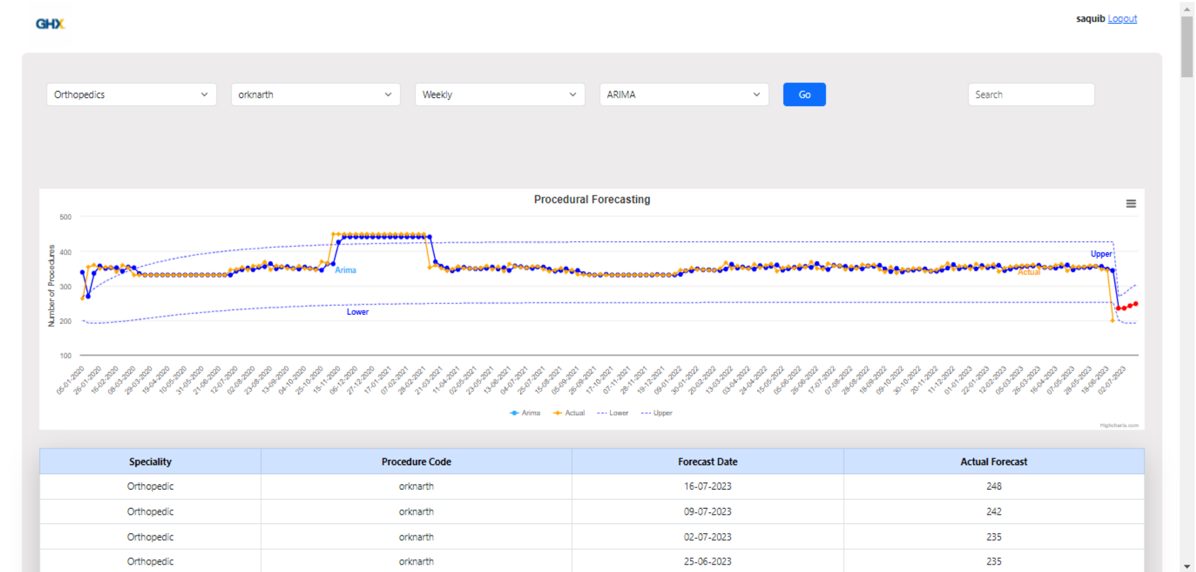

Fig 5.7 ARIMA Model

Simple Exponential Smoothing (SES)

This image depicts a visually insightful graph generated by the Simple Exponential Smoothing model for monthly forecasting within the web app interface. The x-axis represents the time dimension, showcasing weeks, while the y-axis signifies the forecasted demand for a chosen medical specialty and procedure. The graph exhibits a smooth representation of predicted values, capturing both short-term trends and recurring patterns inherent in weekly demand fluctuations.

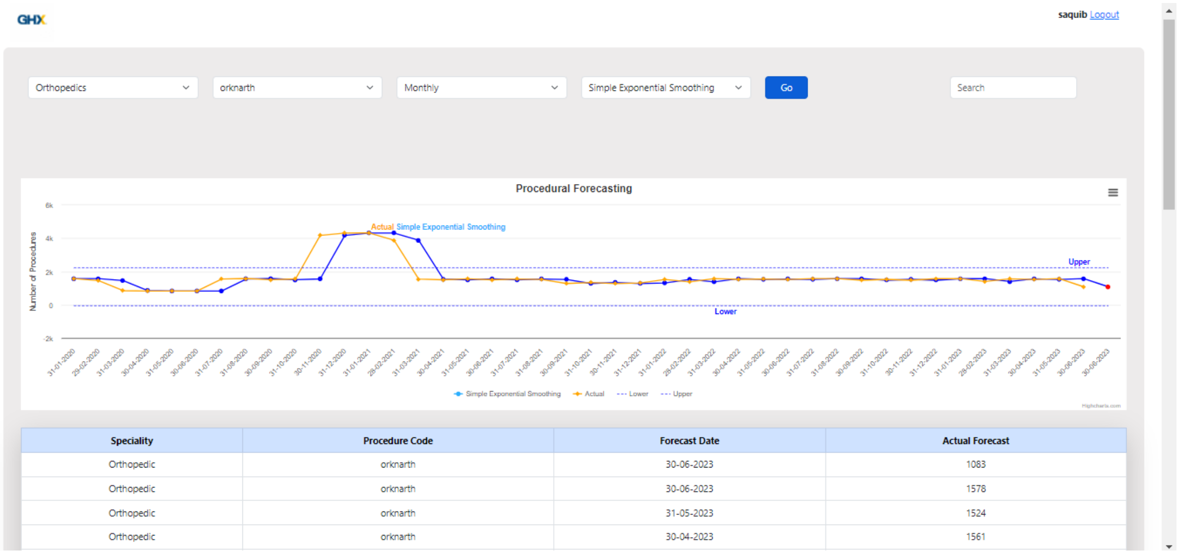

Fig 5.8 SES Model

Moving Averages

This image depicts a visually insightful graph generated by the Moving Average model for monthly forecasting within the web app interface. The x-axis represents the time dimension, showcasing weeks, while the y-axis signifies the forecasted demand for a chosen medical specialty and procedure. The graph exhibits a smooth representation of predicted values, capturing both short-term trends and recurring patterns inherent in weekly demand fluctuations.

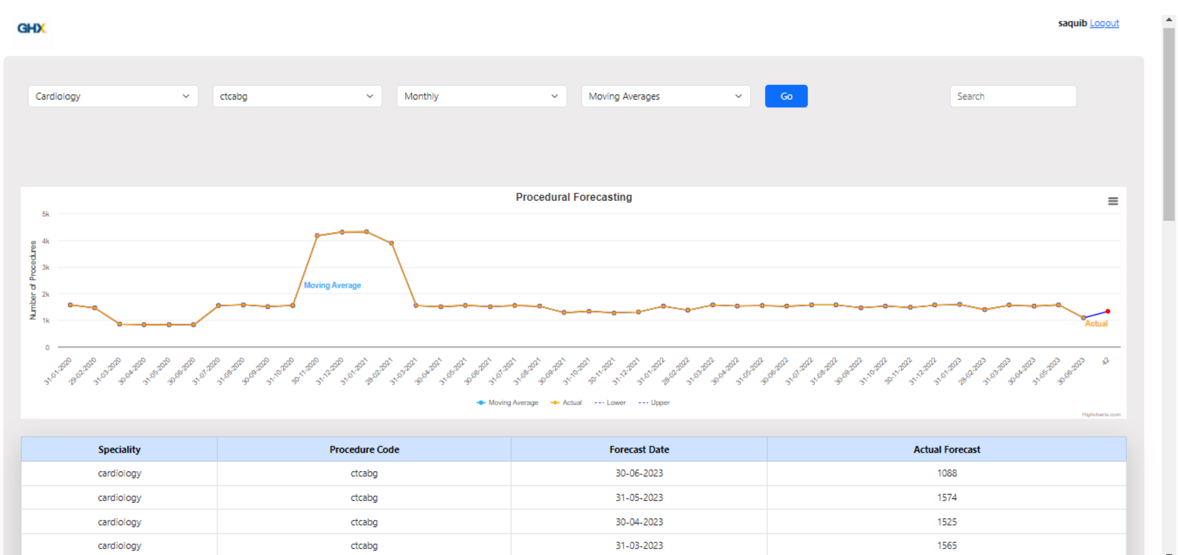

Fig 5.9 Moving Averages Model

Holt-Winters

This image depicts a visually insightful graph generated by the Holt-Winters model for monthly forecasting within the web app interface. The x-axis represents the time dimension, showcasing weeks, while the y-axis signifies the forecasted demand for a chosen medical specialty and procedure. The graph exhibits a smooth representation of predicted values, capturing both short-term trends and recurring patterns inherent in weekly demand fluctuations.

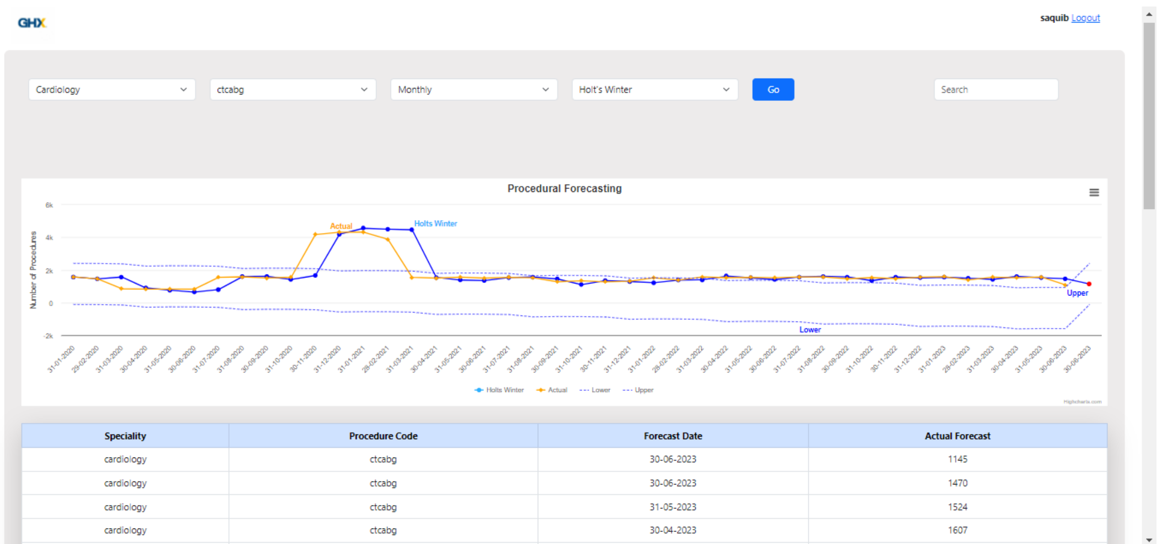

Fig 5.10 Holt-Winter Model

CHAPTER VI
CONCLUSION

In summary, the project has successfully delivered a forecasting app tailored for precise procedure demand predictions in hospital supply chains. Leveraging machine learning models, including ARIMA, SES, MA, and Holt-Winters, the app demonstrates versatility in handling different forecasting scenarios. Rigorous evaluations, utilizing metrics like mean absolute error and mean squared error, highlight the effectiveness of the models.

The project addresses critical needs in hospital supply chain management, aligning with broader objectives of cost reduction, improved patient care, and enhanced operational efficiency. The comprehensive scope, covering data collection, model development, user-friendly app implementation, and continuous monitoring, reflects a holistic approach. Literature surveys on hospital supply chain management, forecasting in healthcare, and machine learning models provide a solid foundation, ensuring the project aligns with industry best practices.

The successful integration of the forecasting app into GHX's supply chain management platform and its adaptability to diverse healthcare settings showcase its potential to significantly impact the industry by facilitating timely and effective resource allocation.

Future Enhancements:

While the developed website provides a robust solution for forecasting, there are several potential areas for future enhancements and advancements. These include:

- Expanding the Training Dataset: Increasing the size and diversity of the training dataset can further improve the system's performance and generalization capabilities.

- Advanced Machine Learning Techniques: Explore the use of more sophisticated machine learning algorithms, such as deep learning models, ensemble methods, or anomaly detection algorithms. These advanced techniques can further improve the accuracy and effectiveness of the system.

- Integration of Real-Time Data: Elevate the capabilities of the forecasting app by incorporating real-time data feeds. This will empower the system to swiftly adapt to unforeseen changes, such as unexpected patient influxes or shifts in treatment protocols. By integrating data streams from electronic health records and other relevant sources, a more dynamic and precise forecasting model can be established.
- Incorporation of External Factors: Expand the forecasting models to include external factors that could impact demand, such as weather conditions, public health trends, or regional events. By incorporating these variables, the forecasting app can provide a more comprehensive and contextualized prediction, improving its adaptability to diverse and dynamic healthcare environments.
- Enhanced User Interface and Interactivity: Enhance the user interface of the web-based app to improve overall user experience and interactivity. Implement features such as customizable dashboards and scenario analysis tools, allowing healthcare professionals to easily interpret and utilize forecasted insights. This will not only facilitate a better understanding of forecasts, but also empower healthcare professionals to make informed decisions based on the presented data.

.

REFERENCES

[1] Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: principles and practice.

[2] Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control.

[3] Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting.

[4] Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). Forecasting: Methods and Applications.

[5] https://www.kaggle.com

[6] https://www.tutorialspoint.com

[7] https://www.analyticsvidhya.com

[7] https://www.tableau.com

APPENDIX

File name:app.py

```
from flask import Flask, request, jsonify, render_template,
redirect,session
from server.Helper_Functions import train_test_split,
read_file
import pandas as pd
from server.Arima import Arima
from server.Simple_Exponential_Smoothing import
SimpleExponentialSmoothing
from server.Holts_Winter import HoltWinter
from server.Moving_Average import MovingAverage
from server.route_helpers import helper_predict_arima,
helper_predict_simple_exponential_smoothing,
helper_predict_holt_winter,helper_predict_moving_average
import sqlite3
import pickle
import os
import json

app = Flask(__name__, template_folder='client/template',
static_folder='client/static')
app.secret_key = 'BAC'

LOGIN_TEMPLATE = 'login.html'
SQLITE_DB_FILE_PATH = 'Forecast-master/data/'
SQLITE_DB_FILE_NAME = 'Sqlite3.db'
DATE_FORMAT = "%d-%m-%Y"
PICKLE_FILE_PATH = 'Forecast-master/Artifacts/'

@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = ''
    if request.method == 'POST' and 'name' in request.form
and 'password' in request.form:
        name = request.form['name']
        password = request.form['password']

        if not name or not password:
            error = 'Please enter both Name and Password'
            return render_template(LOGIN_TEMPLATE,
error=error)

        print(SQLITE_DB_FILE_PATH + SQLITE_DB_FILE_NAME)
```

```
        connection = sqlite3.connect(SQLITE_DB_FILE_PATH +
SQLITE_DB_FILE_NAME)
        cursor = connection.cursor()

        query = "SELECT name, password from users WHERE name
= ? AND password = ? "
        cursor.execute(query, (name, password))
        # cursor.execute('SELECT  name, password FROM users
WHERE name = %s AND password = %s', (name, password, ))
        results = cursor.fetchone()

        if results is None:
            error = 'Sorry, Incorrect Credentials'
            return render_template(LOGIN_TEMPLATE,
error=error)
        else:
            session['loggedin'] = True
            session['username'] = name
            return redirect('/main')

    return render_template(LOGIN_TEMPLATE)

@app.route('/main', methods=["GET", "POST"])
def main_page():
    if not session.get('loggedin'):
        return redirect('/login')

    return render_template('main_page.html')

@app.route('/logout')
def logout():
   # remove the username from the session if it is there
   session.pop('loggedin', None)
   return redirect("/login")

@app.route('/main/predict_arima', methods=['GET', 'POST'])
def predict_arima():
    """
    Route for predicting with ARIMA model.
    """

    data = request.json
    specialty = data['specialty']
    procedure = data['procedure']
    frequency = data['frequency']
    response = helper_predict_arima(specialty, procedure,
frequency)
```

```
    return jsonify(response)

@app.route('/main/predict_holt_winter', methods=['GET',
'POST'])
def predict_holt_winter():
    """
    Route for predicting with Holt-Winters model.
    """
    data = request.json
    specialty = data['specialty']
    procedure = data['procedure']
    frequency = data['frequency']
    response = helper_predict_holt_winter(specialty,
procedure, frequency)
    return jsonify(response)
@app.route('/main/predict_simple_exponential_smoothing',
methods=['GET', 'POST'])
def predict_simple_exponential_smoothing():
    """
    Route for predicting with Simple Exponential Smoothing
model.
    """
    data = request.json
    specialty = data['specialty']
    procedure = data['procedure']
    frequency = data['frequency']
    response =
helper_predict_simple_exponential_smoothing(specialty,
procedure, frequency)
    return jsonify(response)

@app.route('/main/predict_moving_average', methods=['GET',
'POST'])
def predict_moving_average():
    """
    Route for predicting with Moving Average model.
    """
    data = request.json
    specialty = data['specialty']
    procedure = data['procedure']
    frequency = data['frequency']
    response = helper_predict_moving_average(specialty,
procedure, frequency)
    return jsonify(response)

if __name__ == '__main__':
    app.run(debug=True)
```