Optimizing Operating Theatre Scheduling

Efficient scheduling of operating theatres (OTs) is critical for modern healthcare systems, ensuring optimal utilization of resources, minimizing patient wait times, and improving overall hospital efficiency. This white paper presents a comprehensive framework for addressing the multifaceted problem of OT scheduling. We begin with an in-depth analysis of the problem, identifying constraints such as surgeon availability, procedure durations, and emergency cases. Following this, the development and exploration of algorithmic solutions are detailed, including linear programming and heuristic methods tailored for real-world applications. The project also introduces a robust software framework that integrates these algorithms into an intuitive user interface.

Data analysis plays a central role, guiding algorithmic development and validating the model's effectiveness. Comparative performance metrics highlight significant improvements in OT utilization and patient outcomes. The paper concludes with a discussion of results, challenges faced during implementation, and prospects for scaling the solution to larger hospital networks. This work provides a holistic approach to addressing OT scheduling inefficiencies and sets the groundwork for future advancements in healthcare resource management.

Introduction

Operating theatres (OTs) are the cornerstone of surgical care in hospitals, representing a critical intersection of healthcare delivery, resource management, and patient outcomes. Their efficient operation is vital for ensuring that surgeries are performed on time, resources are fully utilized, and patients receive the best possible care. However, the scheduling of OTs remains a persistent challenge, owing to the intricate balance required between resource availability, patient needs, and the unpredictable nature of hospital operations. As one of the most expensive resources in any healthcare facility, operating theatres must be scheduled with precision to justify their high operational costs while meeting the demands of diverse stakeholders.

OT scheduling is not merely about assigning surgeries to time slots; it involves aligning the availability of surgeons, anesthetists, nurses, and specialized equipment while accounting for downstream resources such as post-operative care units. The process is further complicated by uncertainties, such as the variable duration of surgeries, last-minute cancellations, and the arrival of emergency cases that disrupt pre-planned schedules. The interplay of these factors makes OT scheduling a dynamic and multifaceted problem, requiring solutions that go beyond manual planning or static scheduling systems.

The importance of addressing these challenges cannot be overstated. Inefficient OT scheduling has significant economic, operational, and human consequences. Idle or underutilized operating theatres result in substantial financial losses for hospitals, as the fixed costs of maintaining these facilities are high. Delayed or mismanaged schedules lead to extended patient wait times, which can erode patient trust, exacerbate medical conditions, and, in some cases, have life-threatening implications. Additionally, poorly planned schedules create unnecessary stress for healthcare professionals, contributing to staff burnout and reducing overall productivity.

Currently, most hospitals rely on a combination of manual processes and basic rule-based software for OT scheduling. While these methods offer some degree of structure, they often fail to adapt to the dynamic conditions of real-world hospital environments. Manual systems are time-consuming, prone to errors, and unsuitable for large-scale operations, while rule-based approaches are often too rigid to accommodate the nuanced priorities of modern healthcare systems. Advanced solutions, such as heuristic or optimization-based models, have been explored in research settings, but their adoption in practice is limited by challenges in scalability, usability, and integration with existing hospital workflows.

To address these shortcomings, there is a growing need for a more robust, adaptive, and comprehensive approach to OT scheduling. Such a solution must not only optimize theatre utilization but also balance competing objectives, such as minimizing patient wait times, prioritizing urgent cases, and ensuring fair distribution of workloads among medical staff. Furthermore, it must be able to adapt in real time to handle unexpected disruptions, such as emergency cases or extended surgery durations, without compromising the overall schedule's integrity.

This white paper aims to present a holistic framework for tackling the OT scheduling problem, combining insights from data analysis, algorithm development, and software engineering. By leveraging advanced computational techniques and real-world data, the proposed solution seeks to improve operational efficiency, enhance patient outcomes, and reduce the stress on healthcare staff. Through the integration of these methods into a scalable and user-friendly software platform, the work outlined in this paper sets the foundation for a transformative approach to OT scheduling, bridging the gap between theoretical models and practical implementation.

Problem Statement

Efficient scheduling of operating theatres (OTs) is a critical challenge for healthcare systems worldwide, as it directly impacts operational efficiency, financial sustainability, and patient outcomes. Operating theatres are among the most resource-intensive units in a hospital, requiring meticulous coordination to align diverse constraints such as the availability of surgeons, nurses, and specialized equipment. The task becomes even more complex in the face of unpredictable factors like emergency cases, variable surgery durations, and last-minute cancellations or delays. A failure to address these complexities often leads to inefficient schedules that result in underutilized theatres, prolonged patient wait times, and increased stress on healthcare staff.

The dynamic nature of OT scheduling further exacerbates the problem. Elective surgeries must be scheduled alongside emergency cases that demand immediate attention, disrupting pre-planned schedules and creating cascading delays. Additionally, surgical procedures often take longer or shorter than initially estimated, creating scheduling gaps that are challenging to address in real-time. The unpredictability of these events makes manual or static rule-based scheduling systems inadequate for meeting the needs of modern hospitals, especially those operating under tight resource constraints.

The consequences of ineffective OT scheduling are significant. Economically, unused or idle OT capacity can lead to millions of dollars in annual losses for hospitals, as these facilities incur high fixed costs regardless of usage. Operational inefficiencies in the theatre often disrupt downstream hospital workflows, such as recovery rooms and discharge processes, further straining resources and reducing overall hospital throughput. From a patient perspective, delays in surgical procedures not only create dissatisfaction but can also lead to worsening health conditions and, in some cases, life-threatening complications. Moreover, poorly managed schedules contribute to staff burnout and reduced productivity, negatively affecting the morale of healthcare workers.

Despite its critical importance, many hospitals still rely on manual scheduling or basic rule-based systems that are inflexible and incapable of handling the complexity of real-world scenarios. While heuristic or algorithm-driven approaches have been explored in academic and commercial settings, they often lack scalability or fail to address the diverse objectives of OT scheduling. For instance, many models optimize for a single objective, such as resource utilization, while neglecting other critical factors like patient prioritization or staff workload balance. This fragmentation underscores the need for a more comprehensive, adaptive, and scalable solution that integrates these various considerations into a unified framework.

In this project, the OT scheduling problem is framed as a multi-objective optimization challenge, requiring a balance between competing goals such as maximizing resource utilization, and minimizing patient wait times. The ultimate aim is to design a system that not only adapts to the dynamic nature of real-time hospital operations but also delivers tangible improvements in both operational efficiency and patient outcomes. By leveraging advanced algorithms, data analysis, and software development, this work seeks to address the persistent gaps in current OT scheduling practices and provide a robust, deployable solution for hospitals.

Algorithm Exploration

Efficient scheduling of operating theatres (OTs) is a complex optimization problem that requires careful consideration of multiple constraints and objectives. In this project, a range of algorithms was explored to identify the most effective approach for addressing the unique challenges of OT scheduling. The algorithms examined include the greedy algorithm, weighted scheduling, and dynamic programming. These methods were evaluated for their ability to handle the unique constraints and priorities associated with OT scheduling. After extensive testing and comparative analysis, a modified weighted greedy algorithm was chosen as the final solution for its balance between computational efficiency and performance in real-world scenarios. Additionally, XGBoost, a machine learning algorithm, was integrated into the framework to address the specific issue of patient attendance uncertainty. By predicting the probability of patients showing up for their scheduled surgeries, XGBoost significantly enhanced the reliability and efficiency of the scheduling process.

Greedy Algorithm

The greedy algorithm offers a straightforward, iterative approach to scheduling. At each step, the algorithm selects the next best option based on a predefined criterion, such as the earliest end time or highest priority. While computationally efficient, this method often fails to find globally optimal solutions, especially in complex scenarios where overlapping constraints need to be balanced. For instance, prioritizing surgeries solely based on earliest completion may leave critical surgeries unscheduled or result in underutilized operating theatre capacity.

Weighted Scheduling

Weighted scheduling introduces the concept of assigning weights to each surgery based on factors like urgency, resource requirements, or potential delays. This approach is an enhancement over the greedy algorithm as it incorporates prioritization based on custom-defined metrics. While it improves upon unweighted greedy algorithms by accounting for differing levels of importance among tasks, it can still fall short in handling complex dependencies between surgeries or dynamic, real-time adjustments.

Dynamic Programming

Dynamic programming (DP) approaches are well-suited for solving combinatorial optimization problems by breaking them down into smaller subproblems and using the results of solved subproblems to build solutions to larger ones. In OT scheduling, DP can be used to explore all possible combinations of scheduling decisions, ensuring an optimal solution. However, the computational cost of DP grows exponentially with the number of surgeries and constraints, making it impractical for large-scale, real-time scheduling in hospitals.

Rationale for Choosing Weighted Greedy Algorithm

The final choice for this project was a **weighted greedy algorithm**. This approach was selected for several reasons:

1. **Balance of Efficiency and Performance:** The weighted greedy algorithm offers computational efficiency that is critical in real-world hospital environments where schedules may need to be adjusted dynamically. Unlike dynamic programming, it can handle large datasets without excessive computational overhead.
2. **Customization via Weights:** By assigning weights to surgeries based on probability of patient showing up, expected duration, and resource requirements, this algorithm accommodates diverse priorities and constraints. This customization makes it a practical choice for OT scheduling, where different surgeries may have vastly different levels of criticality.
3. **Scalability and Adaptability:** The algorithm is scalable to larger hospital systems with multiple operating theatres. It is also adaptable to incorporate new constraints or prioritize specific objectives, such as minimizing patient wait times or balancing workloads across staff.

XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a powerful and efficient machine learning algorithm designed for supervised learning tasks, such as regression and classification. It is an implementation of gradient-boosted decision trees (GBDTs) that emphasizes speed and performance. Due to its scalability and ability to handle complex datasets, XGBoost has become a go-to choice in many machine learning competitions and real world applications.

In the context of OT scheduling, XGBoost was employed to predict the probability of patients showing up for their scheduled surgeries. Patient attendance is a critical factor influencing the reliability of OT schedules. When patients fail to show up, valuable operating theater resources, including time, staff, and equipment, are wasted. Moreover, idle periods disrupt hospital workflows and negatively impact financial performance. By integrating XGBoost into the scheduling framework, the system can proactively address no-shows, enabling dynamic adjustments to schedules and improving resource utilization.

Rationale for Choosing XGBoost Algorithm

This algorithm was selected for the following reasons:

1. **High Predictive Accuracy:**
   XGBoost's robust boosting framework ensures accurate predictions, even when the data contains complex and non-linear patterns.
2. **Feature Importance Analysis:**
   XGBoost identifies the most influential features contributing to predictions, such as patient demographics, history of no-shows, and notification patterns, providing actionable insights for improving attendance.
3. **Efficiency and Scalability:**
   The algorithm is computationally efficient, allowing it to handle large datasets and make real-time predictions, which are essential for dynamic scheduling.
4. **Adaptability:**
   XGBoost supports continuous learning and can be retained with new data, ensuring that predictions remain accurate as patient behavior evolves.

By combining predictive insights from XGBoost with the modified weighted greedy algorithm, the system balances computational efficiency with data-driven adaptability, addressing both the operational and probabilistic aspects of OT scheduling.

Data Analysis

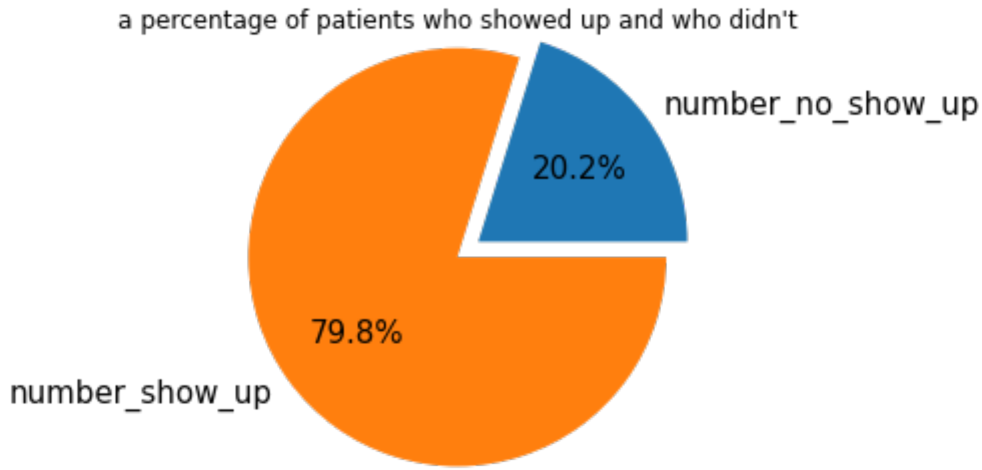
Data analysis forms the backbone of this OT scheduling project, as it provides the insights and metrics needed to design, validate, and optimize the chosen algorithm. By analyzing historical and simulated data, we gain a deeper understanding of key variables such as patient priorities, surgery durations, surgeon availability, and operating theatre utilization patterns etc,. This section outlines the data sources, preprocessing methods, visualization techniques, and statistical analyses conducted to support the development and evaluation of the scheduling system.

Data Sources

The dataset used for this project was derived from the following sources:

1. **Medical Appointment No Shows dataset:**
   This dataset has 110.527 medical appointments with 14 associated variables (characteristics). The most important one is if the patient shows-up or no-show to the appointment. These are the variables in the dataset:
   - PatientId: Identification of a patient that is unique for each person.
   - AppointmentID: Identification of each appointment.
   - Gender: Male or Female.
   - AppointmentDay: The day of the actual appointment, when they have to visit the doctor.
   - ScheduledDay: The day someone called or registered the appointment, this is before the appointment of course.
   - Age: How old is the patient.
   - Neighbourhood: Where the appointment takes place.
   - Scholarship: 1 or 0.(this is a program in Brazil to support poor people with their cost of living)
   - Hipertension: 1 or 0.
   - Diabetes: 1 or 0.
   - Alcoholism: 1 or 0.
   - Handicap: 1 or 0.
   - SMS_received: 1 or more messages sent to the patient.
   - No-show: "Yes" or "No". ("No" means they showed up on their appointments while "Yes" means they didn't!).

From these variables, we are removing some of the variables which are not relevant to our use case like AppointmentID, AppointmentDay, ScheduledDay and Neighbourhood. We use the remaining variables to predict the No-show rate. The distribution of patients who showed is displayed in the following diagram.



2. **Simulated Data:**
The public dataset was used to calculate the probability of a patient showing up to his appointment. For the next step of assigning a schedule for the operation of the patient, a synthetic dataset was generated using statistical distributions derived from historical data. This included:
    - Randomly generated operation ID unique to each patient.
    - Variations in procedure durations to simulate real-world unpredictability.
    - Availabilities of parameters like equipment, staff and special surgeon which play a crucial role in surgeries.

Depending on the number of records we extract from the public dataset, we generate the synthetic data to match the count of records.

Data Preprocessing

Data preprocessing was critical to ensure that the dataset was clean, consistent, and ready for analysis. The steps involved were:

- Removal of invalid or inconsistent records, such as age less than zero or duplicated records like scheduled days
- Converting categorical variables like gender and No-show to numerical type.
- Extracting ‘Waiting days’ from the scheduled day and appointment day columns.
- Finally filtering out only the required columns from the data and using them to build a predictive model.

Predictive Modeling

Once the data preprocessed, regression models were applied on all the features except the No-show column which is our target variable. We have tried many models like logistic regression, decision tree based models like random forests and XGBoost. Principal Component Analysis was applied on the features to select the features which explain the most variance in the dataset. XG-Boost model was finalized based on the metrics like PR-curve, accuracy, ROC curve. The trained XGBoost model predicts the probability of a patient showing up for their scheduled surgery. A threshold is used to classify patients into likely or unlikely to attend.

Scheduling

Once we obtain the probability of showing up for each patient for the public dataset, we perform various operations like filtering the records whose probability is above the median value (0.78%), dropping duplicates and finally sampling a small size of records (100) with unique probabilities and them merging this dataset with the synthetic dataset to build a final dataset which is used as input to the scheduling logic. Show-up probabilities generated by XGBoost are used as additional weights in the modified weighted greedy algorithm. This ensures that scheduling decisions prioritize patients with a high likelihood of attendance, maximizing theatre utilization. Based on this the schedules are generated and assigned to regular and emergency operating theaters.

Software Framework

The success of the OT scheduling project relies not only on the underlying algorithm but also on the design and implementation of a robust software framework that integrates the scheduling logic, provides a user-friendly interface, and offers insightful data visualizations. This section describes the technical architecture, tools, and features of the implemented system, as well as its deployment using Render, a cloud platform for hosting applications.

Overview of the Software Framework

The scheduling system was built using Python for implementing the core scheduling logic, while Streamlit was utilized for developing the interactive frontend. To enhance user engagement and decision-making, Plotly was employed for creating dynamic and interactive visualizations of schedules, resource utilization, and performance metrics. The system was deployed on Render, ensuring accessibility, scalability, and ease of use.

Key Components of the Framework

1. **Backend (Predictive model and Scheduling Logic in Python):**

- The predictive model(XG-Boost Regressor) was implemented in python.
- The scheduling logic was implemented in Python using the modified weighted greedy algorithm developed during the research phase.
- The backend handles:
  - Parsing input public data, preprocessing it, performing regression on the cleaned dataset.
  - Generating a synthetic dataset,
  - Applying the weighted greedy algorithm to generate an optimal schedule.

- Python libraries such as Pandas and NumPy were used for efficient data manipulation, while Scikit-learn was leveraged for preprocessing and prediction tasks and calculating the metrics.

2. **Frontend (Streamlit):**
   - Streamlit provided a lightweight and intuitive framework for creating a web-based user interface.
   - The interface allows users to:
     - Load the public dataset and view the metrics of the predictive model
     - View the generated schedule in tabular and visual formats.

3. **Visualization (Plotly):**
   - To improve the interpretability of the schedule and overall resource usage, Plotly was used for interactive and visually appealing charts.
   - Plotly's interactive features allow users to zoom, hover, and filter data points for deeper insights.
   - The schedules are color coded into three groups : Blue for top 10 patients with highest probability of showing up to their appointments, Red for the bottom 10 patients and Green for the rest of the patients.

Deployment Using Render

To ensure accessibility and reliability, the entire application was deployed on Render, a modern cloud platform that supports scalable and continuous deployment of web applications. Key advantages of using Render include:

1. **Ease of Deployment:**
   - The integration of Render with Github repositories streamlined the deployment process. Code changes pushed to the repository are automatically built and deployed on Render.

2. **Scalability:**
    - Render dynamically scales resources based on the user demand, ensuring the application can handle larger datasets and concurrent users without compromising performance.
3. **Cost-Effectiveness:**
    - Render offers a cost-effective hosting solution with pay-as-you-go pricing, making it suitable for both prototype development and production deployment.
4. **Custom Domain Support:**
    - The application can be accessed using a custom domain, simplifying access for hospital administrators and staff.

Workflow of the Application

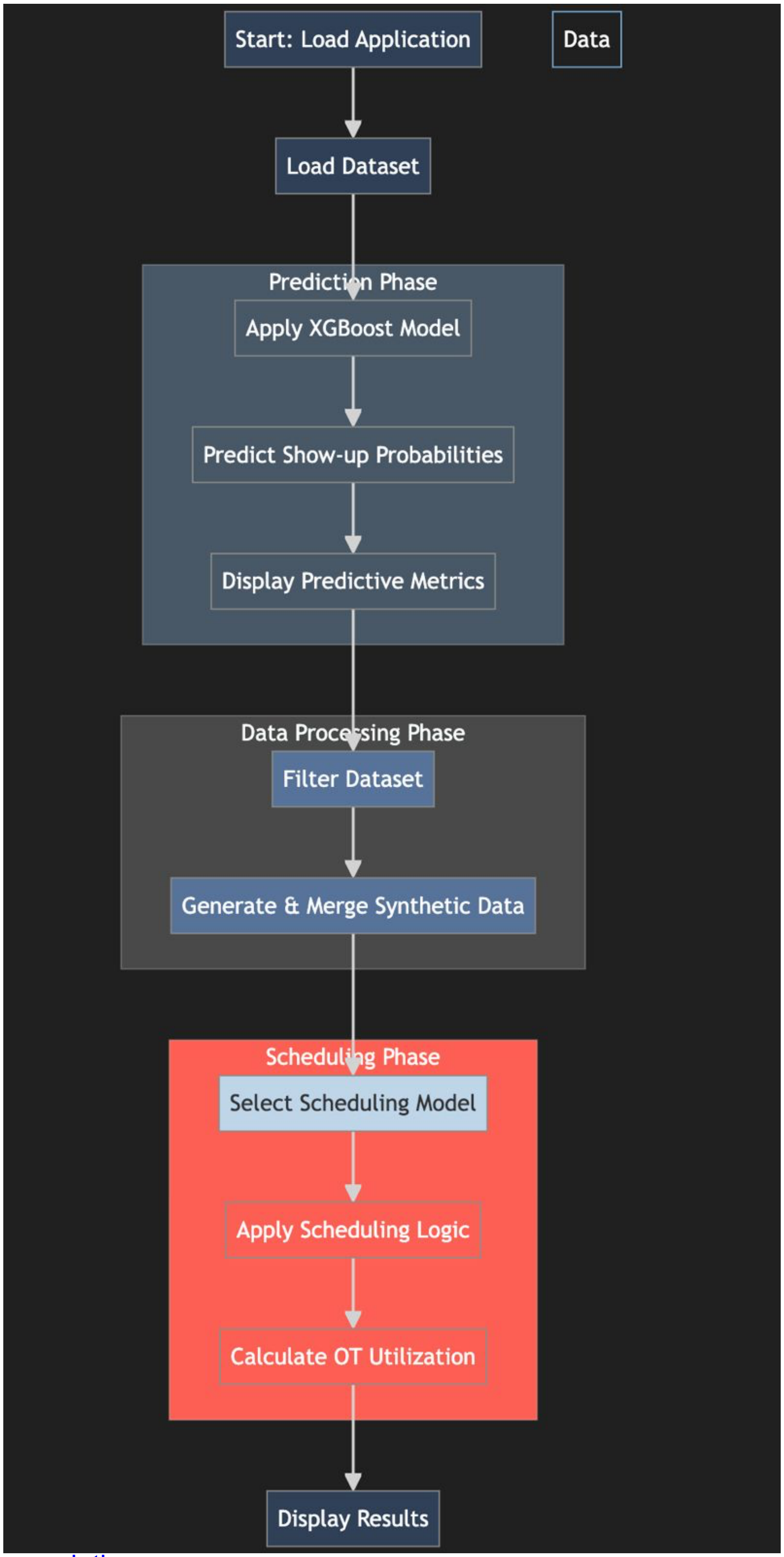
The application workflow is designed to seamlessly guide users through the process of loading data, applying the predictive model, and scheduling operations while providing key metrics and visualizations. This section details the step-by-step workflow, highlighting the interactions between data processing, predictive modeling, and scheduling logic.

Step-by-Step Workflow

1. **Loading the Dataset:**
    - When the application is loaded, the user is presented with a Load Dataset button.
    - Upon clicking the button:
        - The cleaned dataset is loaded into the system.
        - The pre-trained XGBoost model is loaded and applied to the dataset to predict the probability of patients showing up for their scheduled appointments.
2. **Predictive Model Output:**
    - The XGBoost model processes the dataset and generates show-up probabilities for each patient.
    - The predicted probabilities are used to filter the dataset, creating a dataframe with 100 samples. This filtered data frame contains high-probability cases suitable for scheduling.
3. **Synthetic Data Generation:**
    - To stimulate a more complex scheduling environment, synthetic data with information like operation durations, staff, equipment and doctor availability, is generated and combined with the filtered dataframe.
    - The merged data frame becomes the final input for the scheduling logic.

4. **Displaying Predictive Model Metrics:**
    - Key metrics of the predictive model are displayed, including:
        - **Precision-Recall(PR) Curve:** Visualizes the trade-off between precision and recall, highlighting the model's ability to balance these two metrics effectively.
        - **Receiver Operating Characteristic (ROC) Curve:** Shows the true positive rate (sensitivity) versus the false positive rate, providing an overall measure of model performance.
        - **Confusion Matrix:** Offers a detailed breakdown of the model's predictions, showing true positives, true negatives, false positives, and false negatives.
        - **Accuracy:** Highlights the model's overall performance by showing the percentage of correct predictions.
5. **Scheduling Operations:**
    - The user selects a scheduling model from a dropdown menu and clicks the Schedule Operations button.
    - The merged data frame, which includes the probability of patient attendance as weights, is passed as input to the scheduling logic.
    - The scheduling algorithm:
        - Allocates surgeries to two operating theaters:
            i. Regular OT
            ii. Emergency OT
        - Calculates the total utilization time for both theaters.
    - Scheduled operations are displayed visually using timeline plots, showing the allocation of surgeries over time.


Operation Scheduling System
Application Start
Launch Application
Load Dataset
Load XGBoost Model
Prediction Phase
XGBoost Model Application
Show-up Probability Prediction
Predictive Metrics
PR Curve
Accuracy
Feature Importance
Data Processing
Dataset Filtering
Select High-probability Samples
Create Final DataFrame
Data Merging
Generate Synthetic Data
Merge with Filtered Data
Scheduling Process
Model Selection
User Interface
Dropdown Menu
Model Options
Schedule Generation
Scheduling Logic Application
Probability-based Weighting
Results & Analytics
Timeline Visualization
Operation Schedule Plots
Utilization Metrics
Regular OT Usage
Emergency OT Usage


Start: Load Application
Data
Load Dataset
Prediction Phase
Apply XGBoost Model
Predict Show-up Probabilities
Display Predictive Metrics
Data Processing Phase
Filter Dataset
Generate & Merge Synthetic Data
Scheduling Phase
Select Scheduling Model
Apply Scheduling Logic
Calculate OT Utilization
Display Results

Results and Discussion

The OT scheduling system, powered by the modified weighted greedy algorithm and enhanced with the predictive capabilities of XGBoost, demonstrated significant improvements in scheduling efficiency and resource utilization. This section presents the key results from the implementation, including an evaluation of the predictive model and scheduling algorithm, as well as a discussion of the system's overall performance, limitations, and potential for further development.

1. **Predictive Model performance**
   The XGBoost model was evaluated using historical data on patient attendance to predict the likelihood of patients showing up for their scheduled surgeries. The model's performance metrics demonstrate its reliability and practical utility in the scheduling workflow.
   - **Accuracy:** The model achieved an accuracy 79.8%, indicating strong overall performance in predicting patient attendance.
   - **Precision-Recall (PR) Curve:** The highest F1 score achieved during the evaluation was 0.45. This metric reflects the balance between precision and recall, with higher values indicating better overall performance in distinguishing between the positive and negative classes. The optimal threshold for classifying the predicted probabilities as positive or negative was determined to be 0.22. At this threshold, the model maximizes the F1 score, ensuring a balanced trade-off between precision and recall.
   - **Receiver Operating Characteristic (ROC) Curve:** The area under the ROC curve (AUC-ROC) was 0.73, reflecting a high true positive rate relative to false positives.
   - **Confusion Matrix:** The matrix revealed the following details:
     - True Positives: Patients correctly identified as likely to attend.
     - True Negatives: Patients correctly identified as likely no-shows.
     - False Positives: Patients incorrectly predicted to attend.
     - False Negatives: Patients incorrectly predicted as no-shows.

     This breakdown emphasized the model's reliability in reducing false negatives, which is critical for effective scheduling.

2. **Scheduling System Results**
   The primary achievement of this project was the development and implementation of a scheduling system that effectively utilizes predictive insights and scheduling logic to

manage operating theatres. The scheduling system, using patient show-up probabilities as weights, was tested with both real-world and synthetic datasets. The following results were observed:

- **Efficient Scheduling of Surgeries:**
  - The scheduling logic effectively allocated surgeries between Regular OT and Emergency OT, ensuring that resources were utilized efficiently while accommodating the unpredictable nature of emergency cases.
  - By dynamically assigning surgeries based on probabilities of patient attendance, the system minimized idle theatre time and improved the flow of operations.
- **Visualization of Scheduled Surgeries:**
  - The system used timeline plots to display the allocation of surgeries across operating theatres, enabling clear visualization of the schedule.
  - The schedules are color coded: the top 10 surgeries (based on patient attendance probability) were highlighted, along with the bottom 10 to differentiate how schedules were assigned based on the probabilities.
- **Total Utilization Time Calculation:**
  - The system calculated and displayed the total utilization time for both Regular and Emergency OTs which provides a snapshot of how well the operating theatres were utilized during a given scheduling cycle.
  - The calculated utilization times allow administrators to monitor theatre usage patterns and identify opportunities to optimize operations further.

Conclusion and Future Work

Efficient operating theatre (OT) scheduling is a critical aspect of hospital management, directly impacting resource utilization, patient outcomes, and staff workload. This project successfully developed and implemented a scheduling system that dynamically allocates surgeries between Regular and Emergency OTs, incorporates predictive insights, and offers intuitive visualizations for better decision-making. By leveraging a modified weighted greedy algorithm and integrating XGBoost for predicting patient attendance probabilities, the system demonstrated the ability to handle complex scheduling challenges effectively.

One of the significant achievements of this project was the seamless integration of predictive modeling into the scheduling workflow. Using patient show-up probabilities as weights in the scheduling algorithm ensured that resources were allocated efficiently, minimizing idle theatre time and reducing the impact of no-shows. The system's ability to calculate total utilization time and visualize scheduled surgeries through timeline plots further enhanced its practical utility. Color-coded priority indicators for the top 10 and bottom 10 surgeries provided administrators with actionable insights, helping them focus on critical cases.

Future work will focus on addressing the limitations to make the system more robust and scalable. Incorporating machine learning techniques, such as retraining the XGBoost model with real-time data, can improve the accuracy of patient attendance predictions. Additionally, expanding the system's capabilities to include multi-hospital scheduling and resource sharing will make it more applicable to larger healthcare networks. Integration with electronic health record (EHR) systems will streamline data inputs and provide real-time updates, further enhancing scheduling accuracy and adaptability. Enhanced visualization tools, such as heatmaps for utilization trends or predictive analytics for workload balancing, can provide administrators with deeper insights into hospital operations.

In conclusion, this project lays a strong foundation for data-driven OT scheduling, addressing key challenges in resource allocation and dynamic scheduling. While there is room for further development, the system's current capabilities demonstrate its potential to improve hospital efficiency and patient care. By building on these achievements, future iterations of the system can offer even greater value to healthcare administrators and patients alike.

References

Kaggle. (n.d.). *Medical Appointment No Shows Dataset*. Retrieved from https://www.kaggle.com/datasets/joniarroba/noshowappointments/data

- This dataset provided insights into patient scheduling challenges, including missed appointments and delays, which informed aspects of this project.

Kleinberg, J., & Tardos, É. (2005). *Algorithm Design*. Pearson Education.

- This textbook was used as a foundational reference for understanding and implementing the algorithms, including greedy and dynamic programming approaches.

Sufahani, S. F., Mohd Razali, S. N. A., & Ismail, Z. (n.d.). *A Scheduling Problem for Hospital Operating Theatre*.

- This paper provided a detailed perspective on hospital operating theatre scheduling challenges and optimization approaches, contributing valuable insights to the theoretical foundation of this project.

Streamlit Documentation. (n.d.). *Streamlit: The fastest way to build and share data apps*. Retrieved from https://docs.streamlit.io/

- Official documentation for Streamlit, detailing the tools and techniques used to build the project's interactive frontend.

Plotly Documentation. (n.d.). *Plotly: Python graphing library*. Retrieved from https://plotly.com/python/

- Provides guidelines on creating interactive visualizations, such as Gantt charts and heatmaps, that were essential to the project.