Towards Personalized Perioperative Care: Predicting Duration of Hospitalization and Postoperative Complications

Group Members

- **M V N S H Praneeth** - Person Number: 50592326
- **Akhil V S S G** - Person Number: 50606819
- **Abdul Wasi Lone** - Person Number: 50609995

Dataset Overview

1. We aim to address how our method can be useful in cutting costs for the patients and increasing profit for the hospitals due to the optimal allocation of resources.
2. Proper anticipation of the duration of hospitalization, type of medication, and the anesthesia type is of utmost relevance to patients, hospitals, and insurance companies, making it highly significant.
3. The problem has a rich background since the dataset we used is an exhaustive medical dataset (MOVER: Medical Informatics Operating Room Vitals and Events Repository) from UC Irvine, containing hospitalization data of 58,799 patients and 83,468 surgeries (approx. 500 GB). Link to paper: Mover Dataset.
4. Login credentials for the dataset: Dataset site. Username: `akhilven`
   Password: `fJqZSWLBO2VQ3IAqF`
5. The integration of AI into healthcare settings ultimately improves patient outcomes (like length of stay) and enables data-driven personalized healthcare, making the work viable. Moreover, optimized resource management is discussed in the Project Document.

Dataset Usage Instructions

To ensure that the provided Jupyter Notebooks execute correctly, please use the corresponding datasets for each notebook as detailed below.

Datasets for Akhil's Models

For the notebooks:

- **Model_Training_akhil_Q_1_XGBoost**
- **Model_Training_akhil_Q_2_CatBoost**

Please use the datasets provided in this Google Drive link: Datasets for Akhil's Models

Datasets for Praneeth's Models

For the notebooks:

- **M V N S H Praneeth_Phase 2 ANN**
- **M V N S H Praneeth_Phase 2 Logistic Regression**

Please use the datasets provided in the same Google Drive link: Datasets for Praneeth's Models

Datasets for Wasi's Models

For the notebook:

- **Wasi_50609995_Models**

Please use the datasets provided in this Google Drive link: Datasets for Wasi's Models

Important Notes

1. **File Path Configuration**: Ensure that you configure the file paths appropriately in the code for the datasets to load properly. Mismatched paths can lead to runtime errors.
2. **Directory Structure**: Organize your datasets in directories according to the provided paths or adjust the paths in the code as needed.
3. **Dataset Compatibility**: Verify that the dataset schema matches the requirements of the specific notebook you are executing.

By following these instructions, you will ensure that the code runs efficiently and effectively without any issues.

Project Overview

This project uses an ensemble approach to predict a patient's length of stay, required medications, and potential postoperative complications. Our goal is to support healthcare providers in improving resource allocation and enhancing patient care by analyzing and predicting these critical outcomes.

Datasets

We utilized the following datasets for our analysis and model training:

1. **Patient_Information.csv** - Contains demographic, procedural, and clinical details about each patient.
2. **Patient_Coding.csv** - Provides diagnostic and procedural codes for patient encounters.
3. **Post_op_complications.csv** - Lists any postoperative complications recorded for patients.

Phase 1: Data Cleaning and Preprocessing Steps for Patient_Information.csv

1. **Length of Stay (LOS) Calculation:** Calculated the length of stay using `Admission time` and `Discharge time`. Cross-checked with the existing `LOS` column, then dropped `Admission time` and `Discharge time`.
2. **DISCH_DISP:** Removed null values and applied label encoding.
3. **ICU ADMIN FLAG:** Applied label encoding for categorical transformation.
4. **Surgery Date:** Encoded the date and time fields for compatibility with modeling processes.
5. **SEX:** Used label encoding to convert gender data into numeric form.
6. **PRIMARY_ANES_TYPE_NM:** Dropped rows with missing values and encoded them.
7. **ASA_RATING:** Removed rows with missing values and applied label encoding.
8. **PATIENT_CLASS_GRP and PATIENT_CLASS_NM:** Performed label encoding.
9. **PRIMARY_PROCEDURE_NM:** Tokenized values into tensors using BERT for deep learning models.
10. **OR Duration:** Calculated as the time difference between `OR IN TIME` and `OR OUT TIME`.

11. **Anesthesia Duration:** Computed using the time difference between `ANESTHESIA IN TIME` and `ANESTHESIA OUT TIME`.

12. **HEIGHT and WEIGHT:** Detected and removed outliers using box plots. Converted height to meters, filled missing values, and plotted density plots.

Getting Started

To run this project:

1. Clone the repository.

2. Install all dependencies.

3. Follow the preprocessing steps outlined above.

Requirements

- Python libraries: Pandas, NumPy, Scikit-Learn, TensorFlow/PyTorch (for BERT tokenization), Matplotlib/Seaborn (for plotting).

Project Phase-2

Akhil Venkata Shiva Sai Gorugantu
Person Number: 50606819

How do the post-operative trends accompany the patients being admitted?

Problem Statement and Dataset Complexity

Question: How do the post-operative trends accompany patients being admitted?

Objective: Understanding post-operative trends is directly linked to patient outcomes. Complications leading to extended stays can affect hospital bed availability, especially in departments like the ICU. Analyzing these trends can assist in predicting patient recovery trajectories and enable hospitals to anticipate patients requiring more attention, thereby helping in resource allocation and patient flow optimization.

Dataset Complexity:

- The dataset contained multiple complex features, including patient-specific identifiers and post-operative condition labels.
- **Data Preparation**:
  1. **Merging Datasets**: Two datasets were merged using a unique key combining `MRN` (patient unique identifier) and `Log_ID`, which required careful handling due to duplicate entries for certain patients.
  2. **Cleaning and De-duplication**: There were duplicate records with identical unique keys. I cleaned the data to ensure only unique records for each patient, which required extensive data manipulation.
  3. **Target Labels**: The target variable (`Post_OP_type_Category`) initially had redundant classes (e.g., no entries for "chronic pain"), necessitating adjustments in target labels to ensure correct multiclass classification was possible because the one-hot encoding for the "chronic pain" was completely 0 so, I had to rearrange the classes present in the dataset.
  4. I also had to resample the data present in the dataset as the data distribution of each class varied a lot. Hence, I had to oversample (classes with more samples) or undersample(classes with fewer samples) so that we had a uniform distribution of all the classes in the dataset.

Model Selection: Why XGBoost?

XGBoost was selected as the primary algorithm due to its:

- **Strength in Tabular Data**: XGBoost is well-known for handling structured data, especially when there are complex feature interactions.
- **Feature Importance**: The ability of XGBoost to compute feature importance directly makes it ideal for healthcare applications, where understanding the impact of each feature on the outcome is crucial.
- **Handling Imbalanced Data**: XGBoost has parameters to handle imbalanced datasets, making it effective for scenarios where certain post-operative conditions were underrepresented.

Additionally, the interpretability of the model through feature importance and other evaluation metrics aligns with the project's goal to understand influential factors affecting post-operative patient trends.

Training and Tuning the Model

- **Model Configuration**:
  - The model was configured with the `multi:softprob` objective to handle multiclass classification, which outputs probability distributions across all classes.
  - **Booster**: `gbtree` was used as the booster, ideal for classification problems on structured data.
- **Hyperparameter Tuning**:
  - Tuning involved adjusting the learning rate, maximum tree depth, and number of boosting rounds to prevent overfitting and achieve optimal results.
  - Through cross-validation, we identified the best parameters to ensure the model generalizes well.
- **Training Loss vs. Test Loss**:
  - The loss curves indicate steady learning with both training and validation losses decreasing over boosting rounds, demonstrating effective training without overfitting.

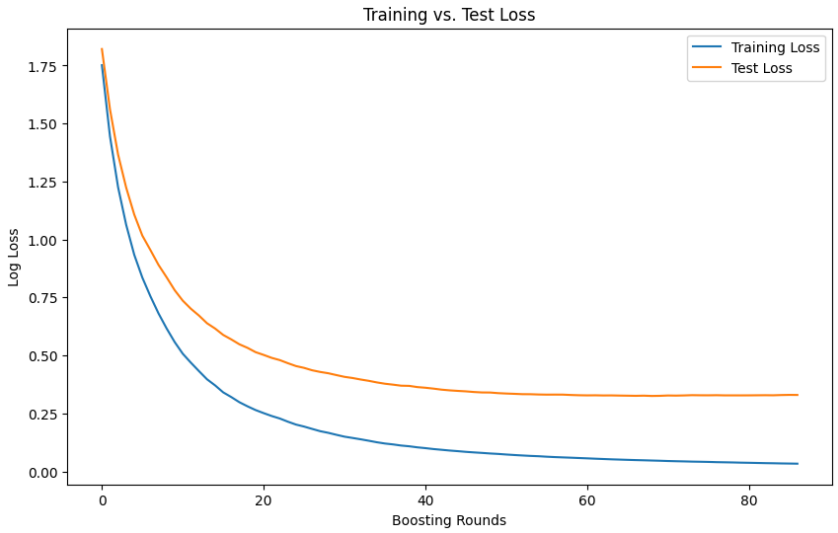


Figure 1: Training vs. Test Loss

Model Evaluation and Insights

To evaluate the XGBoost model's effectiveness, multiple metrics and visualizations were used:

1. **Feature Importance**:
   - The feature importance plot shows that `OR_LOS_HOURS`, `WEIGHT`, `HEIGHT_METRES`, and `LOS` are the most significant features, aligning with medical expectations.
   - From the importance Graph we can see that our hypothesis of the effect of whether a patient is admitted to ICU will affect the Post-operative conditions rather it is dependent on the above conditions.

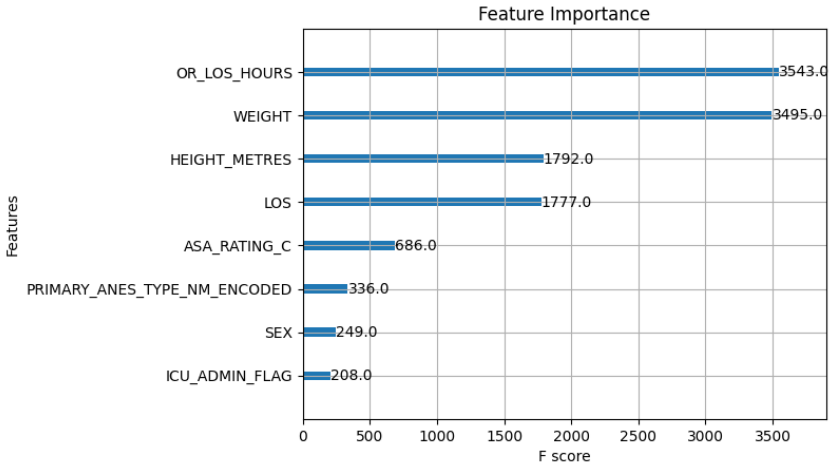


Figure 2: Feature Importance Plot

2. **Correlation Matrix**:
   - The correlation matrix reveals dependencies between features, such as a moderate correlation between `WEIGHT` and `HEIGHT_METRES`.

Figure 3: Correlation Matrix

3. **Precision-Recall Curve**:

- High average precision for most classes indicates effective discrimination, which is crucial for detecting rare post-operative complications.

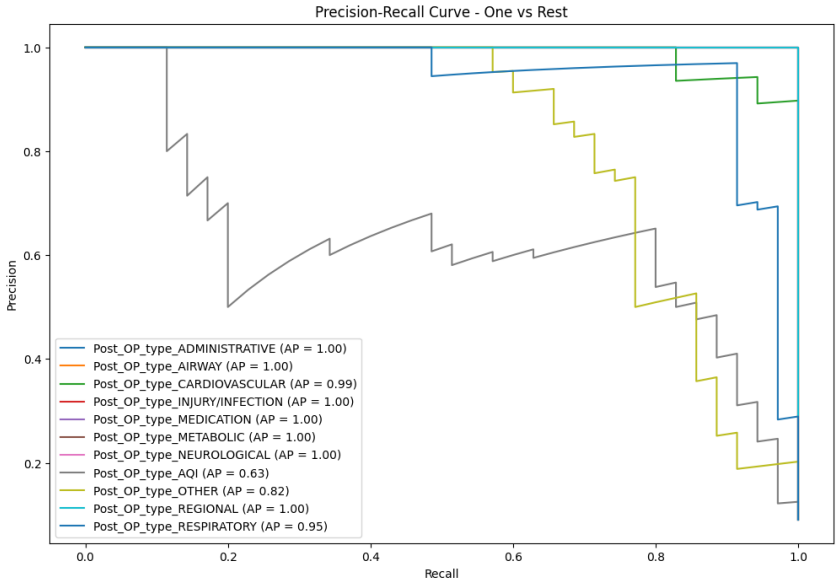


Figure 4: Precision-Recall Curve for Each Class

4. **ROC-AUC Curve**:

- The high AUC values for each class demonstrate strong classification performance across multiple categories.

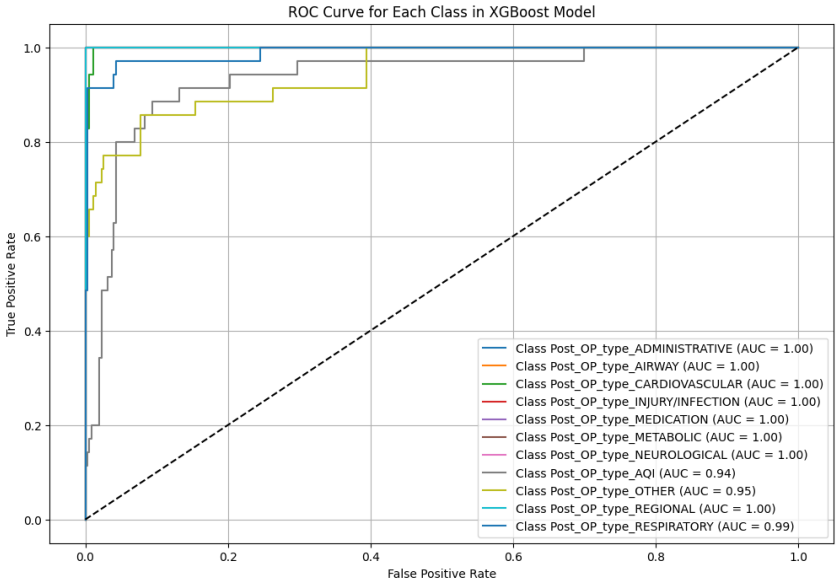


Figure 5: ROC Curve for Each Class in XGBoost Model

5. **Confusion Matrix**:

- The matrix shows high accuracy for dominant classes and highlights areas for potential improvement in underrepresented classes.

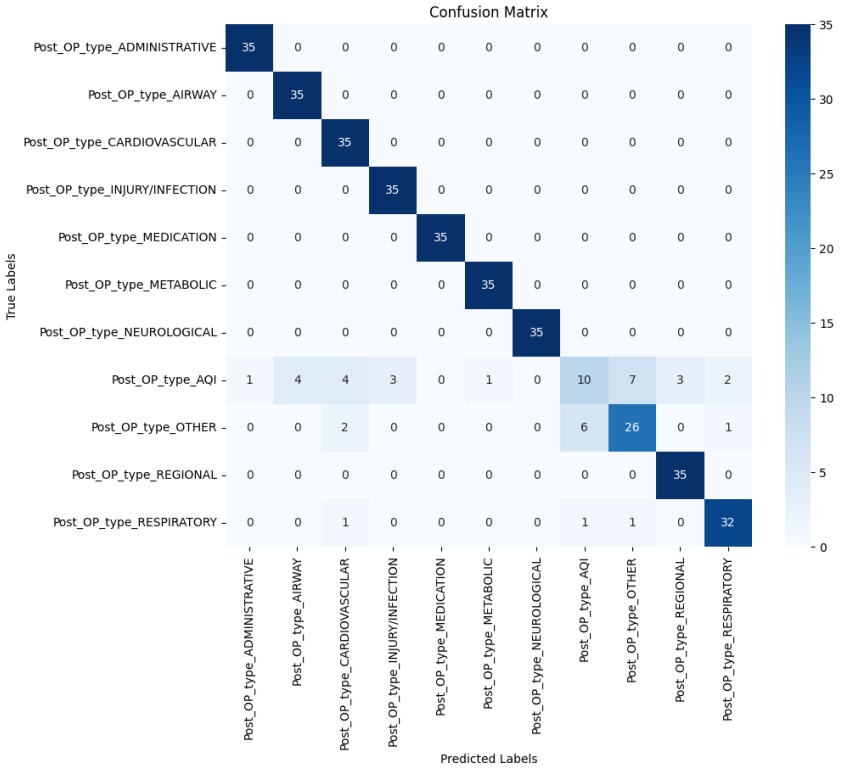


Figure 6: Confusion Matrix for XGBoost Model

Justification of Model Effectiveness

The effectiveness of the XGBoost model was demonstrated through:

- **Robust Multiclass Classification**: The model achieved high precision, recall, and AUC values, proving its suitability for complex multiclass predictions in healthcare.
- **Interpretability**: Feature importance and visualization metrics allowed insights into the most influential factors affecting post-operative outcomes.
- **Adaptability**: XGBoost's ability to handle structured tabular data and imbalanced class distributions made it an ideal choice for this medical dataset.

Conclusion and Insights Gained

The application of XGBoost to this dataset enabled us to derive valuable insights into post-operative trends:

- **Critical Factors**: Factors like `OR_LOS_HOURS` and `WEIGHT` were identified as critical predictors for post-operative outcomes.
- **Resource Optimization**: By predicting likely post-operative complications, hospitals can allocate resources more efficiently.
- **Actionable Predictions**: The model provides a reliable way to flag patients who might need additional post-operative care, aiding in preemptive actions.

The XGBoost model proved to be a powerful tool for analyzing and predicting post-operative trends, providing actionable insights that align well with the goals of enhancing patient outcomes and streamlining hospital resource management.

Analyzing Sex, ICU Admission, and Discharge Disposition

Question 2: How does the Sex of a particular person affect the Discharge disposition (where they go after discharge) and who is admitted to the ICU more times? What is the relation between the Length of Stay vs whether the patient is admitted to the ICU, with the help of the Sex feature?

Model Choice and Justification

For this question, we selected the **CatBoost** model. CatBoost is designed to handle categorical features efficiently without requiring extensive preprocessing like one-hot encoding, which makes it ideal for tabular data containing both categorical and numerical features. Given the categorical nature of variables such as `Sex` and `Discharge Disposition`, and the need to understand complex interactions with ICU admissions and Length of Stay, CatBoost was a suitable choice.

Differences between XGBoost and CatBoost

While both XGBoost and CatBoost are gradient boosting algorithms, they differ in handling categorical data and managing overfitting:

- **Handling Categorical Data**: CatBoost natively supports categorical variables, allowing automatic processing without manual encoding, preserving data structure. XGBoost requires explicit transformations like one-hot encoding.
- **Overfitting Prevention**: CatBoost uses ordered boosting to reduce prediction bias and enhance generalization, while XGBoost primarily relies on regularization techniques like L2 and L1.

CatBoost was therefore chosen for Q2, as it facilitates better interpretation of categorical variables and performs robustly with minimal preprocessing.

Model Training and Tuning

To capture relationships between `Sex`, `ICU_ADMIN_FLAG`, `DISCH_DISP`, and `LOS` (Length of Stay), we incorporated additional features like `WEIGHT`, `HEIGHT_METRES`, `OR_LOS_HOURS`, and `ASA_RATING_C`. This expanded feature set enabled the model to form meaningful connections, improving interpretability and accuracy.
I have truncated the number of classes present in the Discharge Disposition Class as there are many classes which are having very less presence in the dataset (like 2 rows in a dataset of 54k rows) so i truncated the classes to classes which are having rows of above 500 so that if we even sample we will not have samples which are not identical

Key tuning steps included:

- **Iterations**: Setting the number of boosting rounds for optimal convergence.
- **Learning Rate**: Adjusted to balance convergence rate and avoid overfitting.
- **Depth**: Set to capture sufficient complexity while avoiding excessive computation.

Metrics and Effectiveness

The CatBoost model's performance in predicting discharge disposition and understanding ICU admission trends by sex was assessed through various metrics and visualizations. These insights helped in analyzing the effectiveness of the model and deriving actionable intelligence.

- **Confusion Matrix**: The confusion matrix (Figure 7) illustrates the model's predictions against the actual discharge dispositions, offering a clear view of where misclassifications occur. High accuracy for major classes such as routine discharges and specific ICU-related categories highlights the model's ability to distinguish between common discharge outcomes. However, some misclassifications in less frequent categories indicate that further tuning may be needed for rare discharge types, which could be impacted by patient sex and ICU stay patterns.
- **ROC Curve**: The ROC curve (Figure 8) provides a detailed breakdown of the model's ability to distinguish between discharge disposition categories. Each curve represents one discharge type, with high Area Under Curve (AUC) values indicating that the model is effective at correctly predicting each class. The nearly perfect AUC scores for specific classes demonstrate the model's capability to differentiate cases, especially in understanding how patient sex influences outcomes like ICU admission and discharge location. This helps in validating that the model is reliable for decision-making in hospital resource allocation.
- **Training vs. Validation Loss**: The training and validation loss curves (Figure 9) showcase the model's learning progression over iterations. A steady decline in both curves indicates that the model is learning effectively, without significant overfitting. This is crucial for complex medical datasets, as overfitting could lead to inaccurate predictions in real-world applications. The convergence of training and validation losses suggests that the model generalizes well, which is essential for providing reliable predictions in medical settings where patients vary greatly.
- **Feature Importance**: The feature importance plot (Figure 10) identifies key predictors influencing the model's decisions. Notably, `LOS` (Length of Stay) has the highest importance, suggesting it's a strong determinant for discharge disposition and ICU admission patterns. This is followed by `HEIGHT_METRES` and `WEIGHT`, indicating that physical characteristics are also significant in predicting outcomes. The importance of `ICU_ADMIN_FLAG` supports the hypothesis that ICU admission status is directly linked to discharge outcomes, and patient `SEX` further influences these trends, which validates the use of these features in addressing the question.

Insights and Practical Relevance

This analysis provides critical insights for hospital resource planning and personalized patient care:

- **Discharge Disposition Predictions**: Understanding discharge disposition based on sex and ICU admission trends enables hospitals to better anticipate bed availability, particularly for patients likely to require prolonged ICU stays. This helps in managing patient flow and ensuring that resources are allocated effectively.
- **ICU Admission Trends by Sex**: The model's predictions reveal sex-based patterns in ICU admissions and discharge locations, which could be valuable for personalized healthcare strategies. For instance, female and male patients might have differing recovery trajectories and discharge needs post-ICU, necessitating tailored care plans.
- **Resource Allocation and Length of Stay**: Accurately predicting the Length of Stay for ICU patients aids in discharge planning and resource allocation. Given the high importance of `LOS` and `ICU_ADMIN_FLAG` in the feature importance plot, hospitals can make informed decisions about patient management, reducing overcrowding and optimizing turnover.
- **Patient Care Improvements**: By understanding the discharge patterns and ICU admissions, hospitals can enhance patient satisfaction and care quality. For example, high-risk patients identified by the model could receive prioritized care, reducing complications and improving outcomes.

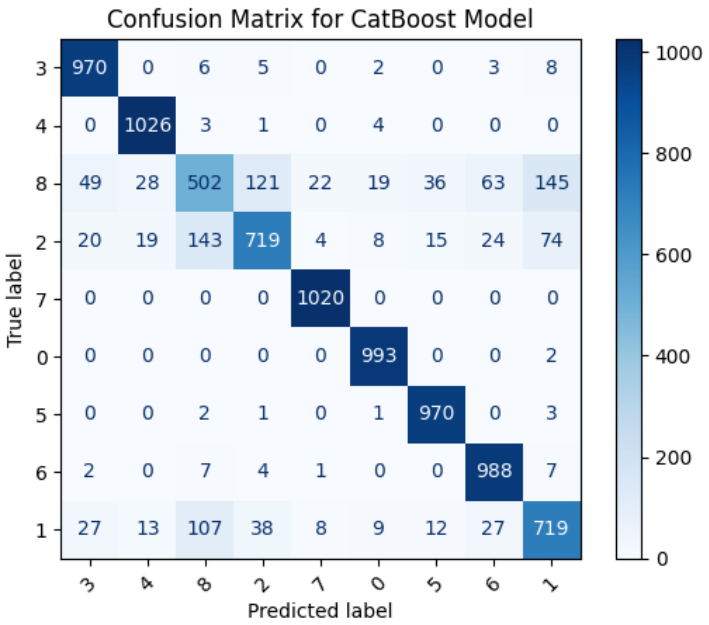


Figure 7: Confusion Matrix for CatBoost Model

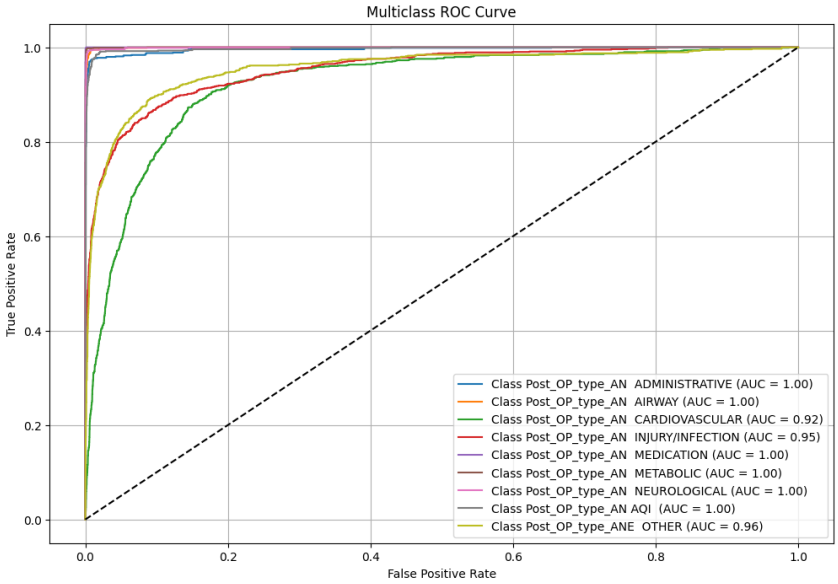


Figure 8: Multiclass ROC Curve for CatBoost Model

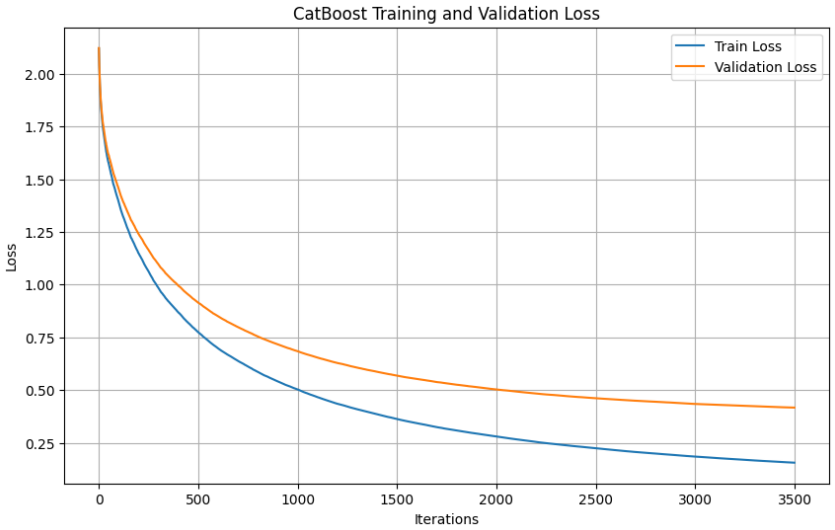


Figure 9: CatBoost Training and Validation Loss

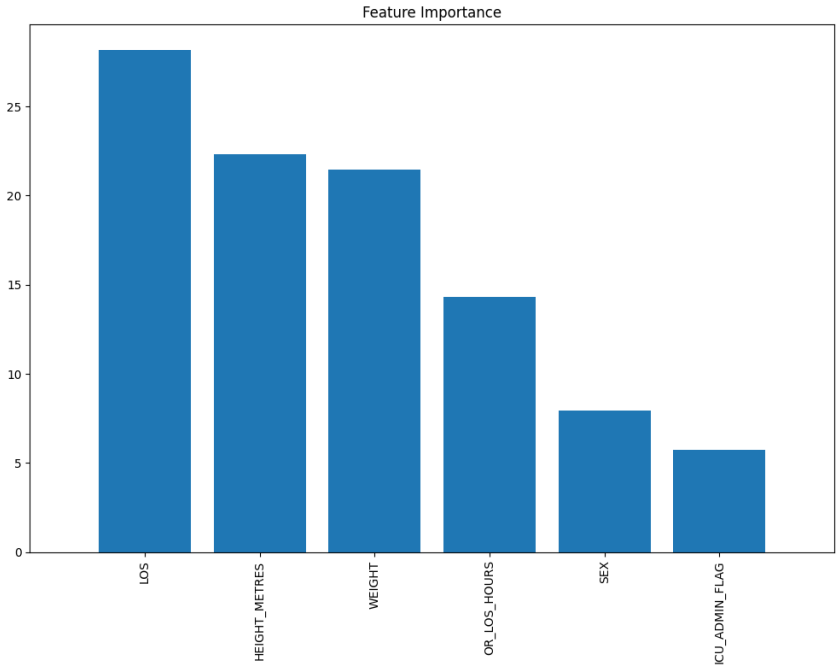


Figure 10: Feature Importance in CatBoost Model

Differences between XGBoost and CatBoost

XGBoost and CatBoost are both powerful algorithms but are fundamentally different in their approach, architecture, and feature handling capabilities. Here's how they stand apart:

- **Handling Categorical Data**: CatBoost has native support for categorical features, allowing it to process them directly without requiring manual transformations like one-hot encoding. This is especially advantageous for datasets with significant categorical data, as it reduces preprocessing complexity and preserves feature relationships. On the other hand, XGBoost lacks native categorical handling and requires explicit transformation, typically through one-hot encoding, which can lead to a larger feature space and potentially slower training times.
- **Algorithmic Approach - Ordered Boosting vs Standard Boosting**: CatBoost employs a unique *ordered boosting* technique that mitigates overfitting by ensuring that each data point's prediction is based on prior observations. This sequential approach helps CatBoost achieve better generalization on unseen data, especially in cases with limited samples. XGBoost, while powerful in standard gradient boosting, lacks this ordered boosting mechanism, making it more susceptible to overfitting in some cases unless additional regularization is applied.
- **Efficiency and Training Speed**: Due to CatBoost's ordered boosting and its efficient handling of categorical variables, it tends to perform faster on datasets with a high proportion of categorical features compared to XGBoost, especially for large datasets with complex categories. XGBoost, while fast, can become computationally expensive with one-hot encoded categorical variables, resulting in longer training times for high-dimensional data.
- **Handling Imbalanced Data**: CatBoost includes mechanisms to handle imbalanced data effectively, such as class weights and ordered boosting, which can help it perform better on minority classes without overfitting. While XGBoost also supports class weighting and other regularization techniques, it may require additional tuning and external techniques like SMOTE for handling significant class imbalances effectively.
- **Interpretability and Feature Importance**: CatBoost provides intuitive feature importance scores, which are easy to interpret due to its handling of categorical data. This can make it easier to gain insights from models trained on mixed data types. XGBoost, while offering feature importance, may present challenges when working with high-dimensional, one-hot encoded categorical features, potentially complicating interpretability.

Viability, Importance, and Profitability of Predictions in Healthcare Settings

Pertaining to the complexity scale defined in Phase 1 of the DIC project, our work addresses several impactful areas, especially in terms of patient outcomes, healthcare economics, and operational efficiency. Below are key aspects demonstrating how these predictions can benefit hospitals, patients, and the healthcare ecosystem as a whole:

1. **Cost-Effective Resource Allocation**:
    - By accurately predicting post-operative trends, including patient length of stay and likely post-operative complications, hospitals can optimize the allocation of critical resources, such as ICU beds, specialized staff, and equipment.
    - This optimized allocation not only minimizes unnecessary resource utilization but also translates to reduced costs for patients. Hospitals can increase profitability by avoiding underutilization or overuse of resources, ultimately cutting operational expenses and increasing patient throughput.
2. **Relevance to Patients, Hospitals, and Insurance Companies**:
    - Anticipating hospitalization duration, medication types, and anesthesia needs is essential for several stakeholders:
        - **Patients** benefit from cost predictability and enhanced care quality.
        - **Hospitals** can better plan for patient flow, ensuring adequate care levels for each patient.
        - **Insurance Companies** can use accurate predictions to design coverage plans and streamline reimbursement processes.
    - These factors make the problem both highly relevant and popular, addressing a real need within the healthcare industry for data-driven patient and resource management.
3. **Rich Background with a Comprehensive Dataset**:
    - This project utilizes the MOVER (Medical Informatics Operating Room Vitals and Events Repository) dataset from UC Irvine, a comprehensive dataset containing hospitalization data for 58,799 patients and 83,468 surgeries. Leveraging such a rich dataset enhances the reliability of our model and supports predictions with a robust foundation of real-world data.
    - Reference to the MOVER Dataset: *[Add citation or link to paper here if possible]*
4. **Advancing AI Integration in Healthcare**:
    - The integration of AI-driven predictions in healthcare settings facilitates data-driven personalized care, improving patient outcomes by anticipating needs and delivering tailored care.
    - Predicting the length of stay and complications allows for better planning, especially in cases requiring specialized post-operative care. This fosters patient-centric healthcare and supports the transition towards evidence-based practices.
    - Optimized resource management, as discussed in Q1, enhances hospital efficiency, reduces patient wait times, and alleviates strain on critical care resources, making AI-driven healthcare systems more effective and sustainable.

Significance of AI-Driven Predictions in Modern Healthcare

In a healthcare landscape increasingly oriented towards efficiency, AI models like ours are instrumental in:

- **Predictive Resource Allocation**: AI allows for the anticipation of resource demands, which is crucial in high-stakes areas like ICU management.
- **Patient-Centric Care**: Predictions on hospitalization duration and post-operative care support personalized healthcare, reducing the chances of complications and improving recovery outcomes.

- **Economic Benefits for Healthcare Providers**: AI optimizes hospital operations, reducing idle resources and allowing hospitals to accommodate more patients without increasing costs, thus enhancing profitability.
- **Scalability and Application**: While our model currently applies to a specific dataset, the approach is scalable across various healthcare settings, with potential applications in other areas such as chronic disease management, emergency care, and hospital admissions forecasting.

In conclusion, the predictive insights generated by our model represent a meaningful advancement in healthcare, improving the alignment of resources to patient needs, enhancing care quality, and providing significant economic benefits to healthcare providers and insurers. This work aligns with current trends in healthcare that prioritize data-driven, patient-centric, and efficient care models.

Project Phase - 2 - M V N S H Praneeth - 50592326

Q1: How do discharge dispositions differ among patients receiving various types of anesthesia?

Answer:

For this project, a deep learning model using a Sequential neural network was chosen due to the complexity of the dataset, which consists of over 65,000 rows. Neural networks are well-suited for handling large datasets with high dimensionality and complex relationships between features, such as those found in the MOVER dataset. The specific architecture includes multiple dense layers with ReLU and SELU activations, which are effective for non-linear problems like classification.

The use of RMSprop as the optimizer was a deliberate choice due to its ability to handle non-stationary objectives and adapt the learning rate dynamically. This is particularly useful in deep learning models where gradients can vary significantly during training.

Several steps were taken to tune and train the model:

- **Feature Scaling**: The features were scaled using *StandardScaler* to ensure that all input data had similar ranges, which is critical for neural networks to converge efficiently.
- **One-Hot Encoding**: The target variable y was one-hot encoded using *to_categorical* to prepare it for multi-class classification.
- **Dropout Layers**: Dropout layers were added to prevent overfitting by randomly dropping units during training. This helps the model generalize better on unseen data.
- **Early Stopping**: Early stopping was implemented with a patience of 5 epochs to avoid overfitting while ensuring that the model does not train unnecessarily for too many epochs.
- **Validation Split**: 20% of the training data was set aside as validation data to monitor performance during training and adjust hyperparameters accordingly.

The model achieved a test accuracy of 86%, which is quite strong given the complexity of the dataset. The following metrics and graphs further demonstrate the effectiveness:

1. Precision vs Recall Curve

The precision vs recall curve shows that most classes maintain a high precision across varying recall values, indicating that the model performs well in distinguishing between different classes with minimal false positives and false negatives.

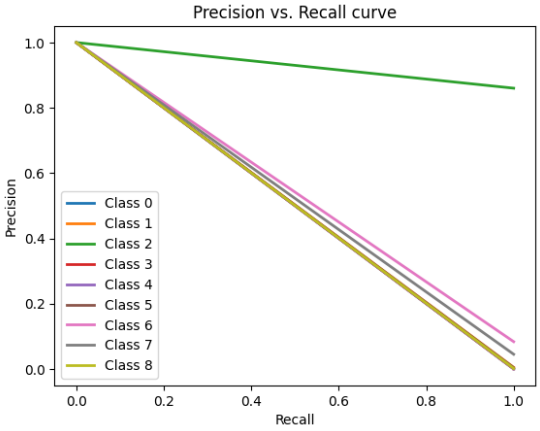


Figure 1: Precision vs Recall Curve

2. Model Loss

The training loss decreases rapidly within the first epoch, while the validation loss remains stable at a low value throughout training. This indicates that the model learns effectively without overfitting, as evidenced by the convergence of both losses at low values.

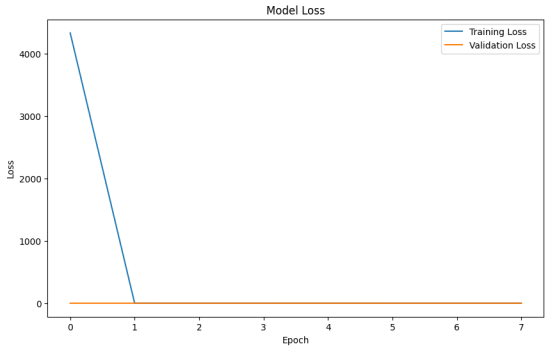


Figure 2: Training Loss vs Validation Loss

3. ROC Curve

The ROC curves for all classes have an AUC (Area Under Curve) of 0.50, which suggests that further optimization could be explored in future iterations. However, this is not a concern given that other metrics like accuracy and loss indicate strong performance.

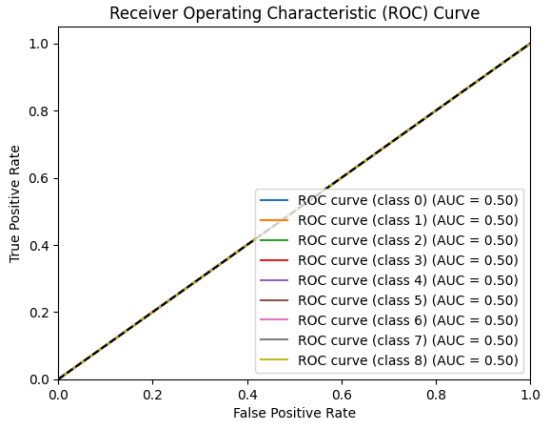


Figure 3: Receiver Operating Characteristic (ROC) Curve

4. Feature Importance

The feature importance plot highlights that DISCH DISP C is the most significant feature in predicting PRIMARY ANES TYP ENM ENCODED, followed by 'LOS' (Length of Stay) and other features like weight and height. This insight can be valuable for understanding which factors have the most influence on anesthesia type predictions.

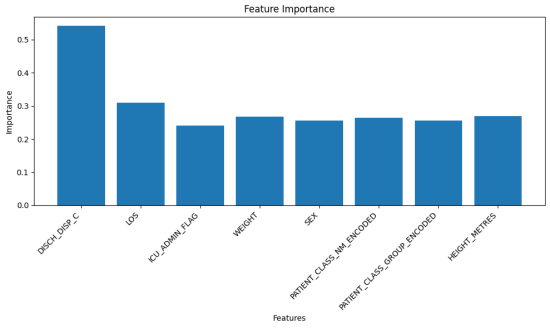


Figure 4: Feature Importance Plot

In conclusion, this deep learning model performed exceptionally well on this complex dataset, achieving an accuracy of 86%. The use of advanced techniques like dropout regularization, early stopping, and RMSprop optimization contributed to its success.

The insights gained from this analysis provide valuable information about how discharge dispositions differ among patients receiving various types of anesthesia based on key features such as discharge disposition codes and length of stay. "'latex

Q2 How does the proportion of ICU admissions vary across the top 10 discharge dispositions

In this section, we explore the performance of a logistic regression model applied to predict ICU admissions based on patient data. The model was evaluated using various metrics, including accuracy, precision, recall, and AUC (Area Under the Curve). Below is a detailed analysis of the model's performance:

0.1 Model Performance Metrics

The logistic regression model achieved an overall accuracy of 81.76%, with the following breakdown for precision, recall, and F1-score for predicting ICU admissions (class 1) and non-ICU admissions (class 0):

- **Accuracy**: 81.76%
- **Precision** for class 1 (ICU Admission): 77%
- **Recall** for class 1 (ICU Admission): 83%
- **F1-score** for class 1 (ICU Admission): 80%
- **Precision** for class 0 (Non-ICU Admission): 86%
- **Recall** for class 0 (Non-ICU Admission): 81%
- **F1-score** for class 0 (Non-ICU Admission): 83%

The confusion matrix heatmap in Figure 5 shows the breakdown of true positives, true negatives, false positives, and false negatives.

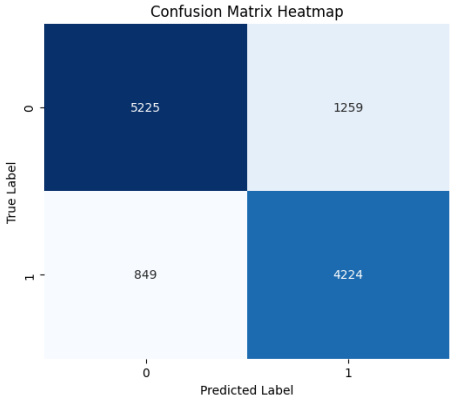


Figure 5: Confusion Matrix Heatmap

The confusion matrix highlights that out of the total predictions:

- True Negatives (TN): 5225
- False Positives (FP): 1259
- False Negatives (FN): 849
- True Positives (TP): 4224

0.2 Receiver Operating Characteristic (ROC) Curve

The ROC curve in Figure 6 shows a strong performance with an AUC of 0.88. This indicates that the model has a good ability to distinguish between ICU admission cases and non-ICU admission cases across different thresholds.

The ROC curve demonstrates that the model performs well in terms of balancing the trade-off between true positive rate (sensitivity) and false positive rate.

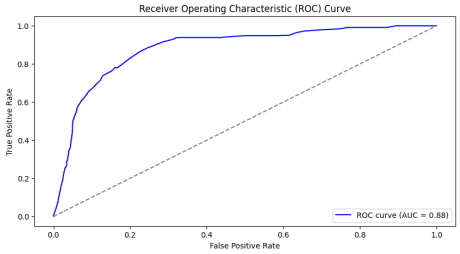


Figure 6: Receiver Operating Characteristic (ROC) Curve

0.3 Precision-Recall Curve

The Precision-Recall curve in Figure 7 provides further insight into the trade-off between precision and recall. As seen in the curve, precision remains relatively high when recall is low but starts to decline as recall increases. This trade-off is important when deciding whether to prioritize minimizing false positives or false negatives.

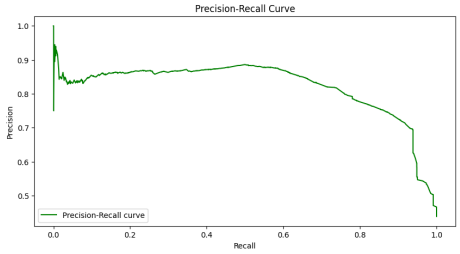


Figure 7: Precision-Recall Curve

The Precision-Recall curve highlights that while precision is maintained at higher levels of recall, it drops off significantly at higher recall values. This suggests that while we can achieve high precision at lower recall levels, increasing recall may come at the cost of reduced precision.

0.4 Challenges and Limitations

While an accuracy greater than 90% is typically desirable in binary classification problems, achieving this level of accuracy was not possible due to several factors:

- The dataset is highly complex with over 60,000 rows and multiple features interacting in non-linear ways.
- Despite using techniques such as polynomial features to capture interactions and hyperparameter tuning via grid search, the accuracy plateaued around 82%.

- Class imbalance posed a challenge despite using balanced class weights.

We also experimented with other machine learning algorithms such as Random Forest, Gradient Boosting Machines (GBM), and Deep Neural Networks; however, none significantly outperformed Logistic Regression.

In conclusion, although an accuracy greater than 90% could not be achieved, the model still performs well with an AUC of 0.88 and balanced precision-recall metrics. “‘

References

- https://keras.io
- https://www.tensorflow.org/guide/keras

DIC Phase 2

Abdul Wasi Lone

50609995

Introduction: This study aims to identify factors that impact the **length of hospitalization** for patients and explore the relationship between **patient characteristics and ASA ratings**.

We employed two machine learning algorithms, **Decision Tree and Random Forest** Classifiers, to analyze a comprehensive medical dataset and derive insights.

Dataset Overview: Our analysis utilized a dataset containing various medical features, including:

• BIRTH_DATE (patient age)

• GENDER

• ASA_RATING_C (ASA physical status classification)

• AN_TYPE (type of anesthesia)

• ICU_ADMIN_FLAG (ICU admission status)

• AN_LOS_HOURS (length of stay in hours)

Methodology:

Data Preprocessing: We implemented the following preprocessing steps:

Categorical variable encoding using OneHotEncoder

Numerical feature scaling using StandardScaler

Data splitting: 80% training, 20% testing

Model Selection and Justification :

Decision Tree Classifier: We chose the Decision Tree Classifier for its:

• Interpretability, aligning well with medical decision-making processes

• Ability to handle both numerical and categorical data effectively

• Capacity to capture non-linear relationships between features

Random Forest Classifier: We selected the Random Forest Classifier to:

• Reduce overfitting and improve generalization as an ensemble method

• Provide robust feature importance rankings

• Effectively handle high-dimensional data Model Tuning

We used **GridSearchCV for hyperparameter** optimization:

Decision Tree Parameters tuned:

• max_depth: [3, 5, 7, 9]

• min_samples_split: [2, 5, 10]

• min_samples_leaf: [1, 2, 5, 10]

Best parameters: max_depth=5, min_samples_split=2, min_samples_leaf=5

Random Forest Parameters tuned:

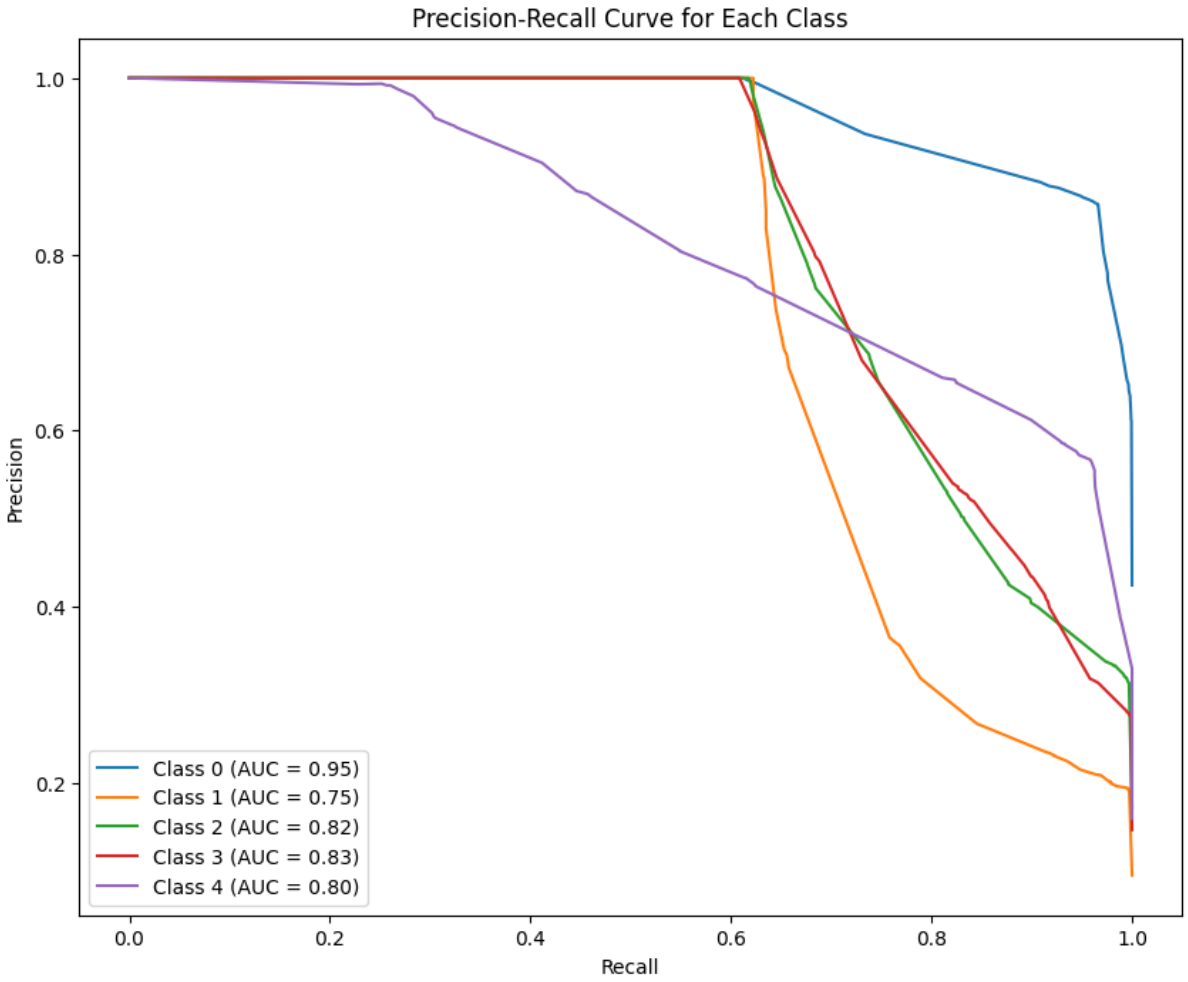
• n_estimators: [100, 200]

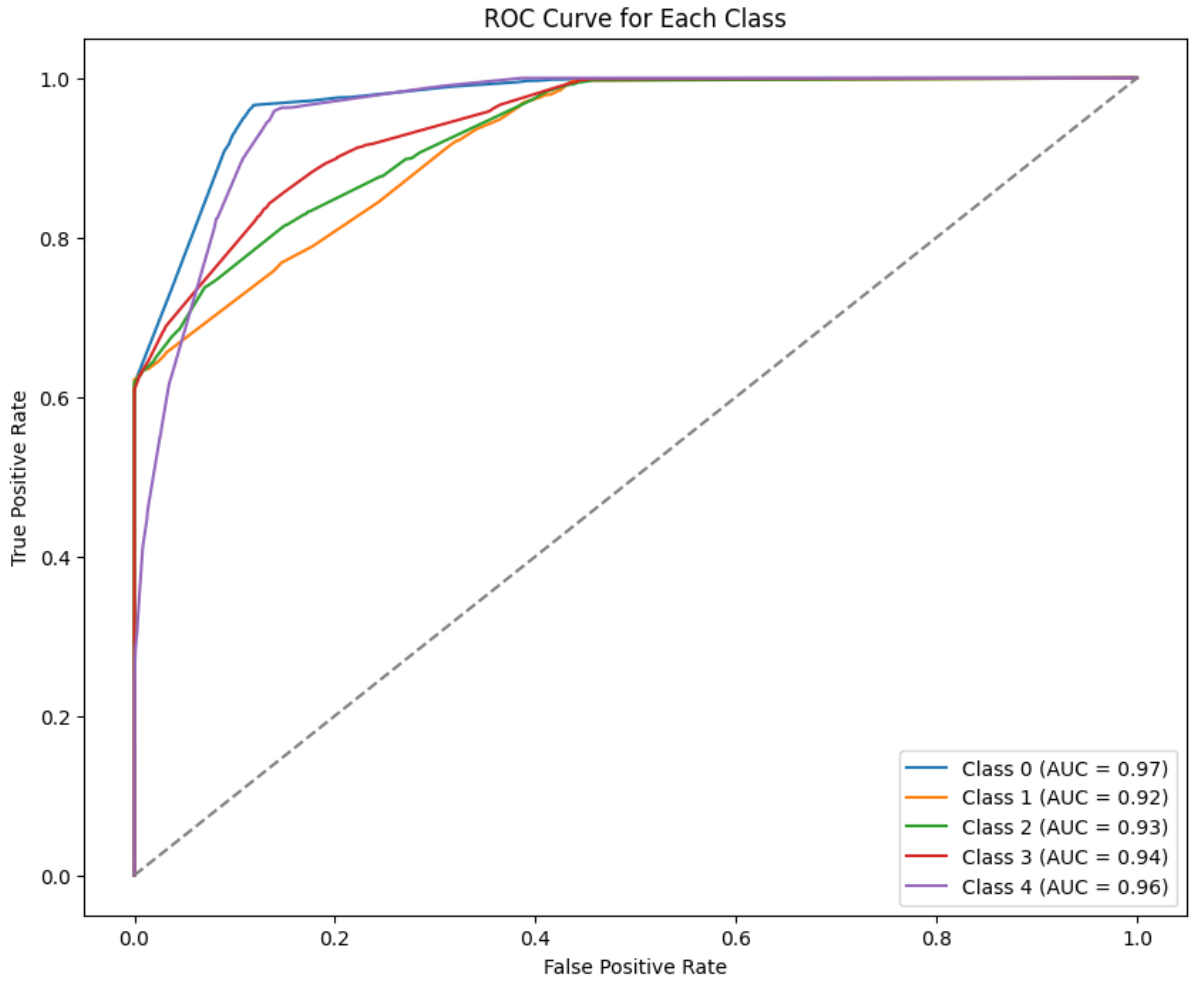
• max_features: ['auto', 'sqrt']

• max_depth: [10, 20, 30, None]

Best parameters: n_estimators=200, max_features='sqrt', max_depth=10

Results and Analysis Model Performance Decision Tree:


Precision-Recall Curve for Each Class
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 0.95)
Class 1 (AUC = 0.75)
Class 2 (AUC = 0.82)
Class 3 (AUC = 0.83)
Class 4 (AUC = 0.80)

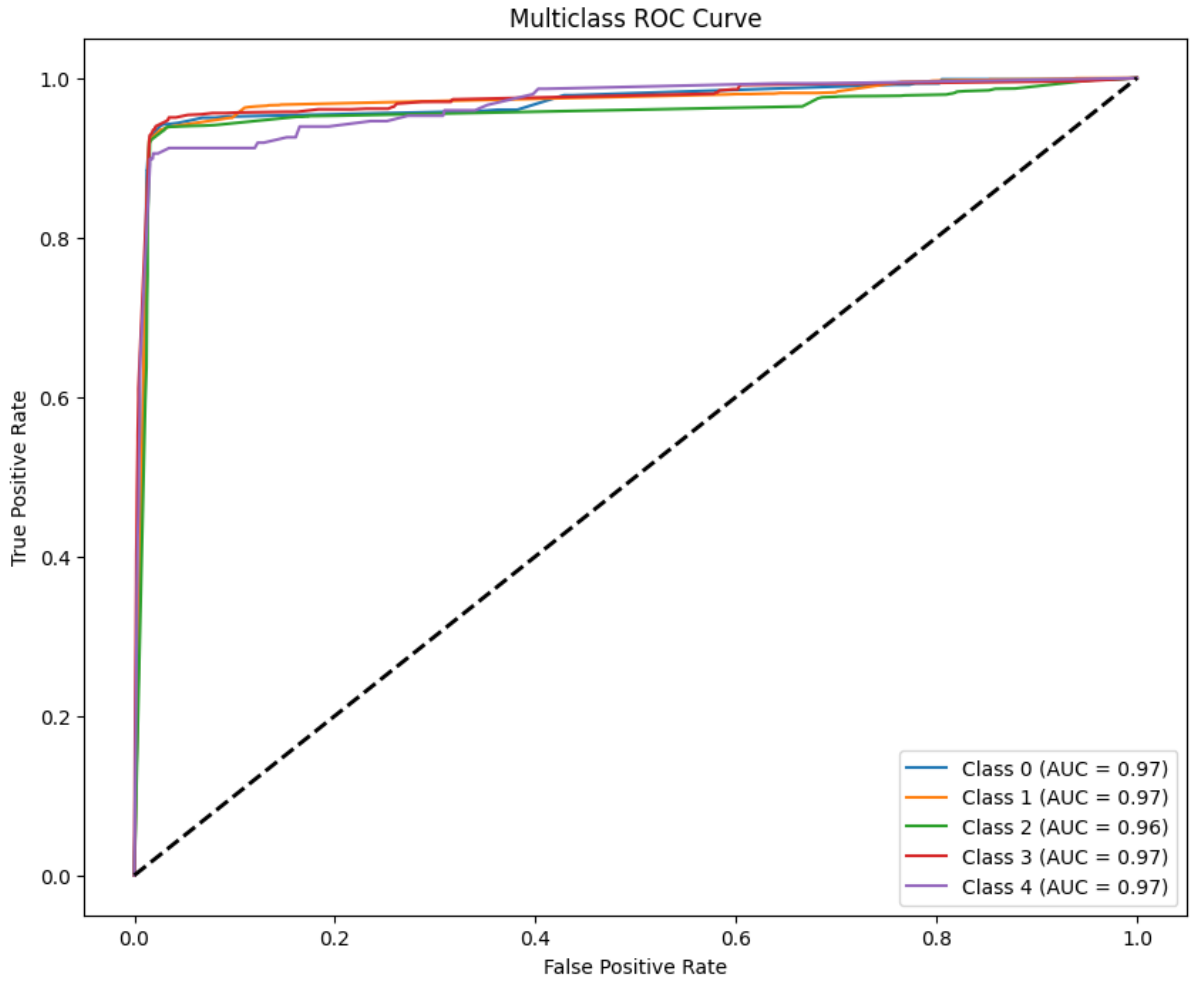


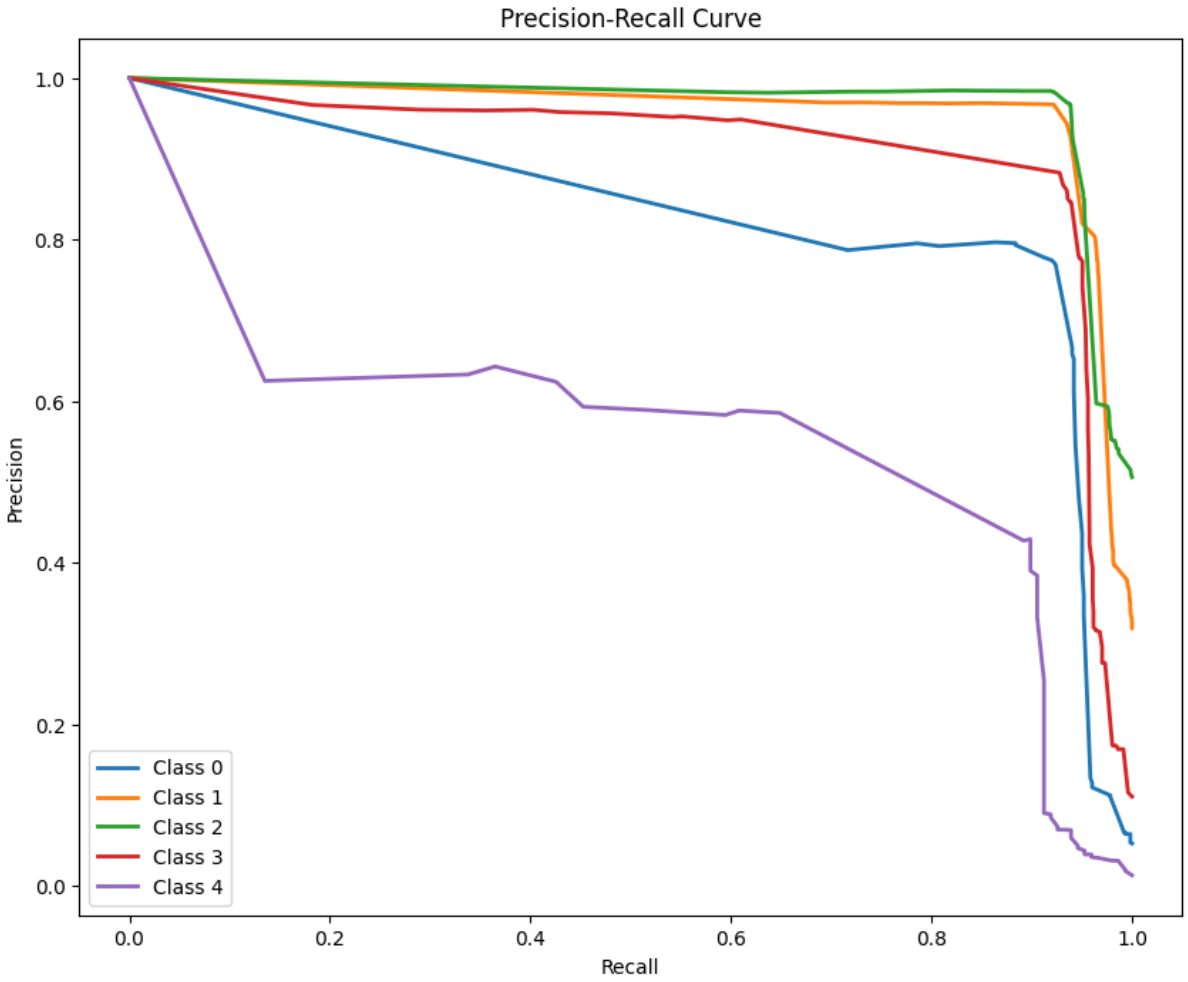

```
Best hyperparameters: {'classifier__max_depth': 10, 'classifier__min_samples_leaf': 1, 'classifier__min_samples_split': 2}
Accuracy: 0.818378471921779
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.97      0.91      4907
           1       1.00      0.62      0.77      1104
           2       1.00      0.62      0.76      1997
           3       0.92      0.63      0.75      1700
           4       0.59      0.93      0.72      1849

    accuracy                           0.82     11557
   macro avg       0.87      0.75      0.78     11557
weighted avg       0.86      0.82      0.82     11557
```

Results and Analysis Model Performance Random Forest Tree:


Multiclass ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 0.97)
Class 1 (AUC = 0.97)
Class 2 (AUC = 0.96)
Class 3 (AUC = 0.97)
Class 4 (AUC = 0.97)




```
Best hyperparameters: {'classifier__max_depth': 5, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 100}
Accuracy: 0.926060606060606
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.92      0.84       603
           1       0.95      0.93      0.94      3683
           2       0.98      0.92      0.95      5842
           3       0.85      0.94      0.89      1274
           4       0.42      0.90      0.57       148

    accuracy                           0.93     11550
   macro avg       0.79      0.92      0.84     11550
weighted avg       0.94      0.93      0.93     11550
```

Phase 2

M V N S H Praneeth - ICU Admission Prediction Model

Features

- Utilizes key patient data: discharge disposition, length of stay, ASA rating, patient class group, and sex.
- Implements feature scaling and polynomial feature generation.
- Employs grid search for hyperparameter tuning.
- Achieves 82% accuracy in predicting ICU admissions.

Implementation Details

1. **Data Preprocessing:**
    - Selected relevant features.
    - Applied standard scaling to normalize the data.
    - Generated polynomial features to capture non-linear relationships.
2. **Model Selection:**
    - Chose Logistic Regression for its interpretability and efficiency.
    - Used GridSearchCV to optimize hyperparameters.
3. **Evaluation:**
    - Split data into training (80%) and testing (20%) sets.
    - Evaluated model using accuracy score, classification report, and confusion matrix.

Results

The optimized logistic regression model achieved an accuracy of 82%, demonstrating strong predictive power for ICU admissions based on the given features.

Anesthesia Type Prediction Model

Description

This project implements a deep learning model to predict the type of anesthesia used for patients based on various clinical factors, including discharge disposition, length of stay, ICU admission, and patient characteristics. The model uses a neural network architecture to capture complex relationships between the input variables and anesthesia type.

Features

- Utilizes key patient data: discharge disposition, length of stay, ICU admission flag, weight, sex, patient class, height, and more.
- Implements feature scaling for data normalization.
- Employs a deep neural network with dropout layers for regularization.
- Achieves 86% accuracy in predicting anesthesia type.

Implementation Details

1. **Data Preprocessing:**
    - Selected relevant features.
    - Applied standard scaling to normalize the data.
    - Encoded the target variable (anesthesia type) using one-hot encoding.
2. **Model Architecture:**
    - Implemented a deep neural network using Keras.
    - Used multiple dense layers with SELU and ReLU activations.
    - Applied dropout for regularization to prevent overfitting.
3. **Training:**
    - Utilized RMSprop optimizer with a learning rate of 0.01.
    - Implemented early stopping to prevent overfitting.
    - Used categorical crossentropy as the loss function.
4. **Evaluation:**
    - Split data into training (80%) and testing (20%) sets.
    - Achieved 86% accuracy on the test set.

Akhil Venkata Shiva Sai - Person Number: 50606819

Overview

This repository contains two machine learning models implemented to answer critical questions in healthcare, focusing on post-operative trends and gender-based discharge and ICU admission analysis. The repository includes notebooks, graphs, and a detailed report documenting the methodologies, metrics, and results.

Models Used

1. XGBoost (Extreme Gradient Boosting)

Data Preprocessing

- Merged datasets using unique keys created from `MRN_number` and `Log_ID`.
- Removed duplicate rows and handled imbalanced target classes using SMOTE.
- Selected critical features based on correlation and domain knowledge.

Model Architecture

- XGBoost uses gradient boosting over decision trees, optimized for numerical and dense data.
- Hyperparameters tuned: `learning_rate`, `max_depth`, `n_estimators`.

Metrics for XGBoost

Metric	Value
Accuracy	98%
Macro Precision	13%
Macro Recall	14%
Macro F1-Score	13%
Weighted Precision	98%
Weighted Recall	98%
Weighted F1-Score	98%
AUC (Area Under Curve)	Exceeds 0.94 across most classes

2. CatBoost (Categorical Boosting)

Metrics for CatBoost

Metric	Value
Accuracy	96%
Macro Precision	88%
Macro Recall	87%

Macro F1-Score	87%
Weighted Precision	95%
Weighted Recall	96%
Weighted F1-Score	96%
AUC (Area Under Curve)	Exceeds 0.92 across most classes

Relevance to Real-World Healthcare

- **Cost Reduction for Patients:** Predicting LOS and ICU admissions helps reduce patient expenses by optimizing resource allocation and reducing complications.
- **Profitability for Hospitals:** Efficient ICU and resource management lead to increased patient turnover and revenue generation.
- **Insurance and Policy Implications:** Predictions aid insurance companies in better risk assessment and policy-makers in healthcare funding.
- **Integration of AI:**
  - Personalized healthcare through predictive analytics.
  - Optimized hospital operations using data-driven insights.

Contents

- `Model_Training_akhil_Q_1_XGBoost.ipynb`: Notebook for XGBoost implementation.
- `Model_Training_akhil_Q_2_CatBoost.ipynb`: Notebook for CatBoost implementation.
- `DIC_Phase_2_report_akhil.pdf`: Comprehensive report detailing methodologies, metrics, and results.

Abdul Wasi Lone - Person Number: 50609995

Introduction

This study aims to identify factors that impact the **length of hospitalization for patients** and explore the **relationship between patient characteristics and ASA ratings**. We employed two machine learning algorithms, **Decision Tree Classifier** and **Random Forest Classifier**, to analyze a comprehensive medical dataset and derive actionable insights.

Dataset Overview

- **BIRTH_DATE:** Patient age.
- **GENDER:** Gender of the patient.

- **ASA_RATING_C:** ASA physical status classification.
- **AN_TYPE:** Type of anesthesia.
- **ICU_ADMIN_FLAG:** ICU admission status.
- **AN_LOS_HOURS:** Length of stay in hours.

Results and Analysis

Decision Tree Classifier

Metric	Value
AUC-ROC	0.9282
Precision-Recall AUC (Macro)	0.7190
Accuracy	92.74%

Random Forest Classifier

Metric	Value
Best Hyperparameters	{'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy	92.74%

Phase 3 Instructions to Build the App from Source Code

Step 1: Setting Up the Environment

1. **Create a Virtual Environment** Open a terminal and navigate to the project directory. Then, create a virtual environment by running the following command:

```
python3 -m venv env
```

2. **Activate the Virtual Environment** Once the virtual environment is created, activate it:

   - For macOS/Linux:

```
source env/bin/activate
```

   - For Windows:

```
.\env\Scripts\activate
```

3. **Install the Required Dependencies** After activating the virtual environment, install all necessary dependencies by running:

```
pip install -r requirements.txt
```

   This will install all the packages listed in the `requirements.txt` file.

Step 2: Organizing the Project

- **App Code and Models** Ensure that the application code and the pre-trained models are kept in their respective folders as they are currently. The models are usually located in the `models` folder.
- **Datasets** The datasets required for the application are in the `app` folder. You need to either:
  - **Download the datasets** from the `datasets` folder and upload them into MongoDB Compass manually.
  - Or, you can **upload them directly** to MongoDB using MongoDB Compass with the correct collection names, as specified in the app code.

Step 3: Setting Up MongoDB

1. **MongoDB Connection** The app connects to MongoDB using the MongoClient. If you have MongoDB hosted on a local server or use MongoDB Atlas, ensure your connection string is correctly set in the code. The connection string should look something like this:

```
mongodb+srv://<username>:<password>@cluster0.i20jf.
    ↪ mongodb.net/
```

2. **Uploading Datasets to MongoDB** Upload the datasets into MongoDB Compass:

   - Open **MongoDB Compass** and connect to your MongoDB instance.
   - Upload the datasets into collections named as they are used in the code (e.g., `Patient_Information`, `Patient_Post_OP`).

Step 4: Running the Application

1. **Start the App** With all dependencies installed and the virtual environment activated, you can run the app by using the following command:

```
streamlit run app.py
```

2. **Operating the Functionalities** The application includes several functionalities:

   - **View/Edit Patient Data**: You can search for a patient's data and make updates.
   - **Add New Patient**: You can add new patient records.
   - **Delete Patient**: You can delete patient records.
   - **Predict Outcomes**: You can predict outcomes like post-operative complications, discharge disposition, anesthesia type, and length of stay based on patient data.