# ETC5543 Business Analytics Creative Activity

Surgery Operating Length Prediction: A Patients Finder Tool

Phuong Anh Mai

2024-10-19

## Abstract

This report presents a predictive modeling project undertaken during an internship with **Monash Health** which also known as Victoria's largest public health service. This whole project initially aims to enhance operational efficiency which aligns with Monash Health's mission to provide high-quality, patient-centered healthcare across diverse communities. (1) In this project, we will explore and predict a target which is surgery duration. The target variable is calculated based on patient demographics, procedure type, and other relevant factors obtained from Monash Health database. Various machine learning models were applied in this project including Random Forest, Ridge, Lasso, and Gradient Boosting Machine (GBM) to estimate surgery lengths. In the end of this project, I have archived Random Forest as the best model, and created a demo version of a patient searching tool which potentially helps Monash Health medical staff prioritize patients from the waiting list to fill available theatre slots.

## Background and Motivation

The allocation of operating theatre resources is one of the most crucial logistical challenges faced by hospitals. The operating room or we can call them as surgery theatre can be accounted for up to 40% of the resource cost in hospitals (2), thus, successfully manage this is a milestone for every health centre. In high-demand healthcare settings like Monash Health, Victoria's largest public health provider, managing this scheduling complexity is key to both patient care and operational efficiency. Monash Health services a vast network, including Monash Medical Centre, Casey Hospital, and Dandenong Hospital, providing healthcare for all stages of life from newborns to elderly patients. Efficient theatre utilization is necessary not only to maximize the number of patients treated but also to manage costs, optimize staff resources, and reduce patient wait times for essential and sometimes life-saving procedures.

Based on the report from Agency for clinical innovation in New South Wales, the number of emergency case that public hospital get annually is around 45% (3), which raised the problem for every hospital to efficiently manage their schedule for operating room. Typically, hospitals prioritize emergency and high-risk cases, often leaving shorter time slots unused or underutilized due to the difficulty of scheduling additional cases in the available time. These unfilled gaps in theatre scheduling are a persistent inefficiency that translates into increased operational costs and delayed treatment for other patients. This issue is magnified in large health systems like Monash Health, where theatre slots are often booked weeks to months in advance.

To address these scheduling inefficiencies, this predictive modeling project is embedded in the Monash Health Surgery Division's larger initiative to maximize theatre efficiency by adding 10-20% more cases to planned schedules. If implemented successfully, the model ideally could enable an additional 1,300 to 2,500 cases per year based on Monash Health estimation. By predicting surgery durations based on available hospital's waiting patient data, combine it with thir surgery data, the model can assist hospital staffs in searching for appropriately matched patients, thus minimizing downtime and improving patient care outcomes.

## Objectives and Significance

The key objectives of this predictive modeling project are as follows:

1. **To accurately predict surgery lengths based on patient and surgery characteristics**: By understanding how different factors such as age, specialty, and priority code influence surgery duration, we can get more effectively match patients to available theatre time slots.

2. **To optimize the utilization of surgical theatres**: The model we fitted will help to develop a front-end tool which directly aids in scheduling adjustments by recommending patients from the waiting list who align with the available theatre time slots, reducing idle time and potentially increasing the annual surgical volume.

The significance of this project extends beyond mere predictive modeling. By improving scheduling accuracy, this initiative has the potential to streamline hospital operations, reduce patient wait times, and maximize the available resources. Additionally, the model supports the decision-making process for healthcare providers, allowing them to prioritize and schedule patients based on data-driven insights. This approach is both innovative and impactful, as it blends data science with practical healthcare applications to address a pressing operational challenge.

## Methodology, Data and Result discussion

### 3.1 Data Collection and Preprocessing

As our purpose is developing a model that helps predict the operating time, we'll rely on Monash Health's comprehensive operational data, specifically focus on waiting list and surgery records over the past four years (2020-2024). The primary reason behind this time period is because we want to have the most recent dataset pattern, meanwhile, we also want to avoid bias caused by external factor such as Covid-19, which definitely affected the waiting time of patients. These records used provides an in-depth look at patient patterns and actual surgery information, which we can use to refine and enhance scheduling predictions. Based on the patient privacy policy from Monash Health, I was not allowed to publish the actual datasets, however, we can discuss through about the datasets I have applied below, thus, we can have a broader view about this project even without the real dataset.

Our primary datasets were:

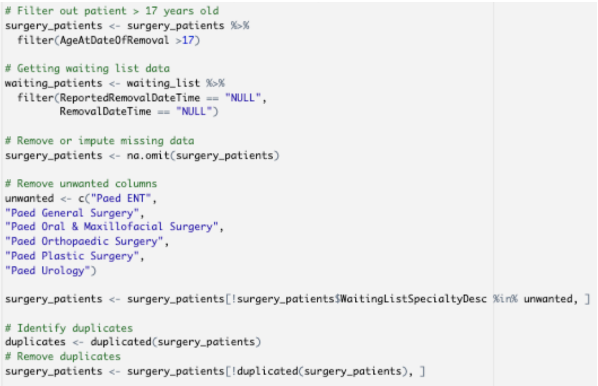
1. **Waiting List Data**: This dataset includes patient demographics, assigned priority codes, specialties, and anticipated stay types for all patients awaiting surgery. The use of this dataset was essential to capture the characteristics of patients who may potentially fit available theatre slots. This dataset with the information of patients who are waiting, which will later become our predict sample.
2. **Surgery Patients Data**: In this dataset, records on completed surgeries were integrated with the group of patients who used to be in the waiting list, only completed cases were used in this dataset. We eliminate every other waited cases that patients were removed from the waiting list with other reasons such as pass away, move to other hospital... This dataset captures every single detail of surgery cases, patient information and hospital treatment specifics. This dataset served as our primary source for actual surgery duration, the target variable in the predictive model.

From these datasets, we move to the preprocessing stage which began with basic data cleaning, where we ensured only adult cases (patients older than 17) were included, removed non-theatre cases and cancellations, and handled missing values for key variables to maintain data integrity. Ultimately, these datasets were prepared and merged, capturing cases performed at Monash Health facilities including the Monash Medical Centre, Dandenong Hospital, and Casey Hospital.

### Data Preprocessing

Our preprocessing included filtering and additional transformations to improve the predictive quality of our data. Key steps included:

- **Data Cleaning and Filtering**: We excluded cases not relevant to routine theatre scheduling, such as pediatric or non-theatre cases, ensuring only elective or scheduled surgeries were retained.



```
# Filter out patient > 17 years old
surgery_patients <- surgery_patients %>%
  filter(AgeAtDateOfRemoval >17)

# Getting waiting list data
waiting_patients <- waiting_list %>%
  filter(ReportedRemovalDateTime == "NULL",
         RemovalDateTime == "NULL")

# Remove or impute missing data
surgery_patients <- na.omit(surgery_patients)

# Remove unwanted columns
unwanted <- c("Paed ENT",
"Paed General Surgery",
"Paed Oral & Maxillofacial Surgery",
"Paed Orthopaedic Surgery",
"Paed Plastic Surgery",
"Paed Urology")

surgery_patients <- surgery_patients[!surgery_patients$WaitingListSpecialtyDesc %in% unwanted, ]

# Identify duplicates
duplicates <- duplicated(surgery_patients)
# Remove duplicates
surgery_patients <- surgery_patients[!duplicated(surgery_patients), ]
```

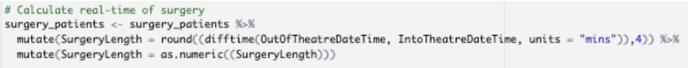
Figure 1: Data Wrangling Steps

- **Generating the target variable Surgery Length (Figure 2) and additional feature (Figure 3)**:
  - **SurgeryLength**: This is our target variable which is calculated using the dataset **Surgery Patients Data** with the following equation:

    **SurgeryLength = OutOfTheatreDateTime - IntoTheatreDateTime**

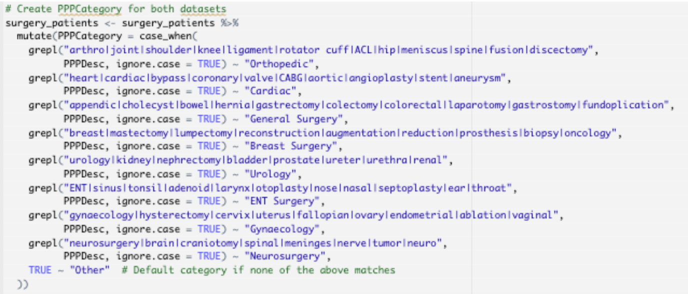
    Where:

    - ∗ **OutOfTheatreDateTime**: The time when past patients were officially recorded as having been removed from the operating room.
    - ∗ **IntoTheatreDateTime**: The time when past patients were officially recorded as having entered the operating room.
  - **Principal Prescribed Procedure Category (PPPCategory)**: This variable categorized procedures based on the descriptive keywords in the original Principal Prescribed Procedure Description column in my dataset. This is created since some model such as Random Forest cannot capture the amount of unique values in a variable.



```r
# Calculate real-time of surgery
surgery_patients <- surgery_patients %>%
  mutate(SurgeryLength = round((difftime(OutOfTheatreDateTime, IntoTheatreDateTime, units = "mins")),4)) %>%
  mutate(SurgeryLength = as.numeric((SurgeryLength)))
```

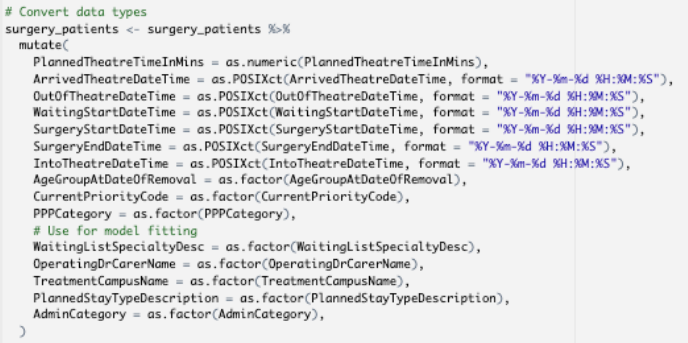
Figure 2: Surgery Length calculation



```r
# Create PPPCategory for both datasets
surgery_patients <- surgery_patients %>%
  mutate(PPPCategory = case_when(
    grepl("arthro|joint|shoulder|knee|ligament|rotator cuff|ACL|hip|meniscus|spine|fusion|discectomy",
          PPPDesc, ignore.case = TRUE) ~ "Orthopedic",
    grepl("heart|cardiac|bypass|coronary|valve|CABG|aortic|angioplasty|stent|aneurysm",
          PPPDesc, ignore.case = TRUE) ~ "Cardiac",
    grepl("appendic|cholecyst|bowel|hernia|gastrectomy|colectomy|colorectal|laparotomy|gastrostomy|fundoplication",
          PPPDesc, ignore.case = TRUE) ~ "General Surgery",
    grepl("breast|mastectomy|lumpectomy|reconstruction|augmentation|reduction|prosthesis|biopsy|oncology",
          PPPDesc, ignore.case = TRUE) ~ "Breast Surgery",
    grepl("urology|kidney|nephrectomy|bladder|prostate|ureter|urethra|renal",
          PPPDesc, ignore.case = TRUE) ~ "Urology",
    grepl("ENT|sinus|tonsil|adenoid|larynx|otoplasty|nose|nasal|septoplasty|ear|throat",
          PPPDesc, ignore.case = TRUE) ~ "ENT Surgery",
    grepl("gynaecology|hysterectomy|cervix|uterus|fallopian|ovary|endometrial|ablation|vaginal",
          PPPDesc, ignore.case = TRUE) ~ "Gynaecology",
    grepl("neurosurgery|brain|craniotomy|spinal|meninges|nerve|tumor|neuro",
          PPPDesc, ignore.case = TRUE) ~ "Neurosurgery",
    TRUE ~ "Other"  # Default category if none of the above matches
  ))
```

Figure 3: PPP Category Created

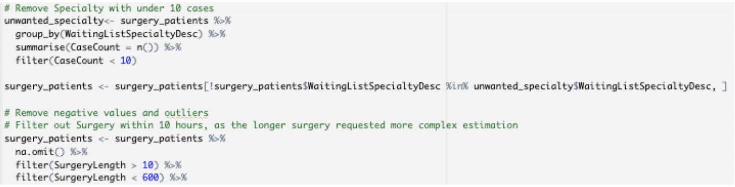
- **Data Transformations for Modeling**: Several categorical variables, such as `PlannedStayTypeDescription` and `TreatmentCampusName`, was converted into factors to facilitate more accurate analysis in machine learning models for both dataset.



```r
# Convert data types
surgery_patients <- surgery_patients %>%
  mutate(
    PlannedTheatreTimeInMins = as.numeric(PlannedTheatreTimeInMins),
    ArrivedTheatreDateTime = as.POSIXct(ArrivedTheatreDateTime, format = "%Y-%m-%d %H:%M:%S"),
    OutOfTheatreDateTime = as.POSIXct(OutOfTheatreDateTime, format = "%Y-%m-%d %H:%M:%S"),
    WaitingStartDateTime = as.POSIXct(WaitingStartDateTime, format = "%Y-%m-%d %H:%M:%S"),
    SurgeryStartDateTime = as.POSIXct(SurgeryStartDateTime, format = "%Y-%m-%d %H:%M:%S"),
    SurgeryEndDateTime = as.POSIXct(SurgeryEndDateTime, format = "%Y-%m-%d %H:%M:%S"),
    IntoTheatreDateTime = as.POSIXct(IntoTheatreDateTime, format = "%Y-%m-%d %H:%M:%S"),
    AgeGroupAtDateOfRemoval = as.factor(AgeGroupAtDateOfRemoval),
    CurrentPriorityCode = as.factor(CurrentPriorityCode),
    PPPCategory = as.factor(PPPCategory),
    # Use for model fitting
    WaitingListSpecialtyDesc = as.factor(WaitingListSpecialtyDesc),
    OperatingDrCarerName = as.factor(OperatingDrCarerName),
    TreatmentCampusName = as.factor(TreatmentCampusName),
    PlannedStayTypeDescription = as.factor(PlannedStayTypeDescription),
    AdminCategory = as.factor(AdminCategory),
  )
```

Figure 4: Data Types Converted

- **Further Refines**: After created the Surgery Length, I also refined the Surgery Length variable to range 10 to 600 minutes with number of total case by specialty more than 10, as the group outside this range is considered as outliers and may affect the model's performance. In addition, surgery duration that exceed 10 hours may need more complex approach for predicting as it also need medical professional to consider the patient's health status.



```
# Remove Specialty with under 10 cases
unwanted_specialty<- surgery_patients %>%
  group_by(WaitingListSpecialtyDesc) %>%
  summarise(CaseCount = n()) %>%
  filter(CaseCount < 10)

surgery_patients <- surgery_patients[!surgery_patients$WaitingListSpecialtyDesc %in% unwanted_specialty$WaitingListSpecialtyDesc, ]

# Remove negative values and outliers
# Filter out Surgery within 10 hours, as the longer surgery requested more complex estimation
surgery_patients <- surgery_patients %>%
  na.omit() %>%
  filter(SurgeryLength > 10) %>%
  filter(SurgeryLength < 600) %>%
```

Figure 5: Further Refined Steps

Each transformation in this data wrangling proceess was tailored to capture potential influences on surgery duration while maintaining data clarity.

### 3.2 Exploratory Data Analysis (EDA)

Next, we move to the Exploratory data analysis to investigate insights into surgery length patterns based on patient demographics, specialties, and operational variables:

- **The Completeness of data after cleaning:** From the plot, we can investigate the completeness of data with `Surgery Patients Data` on the left hand side and `Waiting List Data` on the right hand side. Our datasets included `character`, `factor`, `numeric`, and `POSIXct` (date-time) types, indicating a mix of categorical, numeric, and time-based variables, all prepared for analysis. However, most of our analysis will based on the categorical variables in the dataset, numeric variable is only represent for the quantitative ones. We can also observe some missing data in the plot, this is due to the inherent data limitations as medical data required human manual record rather than error, and this will not affect our analysis since all needed variables have had null values removed, thus, we will keep it as it is.

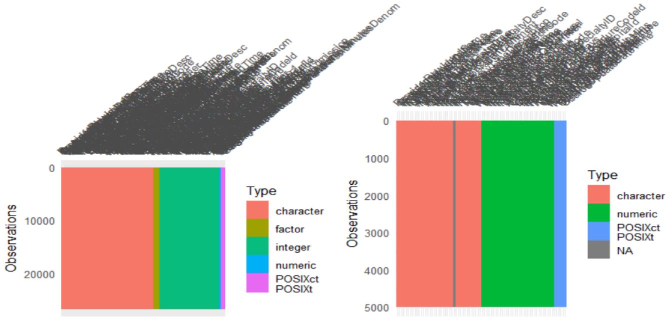


Figure 6: Data Completeness

- **Distribution of Human estimation and Real-time Surgery Duration**: The distribution of **Planned Theatre Time** (top) shows a right-skewed pattern with most estimates clustered under 100 minutes, suggesting that planned durations are generally conservative. In contrast, **Actual Surgery Time** (bottom) also has a right-skewed distribution but with a broader spread, indicating that surgeries often take longer than initially estimated, with some extending up to 600 minutes. This comparison highlights a gap between estimated and actual surgery times, underscoring the importance of improving scheduling accuracy which is also our goal for this project.

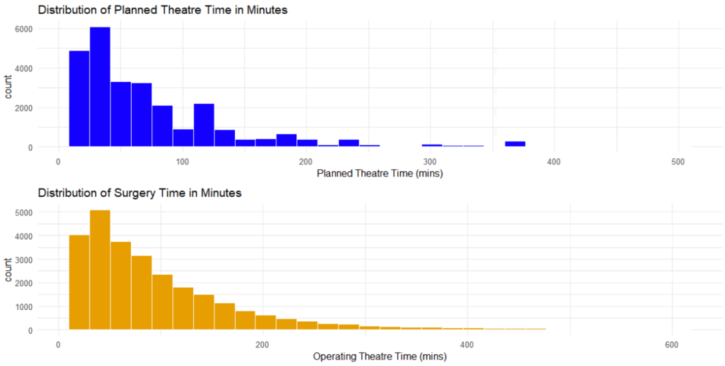


Figure 7: The distribution of Surgery Duration between Human Estimation and Real-time records

- **Surgery Duration and Specialty**: Cardiac and Neurosurgery cases had longer durations, typically exceeding 300 minutes, indicating the need for extended theatre allocation. Shorter durations were observed in specialties like Ophthalmology and Urology, suggesting these may be suitable for filling shorter gaps in theatre schedules. From this, we can conclude that the more complex surgery, the longer operating time it will take. In addition, we can also observe here that we have serveral outliers in different specialty, this may based on the status of patients.

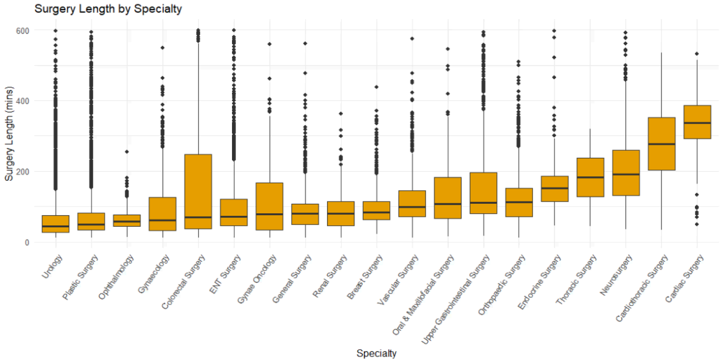


Figure 8: Real-time Surgery length by specialty

- **Waiting Time Segmentation**: The analysis reveals that 84% of surgeries were completed within a year, leaving 16% of cases with extended waiting times. This distribution reflects not only the limitations of the available dataset but is also corroborated by Monash Health, which faces challenges in gathering long-wait data. This limitation raises concerns about the model's predictive efficiency for long-term waits. Additionally, as shown in the plot below, there is no clear relationship between the variety in waiting days and the complexity of the specialty or the number of cases, represented by the size of each point.

| | |
|---|---|
| Proportion of Patient waiting under 1 year | 0.8405 |

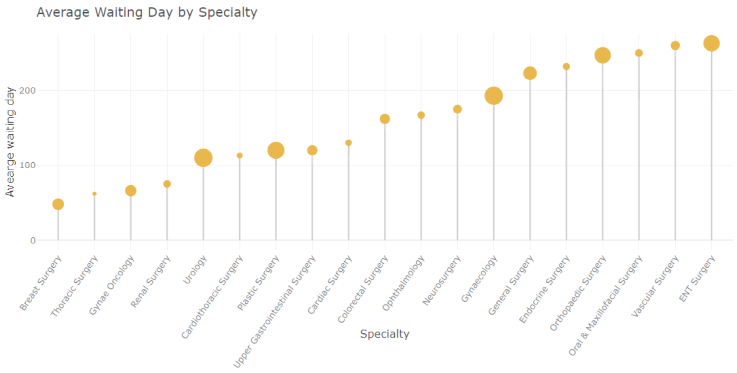


Figure 9: Average patient waiting day by specialty

### 3.3 7 Features Engineering

Feature engineering was critical for improving the predictive power of the model by extracting meaningful patterns and relationships in the data that might influence surgery duration. Based on domain knowledge and exploratory analysis, seven key features were developed or transformed to aid in predictions. Below is an in-depth explanation of each feature we use in this model development:

1. **AgeGroupAtDateOfRemoval**:
   This feature groups patients by age, capturing different age ranges that may influence surgery duration. As age impacted surgery duration, we can observe this from Figure 10, with **younger patients (aged 18-64)** generally requiring longer surgeries compared to **older patients (above 84)**. This trend possibly due to the types of procedures and the intensity of treatment for each age group. We made an assumption here that this may caused by the health status of the patients, as the older patients may have more health issues that need shorten the surgery time.

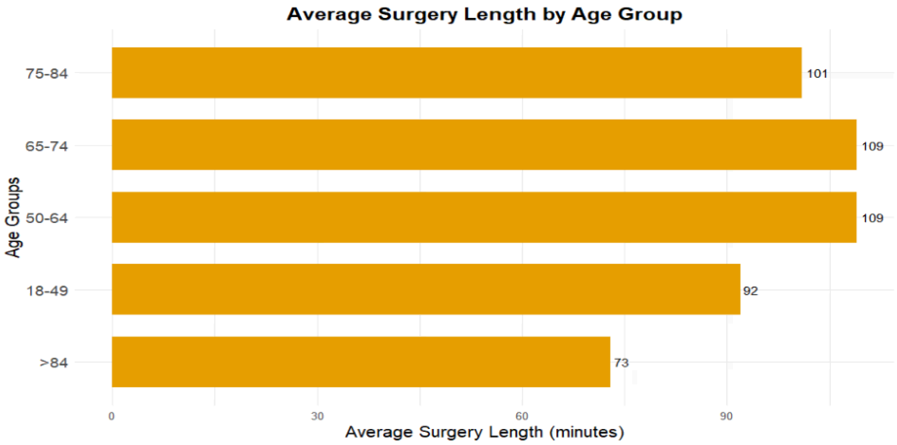


Figure 10: Average Surgery Length by Age

2. **CurrentPriorityCode**:
   The priority code indicates the urgency level of a case which may directly impacting surgery scheduling. This feature is crucial as higher-priority surgeries (e.g., urgent or high-risk cases) may require longer durations due to the complexity and additional safety measures involved. However, from our plot (@fig-avgcode), we can see that even when there's various priority code, the surgery time for these priority code does not significantly differ.

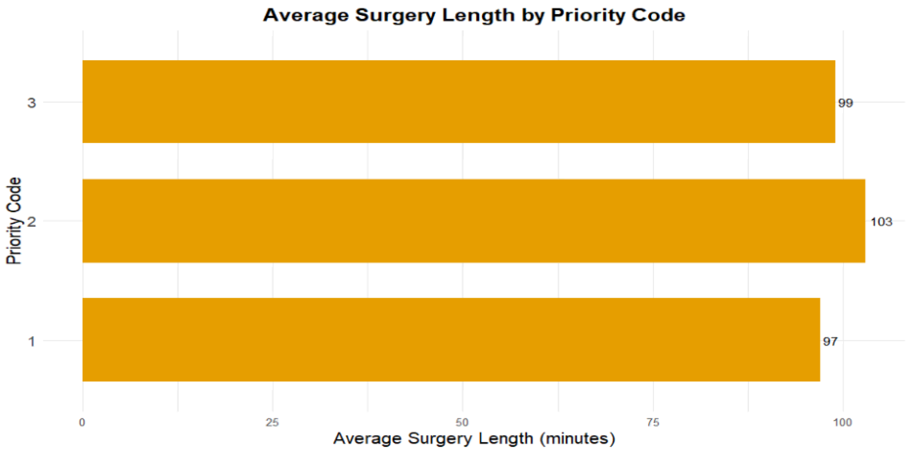


Figure 11: Average Surgery Length by Current Priority Code

3. **WaitingListSpecialtyDesc**:
   Each specialty has different procedural norms and expected time requirements; for example, Cardiac and Neurosurgery typically involve longer, complex surgeries, while Ophthalmology often includes shorter, more routine procedures. This categorical variable captures the specialty assigned to each case, enabling the model to consider the type of surgery as a factor in predicting duration. By leveraging this feature, the model can differentiate between specialties, leading to more precise predictions tailored to specific surgery types. As we also mentioned above, the more complex surgery is, the longer time it will take.

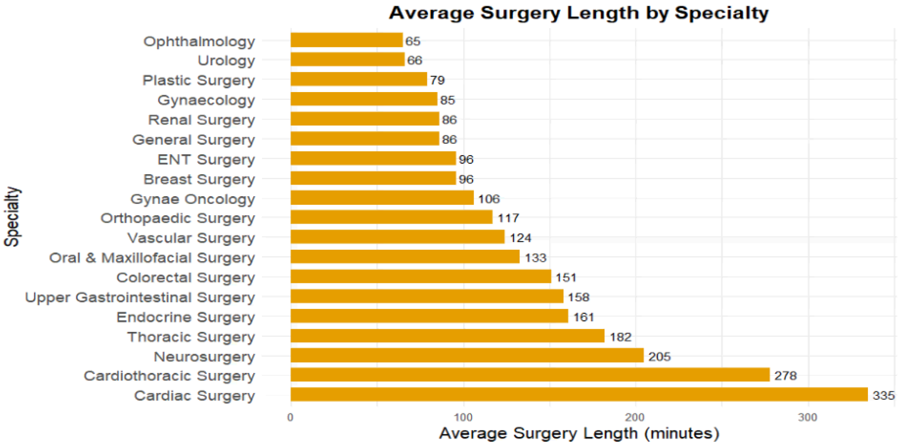
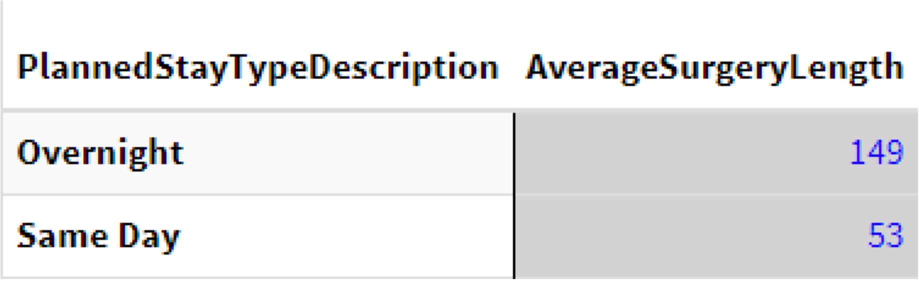


Figure 12: Average Surgery Length by Specialty

4. **PlannedStayTypeDescription**:
   This feature categorizes whether a surgery requires an overnight stay or is a same-day procedure. Typically, surgeries associated with overnight stays are more complex and time-consuming, often requiring extended recovery periods before discharge. This feature allows the model to incorporate the planned hospital stay as an indicator of surgery complexity, improving the accuracy of predictions for complex surgeries that necessitate longer theatre times.



| PlannedStayTypeDescription | AverageSurgeryLength |
|---|---|
| Overnight | 149 |
| Same Day | 53 |

Figure 13: Average Surgery Length by Planned Stay

5. **PPPCategory**:
   Derived from keywords in the `PPPDesc` (Principal Prescribed Procedure Description) column, `PPPCategory`groups surgeries into broad categories (e.g., Orthopedic, Cardiac, ENT) based on the nature of the procedure. This variable enhances the model's interpretive power by consolidating diverse surgical types into general categories that exhibit consistent patterns in duration. It serves as a high-level indicator of the surgical approach, helping the model generalize patterns across similar types of surgeries.

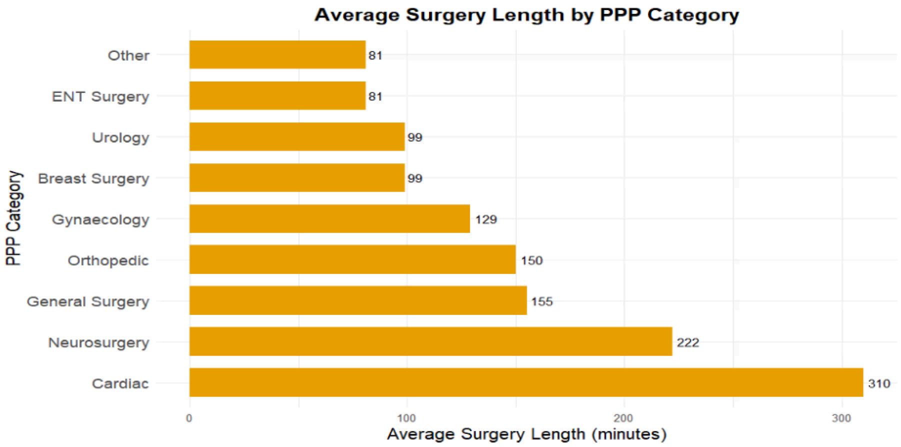


Figure 14: Average Surgery Length by PPP Category

6. **TreatmentCampusName**:
   Different campuses within Monash Health may have unique operational efficiencies, surgical teams, or available resources, potentially influencing surgery duration. Including `TreatmentCampusName` as a feature allows the model to adjust for campus-specific factors, thus accounting for location-based variations in surgery length.

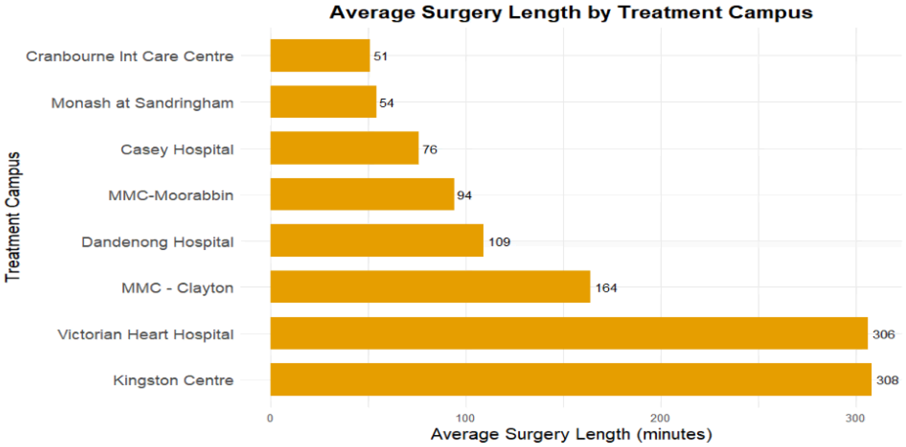


Figure 15: Average Surgery Length by Treament campus

7. **AdminCategory**:
   The Admin Category feature classifies surgeries based on the administrative context of patient admissions, such as work cover cases, asylum seekers, or private patients. This classification is essential for understanding resource allocation and planning, as different administrative categories often reflect distinct patient needs and hospital priorities. As shown in the graph, certain categories, like **Private** and **TAC − Acute**, have longer average surgery times, potentially due to more complex or elective procedures, while categories such as **DVA − Acute** and **Public − Eligible** tend to have shorter durations. This information helps in tailoring resource allocation to better fit the expected duration for each administrative category.

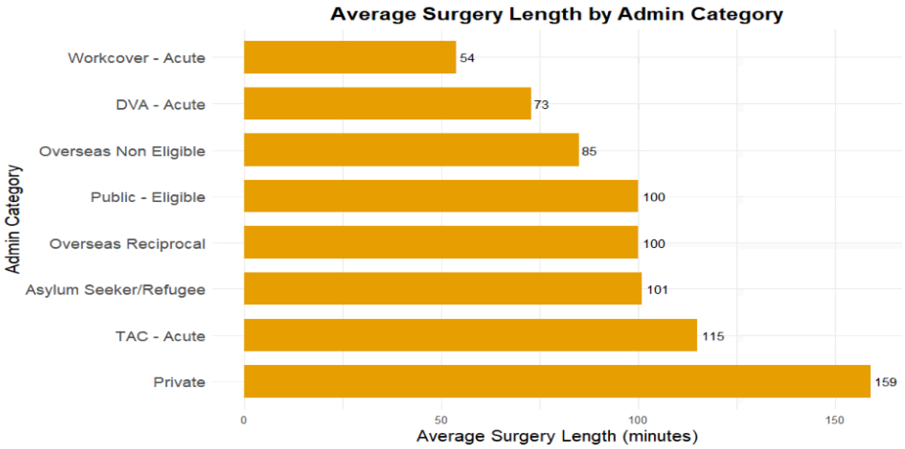


Figure 16: Average Surgery Length by Admin Category

### Additional Transformations and Justifications

To maximize the predictive power of these features we make sure that all of them were encoded as factors as mentioned above in data preprocessing steps to maintain interpretability and ensure compatibility with machine learning algorithms.

Together, these features allow the model to interpret the relationships between patient demographics, surgical needs, and hospital operational factors, ultimately enhancing the accuracy and robustness of the surgery duration predictions.

### 3.4 Model Fit

This section describes our step-by-step model fitting including:

1. **Data preparation**
2. **Cross-validation set up**
3. **Model training and hyperparameter tuning**
4. **Prediction and evaluation**
5. **Final chosen model and apply it to predict the target variable with Waiting List dataset**

We first split the primary dataset of Surgery Patients into training and test sets using stratification by `WaitingListSpecialtyDesc`, which represents the specialty. We then applied cross-validation to split our training set into 5 folds for more robust analysis. Meanwhile, we also transformed the target variable, `Surgery Length` using the logarithmic transformation (`log`) to handle skewness and improve model stability as we can observe Surgery Length is right skewed in our Exploratory Data Analysis part. However, the final prediction on the `Waiting List` dataset will be converted back to the original scale using the exponential function (`exp`).

Five models were employed to determine the best predictive performance: Linear Regression, Ridge Regression, Lasso Regression, Random Forest, and Gradient Boosting Machine (GBM). Each model was selected for its unique capacity to capture different relationships within the data, from linear and regularized models to complex ensemble methods. The process included hyperparameter tuning and careful analysis of model metrics to identify the optimal model for this dataset.

#### Model Selection and Hypertuning parameter explanation

#### Linear Regression

Linear regression was selected as the baseline model, offering a straightforward, interpretable approach. By assuming a linear relationship between the features and the target variable (Surgery Length), it provides a benchmark against which more advanced models can be compared. Although linear regression may not fully capture the non-linear patterns within the surgery data, it serves as an essential reference, allowing us to evaluate the additional gains achieved by more complex models.

The code applies linear regression to the log-transformed surgery duration, addressing skewness in the data. The log transformation standardizes the distribution, reducing the effect of outliers and aligning the data more closely with the assumptions of a linear model.

### Ridge Regression

This model addresses multicollinearity among predictors by applying an $L_2$ regularization penalty. Ridge was chosen to retain all predictors while reducing the influence of less impactful ones, making it suitable when multicollinearity might distort variable importance.

In the code, `cv.glmnet` is used with `alpha = 0` to perform ridge regression, tuning the lambda parameter over a specified `lambda_grid`. Cross-validation is applied to find the optimal `lambda` value, which balances the trade-off between fitting the data well and penalizing large coefficients. By using `lambda.min` as the best value, the model ensures robust generalization on unseen data. Each fold's performance is averaged to select the `lambda` that minimizes errors across validation sets.

### Lasso Regression

Lasso's ability to perform feature selection was a key reason for its inclusion. By applying an $L_1$ penalty, Lasso regression drives certain coefficients to zero, excluding less important predictors (e.g., Priority code) and creating a sparse model. This feature selection is beneficial for interpretability and model simplicity, especially given the diverse predictors in the dataset.

Using `cv.glmnet` with `alpha = 1`, the code sets up Lasso regression with a cross-validated `lambda_grid` to explore a range of penalty values. This tuning process identifies the optimal `lambda` that balances sparsity and predictive accuracy. The selected lambda value excludes non-informative predictors, focusing the model on the most significant features, thus improving generalization. Similar to ridge, the best `lambda` is averaged across folds to ensure consistency.

### Random Forest

Random Forest was selected for its ability to handle non-linear relationships, which are likely present in the dataset. As an ensemble model, it constructs multiple decision trees, each trained on different subsets of the data, then averages their predictions. This ensemble approach reduces variance and improves robustness, which is particularly valuable in capturing the complex interactions among features that influence surgery duration.

In the code, Random Forest hyperparameters are tuned through grid search:

- **`mtry`**: Controls the number of features randomly selected at each tree split, striking a balance between depth and generalizability.
- **`min_n`**: Sets the minimum number of observations required for a node to split, preventing overfitting by setting a threshold for node purity.
- **`ntree`**: Defines the number of trees in the forest. A higher `ntree` generally improves accuracy but increases computation time.

The code evaluates the performance of each parameter combination, identifying the model that achieves the lowest RMSE. Random Forest's ability to capture feature importance also provides valuable insights into which variables most influence surgery durations.

### Gradient Boosting Machine (GBM)

GBM was chosen to leverage boosting, a technique that sequentially corrects errors from previous iterations, making it highly effective in capturing complex, subtle patterns within the data. By adjusting for residual errors at each step, GBM builds a strong predictive model, ideal for non-linear data such as surgery durations where multiple factors interact in complex ways.

The following hyperparameters are tuned for GBM:

- **`n.trees`**: The number of boosting iterations, where more trees generally improve accuracy but increase computation time.
- **`interaction.depth`**: Controls the depth of each tree, influencing the complexity and flexibility of the model.
- **`shrinkage`**: A learning rate parameter that scales each tree's contribution, helping to prevent overfitting by slowing down the learning process.
- **`n.minobsinnode`**: Sets the minimum number of observations in each terminal node, providing control over tree complexity and reducing overfitting risks.

Through grid search, the code identifies the optimal combination of these parameters, selecting the model with the lowest RMSE. GBM's iterative approach, combined with its flexibility in handling complex interactions, makes it an ideal choice for refining predictions on data with nuanced, multi-level relationships among features.

Each model contributes unique strengths, from the simplicity and interpretability of linear regression to the flexibility and accuracy of ensemble methods like Random Forest and GBM. By leveraging a combination of linear models (Ridge and Lasso) and ensemble models, the model selection process addresses potential data challenges, including multicollinearity, non-linear dependencies, and high-dimensionality, ultimately improving the accuracy and robustness of surgery duration predictions.

### Model Results

Following model training and hyperparameter tuning, each model was evaluated using three key performance metrics: **Root Mean Square Error (RMSE)**, **Mean Absolute Error (MAE)**, and **R-squared**. These metrics help to assess prediction accuracy, consistency, and the model's ability to explain variance in surgery durations. The table below summarizes the performance metrics for each model.

The result after fitting model into cross validation:

| Model | RMSE | MAE | R-squared |
|---|---|---|---|
| Linear Regression | 66.90015 | 40.50439 | 0.4293917 |
| Ridge Regression | 66.88930 | 40.49791 | 0.4296786 |
| Lasso Regression | 66.88276 | 40.49139 | 0.4299487 |
| Random Forest | 59.14312 | 35.27976 | 0.5497279 |
| Gradient Boosting | 59.97960 | 35.91501 | 0.5394331 |

The above result is found by calculating the average of invidual metric for every folds with each model. We can investigate the result of each folds by using the plot below.

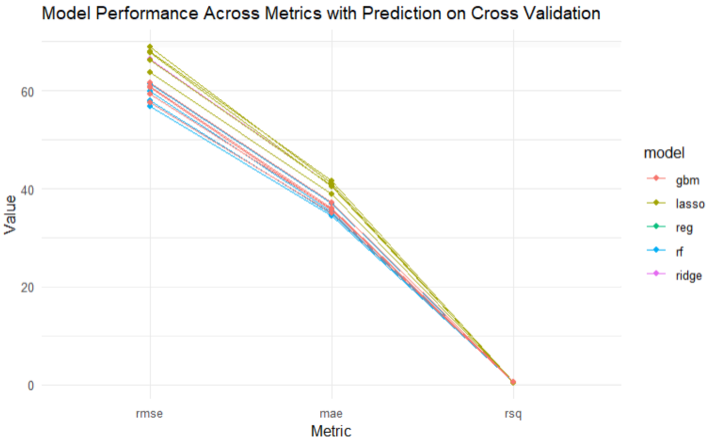


Figure 17: Plot for Cross Validation Folds metrics for different model applied

After investing the result, I have decided to remove Linear Regression model as it obviously not capture the target variable Surgery Length as good as the Ridge and Lasso Regression model.

The result after fitting model to the testing sample (**Waiting list patients**):

| Model | RMSE | MAE | R-squared |
|---|---|---|---|
| Ridge Regression | 68.36267 | 41.01024 | 0.4086466 |
| Lasso Regression | 68.36262 | 41.00753 | 0.4087569 |
| Random Forest | 59.71910 | 35.52226 | 0.5437127 |
| Gradient Boosting | 60.77043 | 36.20915 | 0.5296202 |

---

### 3.5.1 Ridge Regression Results

- **RMSE**: 68.36267
- **MAE**: 41.01024
- **R-squared**: 0.4086466

Ridge regression showed improvement over standard linear models as we can see from the cross validation metrics, demonstrating better control of multicollinearity through regularization. The model's RMSE and MAE indicate reduced prediction error, and the R-squared score of 0.4086466 suggests that Ridge explained around 40.9% of the variance in surgery durations.

**Interpretation**: Ridge regression effectively minimized the impact of correlated features, yet its inability to fully capture non-linear relationships limited its overall accuracy compared to ensemble models.

### 3.5.2 Lasso Regression Results

- **RMSE**: 68.36262
- **MAE**: 41.00753
- **R-squared**: 0.4087569

Lasso regression outperformed Ridge slightly by setting irrelevant coefficients to zero, thereby focusing only on impactful features. The slightly lower RMSE and MAE values suggest that Lasso was able to make slightly more accurate predictions. With an R-squared of 0.4087569, Lasso explained more of the variance in surgery duration than Ridge, highlighting the model's advantage in feature selection.

**Interpretation**: Lasso's ability to exclude non-contributory features provided a slight edge over Ridge, though it still lacked the complexity to fully capture the nuanced relationships in the dataset.

### 3.5.3 Random Forest Results

- **RMSE**: 59.71910
- **MAE**: 35.52226
- **R-squared**: 0.5437127

Random Forest performed strongly across all metrics, achieving the lowest RMSE and MAE, along with the highest R-squared value of approximately 0.543. This indicates that Random Forest captured over 54% of the variance in surgery durations. Its ensemble nature, leveraging multiple trees to capture non-linear relationships and feature interactions, contributed to this enhanced predictive power.

**Interpretation**: Random Forest's ability to capture complex relationships and interactions within the data made it highly effective. The model's superior performance across metrics suggests it is well-suited for predicting surgery durations in this dataset.

### 3.5.4 Gradient Boosting Machine (GBM) Results

- **RMSE**: 60.77043
- **MAE**: 36.20915
- **R-squared**: 0.5296202

GBM's results were comparable to Random Forest, with a slightly higher RMSE and MAE but a near-identical R-squared of 0.5296202. The boosting algorithm in GBM enabled it to incrementally improve predictions by reducing residual errors, which allowed it to capture complex patterns in the data.

**Interpretation**: GBM provided competitive performance with Random Forest, though it required more tuning and computational resources. Given its slight disadvantage in computational efficiency, Random Forest may be the preferred model for deployment.

### 3.5.5 Final selection

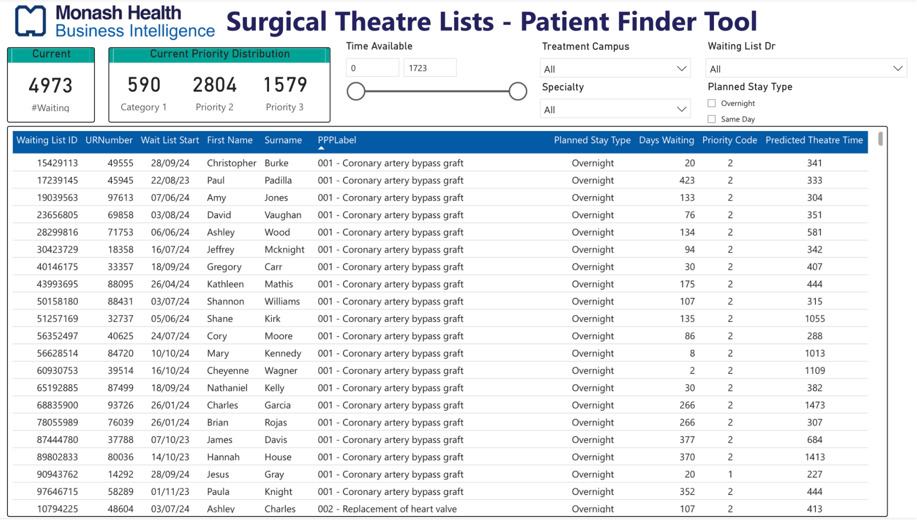
Random Forest and Gradient Boosting Machine were the top-performing models in terms of accuracy and variance explained. Both models are capable of capturing complex, non-linear relationships in the data, making them highly suitable for predicting surgery durations. However, Random Forest's balance of performance and efficiency makes it the most practical model for this application. We can clearly see that the Random Forest model outperformed other models with an RMSE of around 59.72, an R-squared of approximately 0.543, and an MAE of 35.52, indicating its robustness in predicting surgery lengths:

- **RMSE**: Indicates that predictions are, on average, about 60 minutes off the actual duration.

- **R-squared**: Shows that the model explains about 54.3% of the variance in surgery lengths.
- **MAE**: The mean absolute error of 35.5 minutes suggests that, on average, predictions are within this range of the actual surgery length.

### Front-End Tool (Demo Version)

The front-end tool developed for this project serves as a demonstration platform, designed to illustrate how predictive modeling could be integrated into Monash Health's theatre scheduling system. Currently using simulated data, the tool provides an interactive experience for the scheduling team and nurses, showcasing the potential real-time application of the Random Forest model's predictions.



Surgical Theatre Lists - Patient Finder Tool

Current: 4973 #Waiting

Current Priority Distribution: 590 Category 1 | 2804 Priority 2 | 1579 Priority 3

Time Available: 0 – 1723

Treatment Campus: All

Specialty: All

Waiting List Dr: All

Planned Stay Type: □ Overnight □ Same Day

| Waiting List ID | URNumber | Wait List Start | First Name | Surname | PPPLabel | Planned Stay Type | Days Waiting | Priority Code | Predicted Theatre Time |
|---|---|---|---|---|---|---|---|---|---|
| 15429113 | 49555 | 28/09/24 | Christopher | Burke | 001 - Coronary artery bypass graft | Overnight | 20 | 2 | 341 |
| 17239145 | 45945 | 22/08/23 | Paul | Padilla | 001 - Coronary artery bypass graft | Overnight | 423 | 2 | 333 |
| 19039563 | 97613 | 07/06/24 | Amy | Jones | 001 - Coronary artery bypass graft | Overnight | 133 | 2 | 304 |
| 23656805 | 69858 | 03/08/24 | David | Vaughan | 001 - Coronary artery bypass graft | Overnight | 76 | 2 | 351 |
| 28299816 | 71753 | 06/06/24 | Ashley | Wood | 001 - Coronary artery bypass graft | Overnight | 134 | 2 | 581 |
| 30423729 | 18358 | 16/07/24 | Jeffrey | Mcknight | 001 - Coronary artery bypass graft | Overnight | 94 | 2 | 342 |
| 40146175 | 33357 | 18/09/24 | Gregory | Carr | 001 - Coronary artery bypass graft | Overnight | 30 | 2 | 407 |
| 43993695 | 88095 | 26/04/24 | Kathleen | Mathis | 001 - Coronary artery bypass graft | Overnight | 175 | 2 | 444 |
| 50158180 | 88431 | 03/07/24 | Shannon | Williams | 001 - Coronary artery bypass graft | Overnight | 107 | 2 | 315 |
| 51257169 | 32737 | 05/06/24 | Shane | Kirk | 001 - Coronary artery bypass graft | Overnight | 135 | 2 | 1055 |
| 56352497 | 40625 | 24/07/24 | Cory | Moore | 001 - Coronary artery bypass graft | Overnight | 86 | 2 | 288 |
| 56628514 | 84720 | 10/10/24 | Mary | Kennedy | 001 - Coronary artery bypass graft | Overnight | 8 | 2 | 1013 |
| 60930753 | 39514 | 16/10/24 | Cheyenne | Wagner | 001 - Coronary artery bypass graft | Overnight | 2 | 2 | 1109 |
| 65192885 | 87499 | 18/09/24 | Nathaniel | Kelly | 001 - Coronary artery bypass graft | Overnight | 30 | 2 | 382 |
| 68835900 | 93726 | 26/01/24 | Charles | Garcia | 001 - Coronary artery bypass graft | Overnight | 266 | 2 | 1473 |
| 78055989 | 76039 | 26/01/24 | Brian | Rojas | 001 - Coronary artery bypass graft | Overnight | 266 | 2 | 307 |
| 87444780 | 37788 | 07/10/23 | James | Davis | 001 - Coronary artery bypass graft | Overnight | 377 | 2 | 684 |
| 89802833 | 80036 | 14/10/23 | Hannah | House | 001 - Coronary artery bypass graft | Overnight | 370 | 2 | 1413 |
| 90943762 | 14292 | 28/09/24 | Jesus | Gray | 001 - Coronary artery bypass graft | Overnight | 20 | 1 | 227 |
| 97646715 | 58289 | 01/11/23 | Paula | Knight | 001 - Coronary artery bypass graft | Overnight | 352 | 2 | 444 |
| 10794225 | 48604 | 03/07/24 | Ashley | Charles | 002 - Replacement of heart valve | Overnight | 107 | 2 | 413 |

Figure 18: Overview of the Front-end Tool

### Features and Design of the Demo Tool

1. **User-Friendly Interface**: The tool's interface was designed with simplicity in mind, ensuring that users can quickly navigate between fields and enter patient-specific details efficiently. The interface includes dropdown menus and auto-populated fields for critical features such as:

   - `WaitingListSpecialtyDesc`: specialty classification for each patient, which significantly influences surgery duration.
   - `CurrentPriorityCode`: urgency level that helps prioritize scheduling based on clinical need.
   - `Time available`: allows staff to enter the available time they want to search for patients

   Each input field is designed to streamline data entry, allowing the scheduling team to generate surgery duration estimates based on patient profiles without navigating complex workflows.

2. **Simulated Data for Predictive Feedback**: Although the tool is currently based on simulated data, it closely mimics the intended real-time functionality. By entering simulated patient cases, scheduling staff can observe how the tool would ideally predict surgery durations, rank patients, and recommend optimal scheduling times in a live setting. This demonstration allows users to familiarize themselves with the system's interface and simulated outputs, setting expectations for the tool's future integration with actual model results.

3. **Prioritization and Slot Matching**: Using simulated predictions, the tool demonstrates how patients could be prioritized based on urgency, wait time, and estimated surgery duration. This functionality helps the scheduling team align open slots with the needs of high-priority cases, reducing wait times for urgent patients. For instance, shorter simulated surgery durations can be matched to shorter time slots, which can enhance patient flow and ensure optimal resource utilization.

### Future Integration with Actual Predictive Model Results

Once connected to the predictive model's actual outputs, the front-end tool would enable Monash Health to use accurate, data-driven estimates for scheduling in real-time. This integration would support informed decision-making by allowing the scheduling team to view, at a glance, which patients are best suited for upcoming theatre slots based on their specific case details. Meanwhile, we will also have more features plug in for more convenience usage of the tools such as: interpreter needs...

### Practical Impact and Operational Benefits

The front-end tool's design addresses key scheduling challenges by providing a straightforward and accessible platform for data-driven theatre management. While currently in a demonstration phase with simulated data, the tool highlights Monash Health's commitment to modernizing scheduling through predictive analytics. The practical impacts include:

- **Enhanced Scheduling Precision**: With model-based predictions embedded directly into the tool, the scheduling team can improve theatre utilization by aligning available slots with patient requirements.
- **Dynamic Adjustments**: Once fully implemented, the tool will support dynamic adjustments, allowing staff to respond to real-time changes in patient admissions or cancellations efficiently.
- **Financial Benefits**: The front-end tool can also reduce the operating fee of Monash Health with better schedule management.

## Conclusion

This project successfully developed a predictive Random Forest model for estimating surgery durations, achieving an RMSE of 59.72 and an R-squared of 0.543. These metrics demonstrate the model's partial robustness in predicting surgery times, particularly for standard elective cases, providing Monash Health with a reliable basis for scheduling optimization.

### Key Findings

The Random Forest model emerged as the top performer, capturing over 54% of the variance in surgery durations, with low error metrics relative to other models tested. This result reflects the model's capability to handle complex relationships and interactions within the dataset, making it a suitable choice for resource planning in theatre scheduling which helps us to solve the operating room inefficiency management.

### Practical Implications

Integrating this predictive model into Monash Health's scheduling workflow offers multiple benefits:

- **Enhanced Theatre Utilization**: By identifying which patients are best suited for open time slots, the model has the potential to reduce idle theatre time and increase the efficiency of resource allocation.
- **Improved Patient Flow**: The model's prioritization of patients based on predicted surgery duration and urgency level can help minimize waiting times, improving patient experience and care delivery.
- **Financial Benefits**: The front-end tool offers Monash Health potential cost savings by optimizing surgery scheduling, which can lead to more efficient resource use and reduced operating expenses.

### Front-End Tool as a Demonstration

A **front-end tool** was developed to illustrate the model's potential within Monash Health's scheduling system. Currently a **demo version using simulated data**, this tool demonstrates the workflow and interface that would allow scheduling staff to input patient and procedure details, view simulated duration predictions, and make scheduling adjustments in real time. This demo tool is a valuable prototype, showcasing how a fully integrated system could operate, while setting the foundation for full implementation with live model outputs.

### Limitations and Future Directions

While the model performed well on standard cases, additional refinement could improve predictions for surgeries with high variability in duration. Key areas for future enhancement include:

- **Integrating More Patient Data**: Adding more features like patient history or procedure complexity could yield even greater accuracy.
- **Real-Time Data Integration**: Incorporating APIs to enable real-time data updates would allow the tool to respond dynamically to scheduling changes, such as cancellations or new patient admissions.
- **Specialty-Specific Models**: Further training the model on specific surgery types and segmentation analysis could enhance precision, making predictions even more useful for resource allocation.

In conclusion, this project highlights the potential of predictive modeling and prototype tool development for improving healthcare operations. A fully integrated model and tool could greatly support Monash Health's scheduling team, ultimately leading to better patient outcomes and more efficient theatre management.

## Reference

1. Monash Health. (2024). *Caring for you.* Retrieved October 25, 2024, from https://monashhealth.org/about/caring-for-you/

2. Schouten, A. M., Flipse, S. M., van Nieuwenhuizen, K. E., Jansen, F. W., van der Eijk, A. C., & van den Dobbelsteen, J. J. (2023). Operating room performance optimization metrics: A systematic review. *Journal of Medical Systems, 47*(1), 19. https://doi.org/10.1007/s10916-023-01912-9

3. Agency for Clinical Innovation. (n.d.). *Operating theatre efficiency: Clinical practice guide.* Retrieved October 27, 2024, from https://aci.health.nsw.gov.au/___data/assets/pdf_file/0004/252436/ACI-Operating-theatre-efficiency-clinical-practice-guide.pdf