TEXAS

The University of Texas at Austin

UT ORIE Applied Projects

Summer 2024

Leveraging Machine Learning to Predict Post Surgery Length – of – Stay to optimize the assignment of Patients to Inpatient and Outpatient Care

12th August 2024

Guided By:

Dr Jeffrey Siewerdsen, Professor Imaging Physics,

Aaron Milhorn, Data Scientist, Imaging Physics,

MD Anderson Cancer Centre

Supervised By:

Prof. Eric Bickel and Prof Erhan Kutanoglu,

Professors, Department of Operations Research and Industrial Engineering

The University of Texas at Austin

Submitted By:

Prudhvinath Guduru (UT EID: gp23259)

Soorya Sriram (UT EID: s9623)

Graduate Students from Operations Research and Industrial Engineering Department

Introduction

1.1 Problem Background:

A critical challenge with evolving healthcare demands is to strategically plan the surgical capacity to meet future demands. By leveraging mathematical models and ML techniques, the goal of the project is to understand, address and plan for the nuances introduced by location-based regulations including distinctions between inpatient and outpatient care.

With a dataset comprising 139,634 records and 43 unique features, the goal was to employ the state of the Machine Learning techniques to ensure optimal assignment of care facility.

1.2 Problem Statement:

A robust Machine Learning solution to predict the Post-Surgery Length of Stay into 4 categories to optimize the assignment of patients between inpatient and outpatient care based on a set of features known before, after and accommodating to the changes that might occur during the surgery.

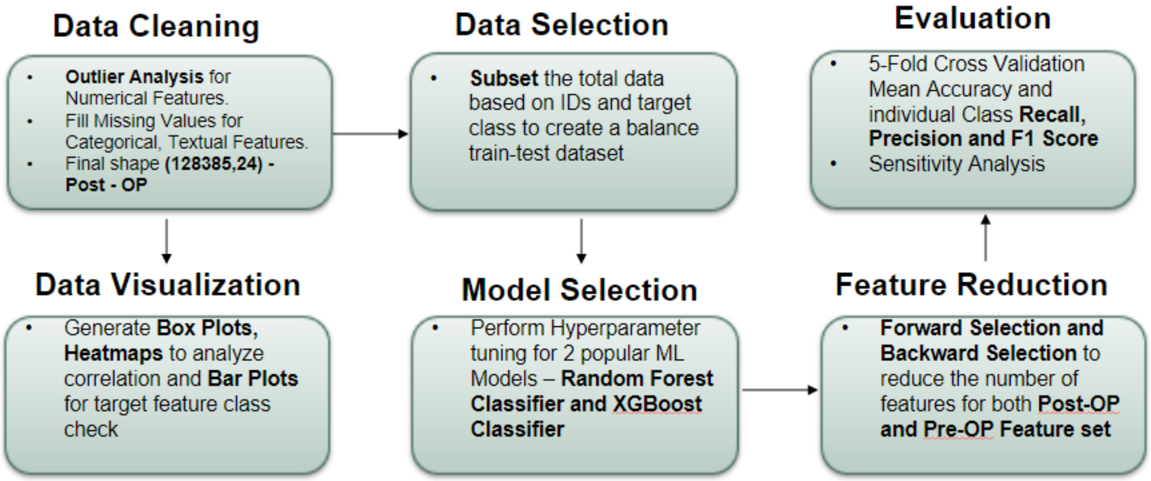
1.3 Problem Objective:

The project's objective is to find out which approach whether training the model on features known before the surgery and testing it on the same or to train the model on data known after the surgery and test it on the same set of features.

1.4 Problem Breakdown:

1. Preprocess unstructured data and identify features suitable for training and testing.
2. Perform Exploratory Data Analysis and visualizations to distinguish correlated features.
3. Feature Selection to reduce the total number of features and use feature importance techniques on the feature known before and after the surgery.
4. Evaluate using Metrics such as Precision, Recall and F1 score.
5. Sensitivity Analysis to explore the impact of features on the target predictions.

1.5 Solution - End to End Flow:



Data Cleaning

- Outlier Analysis for Numerical Features.
- Fill Missing Values for Categorical, Textual Features.
- Final shape **(128385,24) - Post - OP**

Data Selection

- **Subset** the total data based on IDs and target class to create a balance train-test dataset

Evaluation

- 5-Fold Cross Validation Mean Accuracy and individual Class **Recall, Precision and F1 Score**
- Sensitivity Analysis

Data Visualization

- Generate **Box Plots, Heatmaps** to analyze correlation and **Bar Plots** for target feature class check

Model Selection

- Perform Hyperparameter tuning for 2 popular ML Models – **Random Forest Classifier and XGBoost Classifier**

Feature Reduction

- **Forward Selection and Backward Selection** to reduce the number of features for both **Post-OP and Pre-OP Feature set**

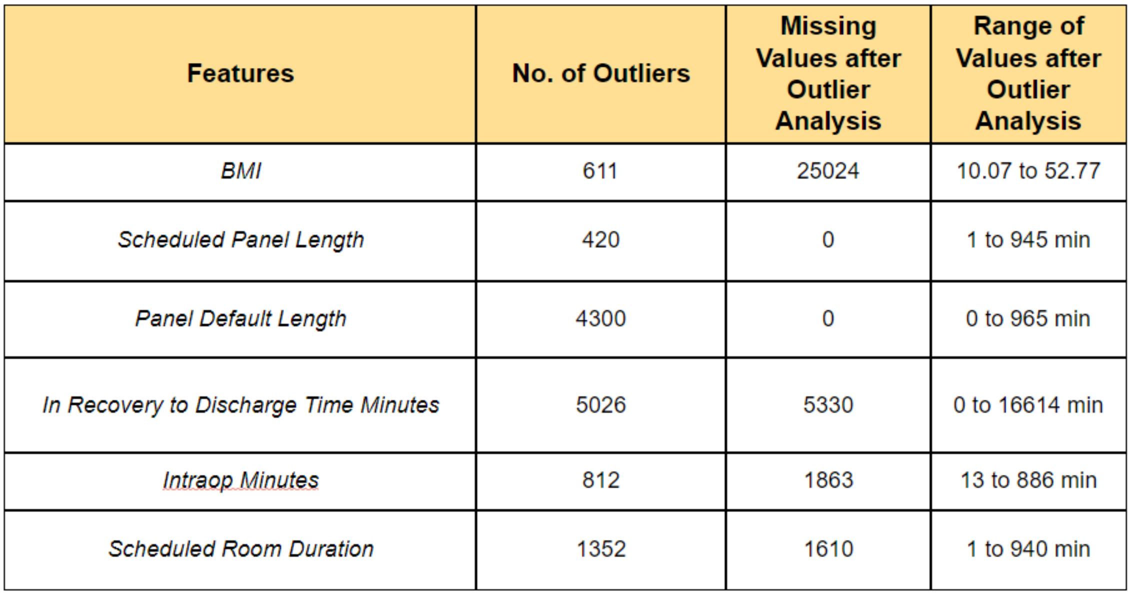
Methodology

2.1 Data Preprocessing:

The dataset provided to us had plenty of missing values, redundant as well as highly correlated features posing a challenge for the Machine Learning models to grasp the context appropriately. Consequently, effective preprocessing seeks to standardize the data to enhance the performance of the Machine Learning models being used. We took the following steps to ensure this:

a) Numerical Features:

Inter Quartile Range: IQR was used to measure the spread of the middle 50% of the data. Identifying a threshold for the IQR multiplier to balance outlier handling and data retention was the objective and the range 0f 1 to 3 was experimented with 0.25 increments before settling on 2.5 as it was found to have the right balance between removing outliers as well as having some of the variance still present.



Features	No. of Outliers	Missing Values after Outlier Analysis	Range of Values after Outlier Analysis
BMI	611	25024	10.07 to 52.77
Scheduled Panel Length	420	0	1 to 945 min
Panel Default Length	4300	0	0 to 965 min
In Recovery to Discharge Time Minutes	5026	5330	0 to 16614 min
Intraop Minutes	812	1863	13 to 886 min
Scheduled Room Duration	1352	1610	1 to 940 min

- The BMI for 10 age buckets starting from 0 – 9, 10 – 19 and so forth was created. The missing values were then filled by further grouping by sex and then median values were used to fill them up.
- The data was grouped by the Primary Service and missing values in the Scheduled Room Duration feature was filled with the median value for each subset.
- The data was subset by Panel Service and the median value to fill in missing entries in the Intra-Op Minutes.
- To remove any negative values, absolute values were used, and the missing values were filled based on Panel Service for In Recovery to Discharge Time Minutes.
- Hours to Recovery in Discharge was in turn calculated from In Recovery to Discharge Time Minutes and the missing the LOS 4 groups were also filled accordingly.
- The 5 respective Scheduled Panel Lengths and Panel Default Lengths were summated to create two features Total Panel Default Length and Total Scheduled Panel Length.

12,521 rows were removed after outlier analysis, and the range of these features was suitable to fill in missing values.

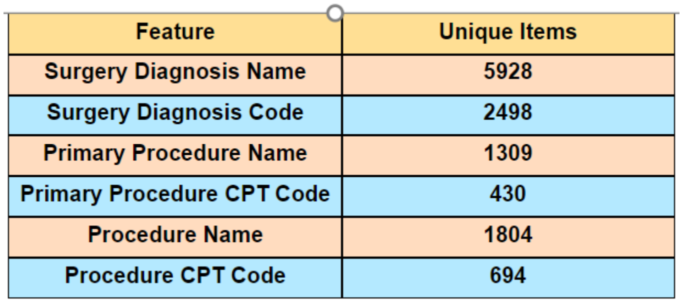
b) **Categorical Features**:

- Relabeled 'Unknown Ethnicity' and 'Decline to Answer' as 'Unknown’ for the Ethnicity Feature.
- Replaced missing values, 'Declined to Answer,' and 'Unknown' categories with 'Unknown’ for the Primary Race Feature.
- Anesthesia Type: Filled with 'General' (74.5% Prevalence)
- ASA Status: Filled with 'Severe Systemic Disease' (68.3% Prevalence).
- Patient Class: Missing values belonged to a single category and were filled with Surgery Admit.

c) **Textual Features**:

- Surgery Diagnosis Name feature was handled such that the rows with missing values were grouped by the Primary Surgeon ID and were filled with commonly occurring Surgery Diagnosis Name for that Primary Surgeon ID.
- Missing Primary Procedure Names were grouped by the Surgery Diagnosis Name. The Most commonly occurring primary procedure name for that surgery Diagnosis Name was then used to fill the missing values.

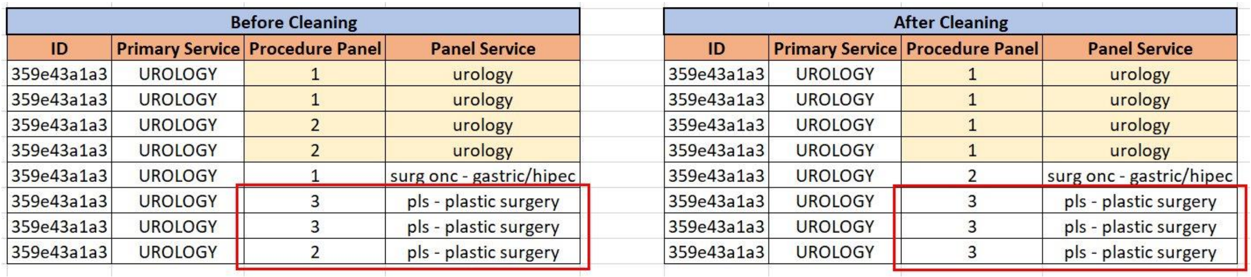
The following table describes the number of unique occurrences of the features.



Feature	Unique Items
Surgery Diagnosis Name	5928
Surgery Diagnosis Code	2498
Primary Procedure Name	1309
Primary Procedure CPT Code	430
Procedure Name	1804
Procedure CPT Code	694

In summary, the respective code columns do not capture the same diversity as the respective feature columns and were considered redundant for further experiments.

Data Inconsistency:



Before Cleaning			
ID	Primary Service	Procedure Panel	Panel Service
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	2	urology
359e43a1a3	UROLOGY	2	urology
359e43a1a3	UROLOGY	1	surg onc - gastric/hipec
359e43a1a3	UROLOGY	3	pls - plastic surgery
359e43a1a3	UROLOGY	3	pls - plastic surgery
359e43a1a3	UROLOGY	2	pls - plastic surgery

After Cleaning			
ID	Primary Service	Procedure Panel	Panel Service
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	1	urology
359e43a1a3	UROLOGY	2	surg onc - gastric/hipec
359e43a1a3	UROLOGY	3	pls - plastic surgery
359e43a1a3	UROLOGY	3	pls - plastic surgery
359e43a1a3	UROLOGY	3	pls - plastic surgery

A severe problem was the inconsistent numbering of Procedure Panel Number. In the above example, the primary service is Urology. However, when the Panel Service Urology occurs, the Procedure Panel number is not unique. Procedure Panel 1 should be assigned when Urology occurs in the Panel Service. For the other unique Panel services, a similar problem occurs.

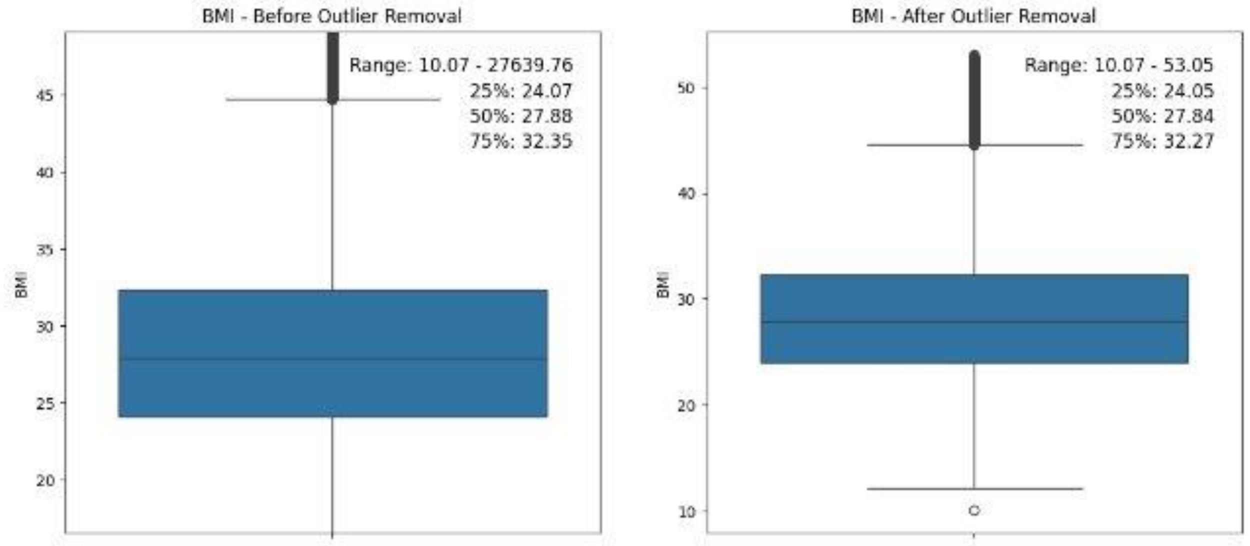
Solution: When the Primary Service was the same as Panel Service, the Procedure Panels were assigned as 1. The other unique panel services were assigned with the most commonly occurring Procedure Panel Number. If a tie was found, that is if the multiple Panel Services have the same Procedure Panel number, the next largest Procedure Panel was assigned.

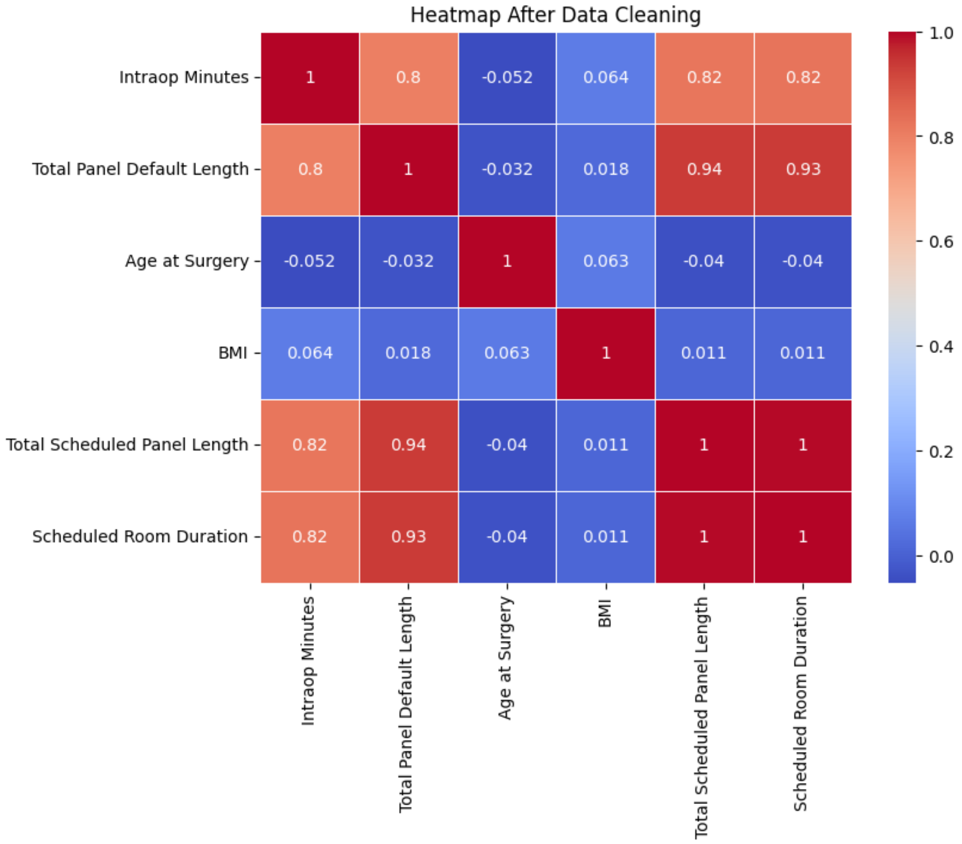
The following features were dropped:

1. Estimated Case Duration – 53% Missing values
2. Height (kg) - Highly correlated with BMI
3. Weight (cm) - Highly Correlated with BMI
4. ID
5. Surgery Diagnosis Code
6. Primary Service (Rolled Up) - A more detailed Primary Service Feature exists.
7. Primary Procedure CPT Code
8. Panel Service – Correlated with Procedure Panel.
9. Panel Service (Rolled Up)
10. Hours in recovery to discharge

2.2 Data Visualization:

The below figure shows the range and percentiles before and after outliers were removed.

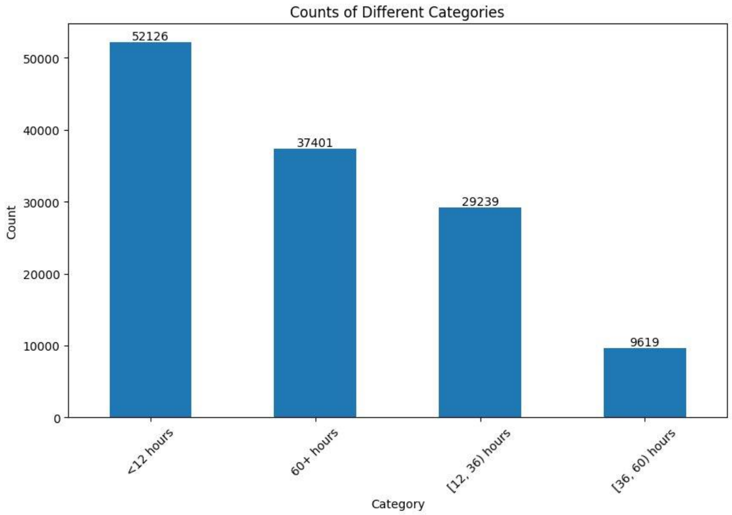




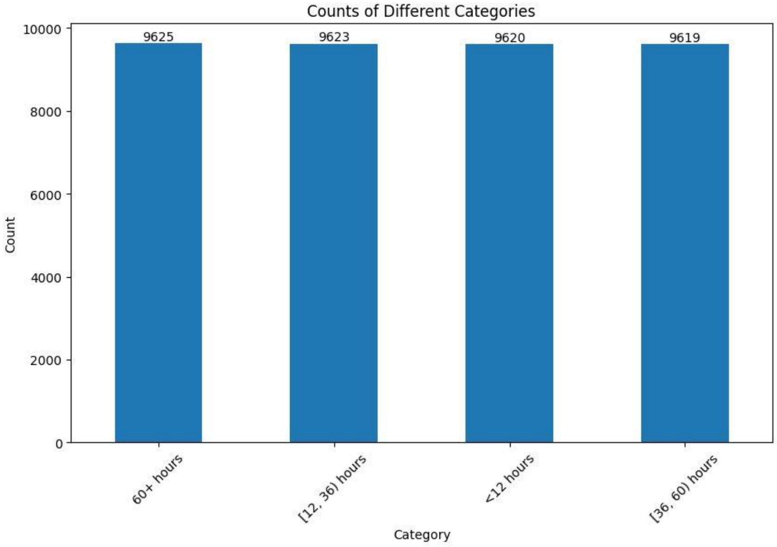
2.3 Data Selection

Post – OP Features:

The graph below shows the count of the different target classes in the total dataset where the number of final rows was 128,385 and the total number of features was 23. Class [36,60) is 7.5% of the total number of rows in that dataset.



To make sure, there is a balanced representation of all classes in the train and test dataset for model training, the dataset was created such that the balanced dataset was created with the class that occurs the least. The total number of rows after performing the cleaning process was 38487 and the number of features selected remained 24 but now with the class [36,60) have a 25% representation.



The final features selected were as follows:

Numerical Features:

1. Age at Surgery
2. BMI
3. Scheduled Room Duration
4. Intraop Minutes
5. Total Scheduled Panel Length
6. Total Panel Default Length

Categorical Features:

1. Sex
2. Ethnicity
3. Primary Race
4. Surgery Diagnosis Name
5. Location
6. Primary Service
7. Patient Class
8. Anesthesia Type
9. ASA Status
10. Primary Surgeon ID
11. Primary Procedure Name
12. Robotic Case?
13. Procedure Name
14. Scheduled?
15. Performed?
16. Procedure Panel
17. Panel Primary Surgeon ID

Target Feature:

1. LOS 4 Groups

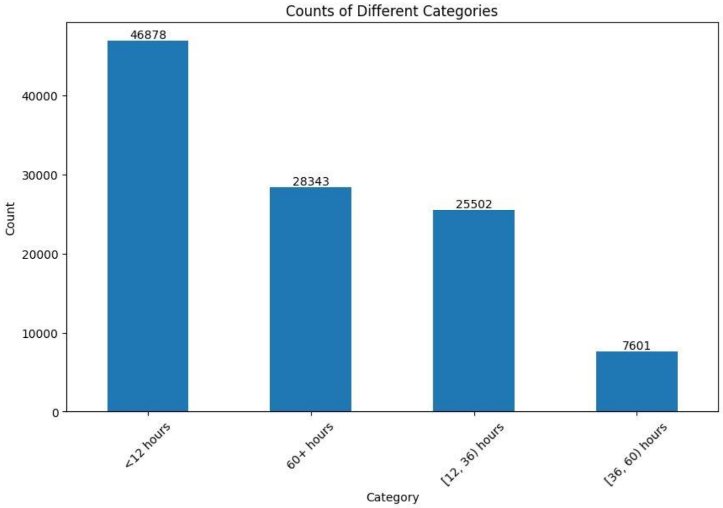
One Hot Encoding:

The process of converting categorical features into binary numerical columns. One hot encoded categorical features and final dataset shape is **(38487, 6158)**.

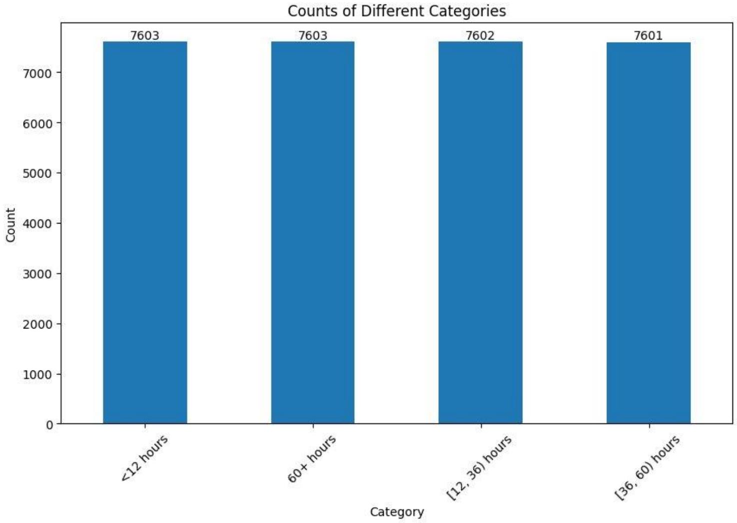
Pre – OP Features:

The following features were dropped –' Intraop Minutes',' Scheduled?', 'Location', 'Patient Class'.

The graph below shows the count of the different target classes in the total dataset where the number of final rows was 108,324 and the total number of features was 19. Class [36,60) is 7% of the total number of rows in that dataset.



To make sure, there is a balanced representation of all classes in the train and test dataset for model training, the dataset was created such that the balanced dataset was created with the class that occurs the least. The total number of rows after performing the cleaning process was 30409 and the number of features selected remained 19 but now with the class [36,60) have a 25% representation.



The final features selected were as follows:

Numerical Features:

1. Age at Surgery
2. BMI
3. Scheduled Room Duration
4. Total Scheduled Panel Length
5. Total Panel Default Length

Categorical Features:

6. Sex
7. Ethnicity
8. Primary Race
9. Surgery Diagnosis Name
10. Primary Service
11. Anesthesia Type
12. ASA Status
13. Primary Surgeon ID
14. Primary Procedure Name
15. Robotic Case?
16. Procedure Name
17. Performed?
18. Procedure Panel
19. Panel Primary Surgeon ID

Target Feature:

1. LOS 4 Groups

One Hot Encoding:

The process of converting categorical features into binary numerical columns. One hot encoded categorical features and final dataset shape is **(30409, 5831)**.

Label Encoding the target Feature as follows:

Class	Label
0	60+ hours
1	<12 hours
2	[12, 36) hours
3	[36, 60) hours

2.4 Model Selection:

Candidate Models

To tackle this classification problem, two powerful and widely used models were selected as candidates: the **Random Forest Classifier (RFC)** and **XGBoost (XGB)**. Both models are known for their robustness and ability to handle a wide variety of data patterns, making them ideal choices for this task.

- **Random Forest Classifier (RFC):** This ensemble method leverages multiple decision trees to improve predictive accuracy and control overfitting. The random forest is known for its simplicity and effectiveness, particularly in handling non-linear relationships and interactions between variables.

- **XGBoost (XGB):** XGBoost is a more sophisticated ensemble technique that builds on the gradient boosting framework. It is known for its efficiency, scalability, and performance, often yielding superior results in competitive machine learning tasks.

Evaluation Metrics

Given the classification nature of the problem, multiple evaluation metrics were selected to ensure a comprehensive assessment of model performance. The chosen metrics were:

- **Accuracy:** Measures the overall correctness of the model by evaluating the proportion of correctly classified instances.
- **Precision:** Focuses on the model's ability to correctly identify positive instances, minimizing false positives.
- **Recall:** Evaluates the model's capacity to detect all relevant positive instances, minimizing false negatives.
- **F1-Score:** Provides a balance between precision and recall, offering a single metric that considers both false positives and false negatives.

These metrics were chosen to ensure that the model not only performs well overall but also addresses the trade-offs between precision and recall.

Cross-Validation

To ensure the reliability and generalizability of the model performance, a 5-fold cross-validation approach was employed. This method involves splitting the dataset into five subsets or "folds". The model is trained on four of the folds and validated on the remaining one, repeating this process five times, each time using a different fold as the validation set.

This cross-validation approach provides a robust estimate of model performance by reducing the variability that might arise from a single train-test split. It also helps in detecting any potential overfitting or underfitting issues, offering a more reliable comparison between the candidate models.

Hyperparameter Tuning

Hyperparameter tuning is a critical step in the model development process, particularly for complex models like Random Forest and XGBoost. While the model’s default parameters often provide a reasonable starting point, they are rarely optimal for a specific dataset and problem.

Without proper tuning, a model might underperform, either by overfitting the training data (leading to poor generalization) or by being too simplistic (failing to capture the underlying patterns in the data). Therefore, hyperparameter tuning is essential to unlocking the full potential of a model, ensuring that it is both accurate and generalizes well to unseen data.

Bayesian Optimization for Hyperparameter Tuning

For our hyperparameter tuning process, we opted for Bayesian Optimization, a sophisticated approach that provides several advantages over traditional methods like Grid Search and Random Search. Unlike these methods, which either exhaustively search a predefined grid of parameters or randomly sample from the parameter space, Bayesian Optimization builds a probabilistic model of the objective function, predicting how changes in hyperparameters will affect model performance.

This approach balances exploration (searching new regions of the hyperparameter space) and exploitation (focusing on regions known to perform well), allowing it to efficiently converge on the optimal set of hyperparameters. This efficiency is particularly important given the computational cost of training models like Random Forest and XGBoost, as it reduces the time and resources required to find the best parameters.

Hyperparameter Tuning for Random Forest Classifier (RFC)

For the Random Forest Classifier, we started with the following default parameters:

- **N_estimators:** 100
- **Min_samples_split:** 2
- **Min_samples_leaf:** 1

Given these defaults, we explored a range of hyperparameters to optimize the model:

- **N_estimators:** (50, 200)
- **Min_samples_split:** (2, 20)
- **Min_samples_leaf:** (1, 10)

The key hyperparameters we tuned included:

- **N_estimators:** The number of trees in the forest.
- **Max_depth:** The maximum depth of the trees.
- **Min_samples_split:** The minimum number of samples required to split an internal node.
- **Min_samples_leaf:** The minimum number of samples required to be at a leaf node.

After running the Bayesian Optimization process, the best hyperparameters found were:

- **Max_depth:** 50
- **N_estimators:** 200
- **Min_samples_split:** 2
- **Min_samples_leaf:** 1

The optimization process took approximately 3 hours.

Hyperparameter Tuning for XGBoost (XGB)

For the XGBoost model, we began with these default parameters:

- **Learning Rate:** 0.3
- **Max_depth:** 6
- **N_estimators:** 100

The ranges for hyperparameter tuning were:

- **N_estimators:** (50, 125)
- **Max_depth:** (4, 7)
- **Learning_rate:** Real (0.05, 0.2, prior='log-uniform')

The key hyperparameters we tuned were:

- **Learning_rate:** Controls how much the model's weights are adjusted with respect to the loss gradient.
- **Max_depth:** Determines the maximum depth of each tree.
- **N_estimators:** Number of boosting rounds (trees in the ensemble).

The Bayesian Optimization process yielded the following best parameters:

- **Learning Rate:** 0.19
- **Max_depth:** 7
- **N_estimators:** 102

This optimization also took about 4 hours to complete. We ran the balanced encoded dataset through both the base models (with default hyperparameters) and the tuned models (with optimized hyperparameters).

2.5 Feature Reduction

Feature reduction is a crucial step in the machine learning pipeline, especially when dealing with many features. Reducing the number of features can lead to more efficient models that are easier to interpret and faster to train. Moreover, it helps in mitigating the curse of dimensionality, where models with too many features can become overcomplicated, leading to overfitting and poor generalization to new data.

Feature reduction also enhances model performance by eliminating redundant or irrelevant features that do not contribute significantly to the model's predictive power. This process can lead to simpler models that generalize better to unseen data, improving the model's ability to make accurate predictions.

Forward and Backward Feature Reduction

To identify the most relevant features for our models, we employed both forward and backward feature reduction techniques:

Forward Feature Reduction: This method starts with an empty set of features and adds features one by one based on their contribution to the model's performance. It continues adding features until no significant improvement is observed.

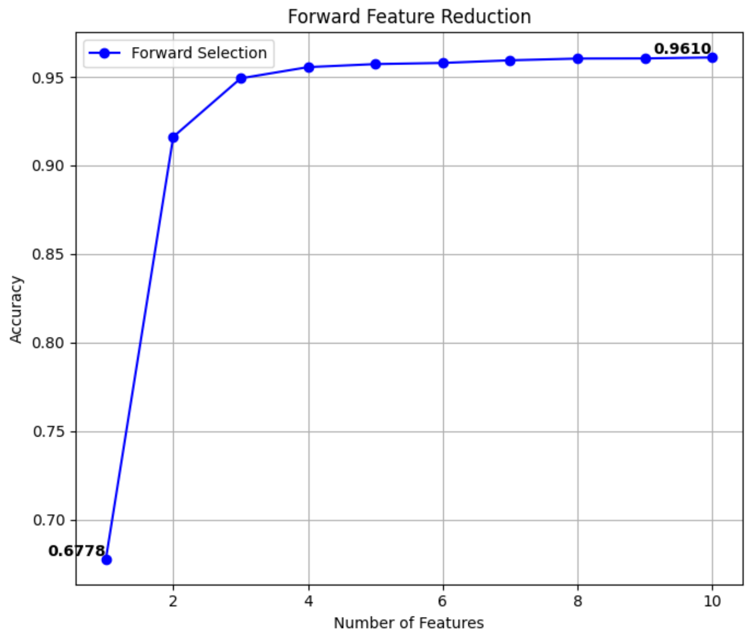
Backward Feature Reduction: Conversely, this method begins with the full set of features and systematically removes the least important ones. Features are removed one at a time until the model's performance no longer improves.

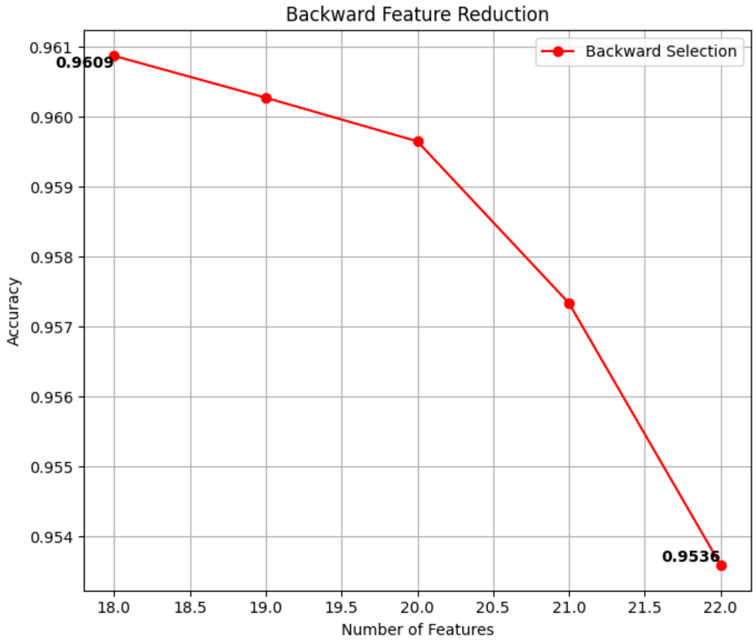
These techniques were applied to both the Random Forest Classifier (RFC) and XGBoost (XGB) models for different datasets to optimize their performance.

Postoperative Model (RFC)

Forward Feature Reduction: Reduced the total number of features from 23 to 10.

Backward Feature Reduction: Reduced the total number of features from 23 to 18.





Features	Feature Name	Mean Accuracy
1	Primary Procedure Name	0.677
2	BMI	0.916
3	Intraop Minutes	0.949
4	Primary Surgeon ID	0.955
5	Total Panel Default Length	0.957
6	Patient Class	0.958
7	Sex	0.959
8	Primary Service	0.9603
9	Age at Surgery	0.9604
10	Primary Race	0.9609

Accuracy Improvements in Forward Selection

Removed Features	Feature Name	Mean Accuracy
1	Procedure Name	0.954
2	Scheduled?	0.957
3	Procedure Panel	0.959
4	Performed?	0.960
5	Ethnicity	0.961

Accuracy Improvements in Backward Selection

The most notable features identified were:

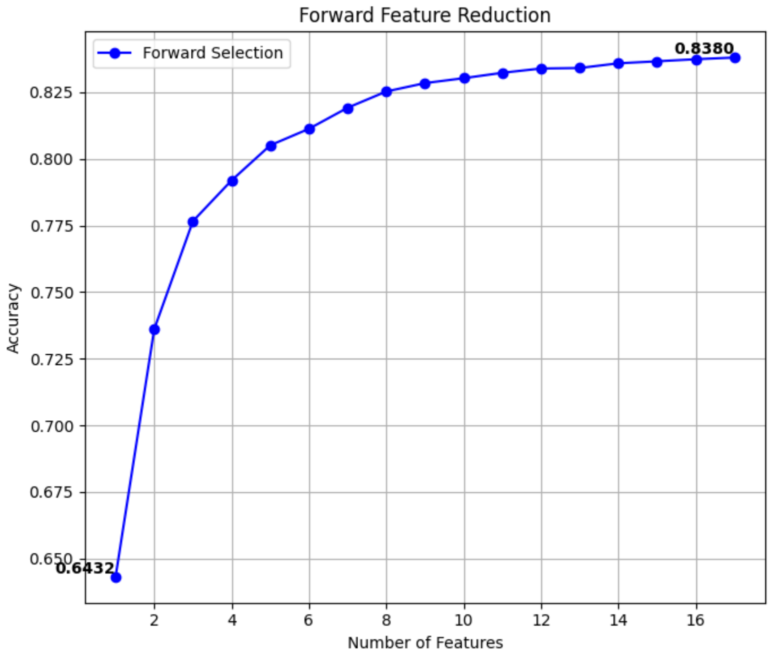
- Primary Procedure Name
- BMI
- Intraop Minutes
- Primary Surgeon ID
- Patient Class
- Total Panel Default Length
- Sex
- Primary Service

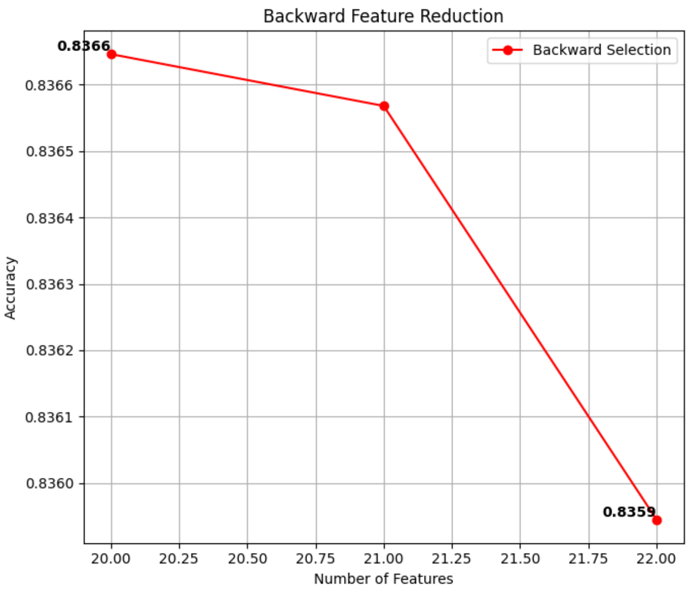
With these features, the RFC model for postoperative predictions achieved an accuracy close to ***96%.***

Postoperative Model (XGB)

Forward Feature Reduction: Reduced the total number of features from 23 to 17.

Backward Feature Reduction: Reduced the total number of features from 23 to 20.





Removed Features	Feature Name	Mean Accuracy
1	Procedure Name	0.836
2	Procedure Panel	0.837
3	Anesthesia Type	0.837

Accuracy Improvements in Backward Selection

Features	Feature Name	Mean Accuracy
1	Primary Procedure Name	0.6431
2	Intraop Minutes	0.736
3	Primary Surgeon ID	0.776
4	Patient Class	0.791
5	Total Panel Default Length	0.805
6	Primary Service	0.811
7	Age at Surgery	0.819
8	BMI	0.825
9	Total Scheduled Panel Length	0.828
10	Robotic Case?	0.830
11	ASA Status	0.832
12	Primary Race	0.833
13	Surgery Diagnosis Name	0.834
14	Anesthesia Type	0.835
15	Sex	0.836
16	Scheduled Rood Duration	0.837
17	Procedure Panel	0.838

Accuracy Improvements in Forward Selection

Key features identified included:

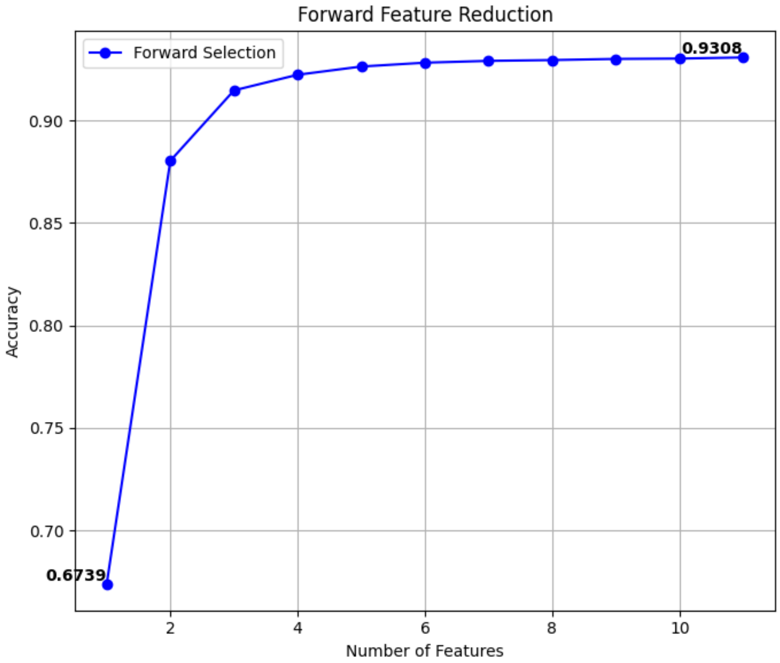
- Primary Procedure Name
- Intraop Minutes
- Primary Surgeon ID
- Patient Class
- Total Panel Default Length
- Primary Service
- Age
- BMI
- Total Scheduled Panel Length
- Robotic Case?

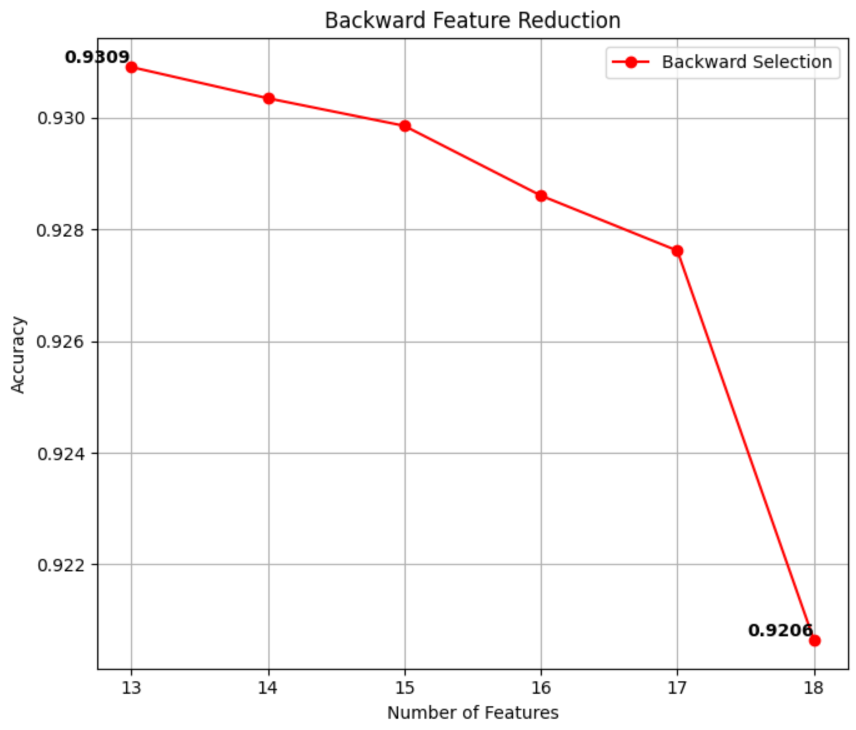
With these features, the XGB model for postoperative predictions achieved an accuracy close to ***84%.***

<u>Preoperative Model (RFC)</u>

Forward Feature Reduction: Reduced the total number of features from 19 to 11.

Backward Feature Reduction: Reduced the total number of features from 19 to 13.





Removed Features	Feature Name	Mean Accuracy
1	Procedure Name	0.9206
2	Panel Primary ID	0.9276
3	Performed?	0.9286
4	Anesthesia Type	0.9299
5	Primary Race	0.9303
6	Procedure Panel	0.9309

Accuracy Improvements in Backward Selection

Features	Feature Name	Mean Accuracy
1	Primary Procedure Name	0.674
2	BMI	0.880
3	Scheduled Room Duration	0.915
4	Primary Surgeon ID	0.922
5	Surgery Diagnosis Name	0.926
6	Age at Surgery	0.928
7	Total Panel Default Length	0.9291
8	Primary Service	0.9295
9	ASA Status	0.9300
10	Total Scheduled Panel Length	0.9302
11	Ethnicity	0.9308

Accuracy Improvements in Forward Selection

The most notable features were:

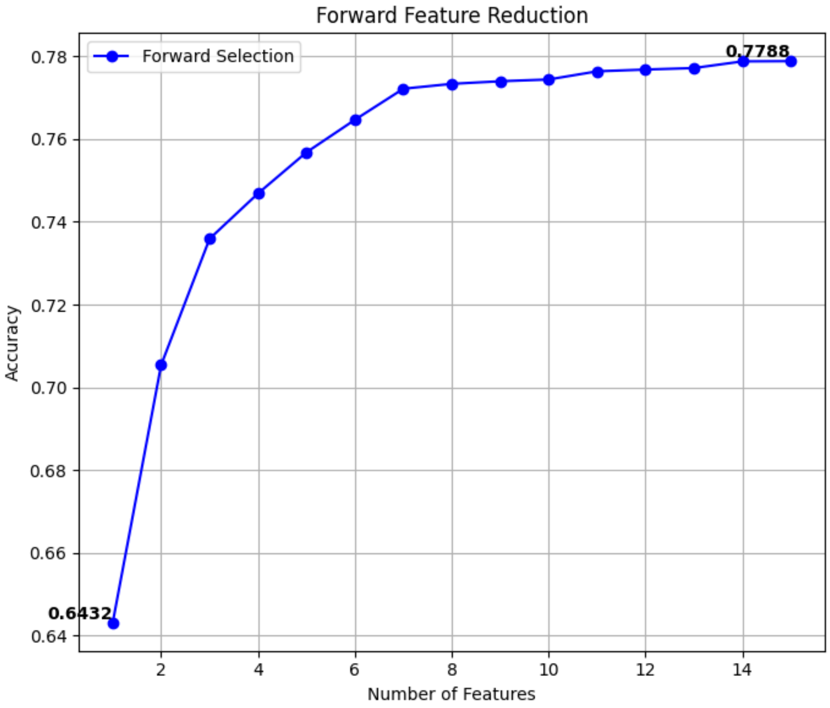
- Primary Procedure Name
- BMI
- Scheduled Room Duration
- Primary Surgeon ID
- Surgery Diagnosis Name
- Age
- Total Panel Default Length

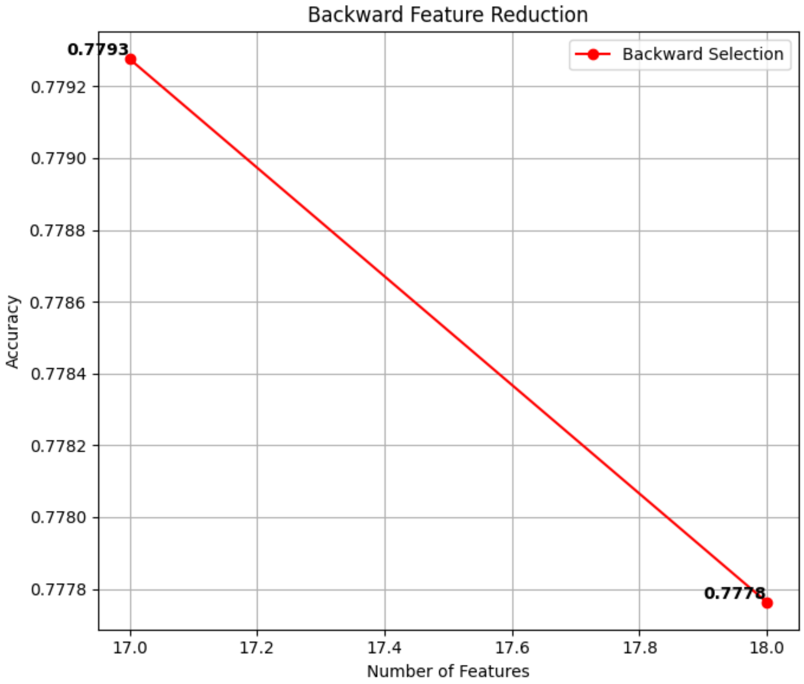
With these features, the RFC model for preoperative predictions achieved an accuracy close to ***93%***.

Preoperative Model (XGB)

Forward Feature Reduction: Reduced the total number of features from 19 to 15.

Backward Feature Reduction: Reduced the total number of features from 19 to 17.





Features	Feature Name	Mean Accuracy
1	Primary Procedure Name	0.6431
2	Total Panel Default Length	0.705
3	Primary Surgeon ID	0.735
4	Scheduled Rood Duration	0.746
5	Primary Service	0.756
6	BMI	0.764
7	Age at Surgery	0.772
8	Primary Race	0.773
9	Sex	0.7739
10	Surgery Diagnosis Name	0.774
11	ASA Status	0.7763
12	Total Scheduled Panel Length	0.7767
13	Robotic Case?	0.777
14	Anesthesia Type	0.778
15	Procedure Panel	0.779

Accuracy Improvements in Forward Selection

Removed Features	Feature Name	Mean Accuracy
1	Procedure Name	0.777
2	Panel Primary ID	0.779

Accuracy Improvements in Backward Selection

Key features included:

- Primary Procedure Name
- Total Panel Default Length
- Primary Surgeon ID
- Scheduled Room Duration
- Primary Service
- BMI
- Age
- Primary Race
- Sex
- Surgery Diagnosis Name

With these features, the XGB model for preoperative predictions achieved an accuracy close to ***78%.***

Results and Evaluations:

Summary of Post OP results

	Model		Accuracy	Standard Deviation	95% Confidence Interval	Recall	F1 Score
Post - OP	RFC	Base	93.43%	0.0050	[93.428211, 93.431789]	93%	93%
		Tuned	87.63%	0.0026	[87.629070, 87.630930]	88%	88%
	XGB	Base	82.46%	0.0041	[82.458533, 82.461467]	83%	83%
		Tuned	83.60%	0.0057	[83.597960, 83.602040]	84%	84%
	RFC with Selected Features (Forward Selection)	Base	96.18%	0.0037	[96.178676, 96.181324]	96%	96%
		Tuned	95.24%	0.0039	[95.238604, 95.241396]	95%	95%
	XGB with Selected Features (Forward Selection)	Base	82.84%	0.0021	[82.839249, 82.840751]	84%	84%
		Tuned	83.09%	0.0036	[83.088712, 83.091288]	84%	84%

Recommendation: RFC, with 10 features after forward feature selection to predict the LOS target features, has 96.18% accuracy as compared to all the other approaches.

Summary of Pre-OP results:

	Model		**Accuracy**	**Standard Deviation of 5-Fold**	**95% Confidence Interval**	**Recall**	**F1 Score**
Pre - OP	RFC	Base	89.39%	0.0049	[89.388247, 89.391753]	89%	89%
		Tuned	89.21%	0.0017	[89.209392, 89.210608]	89%	89%
	XGB	Base	78.49%	0.0038	[78.488640, 78.491360]	78%	78%
		Tuned	78.64%	0.0022	[78.639213, 78.640787]	79%	79%
	RFC with Selected Features (Forward Selection)	Base	**93.29%**	0.0045	[93.288390, 93.291610]	93%	93%
		Tuned	89.21%	0.0039	[89.208604, 89.211396]	89%	89%
	XGB with Selected Features (Forward Selection)	Base	77.72%	0.0017	[77.719392, 77.720608]	78%	78%
		Tuned	77.90%	0.0051	[77.898175, 77.901825]	78%	78%

Recommendation: RFC, with 11 features after forward feature selection to predict the LOS target features, has 93.29% accuracy as compared to all the other approaches.

Discussion

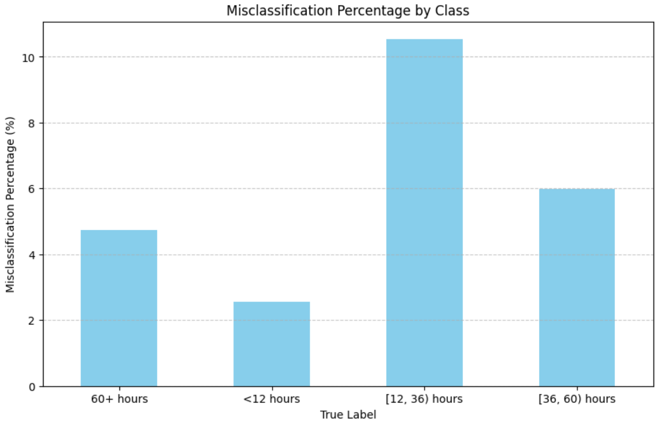
4.1 Misclassification Analysis

Understanding Misclassification

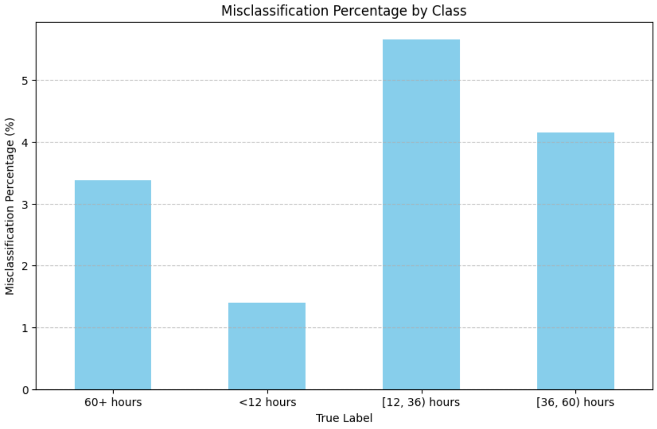
Once the models were trained and evaluated, we delved deeper into the performance by examining misclassifications. Misclassification analysis helps in understanding where the models struggle, and which classes are most frequently confused with others.

Visualization of Misclassification

To visualize the performance of the models, we plotted the misclassification percentage against the misclassification percentage per class. This provided insights into which classes were most frequently misclassified and allowed us to identify patterns and areas for improvement.



RFC Classifier



XGB Classifier

Our analysis revealed that the class ***[12, 36)*** hours was particularly problematic, with a high percentage of misclassifications in both the XGBoost (XGB) and Random Forest Classifier (RFC) models. This indicated that both models had difficulty distinguishing instances within this class from other classes, suggesting a potential area for further investigation and model improvement.

We further investigated the misclassification patterns by calculating the count of the most common misclassified classes within each true class. This breakdown helped in identifying specific classes where misclassifications were most prevalent, providing a clearer picture of the model's weaknesses.

True Class	Most Common Misclassified Class	Count of Misclassifications
60+ hours	[36, 60) hours	48
<12 hours	[12, 36) hours	19
[12, 36) hours	[36, 60) hours	61
[36, 60) hours	[12, 36) hours	37

XGB Classifier

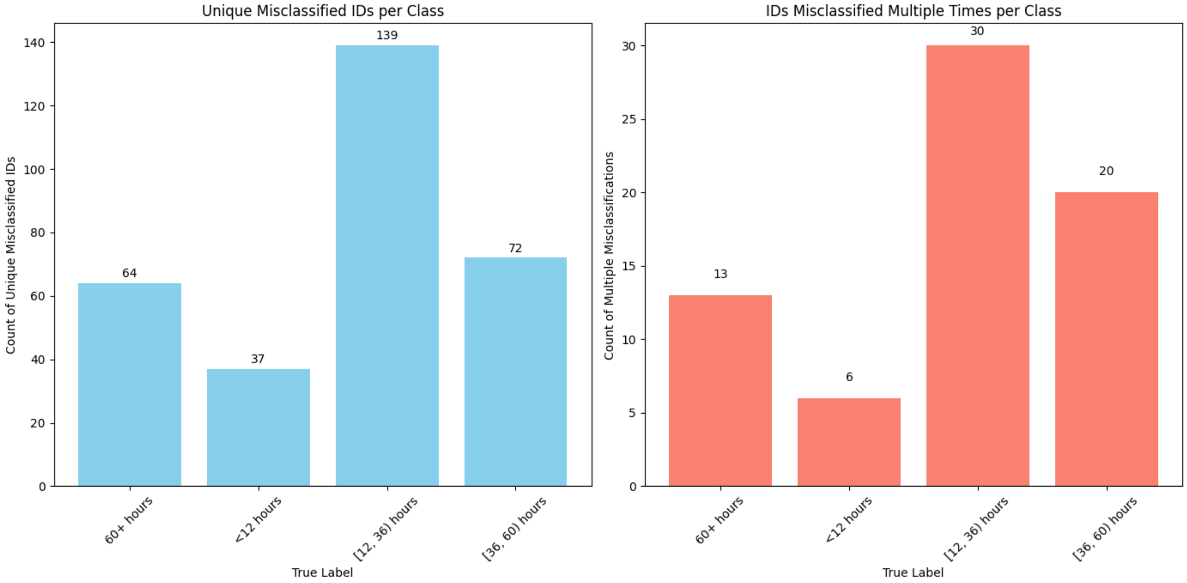
The above table shows Instances that truly belong to the 60+ hours class were frequently misclassified as [36, 60) hours and so on. Similarly for RFC,

True Class	Most Common Misclassified Class	Count of Misclassifications
60+ hours	[36, 60) hours	48
<12 hours	[36, 60) hours	19
[12, 36) hours	60+ hours	61
[36, 60) hours	60+ hours	37

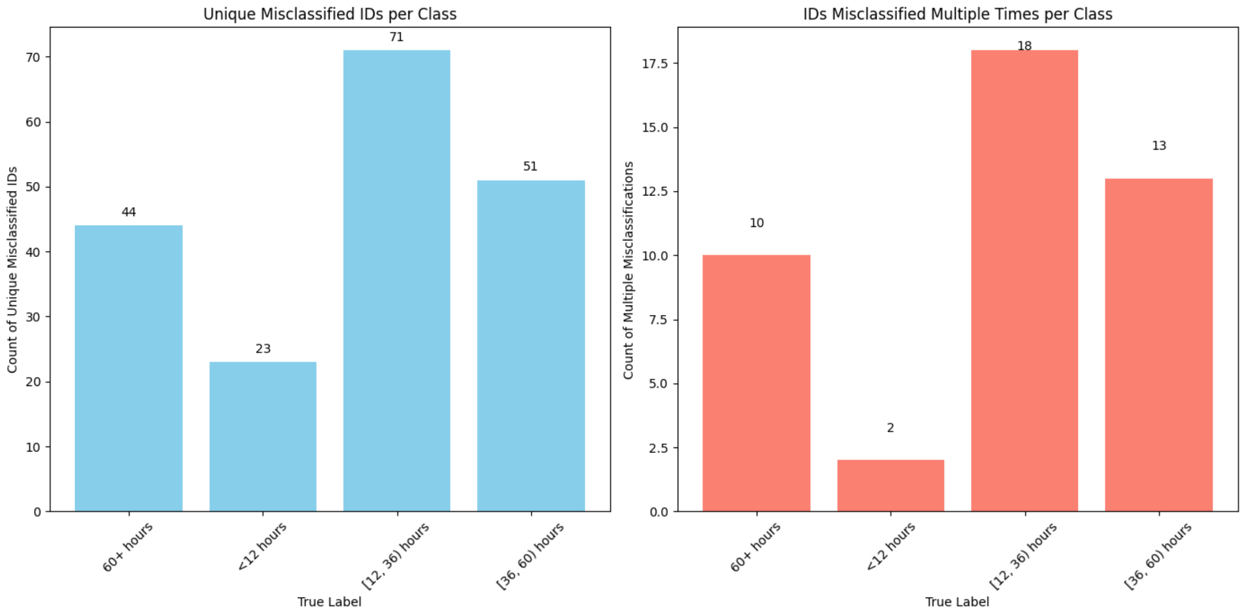
RFC Classifier

The above table shows Instances that truly belong to the 60+ hours class were frequently misclassified as [36, 60) hours and so on.

In addition to understanding the overall misclassification patterns, we aimed to delve deeper by identifying the specific instances (IDs) that were misclassified. This step was essential for uncovering any patterns or commonalities among the misclassified instances, which could indicate underlying issues with the model or data.



RFC Classifier

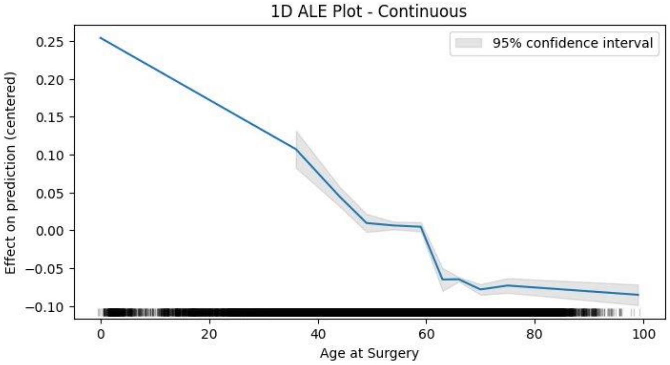


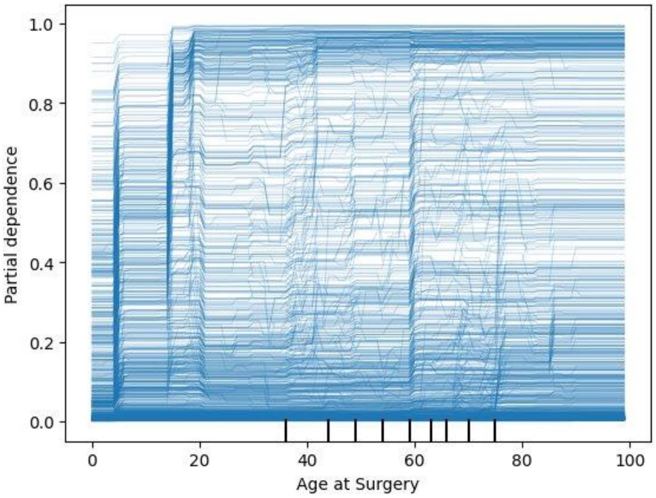
XGB Classifier

To gain a clearer understanding of how often individual IDs were misclassified, we created two bar graphs:

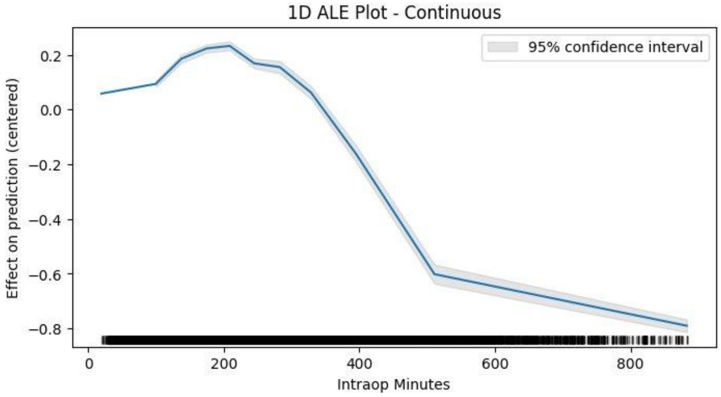
- **IDs Misclassified Only Once:** The first bar graph shows the number of unique IDs that were misclassified only once in each class. This graph helps to highlight the instances where the model made an occasional error, which might be due to subtle data anomalies or borderline cases between classes.
- **IDs Misclassified Multiple Times:** The second bar graph displays the count of IDs where misclassification occurred multiple times within each class. This graph is particularly insightful as it reveals persistent issues with certain IDs, indicating that these instances may consistently confuse the model. Such repeated misclassifications could suggest a need for further investigation into those specific data points.

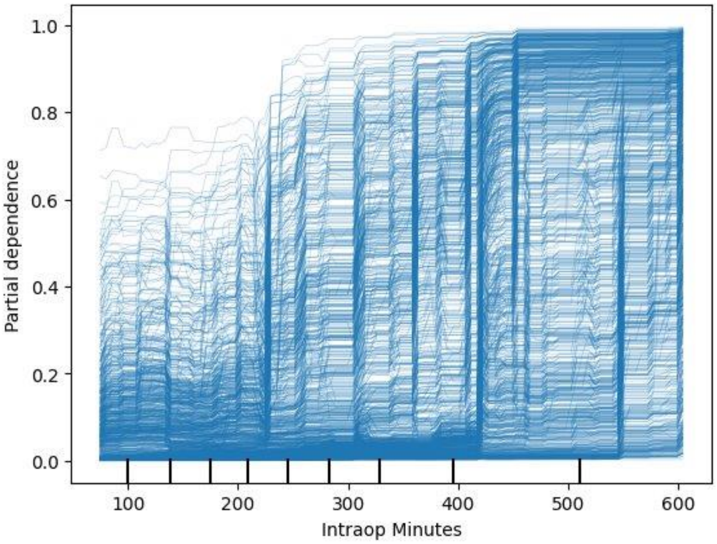
4.2 Sensitivity Analysis – Numerical Features



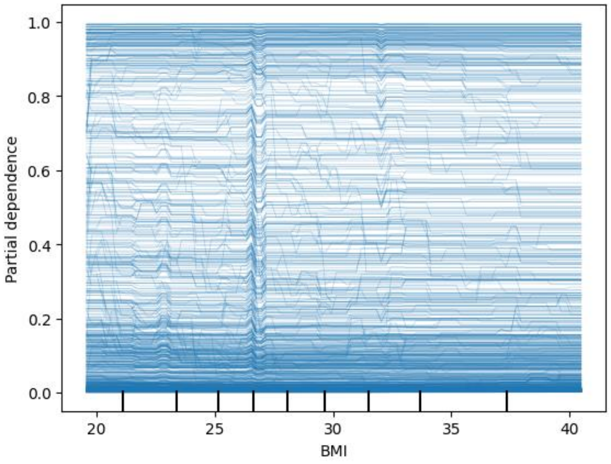


Age at Surgery affects the prediction for LOS group who have an age above 20 very minimally.



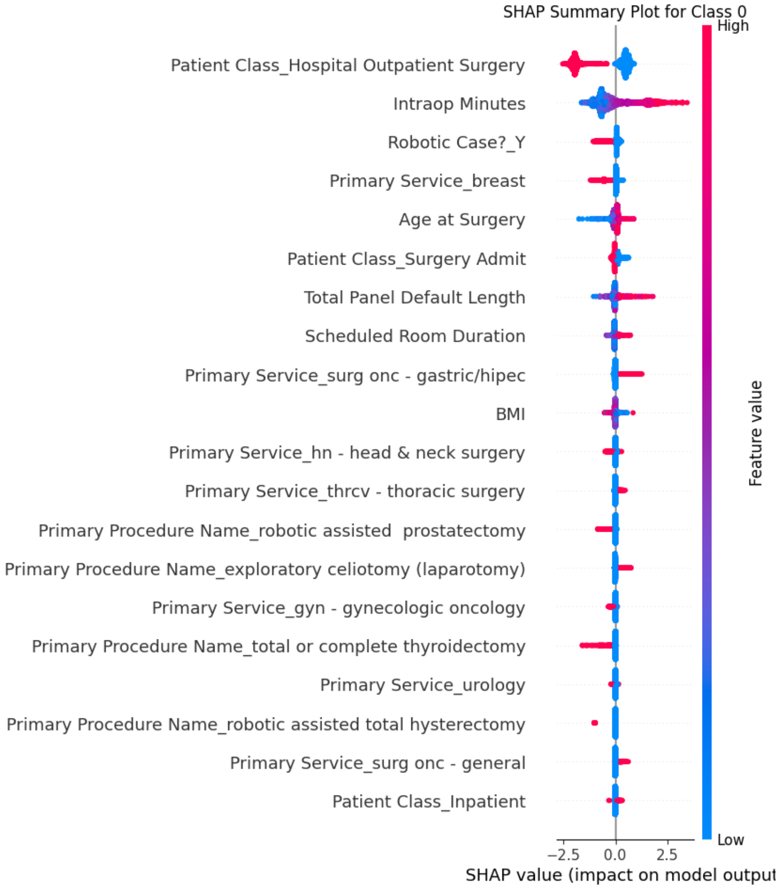


Intraop Minutes affect the prediction for LOS group when the Intraop Minutes is less than 200 minutes (about 7 hours).



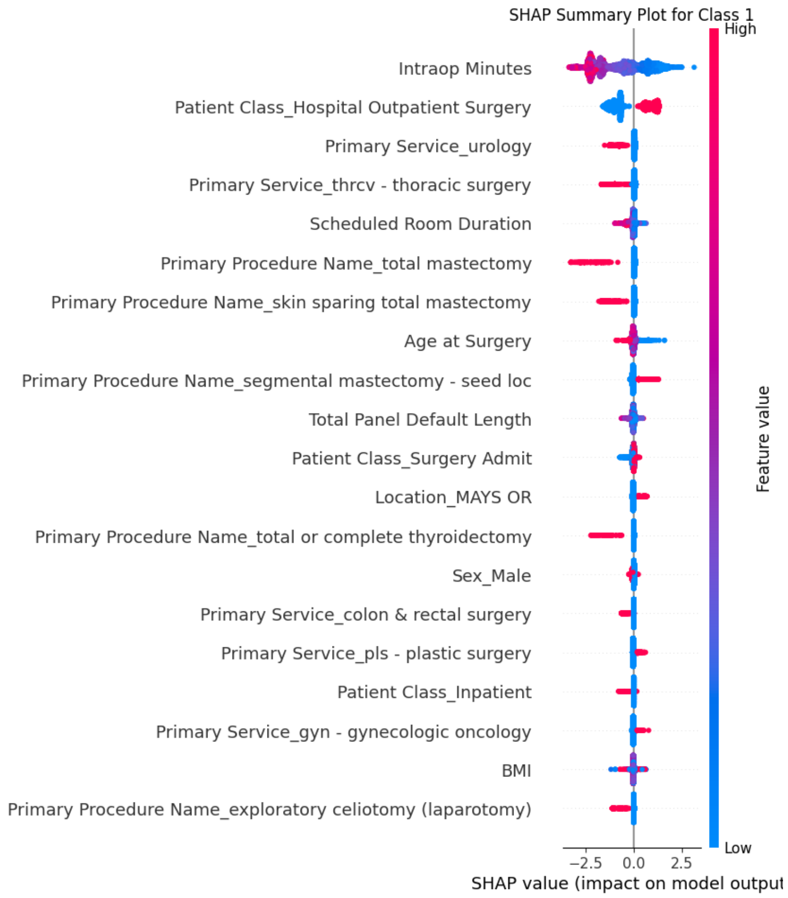
BMI is inconclusive.

SHAP Analysis for each Target Class after Encoding for the POST – OP Features for XGB Classifier



Important Features after Prediction for LOS 60+ hours:

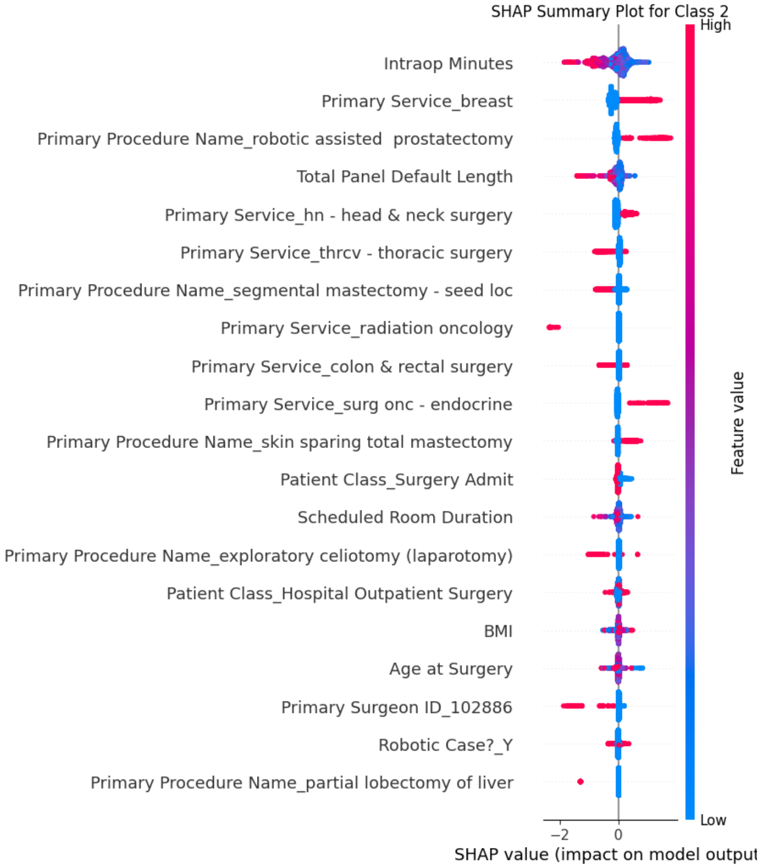
1. Patient Class
2. IntraOP Minutes
3. Robotic Case
4. Primary Service
5. Age at Surgery
6. Patient Class
7. Total Panel Default Length
8. Scheduled Room Duration
9. BMI
10. Primary Procedure Name



Important Features after Prediction for LOS <12 hours:

1. IntraOP Minutes
2. Patient Class
3. Primary Service
4. Scheduled Room Duration
5. Primary Procedure Name
6. Age at Surgery
7. Total Panel Default Length
8. Patient Class
9. Location
10. Sex
11. BMI

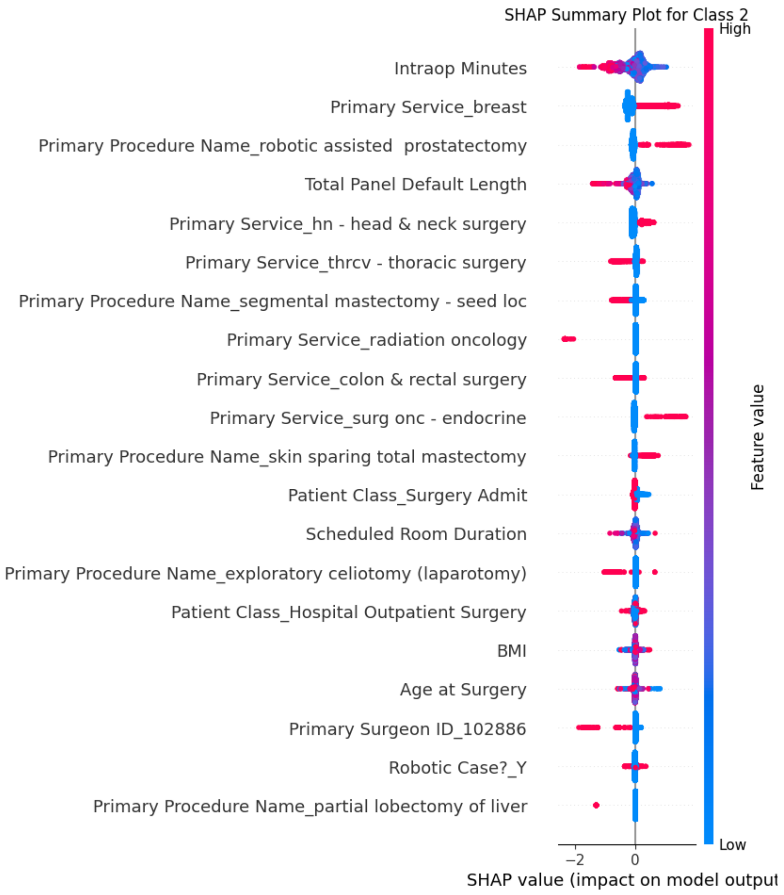
Missing: Robotic Case **New:** Location, Sex



Important Features after Prediction for LOS [12,36) hours:

1. IntraOP Minutes
2. Patient Class
3. Primary Procedure Name
4. Total Panel Default Length
5. Primary Service
6. Scheduled Room Duration
7. Patient Class
8. BMI
9. Age at Surgery
10. Robotic Case
11. Primary Surgeon ID

Missing: Location, Sex, Robotic Case? **New**: Primary Surgeon ID



Important Features after Prediction for LOS [36,60) hours:

1. IntraOP Minutes
2. Primary Service
3. Primary Procedure Name
4. Total Panel Default Length
5. Primary Service
6. Patient Class
7. Scheduled Room Duration
8. BMI
9. Age at Surgery
10. Primary Surgeon ID
11. Robotic Case?

Missing: Sex, Location

Conclusion

1. Random Forest Classifier, with 10 essential features with post-op as well as pre-op provides the best prediction for LOS groups.
2. However, if experiments need to be performed only on Pre – OP features, Random Forest Classifier has the most accurate prediction for LOS groups.
3. Prediction is affected by the Numerical Features – Age at Surgery and Intra OP minutes significantly.
4. Tests such as Random State did not change the accuracy of prediction indicating no overfitting and bias within the model.
5. SHAP Analysis provides key features for each target class and indicates that around 12 unique features are needed to accurately predict the target class.

Future Steps

1. Textual Columns instead of using one-hot encoding could be vectorized using transformers such as Clinical Bert preventing the explosion of features.
2. Categorical Columns such as Procedure Panel and Primary Service which impact heavily as seen from SHAP need to be explored further.
3. Accurately predict the actual number of minutes (regression) instead of classification.
4. Hyper Parameter Tuning of XGB indicates better performance which brings about the question on what the best depth is and whether RFC, XGB is overfitting the data.

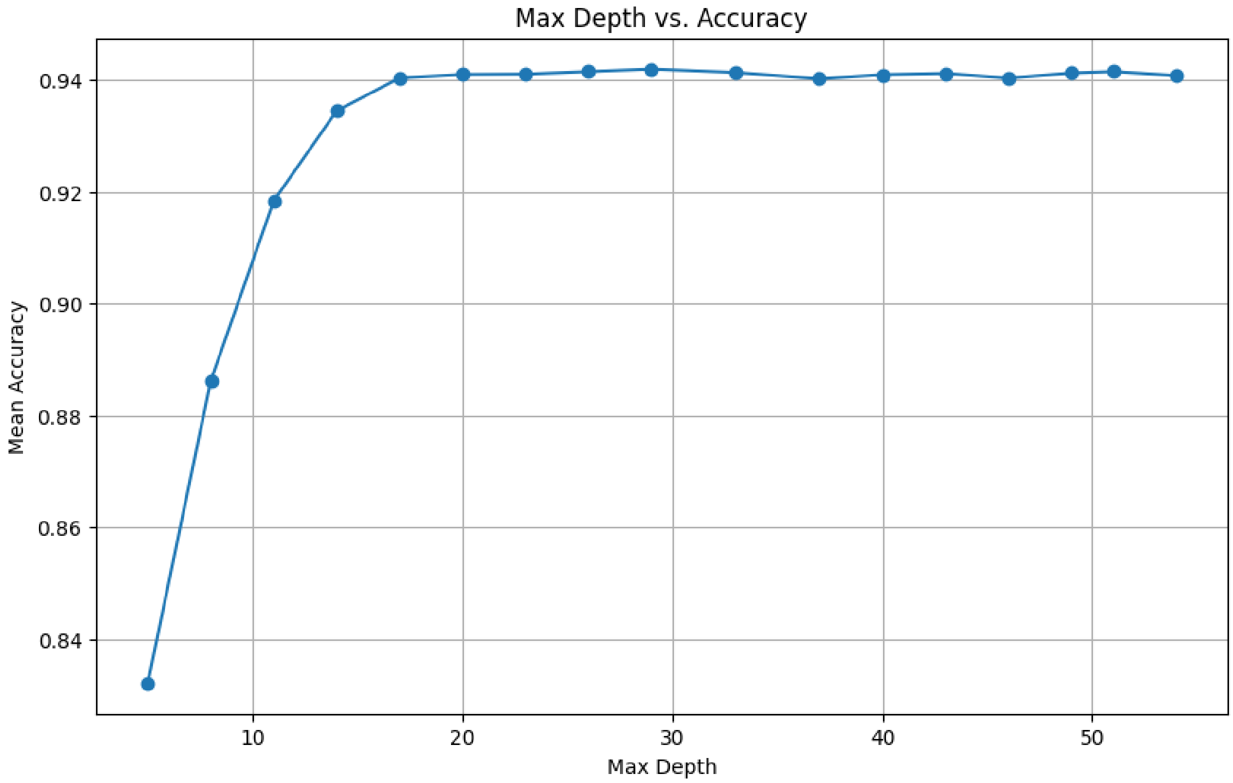
In our quest to build the most accurate classification model, we initially focused on two strong contenders: the Random Forest Classifier (RFC) and XGBoost (XGB). Given the power and flexibility of XGB, it was reasonable to expect that it would perform at least as well as RFC, if not better. However, our initial results pointed us toward recommending RFC as the preferred model.

To ensure we were making the most of XGB's capabilities, we decided to revisit our hyperparameter tuning process. Specifically, we reconsidered the maximum depth of the trees in the XGB model.

In our initial experiments, we set the maximum depth for XGB to 7, while RFC was allowed to explore a maximum depth of 50. This disparity in the depth ranges could have been a limiting factor for XGB. After our final presentation, Aaron suggested that we try increasing the maximum depth for XGB. Taking this advice to heart, we reran our hyperparameter tuning, this time allowing XGB to reach a maximum depth of 50. To our delight, this adjustment allowed XGB to outperform RFC. It became clear that the default max depth of 6 in XGB, which guided our initial tuning range, may have been too conservative.

Disclaimer

We conducted the experiment of varying the max depth for XGBoost after our final presentation, which led to the discovery that XGB could outperform RFC when allowed a greater maximum depth.



XGB Classifier

As we continue to refine our models, one area of potential exploration is overfitting, particularly given the increased complexity allowed by deeper trees. We plan to investigate this further to ensure that the improved performance we've observed with XGB is not at the cost of generalization to unseen data.