# Requirement Specification Document

# Anomaly Detection in Operating Room Performance Metrics and Development of a Bed Occupancy Prediction Model


| | |
|---|---|
| Clinical Advisor: | Dr. Royi Barnea, Assuta Institute for Health Services Research |
| Academic Advisors: | Prof. Aviv Gibali, Applied Mathematics Department, HIT<br>Dr. Yariv Marmor, Industrial Engineering and Management, Braude College |
| Developers: | Dvora Goncharok, Digital Medical Technologies, HIT<br>Arbel Shifman, Digital Medical Technologies, HIT |


Signatures

_____ _____

Clinical Advisor Academic Advisor

# Table of Content

# Introduction

The healthcare system faces increasing demand for efficiency and patient satisfaction. One of the major challenges is optimizing the utilization of operating rooms and inpatient beds. Delays, cancellations, and underutilized resources not only affect patient experience but also hospital profitability, particularly in private institutions like Assuta Ramat HaHayal. This project aims to address operational inefficiencies by detecting anomalies in operating room performance and building predictive models to forecast bed occupancy. The insights will support real-time decision-making and smarter scheduling.

## Background

Effective management of operating room (OR) scheduling and inpatient bed occupancy is one of the most critical yet complex operational challenges faced by hospitals today. Delays, cancellations, and under-utilized ORs significantly impact hospital efficiency, patient satisfaction, and financial outcomes. At Assuta Ramat HaHayal, a private hospital with 17 operating rooms, the current scheduling approach—based on rigid, pre-allocated time blocks assigned to individual surgeons—often results in unused surgical slots due to cancellations and no-shows. These unused slots represent lost opportunities for patient care, wasted hospital resources, and potential revenue loss. Furthermore, the rigidity of this scheduling system makes it challenging to fill gaps with last-minute substitutions, as replacement surgeries require extensive logistical preparation and preoperative management.

Another critical issue contributing to bottlenecks is the limited availability of inpatient and recovery beds, which are essential for post-surgical patient care. At Assuta Ramat HaHayal, there is no provision for hallway beds, unlike in many public hospitals. When bed availability cannot meet patient demand, it leads to delays in surgical operations, as operating rooms remain occupied by patients awaiting recovery beds—causing cascading delays throughout the surgical schedule.

Historically, traditional solutions have been employed to manage operational challenges in hospitals, including fixed block allocations Goldhaber et al. (2023) and simpler regression-based methods like ARIMA Seo et al. (2024). However, these methods lack adaptability, accurate predictive capabilities, and real-time responsiveness. Static scheduling techniques and heuristic-based management do not sufficiently account for unforeseen variability in surgical durations, which is influenced, for instance, by surgical staff experience.

Previous studies, such as Marmor et al. (2011), highlighted the critical importance of accurate forecasting tools for surgical resource allocation, thereby implying the limitations of traditional static approaches. Furthermore, Seo et al. (2024) emphasize the critical role of accurate operational metrics such as utilization rate, turnover time, and predictive accuracy. They also underscore the importance of real-time prediction, particularly through the integration of predictive analytics with dynamic real-time dashboards, to provide actionable insights and enable proactive and efficient management of hospital resources.

The COVID-19 pandemic further underscored the necessity for adaptive resource management due to fluctuating surgical volumes, isolation protocols, and increased operational uncertainty. This environment demonstrated the inadequacy of traditional scheduling buffers and reactive resource management, highlighting a growing need for proactive, predictive approaches Fernández et al. (2022).

Our project addresses these identified gaps by leveraging advanced machine learning (ML) models, specifically CatBoost regression for predicting surgical durations and Bidirectional Long Short-Term Memory (BiLSTM) networks for departmental inpatient bed occupancy predictions. These methods have shown superior performance in capturing complex, non-linear, temporal relationships within hospital data (Seo et al., 2024). Additionally, we developed an optimization model (currently undergoing final validation) to dynamically maximize OR utilization under real-world constraints such as staff availability and room capacities.

Operating rooms are costly resources, yet Assuta's rigid block scheduling and limited inpatient capacity lead to frequent delays and underutilization. These constraints cause recovery room backups, cascading surgical postponements, and ultimately financial and patient satisfaction losses.

Ultimately, by combining predictive modeling, optimization strategies, and dynamic visualization tools within an integrated Power BI dashboard, this project aims to provide hospital administrators and schedulers with actionable, real-time insights for proactive operational management, supported by optimization models that were implemented and validated in this project. Specifically, a rolling-horizon scheduling framework combining greedy heuristics and Constraint Programming (CP) was developed to maximize operating room utilization while ensuring clinically feasible, conflict-free schedules. When applied to real-world data from Assuta Ramat HaHayal, the model nearly doubled mean utilization (from 28.41% to 57.05%) and significantly improved the efficient use of rooms and staff. Statistical testing confirmed the significance of the improvement ($t = 18.68$, $p < 0.00001$). These results validate the impact of incorporating optimization algorithms as a core component of the hospital's operational intelligence toolkit.

## Project goals

The primary objective of this project is to enhance operating room efficiency at Assuta Ramat HaHayal by improving utilization within a safe and effective target range of 81% to 89%. In practice, operating rooms are often underutilized due to rigid block scheduling and unexpected events such as cancellations, leading to wasted resources and revenue loss.

To address this, the project includes the following key goals:

- **Accurate Surgical Duration Predictions**: Develop a machine learning model (CatBoost) to accurately predict individual surgery durations, improving scheduling precision and reducing operational disruptions.

- **Forecasting Bed Occupancy**: Build a BiLSTM-based predictive model to forecast inpatient bed occupancy at the departmental level 5–7 days in advance, supporting better resource planning and preventing bottlenecks caused by limited recovery and inpatient bed availability.
- **Identifying Operational Anomalies**: Automatically detect anomalies in key operational metrics, including extended Length of Stay (LOS), turnover times, and mismatches between planned and actual surgery durations, offering early alerts to inefficiencies.
- **Interactive Real-Time Dashboard**: Integrate all predictive insights and analytics into a user-friendly Power BI dashboard for hospital administrators and clinical teams, enabling data-driven decision-making in real time.
- **Optimization of Surgery Scheduling:** Develop and implement an optimization model using a rolling-horizon approach that combines greedy heuristics with Constraint Programming (CP) to generate conflict-free, efficient surgical schedules. The model aims to dynamically allocate surgeries in a way that maximizes utilization while respecting clinical and logistical constraints such as staff availability, room capacity, and recovery bed limitations.

Ultimately, the project seeks to streamline surgical workflows, reduce idle time, and improve patient throughput while supporting sustainable, high-quality care delivery.

# Data

## Source data

Anonymized patient-level hospital records stored in Excel files, provided by Assuta. The data includes detailed surgical logs with exact timestamps for each phase of surgery (entry, incision, closure, and exit), recovery and hospitalization times, and pre-surgery block schedules assigned to surgeons. It also contains procedure codes, operation types, surgeon and anaesthesiologist identifiers, and in many cases, patient comorbidities and background medical conditions relevant for LOS prediction.

## Subject/Population definition

The timeframe of the study spans from 2017 through 2024. Given the evolving structure and quality of the data across different years, we performed EDA and data cleaning on each annual dataset independently. After handling missing values and outliers, we computed daily operating room utilization based on the formal utilization formula, which required aggregating all procedures performed in the same room and date, while resolving inconsistencies across surgeon blocks, time overlaps, and surgical phases.

Following the cleaning and enrichment of yearly datasets, we consolidated all years into a single integrated dataframe enriched with additional features for use in optimization and surgical duration prediction models.

In parallel, a dedicated dataframe was created specifically for training the inpatient bed occupancy prediction model. This dataset included only patients who were clearly recorded as hospitalized, with valid recovery room exit and inpatient discharge timestamps. Procedures without corresponding hospitalization data were excluded. On this dataset, we constructed a time-series formatted dataframe with engineered features, such as daily bed occupancy, historical occupancy statistics (e.g., 3-day and 7-day rolling averages), seasonal and weekly patterns, holidays, and capacity constraints.

Inclusion: completed surgical or interventional records with valid timestamps for surgical phases and recovery.

Exclusion: records with missing essential time fields that could not be imputed using statistical methods, and outpatient procedures.

Workflow – Population Selection Steps:

1. Total study population: All medical procedures recorded at Assuta Ramat HaHayal from 2017 to 2024 across all departments.
2. Definition filter: Retained only surgical or interventional medical procedures relevant for operating room and recovery analysis.
3. Inclusion filter applied: Kept records that included valid timestamps for surgical stages and recovery.

4. Exclusion filter applied: Removed outpatient procedures and records with critical missing time fields that could not be statistically imputed.
5. Final datasets created:

- Dataset A – This dataset includes 117,698 inpatient surgery records extracted from hospital logs between 2017 and 2024. It was primarily used for calculating daily operating room (OR) utilization and for training predictive models for surgery duration. Significant preprocessing was required to handle missing values, standardize time fields (e.g., surgery start/end times), and reconcile duplicate entries. In addition to raw fields (e.g., patient demographics, surgery type, room number, team members), we engineered key features such as: Number of surgeries per room per day, Block vs. planned ratio, Team-to-duration ratios, Surgeon-level and anesthesia-level historical duration trends
- This dataset also served as the core input for the optimization model, enabling the development of conflict-free surgical schedules that maximize utilization across room-day combinations.
- Dataset B: subset of hospitalized cases with valid admission/discharge data for bed occupancy forecasting, with additional engineered features (rolling occupancy stats, weekdays, holidays, seasons).

## Outcome variable definition

1. Daily Operating Room Utilization Rate: Defined as the ratio between the total duration of all surgical procedures (including anesthesia preparation, incision, closure, and patient setup) performed in a specific operating room on a given day, and the total available operating room time (i.e., scheduled block hours). This variable is computed per room, per day, and is used to evaluate operational efficiency and identify underutilization or overbooking trends. Utilization values are expressed as percentages, with 81–89% considered optimal.
2. Inpatient Bed Occupancy Forecast (5–7 Days Ahead): A predicted continuous variable representing the number of inpatient beds expected to be occupied on each future date (within the next 5–7 days). It is derived using time-series and machine learning models based on historical occupancy levels, scheduled surgeries, day-of-week effects, holidays, seasonal patterns, and recent admission/discharge trends. In addition, the model incorporates rolling occupancy statistics such as 3-day and 7-day moving averages to enhance temporal context and forecasting accuracy.
3. Anomalies in Length of Stay (LOS) and Turnover Time: Binary classification labels indicating whether the LOS or turnover time for a specific patient or surgery deviates significantly from expected values, based on statistical modeling. Anomalies are identified using unsupervised outlier detection models (e.g., Isolation Forest, LOF) and flag situations such as unexpectedly prolonged hospitalization or excessive delays between surgeries.
4. Optimized Surgical Scheduling Score (Post-Optimization Utilization):

This outcome variable reflects the core objective of the optimization model: increasing operating room (OR) utilization by generating efficient, conflict-free surgical schedules. It measures the improvement in daily OR utilization (expressed as a percentage difference between historical and optimized schedules), ensures that no overlapping assignments occur across rooms, surgeons, anesthesiologists, or teams (conflict-free indicator), and tracks the number of ORs used each day to assess resource efficiency. By capturing these elements, this variable directly quantifies the effectiveness of the scheduling algorithm in enhancing OR utilization under real clinical and operational constraints.

## Confounding variables

Several variables may act as confounders, influencing both the predictors and the outcomes of our models:

- Patient age and comorbidities: Older patients or those with chronic illnesses may have longer surgical durations and extended hospital stays, potentially biasing the prediction of LOS and bed occupancy.
- Type and complexity of surgery: Complex procedures inherently take longer and may require longer recovery, skewing operating room utilization and bed forecasts.
- Department-specific workflows: Differences in protocols (e.g., gynecology vs. orthopedics) may lead to systematic variations in surgical pace, staff availability, and discharge timing.
- Surgical team composition: Experience level and coordination among surgeons, anesthesiologists, and nurses may impact turnover time and efficiency.
- Scheduling constraints and block allocations: Variability in daily schedules or emergency cases can introduce irregularities that affect utilization calculations.

To mitigate the influence of these confounders, our models incorporate department and procedure-level controls, and stratified validation is used to ensure robustness across different clinical contexts.

In the optimization model, several of these variables—such as predicted surgery duration, staff availability, and department-specific patterns—are explicitly accounted for as constraints or inputs, ensuring that the generated schedules remain realistic and clinically feasible.

## Source of Bias

- Manual data entry errors: According to discussions with the clinical supervisor, many values were entered manually by staff members. This introduces a high risk of human error, such as typos, missing timestamps, or inconsistent formatting, which may affect model accuracy and reliability.

- Underreporting of delays or clinical events: Events such as infections or recovery delays may not be systematically recorded, leading to biased LOS distributions.
- Missing values in timestamp fields: Incomplete data can reduce sample size or require imputation, introducing potential bias.
- Seasonal or policy changes: External events such as the COVID-19 pandemic or internal hospital policy shifts (e.g., staff restructuring, updated discharge policies) may have impacted procedures and patient flow patterns, confounding temporal analysis. In addition, the 2023 Israel–Hamas war led to increased staff absences due to military reserve duty, further affecting scheduling regularity and operational continuity.

## IRB/Regulatory permissions

Data access is subject to approval by the Assuta hospital and the research room team. All data is anonymized, and no identifiable patient information is present. Permissions were granted under the oversight of Assuta's internal IRB committee and clinical research governance.

## Required features

- Timestamps for surgery stages (entry, incision, closure, exit)
- Recovery and discharge dates
- Department and room codes
- Patient demographics and clinical risk indicators
- Surgery type and duration

# Exploratory Data Analysis (EDA)

EDA was conducted independently for each annual dataset using Python 3.10. The primary libraries used include pandas, numpy, seaborn, matplotlib, scipy, and ydata-profiling. The process included validation of date and time formats, consolidation of patient records into single rows, conversion of time features to numeric values, and profiling of dataset completeness.

## Outliers

Outliers were discovered through both logical checks and statistical techniques. Specific attention was given to detecting temporal inconsistencies in the surgical process, such as cases where the recorded surgery exit time occurred before the closure time. These were flagged and analyzed separately. Additionally, z-score thresholds and interquartile range (IQR) methods were considered for identifying extreme numerical values in continuous variables (e.g., LOS, turnover duration, surgical duration). Outliers were not removed automatically, but either flagged for downstream modeling or transformed into categorical anomaly indicators when clinically meaningful. the identification of inconsistencies in surgery timing (e.g., cases where surgery exit time preceded closure). These cases were flagged for review. No removal was done at this stage to preserve data integrity during cleaning.

## Missing values

To identify the missingness mechanism, we first calculated the proportion of missing values per column and categorized them based on severity.

- Columns with over 70% missingness were considered unreliable and removed from the dataset. All steps were validated through visual inspection (e.g., missingness heatmaps) and a profiling report was generated to ensure consistency across datasets.
- For features with 41–69% missingness, we performed a categorical transformation by introducing an explicit “missing” level.
- For features with <40% missing values, we first determined the missingness mechanism:
    - Numerical features and time variables: compared the distribution of observed vs. missing cases using a two-sample Kolmogorov–Smirnov test.
    - Categorical features: tested dependence between the missingness indicator (0/1) and each category via a Chi-square test.
    - We conducted statistical tests to distinguish between Missing Completely At Random (MCAR) and Missing At Random (MAR). Features were classified as

MCAR when p-value > 0.05 (no evidence of difference between missing and observed) and as MAR when p-value ≤ 0.05 (missingness depends on other covariates).

- For features identified as MCAR, we applied KNN imputation (k=3) based on Euclidean distance in the remaining variables.
- For features identified as MAR, we used Multivariate Imputation by Chained Equations (MICE) with Predictive Mean Matching over 5 iterations to leverage inter-variable relationships for more accurate completion.

## Feature Enrichment

New features were engineered primarily from the surgery timeline and hospitalization data. Time-based variables (e.g., incision time, closure time) were transformed into numerical representations (minutes since midnight) to enable further processing. From these, additional engineered features were derived, including:

- Time-of-day categories (morning, noon, evening, night) based on converted time values.
- Time gaps between phases of surgery (e.g., incision to closure, closure to exit), used to calculate turnover and operating time.
- Flags for temporal inconsistencies or anomalies (e.g., closure time after exit).

For bed occupancy prediction, we engineered time-series features per department:

- Daily occupancy and maximum occupancy capacity
- Rolling averages of occupancy over the previous 3 and 7 days
- Calendar-based features including day of the week, season, and holiday indicators

Numerical variables were scaled when required for model input (e.g., neural networks), and categorical transformations were applied to time-related intervals and selected numeric bins. were converted to minutes-since-midnight and used to derive new variables such as:

- Time-of-day category (morning, noon, evening, night)
- Duration between surgical stages
- Turnover time between surgeries
- Conflict indicators for temporal logic violations

In addition to timeline-derived variables, we engineered a set of clinical and operational features to improve the accuracy of the surgical duration prediction model (CatBoost). These included:

- Calendar features such as day of month, month, and quarter to capture seasonal trends.

- A binary indicator for morning surgeries (before 12:00), based on observed differences in scheduling patterns.
- Surgical workload indicators, including the number of surgeries per room per day and per surgeon per day.
- Block efficiency metrics such as the “plan vs. block” ratio, reflecting how tightly the planned duration fits into the surgeon’s allocated block.
- Patient flow features, including days from admission to surgery and length of preoperative hospitalization.
- Team-related variables such as number of staff per case and ratio of team size to planned duration.
- Comparative ratios, such as each surgery’s planned duration relative to the surgeon’s historical average or standard deviation, to detect anomalies.

For the surgical scheduling optimization model, we created additional engineered inputs and mappings:

- Predicted surgery duration per case (from the CatBoost model)
- Surgeon–room and surgeon–anesthesiologist preference dictionaries based on historical patterns
- Daily availability dictionaries for surgeons based on past usage intervals and merged into conflict-free time blocks per weekday
- Operating room availability profiles per activity type and day of week

These enriched features ensured that both prediction and optimization components were grounded in realistic, data-driven representations of hospital workflows.

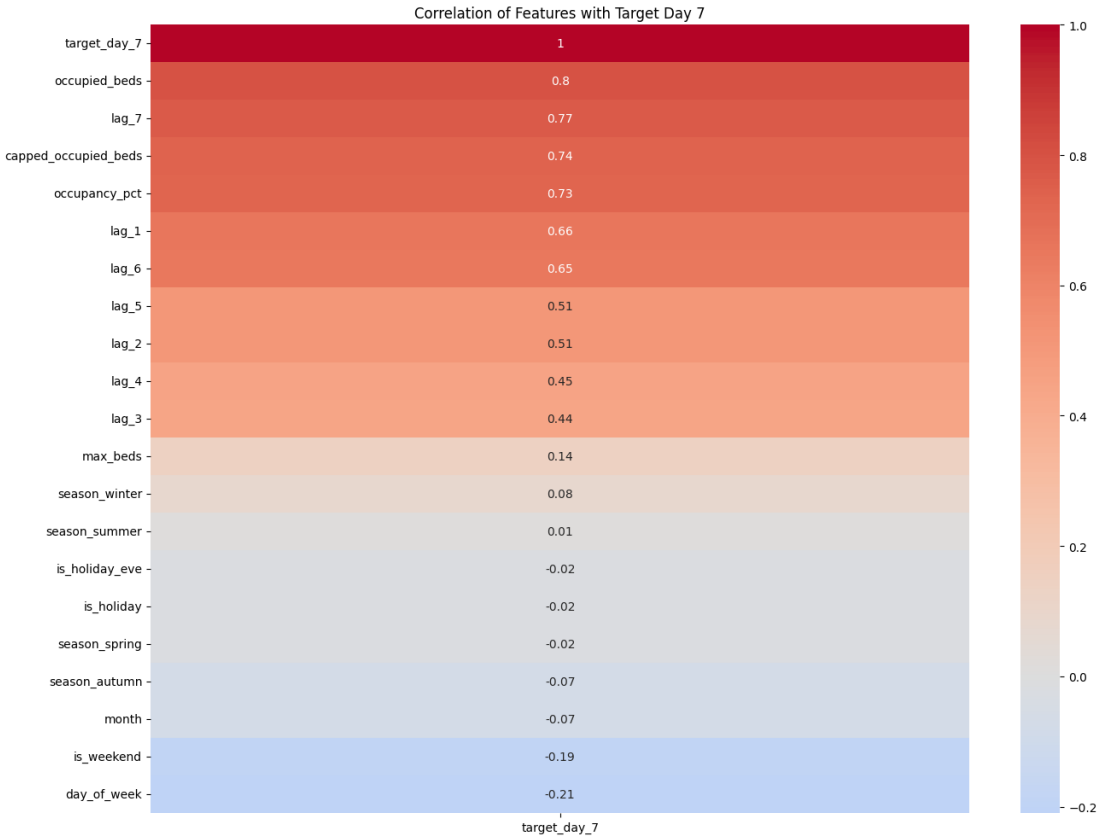
## Feature Selection

Feature selection was approached through a combination of domain knowledge, missingness evaluation, and modeling feedback. During EDA, variables were grouped into time_vars, num_vars, and cat_vars based on data type and clinical relevance. Variables with high missingness or unreliable behavior were dropped or transformed. In preparation for the bed occupancy prediction model, we constructed a dedicated dataframe containing only patients with a confirmed inpatient discharge timestamp. That is, procedures without recorded hospitalization data (i.e., missing recovery room exit or inpatient discharge times) were excluded from this model. From this refined inpatient dataset, we generated a new time-series dataframe incorporating features such as daily bed occupancy, maximum capacity, rolling occupancy averages (3-day, 7-day), and calendar features including day of

week, season, and holiday flags. This allowed us to focus the model on reliable and context-rich data.

To support variable selection, we used correlation analysis and model-based importance evaluation. For example, in the bed occupancy forecasting model using a Bi-LSTM architecture, we visualized the Pearson correlation between candidate features and the target variable (target_day_7). As shown in the correlation heatmap (see figure), features such as occupied_beds, lag_7, and capped_occupied_beds showed high positive correlation (>0.7) with the target, and were therefore prioritized. On the other hand, calendar-related features like day_of_week and is_weekend had weaker or even negative correlations and were included with caution depending on model performance.

This type of visualization helped guide feature selection decisions prior to model training and interpret model behavior. In later stages, advanced techniques such as SHAP values and Recursive Feature Elimination (RFE) may be applied to enhance interpretability and optimize the feature set.



For the surgery duration prediction model, we applied feature selection using a model-based approach with a Random Forest regressor. After filtering out non-numeric variables and rows with missing values, we trained a preliminary RandomForest model to assess the relative importance of each numerical feature.

Only features with an importance score greater than 0.01 were retained for modeling. This threshold-based pruning reduced dimensionality, improved model efficiency, and eliminated redundant or low-signal variables.

The selected features were then used to train and compare five different regression models (RandomForest, XGBoost, LightGBM, CatBoost, ElasticNet) using 5-fold cross-validation. This process ensured that only relevant, interpretable, and stable variables contributed to the final prediction pipeline.

| Feature Name | Importance |
|---|---|
| Activity Code | 0.39 |
| Avg. Planned Duration/Surgeon | 0.03 |
| Surgical Team Size | 0.03 |
| Activity Type Code | 0.03 |
| Surgeon Daily Count | 0.02 |
| Std. Duration per Surgeon | 0.02 |
| Team Size to Duration Ratio | 0.01 |

## Feature Imbalance

In this project, most modeling tasks involved continuous outcome variables (e.g., utilization rates, predicted bed occupancy), which are not subject to class imbalance. No binary classification was performed.

For exploratory analysis of anomalies in Length of Stay (LOS), we observed a small number of extreme cases, including rare outliers such as one patient who was hospitalized for 80 days. These cases were retained in the dataset and flagged as potential clinical anomalies based on logical thresholds (e.g., hospitalization longer than 30 days). No supervised outlier detection models were used.

# Predictive Models

## Dataset partition

To ensure robust evaluation and avoid data leakage, the dataset was partitioned chronologically, simulating a real-world forecasting scenario. Specifically, we used a training period of 80% of the earliest available days, and reserved the remaining 20% as the test set, representing unseen future dates. This approach prevents information from the future leaking into the model during training, which is crucial in time-series forecasting. Beyond time-series forecasting, additional modeling strategies were implemented for different prediction tasks, each requiring a tailored data partitioning approach.

To prevent the introduction of bias, the partitioning was stratified by department, ensuring each department's patterns are preserved across both training and test sets. In cases where data volume per department was limited, we applied department-specific train-test splits to maintain distributional consistency.

Reproducibility was guaranteed by setting a fixed random seed (random_state=42) for all operations involving randomness, including shuffling or train-validation splits (if applied within the training data). In addition, we logged partition metadata such as date ranges and row indices used in each set to ensure full traceability.

The dataset for predicting surgical duration was split randomly into 80% training and 20% test sets using train_test_split with a fixed random_state=42 to ensure reproducibility. Prior to the split, the data was filtered to retain only rows with valid numeric features and a positive target value. Feature selection was performed using a Random Forest regressor, and the selected features were used for training and evaluation of five regression models. The log-transformed target variable was used to stabilize variance and improve predictive performance. This setup allowed for robust model comparison across folds, while the test set served as a final hold-out evaluation.

For the surgery scheduling optimization model, we evaluated performance in two distinct workload conditions:

- High-load scenario: The algorithm was executed on the *busiest week* in the dataset to assess its ability to schedule surgeries under pressure — ensuring resource constraints (e.g., no overlaps for surgeons or rooms) were satisfied while maintaining high utilization.
- Low-load scenario: The model was also applied to the *least busy month*, using a rolling horizon approach. This allowed for evaluation in a more flexible, real-world scheduling context where availability is higher.

These complementary evaluations were designed to test the robustness and adaptability of the system under both extreme and routine operating conditions.

## Metric Selection

To evaluate the predictive performance of our models, we selected three key metrics:

- Mean Absolute Error (MAE): Provides an intuitive interpretation of error magnitude by measuring the average absolute difference between predicted and actual values. It is robust to outliers and is particularly useful when all errors are equally important.
- Root Mean Squared Error (RMSE): Similar to MAE but places greater emphasis on larger errors due to the squaring of residuals. This makes it valuable for highlighting models that fail to capture extreme deviations, which is critical in healthcare operations where large forecasting errors (e.g., in bed occupancy) may have serious consequences.
- R-squared ($R^2$): Expresses how much of the variance in the target variable is explained by the model. It is useful for comparing models across departments and identifying which models generalize well.

By combining these metrics, we gain a balanced view of model accuracy, sensitivity to large errors, and explanatory power—ensuring that selected models are both accurate and operationally reliable.

## Model Selection

This project focuses on supervised learning, using regression models to predict future inpatient bed occupancy levels across multiple hospital departments.

**Bed Occupancy Forecasting (Time-Series Regression)**
We employed a diverse set of models from both traditional machine learning and deep learning categories, in order to benchmark and compare predictive performance across departments:

Machine Learning Models:

- Linear Regression (Baseline)
- Random Forest Regressor
- Gradient Boosting Regressor
- XGBoost Regressor
- K-Nearest Neighbors Regressor (KNN)
- Support Vector Regression (SVR)
- SARIMAX (Statistical Time Series Model)

Deep Learning Models:

- Long Short-Term Memory (LSTM)

- Bidirectional LSTM (Bi-LSTM)
- Gated Recurrent Units (GRU)

Training and Validation Strategy:
The dataset is partitioned chronologically into training and testing sets to preserve temporal dependencies and prevent data leakage. Typically, 80% of the data is used for training and 20% for testing, ensuring that the model only sees past data during training and is evaluated on future unseen data.

To ensure reproducibility, we fixed the random seed (np.random.seed(42)) across all data splits and model training processes.

Model Execution:
Each model is trained per department using relevant features such as lag variables, holidays, day-of-week indicators, and rolling averages. Deep learning models were trained on sequences of 14 previous days to predict 7 future days. Model outputs are stored and compared consistently.

Due to the time-series nature of the task, cross-validation was not used in the traditional k-fold sense. Instead, we relied on walk-forward validation where appropriate, particularly for evaluating models like SARIMAX. Bootstrapping was not applied, as it may violate temporal order and dependencies in time-series data.

Model Selection Criteria:
Models are evaluated using a combination of MAE, RMSE, and $R^2$. The best model per department is selected based on:

- Lowest average MAE and RMSE across 7-day forecasts
- Highest average $R^2$ score
- Stability and generalization across different time windows

This multi-metric evaluation ensures a robust and reliable selection that reflects both accuracy and consistency over time.

**Surgical Duration Prediction**

To accurately estimate the duration of each surgery (critical for scheduling), we developed a dedicated machine learning pipeline:

Models used:

- Random Forest Regressor
- XGBoost
- LightGBM
- CatBoost
- ElasticNet

Feature selection was performed using a Random Forest importance threshold (>0.01), and the target variable was log-transformed for stability. Models were evaluated using 5-fold

cross-validation with MAE and $R^2$, and the best-performing model (typically CatBoost or XGBoost) was used to generate predicted durations for use in the scheduling optimizer.

**Surgery Scheduling Optimization**

To improve operating room efficiency and ensure conflict-free allocation of critical resources (surgeons, anesthesiologists, operating rooms), we developed a two-stage optimization model combining:

- Greedy Scheduling Algorithm: A fast heuristic used in the first stage to schedule as many surgeries as possible based on predicted durations and resource availability. This step prioritizes feasibility and speed, while minimizing overlaps.
- Constraint Programming (CP): In the second stage, leftover unscheduled cases are optimized using Google OR-Tools CP-SAT solver. This model enforces strict constraints such as:
    - No overlaps between surgeries assigned to the same room, surgeon, or anesthesiologist
    - Time buffers between surgeries
    - Preference scores for surgeon-room and surgeon-anesthesiologist combinations
    - Operating room and personnel availability per weekday

Evaluation Strategy:

The optimization model was tested under two real-world workload conditions:

- High-load scenario**:** Applied to the busiest week in the dataset to evaluate performance under scheduling pressure.
- Low-load scenario: Executed using a rolling horizon approach over the least busy month to simulate flexible daily updates with greater resource availability.

The scheduling optimization model was evaluated based on:

- Improvement in Operating Room Utilization: We compared the predicted utilization resulting from the model's scheduling output to historical utilization rates for the same rooms and weekdays. This comparison measured whether the proposed scheduling plan achieves better resource efficiency than actual past usage.
- Feasibility and Constraint Satisfaction: The model ensured that no overlaps occurred between surgeries in the same room, or for the same surgeon or anesthesiologist, and that required buffers between surgeries were maintained.

This evaluation framework allowed us to determine whether the model not only produces feasible schedules but also leads to measurable improvements in OR efficiency compared to real-world historical data.

## Model Fine Tuning

Bed Occupancy Forecasting

Each selected model underwent a fine-tuning process to optimize performance for the specific task of predicting hospital bed occupancy. The tuning strategy varied by model type:

For machine learning models such as Random Forest, Gradient Boosting, XGBoost, and SVR, we applied both grid search and randomized search over defined hyperparameter spaces. The key parameters tuned included the number of trees (n_estimators), tree depth (max_depth), learning rate (for boosting models), kernel type and regularization (C) for SVR, and distance metric and neighbor count (n_neighbors) for KNN. Performance during tuning was assessed using cross-validation on the training set, with final confirmation based on MAE and RMSE on the test set.

For time series models like SARIMAX, we performed grid search over seasonal and non-seasonal parameters (p, d, q) and (P, D, Q, s). Model selection was guided by the Akaike Information Criterion (AIC), followed by out-of-sample validation.

For deep learning models such as LSTM, Bi-LSTM, and GRU, we combined manual tuning (based on validation loss trends) with automated search using Keras Tuner. Hyperparameters tuned included number of layers, units per layer, learning rate, optimizer (e.g., Adam, RMSProp), dropout rate, batch size, and number of epochs. We used early stopping to prevent overfitting, and the best-performing weights were restored based on the lowest validation loss.

**Surgical Duration Prediction**

To improve the precision of predicted surgery durations—used downstream in scheduling optimization—we built a dedicated regression pipeline. Feature selection was first applied using a Random Forest importance threshold (importance > 0.01), and the target variable was log-transformed to stabilize variance.

We evaluated five models: Random Forest, XGBoost, LightGBM, CatBoost, and ElasticNet. Fine-tuning was conducted via 5-fold cross-validation using a performance grid based on MAE and $R^2$. Hyperparameters such as learning rate, depth, and regularization strength were adjusted to maximize generalization. The best model (often CatBoost or XGBoost) was retained for generating duration predictions used in the optimization stage.

**Optimization Scheduling Model**

To enhance operating room efficiency and ensure feasible resource allocation, a two-stage optimization model was developed combining:

1. Greedy Scheduling Algorithm
2. Constraint Programming (CP) using Google OR-Tools

Fine-tuning of this hybrid model included:

- In the Greedy stage:
    - Tuning the utilization threshold (e.g., util_thresh = 0.5) to control how aggressively rooms are filled.

- Adjusting surgery buffer times (e.g., 10 minutes) to prevent scheduling conflicts.
- Prioritizing longer surgeries first to improve packing efficiency under time constraints.

- In the CP stage:
  - Enforcing strict non-overlap constraints for surgeons, anesthesiologists, and rooms using AddNoOverlap.
  - Tuning penalty terms such as penalty_open_room to balance between maximizing scheduled time and minimizing the number of open rooms.
  - Iteratively adjusting solver time limits and availability windows for scalability.

The effectiveness of the optimization was assessed by comparing predicted OR utilization against historical utilization per room and day, while validating feasibility (e.g., no overlapping assignments, valid availability windows). This ensured that the schedules produced by the model were both efficient and realistic.

## External Validation

For the bed occupancy forecasting model, we performed external validation using data from a distinct temporal holdout set that was excluded entirely from model training and tuning. Specifically, the last three months of available data were set aside as a separate test window. This period was not used in any phase of training, feature selection, or hyperparameter tuning. By holding out this future slice of data, we were able to simulate real-world deployment conditions and evaluate how well the models generalize to new, unseen operational environments.
The models were assessed on this holdout set using the same metrics employed during internal evaluation (MAE, RMSE, $R^2$), providing a consistent basis for comparison. This validation strategy helps confirm that the models maintain performance stability beyond the training period and are suitable for implementation in practice.

For the surgical duration prediction model, external validation was conducted using a hold-out test set (20% of the data), fully excluded from training and feature selection. The model's predictions were evaluated on this unseen set using $R^2$ and MAE to ensure its accuracy generalizes to new surgical cases.

For the optimization scheduling model, external evaluation involved applying the final scheduling system to two real-world workload scenarios that were not used during development: the busiest week and the least busy month in the dataset. These scenarios allowed us to assess whether the optimization logic generalizes to both high- and low-load conditions and whether the predicted OR utilization improves upon historical benchmarks.

## Deployment of the model

The final predictive model will be implemented as part of the hospital's decision support system for surgical and bed occupancy management. Deployment will be carried out in several modular and scalable steps:

Data Ingestion and Pre-processing:
The model will be connected to the hospital's data pipeline to receive daily updates of operational and scheduling data (e.g., surgery exit times, recovery room timestamps).

Pre-processing will mirror the training pipeline and include:

- Conversion of time-based fields to derived features (e.g., day_of_week, is_weekend, season)
- Creation of lag variables based on prior 7 days' occupancy
- Handling of categorical fields via one-hot encoding (e.g., seasonality, holidays)
- Scaling of numeric inputs where applicable

Handling Missing or Incomplete Data
In real-time scenarios, the model may receive incomplete inputs (e.g., missing prior occupancy data due to system downtime).

In such cases:

- A fallback strategy will be used: imputing missing lags with rolling means or default department-level averages
- If input coverage falls below a reliability threshold (e.g., more than 3 lag values missing), the system will flag the output as low-confidence and notify the user accordingly
- Alternatively, the system will return a prediction marked as null or unstable, prompting manual review

Prediction Frequency and Update Policy:
Daily predictions will be generated for each surgical department, forecasting occupancy up to 7 days ahead. The model will be retrained and validated on a monthly basis using the most recent data, with version control to track performance changes. Automated retraining pipelines will include performance benchmarking and drift detection checks.

Platform and Interface:
The model will be deployed on the hospital's secure server infrastructure or via a cloud-based container (e.g., Dockerized API using FastAPI or Flask). Predictions will be saved in a dedicated database table and integrated into the hospital's internal dashboard system (e.g., Power BI, Looker Studio). Visualizations will include occupancy trends, predicted vs actual comparisons, and flags for high-risk overload days.

User Interaction and Training:
End users (e.g., hospital operations managers, surgical schedulers) will receive training sessions including:

- Interpreting predictions and confidence levels
- Understanding model limitations and edge cases
- Responding to warning signals or overload forecasts

Training materials (e.g., video tutorials, quick guides) will be provided alongside the dashboard interface.

Monitoring Model Decay:
Model decay will be assessed continuously through:

- Rolling evaluation of prediction error (e.g., MAE/ RMSE on actual occupancy)
- Alerts on significant drops in $R^2$ or rise in residuals
- Periodic correlation analysis between key features and targets to detect feature drift

If performance deteriorates beyond acceptable thresholds, automatic triggers will retrain the model or escalate for manual intervention.

Optimization Model Integration and Workflow

In addition to occupancy forecasting, the system includes a scheduling optimization component that generates full weekly surgery schedules. The optimization model takes as input the list of planned surgeries, room availability, and the availability of surgeons, anesthesiologists, and nursing teams. Preference data (e.g., surgeon-room or surgeon-anesthesiologist preferences) is also incorporated to improve practical feasibility.

Importantly, surgery durations are not provided manually but are instead predicted automatically using a dedicated machine learning model trained on historical data. This ensures consistency and realism in the scheduling process.

The optimization model is designed to operate on availability dictionaries, which are generated from structured staff shift data (e.g., pandas DataFrames of scheduled shifts). While current availability inputs are based on historical data and preferences, the system is built to support periodic updates. Users can upload new availability files—weekly or monthly—and re-run the scheduling process without modifying the codebase. This allows seamless adaptation to frequent staff changes and evolving constraints.

The system's input includes:

- Scheduled surgeries (with department and clinical metadata)
- Predicted durations (from the duration prediction model)
- Staff and OR availability
- Allocation preferences

The output is a structured schedule (DataFrame or .xlsx file) containing all assigned surgeries with exact start and end times, assigned room, and staff. This file can be used operationally by coordinators or connected to a BI tool for visual analysis. Although the current dashboard primarily visualizes OR utilization comparisons (model vs. historical), future iterations can integrate full schedule displays by room, staff, and day.

While real-time deployment is not yet implemented, the architecture supports modular execution based on updated inputs. This enables flexible and recurring use by hospital staff (e.g., weekly planning rounds). As part of a broader decision-support system, the scheduling engine can be integrated with front-end interfaces, staff portals, or alerting systems to support dynamic clinical operations.

Together, these components create an integrated, adaptive framework to support real-time and long-term hospital resource planning.