STAT215 FINAL PAPER

Joseph Kouadio / Vidhi / Zihe / Tianyu

December 2024

1 Introduction

Traumatic Brain Injury (TBI) is a pressing public health concern, particularly among children, where it stands as a leading cause of death and disability worldwide. In the United States alone, pediatric head trauma accounts for approximately 7,400 deaths annually, alongside more than 60,000 hospital admissions and over 600,000 emergency department visits. The effective management of TBI is paramount to minimizing both immediate and long-term health impacts, making it a critical area of research and intervention.

A significant challenge in managing pediatric TBI lies in clinical decision-making, particularly regarding the use of computed tomography (CT) scans for diagnosis. While CT imaging is a valuable diagnostic tool, it exposes children to ionizing radiation, which carries the potential risk of radiation-induced malignancies later in life. Thus, clinicians must balance the benefit of early and accurate detection of serious brain injuries with the potential harm from unnecessary imaging. This tradeoff underscores the need for robust and interpretable clinical decision rules that can accurately identify children at very low risk of clinically important traumatic brain injuries (ciTBI), enabling clinicians to minimize unnecessary CT scans.

Building on the foundational work of Nathan Kuppermann et al. (2009) in The Lancet, this study uses the Pediatric Emergency Care Applied Research Network (PECARN) dataset to perform exploratory data analysis (EDA) and develop predictive models. This work aims to identify actionable insights for improving pediatric emergency care practices. Machine learning algorithms, such as logistic regression (LR), random forests (RF), neural networks (NN), and discriminant analysis methods (LDA, QDA), will be employed to evaluate predictive performance and interpretability, ultimately prioritizing patient safety and health outcomes.

2 Dataset

The dataset focuses on pediatric TBI cases collected through the PECARN study. It contains both clinical and demographic variables that are critical for predicting ciTBI. This prospective observational cohort study involved children under 18 years of age with minor head trauma, evaluated across 25 PECARN emergency departments. The aim was to develop and validate two clinical prediction rules to accurately identify children at near-zero risk of clinically significant traumatic brain injuries (TBI) following blunt trauma—one rule for children under 2 years old and another for those aged 2 years and older. Around 44k subjects were enrolled within 24 hours of head trauma, and various patient-related relevant features were recorded. These validated prediction rules successfully identified children at very low risk of clinically significant TBI, allowing for the routine avoidance of unnecessary CT scans.

3 Data Cleaning Process

Extensive data cleaning was performed to ensure the dataset was accurate and suitable for analysis. The following steps were implemented:

3.1 Initial Data Validation

First, each column (feature) was examined for invalid values. The smallest, largest, and average values were printed for each of the 125 features. Additionally, the unique values and the number of `NaN` entries in each feature were evaluated to ensure the values were meaningful. To further validate the data, histograms of the values for each column were plotted to identify any anomalies.

Upon investigation, it was observed that there was an unusually large number of entries with the value `92`. Referring to the dataset documentation, it was determined that `92` served as a placeholder for *Not Applicable*. This was true for several features. Consequently, all features where `92` was a placeholder for *Not Applicable* were updated by replacing `92` with `NaN`. This modification was implemented in the `clean.py` function as the first cleaning step. Figure **??** illustrates a histogram for the feature *AgeInMonths*, which exhibited no anomalies.

3.2 Handling Placeholders

Further analysis of unique values in the features *EDDisposition* and *Race* revealed that the value *Other* was also used as a placeholder for *Not Applicable*. Therefore, the value *Other* was replaced with `NaN`. This step was also implemented in the `clean.py` function.

3.3 Validation of Glasgow Coma Scale (GCS)

The variable *GCS* was evaluated based on domain knowledge. The Glasgow Coma Scale (GCS) is a clinical tool used to assess a patient's level of consciousness after a brain injury. It evaluates three aspects of a patient's response:

- *GCSEye* (eye opening),
- *GCSMotor* (motor response), and

- *GCSVerbal* (verbal response).

The sum of these features provides the total GCS score (*GCSTotal*). Rows where the sum of *GCSEye*, *GCSMotor*, and *GCSVerbal* did not equal *GCSTotal* were considered invalid. This discrepancy was found in 1,344 patients. Since this criterion is crucial in ICU decision-making, such as determining the need for a CT scan, rows failing this criterion were removed.

3.4 Ensuring Tidy Data Format

The dataset was further assessed to ensure it adhered to the "tidy data" standard, where each row corresponds to a single observational unit and each column represents a unique type of measurement. Types of measurements for individual columns were printed, and it was verified that the data was not duplicated across rows. The dataset met these criteria and was already in a tidy format.

3.5 Feature Selection

The primary goal of the analysis was to determine whether a patient was diagnosed with traumatic brain injury (TBI) following a CT scan. The objective was to reduce unnecessary CT scans using features deducible without performing a CT scan. Based on the documentation:

- All columns after the feature *PosCT* (TBI on CT scan determined by PI) were considered irrelevant to this objective, except for *PosIntFinal*, which is the final declaration of TBI presence and the main variable of interest.
- Consequently, a judgment call was made to remove the last 24 features, based on this criterion, in the `clean.py` function.

3.6 Handling Missing Data

Since analysis could only proceed if values were present for the main variable of interest (*PosIntFinal*), all rows with `NaN` values in this feature were removed. Additionally, features with more than 50% missing values were also dropped.

During data perturbation, the threshold for missing values was adjusted to 90%. Features with more than 90% missing values were removed. The results of this perturbation are detailed in the section *Stability Under Data Perturbation*.

4 Data Exploration

A missing value analysis was conducted to evaluate the completeness of the data and guide subsequent pre-processing decisions. Figure 1 shows that certain variables, such as `Dizzy`, `Ethnicity`, and `ClavFace`, have a large number of missing values (approximately 16,000 records), making them difficult to use without further imputation or exclusion. Variables like `Observed`, `Amnesia_verb`, and

LOCSeparate have moderate missing values, and variables such as SfxPalp, FontBulg, and EDDisposition have very limited missing values. The missing values in these variables can be filled through standard imputation techniques.

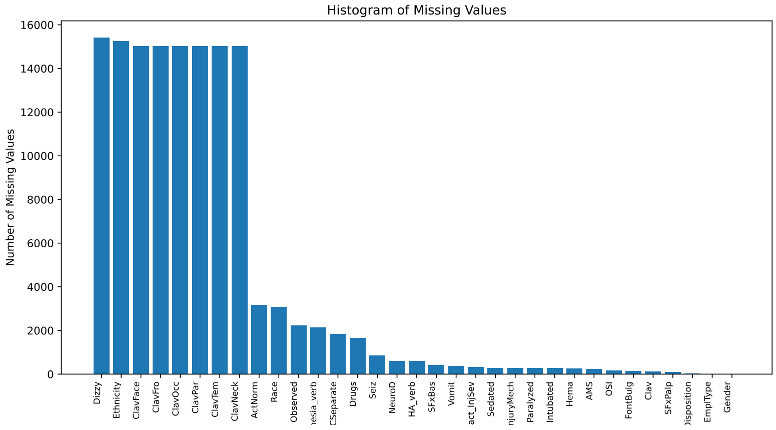

Figure 1: Count of Missing Values

The distribution of the target variable, PosIntFinal, shows a significant class imbalance (Figure 2). The majority class 0, which indicates that there is no clinically important traumatic brain injury, accounts for 98.3% of the data (41,319 records), while the minority class 1, representing clinically important injuries, constitutes only 1.7% (717 records). This imbalance poses challenges for predictive modeling. To address this, techniques such as oversampling the minority class, undersampling the majority class, or employing cost-sensitive learning methods will be considered. Furthermore, evaluation metrics such as false negative rate, precision, recall, and the F1 score will be prioritized to ensure a fair assessment of model performance.

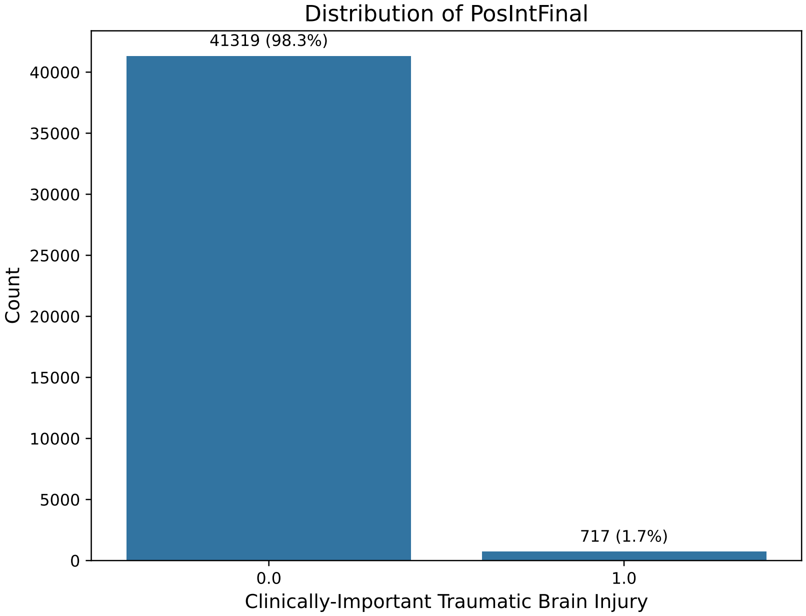

Figure 2: Distribution of the Target Variable

A subset of variables was selected to generate a heatmap based on their numerical or ordinal nature, as heatmaps are most effective for analyzing correlations between such variables. Numerical variables, like `GCSTotal`, and ordinal variables, such as `High_impact_InjSev`, `GCSEye`, `GCSVerbal`, and `GCSMotor`, were chosen for the correlation analysis. Categorical variables without an ordinal relationship were excluded as they may not be able to provide meaningful linear correlations in this context.

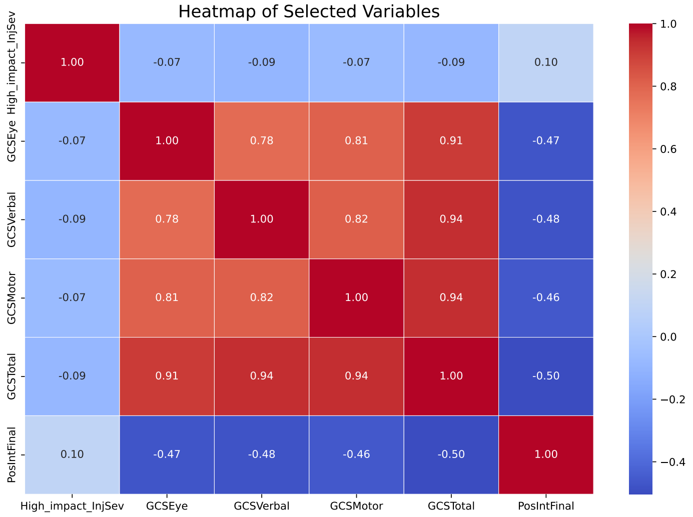

Figure 3: Heatmap of Selected Features

Figure 3 indicates strong positive correlations among the Glasgow Coma Scale (GCS) components (`GCSEye`, `GCSVerbal`, `GCSMotor`) and `GCSTotal`, which is expected since `GCSTotal` is derived from the sum of these components. Given their multicollinearity, using `GCSTotal` alone rather than including all these features in a predictive model may be beneficial to the model training. Additionally, `PostIntFinal` shows negative correlations with the GCS components and `GCSTotal` (approximately -0.46 to -0.50), indicating that higher GCS scores are associated with better neurological outcomes. Meanwhile, `High_impact_InjSev` has weak correlations with both the GCS components and `PostIntFinal` (around 0.10).

5 Train Test Split

To address the issue of class imbalance in the dataset, we implemented an oversampling technique for the minority class. This method involves artificially increasing the representation of the underrepresented class by duplicating its samples until it matches the size of the majority class. By doing so, the training dataset becomes balanced, ensuring the model does not become biased toward the majority class during learning. The oversampling process was applied only to the training data, keeping the test data untouched to maintain its original distribution for unbiased evaluation. This approach helps the model learn the characteristics of the minority class more effectively, improving its ability to correctly predict positive cases while maintaining robust generalization to unseen data.

20% of the entire cleaned data were kept as a test set and the rest was used for training. Individual models then used different ratios of this training data to make a validation set.

6 Models

In this section, the implementation of various prediction models will be discussed, including random forest, logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), convolutional neural networks (CNN), and neural networks (NN). Each model is evaluated in terms of the rationale for its selection, its configuration (e.g., hyperparameters, and architecture for CNN and NN), and the methods employed to address class imbalance in the dataset.

6.1 Random Forest

A Random Forest classifier was trained to predict the target variable *PosIntFinal*. To address the class imbalance in the dataset, a class weighting scheme was utilized, which automatically adjusts weights inversely proportional to class frequencies. This ensured that the minority class was adequately represented during the training process.

The model was configured with 100 decision trees, striking a balance between computational efficiency and predictive accuracy. A random seed was set to ensure the reproducibility of results. The training dataset, which was balanced through oversampling, was used for model training, while the test dataset was reserved for unbiased evaluation.

The Random Forest model provided both class predictions and class probabilities, which were used to evaluate performance metrics such as ROC-AUC, precision, recall, and false-negative rate. This allowed for a comprehensive assessment of the model's ability to correctly identify positive cases while minimizing false negatives.

Random Forest was selected for its robustness, ability to handle high-dimensional data, and inherent capability to estimate feature importance. These characteristics make it an effective choice for predicting *PosIntFinal* in an imbalanced dataset.

6.2 LDA

A Linear Discriminant Analysis (LDA) model was trained to predict the target variable *PosIntFinal*. LDA is a statistical method that assumes features are normally distributed and maximizes the separation between classes by finding a linear combination of features that best discriminates between the target classes. This makes LDA particularly effective for datasets with linear class separability.

The model was trained on a balanced dataset, where class imbalance was addressed through oversampling techniques. To ensure robustness, the LDA model was evaluated on a separate test set, preserving the original class distribution. LDA's probabilistic framework also allowed for the prediction of class probabilities, which were utilized to compute performance metrics such as ROC-AUC, precision, and recall.

LDA was chosen for its simplicity, interpretability, and efficiency, particularly for high-dimensional data with clear decision boundaries. It provided a benchmark for comparison with more complex models while offering insights into the linear separability of the dataset.

6.3 QDA

A Quadratic Discriminant Analysis (QDA) model was implemented to predict the target variable *PosIntFinal*. QDA extends Linear Discriminant Analysis by relaxing the assumption of equal covariance matrices for classes, allowing it to model more complex, non-linear decision boundaries. This flexibility makes QDA particularly suitable for datasets where classes exhibit distinct distributions.

The model was trained on a balanced dataset, with class imbalance addressed through oversampling techniques. QDA computes probabilities for class membership, enabling the evaluation of probabilistic metrics such as ROC-AUC, precision, and recall alongside standard classification metrics.

QDA was selected for its ability to handle non-linear separability while maintaining interpretability. By modeling the covariance structures of the data, QDA provided a deeper understanding of class distribution and offered a valuable comparison to linear models like LDA and more complex algorithms.

6.4 CNN

Missing values in the data are handled by replacing NaNs with column means to ensure data consistency, while padding with zeros is applied to customize the input size for the model. A custom PyTorch TabularDataset class is used to preprocess the data, replacing missing values and reshaping the features into grids suitable for convolutional layers. The model, TabularCNN, is designed with two convolutional layers, each followed by max-pooling, and fully connected layers to extract and classify patterns from the tabular data. To address the issue of imbalanced data, a custom loss function combines Binary Cross-Entropy (BCE) with a penalty for high False Negative Rate (FNR), promoting a more balanced performance.

The input features are padded and reshaped into a 1x5x9 grid format, enabling the CNN to utilize spatial relationships within the tabular data. During training, a weighted loss function is applied, where class weights are calculated based on the ratio of negative to positive samples, ensuring that the model remains sensitive to minority class predictions. Recall and FNR metrics are tracked throughout training to evaluate the model's ability to correctly identify positive samples. The integration of FNR into the custom loss function further encourages the model to minimize missed positives, which is particularly important for applications requiring high sensitivity. This cohesive approach aligns the loss function and evaluation metrics with the project's goal of optimizing performance on imbalanced datasets.

6.5 Neural Network

A neural network was designed and trained to predict the target variable `PosIntFinal`. Missing values in the data are handled by replacing NaNs with column means to ensure data consistency. The dataset was split into training, validation, and test sets in a 60-20-20 ratio to ensure robust evaluation. Features were standardized using a `StandardScaler` to enhance training efficiency. The neural network consisted of three fully connected layers: the input layer with 128 neurons, a hidden layer with 64 neurons, and an output layer with a single neuron using the sigmoid activation function for binary classification.

To address the class imbalance in the dataset, a custom loss function was implemented to penalize false negatives more heavily, ensuring the model prioritized correctly identifying positive cases. Additionally, a custom metric, `false_negative_rate`, was defined to monitor the proportion of false negatives during training, alongside traditional metrics like accuracy and recall. The model was compiled with the Adam optimizer and trained for 50 epochs using a batch size of 32. Validation data was used to monitor performance and prevent overfitting.

To optimize the model's hyperparameters, `GridSearch` was employed to explore combinations of the optimizer, number of epochs, and batch size. The optimizer options included `adam`, `sgd`, and `rmsprop`, while the number of epochs ranged from 20 to 100, and batch sizes varied between 16, 32, and 64. For each combination, the model was trained and validated, and the configuration yielding the best validation recall and lowest false negative rate was selected.

6.6 Logistic Regression

Logistic regression (LR) was selected due to its simplicity, interpretability, and suitability for binary classification in clinical contexts. For predicting `PosIntFinal` (ciTBI), LR offers transparent, actionable decision-making by expressing predictors as odds ratios, which is particularly valuable for clinicians.

Missing values in the dataset were handled by imputing column medians to maintain data consistency. Features were standardized using `StandardScaler` to prevent variables with larger magnitudes from dominating the model. Recursive Feature Elimination (RFE) was used to identify clinically meaningful predictors, such as Glasgow Coma Scale (GCS), age, and loss of consciousness, ensuring a parsimonious model while retaining predictive accuracy.

To address the imbalance in ciTBI cases, the `class_weight='balanced'` parameter was used to adjust weights inversely proportional to class frequencies, reducing the risk of false negatives. Threshold optimization was performed to enhance recall and precision, ensuring the model achieved clinically relevant outcomes.

Stratified 5-fold cross-validation was employed to evaluate model performance, ensuring consistent class representation across folds. Bootstrapping was used to compute confidence intervals for coefficients, enhancing the interpretability and robustness of the results. Stability tests were conducted using varying regularization strengths and alternative feature sets, confirming the model's reliability across scenarios.

The model achieved high recall, effectively minimizing false negatives in predicting ciTBI. ROC and calibration curves demonstrated the model's strong discriminative power and alignment between predicted probabilities and observed outcomes.

Logistic regression's computational efficiency makes it suitable for resource-limited clinical settings, delivering real-time predictions. Its simplicity ensures the model can be easily retrained with updated data, enabling long-term scalability. Overall, logistic regression combines performance, interpretability, and practicality, making it a reliable tool for minimizing false negatives here in this pediatric care study.

7 Model Evaluation

Table 1 summarizes the performance of various models on the test dataset with a classification threshold set to 0.5. As the primary concern is minimizing the False Negative Rate (FNR), which corresponds to missing positive cases, the best neural network (NN) model and non-NN model were selected based on FNR.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}$$

Where:

FN = False Negatives (actual positives incorrectly classified as negatives)

TP = True Positives (actual positives correctly classified)

Table 1: Results for model performances on the test set.

Model	Accuracy	Precision	Recall	F1	AUC	FNR
lda	0.888	0.124	0.838	0.215	0.938	0.162
randomforest	0.986	0.820	0.325	0.465	0.974	0.675
qda	0.922	0.150	0.701	0.248	0.920	0.299
cnn	0.958	0.285	0.864	0.428	0.981	0.136
nn	0.979	0.443	0.481	0.461	0.968	0.519
logreg	0.893	0.129	0.838	0.223	0.943	0.162

The CNN model demonstrated the lowest FNR among NN models, while logistic regression achieved the best FNR among non-NN models, outperforming LDA due to its higher accuracy. Additionally, both CNN and logistic regression exhibited strong performance across other metrics, including AUC, making them the most reliable choices for this task.

Figure 4 shows the ROC curves and the Threshold vs FNR plot for the models. The left plot presents the ROC curves with AUC values, while the right plot visualizes the False Negative Rate (FNR) across varying thresholds.

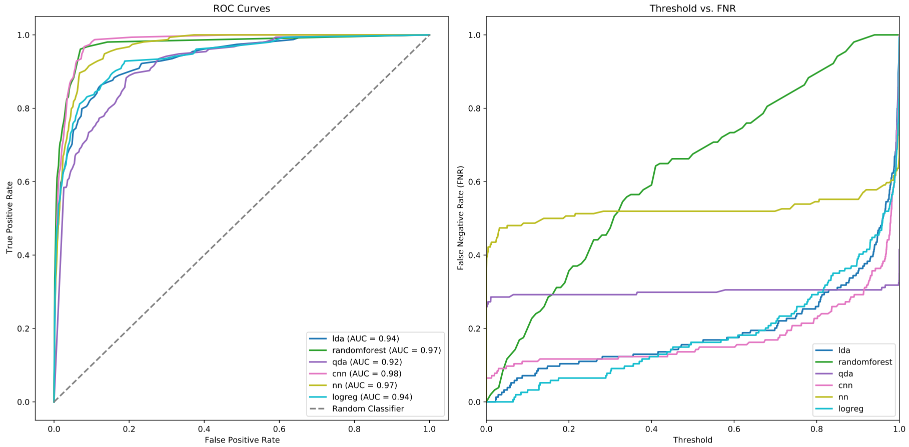

Figure 4: ROC Curves and Threshold vs FNR for Different Models

8 Interpretation

8.1 Convolutional Neural Network (CNN)

The CNN interprets data through its convolutional architecture, designed to uncover patterns and interactions among predictors, making it particularly adept at handling complex, nonlinear relationships. By leveraging convolutional layers, the CNN identifies hierarchical features, enabling it to extract subtle patterns indicative of ciTBI risk.

For instance, the model can capture interactions among features such as age, mechanism of injury, and symptom clusters (e.g., vomiting or abnormal mental status), creating high-level representations that are challenging for traditional models to identify. This capability allows the CNN to move beyond individual predictors and analyze their spatial or contextual relationships, enhancing its ability to predict ciTBI with improved accuracy.

While inherently less interpretable than logistic regression, the CNN's predictions remain clinically relevant by offering insights into feature interactions and improving diagnostic precision. Its ability to identify patterns across multiple dimensions makes it a valuable tool for detecting high-risk cases that might otherwise be missed, providing a complementary perspective to simpler, more interpretable models.

8.2 Logistic Regression (LR)

Logistic regression excels in interpretability, linking statistical predictions to actionable clinical insights. Each coefficient can be directly interpreted as an odds ratio, providing clinicians with a clear understanding of how specific factors influence the likelihood of ciTBI. This transparency makes logistic regression an essential decision-support tool in clinical practice.

A key strength of logistic regression lies in its probabilistic predictions, which enable clinical prioritization. By defining an optimal threshold tailored to the dataset, the model effectively flags high-risk patients while reducing unnecessary interventions. This threshold optimization focuses on minimizing false negatives, ensuring patient safety in critical scenarios where undiagnosed ciTBI could lead to severe outcomes.

The model's simplicity also facilitates a clear understanding of variable contributions. Predictors such as Glasgow Coma Scale (GCS) or loss of consciousness, already well-established in clinical guidelines, align with the model's statistical outputs, reinforcing its acceptance and ease of integration into diagnostic workflows. Unlike more complex models, logistic regression requires minimal retraining and can be seamlessly adapted to evolving clinical needs.

Additionally, stability analyses demonstrate the reliability of logistic regression under varying conditions, such as adjustments in regularization strength or feature subsets. The model consistently identifies key predictors, validating its robustness and ensuring trustworthy predictions in dynamic healthcare environments.

Lastly, logistic regression's computational efficiency supports real-time updates, enabling clinicians to incorporate new data without extensive delays. Its adaptability and scalability make it a practical, long-term solution for resource-constrained settings, ensuring its continued relevance in pediatric trauma care.

By balancing interpretability, clinical alignment, and computational efficiency, logistic regression emerges as a dependable and transparent tool, complementing the CNN's predictive strength to provide a comprehensive approach to improving outcomes in pediatric head trauma cases.

9 Stability under Model Perturbation

Tianyu/Joseph: Analyze how sensitive the models are to changes in architecture or hyperparameters. Discuss the robustness of results and their implications for clinical reliability.

10 Stability under Data Perturbation

We increased the threshold for rejecting features with a high percentage of missing values from 0.5 to 0.9, meaning a feature would only be removed if more than 90% of its values are NaN. This adjustment allowed us to retain more features, potentially capturing additional underlying patterns in the data while still addressing severe sparsity. Under this perturbation, logistic regression demonstrated greater stability, with a modest accuracy improvement of 0.2%, while the CNN showed a more substantial accuracy increase of 1.7%. This outcome may be due to the CNN's ability to leverage the additional features more effectively through its hierarchical learning structure, while logistic regression's simpler nature makes it less sensitive to subtle changes in feature availability. These results highlight the CNN's adaptability and capacity to exploit enriched datasets, while logistic regression benefits from its robustness under stricter missing data thresholds.

Table 2: Model Performance Comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**FNR**
cnn	0.975	0.401	0.701	0.511	0.980	0.299
logreg	0.895	0.130	0.831	0.225	0.946	0.169

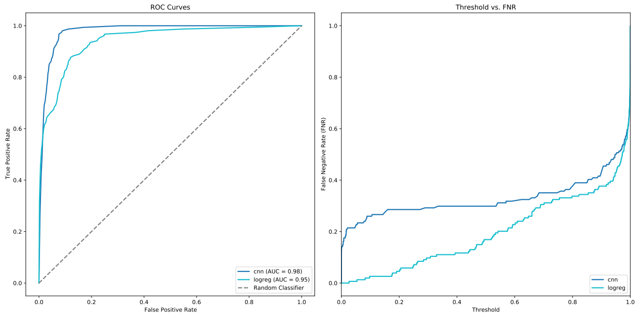

Figure 5: ROC and Threshold after perturbation

11 Reality Check

The predictive models developed in this study show promise in assisting doctors with decision-making but face limitations in real-world clinical applications. Variability in diagnostic equipment, practitioner expertise, and patient populations can significantly influence model performance across diverse healthcare settings. Limited access to high-quality data or advanced tools in resource-constrained environments may further compromise accuracy. Additionally, differences in practitioner training and experience could affect how model outputs are integrated into clinical workflows. Moreover, patient populations may differ from those represented in the dataset used to train the models, raising concerns about generalizability.

To address these challenges, future efforts should focus on external validation using diverse datasets, the development of user-friendly decision-support tools, and close collaboration with clinicians. These steps are essential to ensure the models are robust, adaptable, and effective in enhancing clinical outcomes.

12 Posthoc

12.1 CNN Model

To identify the top features from the CNN model, we can use the saliency map generated during the backpropagation process as Figure 6. This map highlights the features that contribute the most to the prediction.

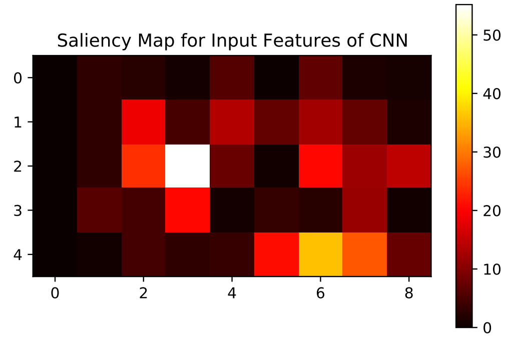

Figure 6: Saliency map for input features of CNN

Table 3 highlights the most important features identified by the CNN model based on the saliency analysis. The top features are ‘EDDisposition‘, ‘AMS‘ and ‘SFxPalp‘. Features such as ‘Drugs‘ and ‘ClavOcc‘ suggest additional predictive relevance. These indicated that we should prioritize these features in clinical evaluations, when deciding whether to perform a CT. float

Table 3: Top Features and Their Importance Scores

Feature	**Importance**
EDDisposition	70.169
AMS	46.525
SFxPalp	23.044
Drugs	19.725
ClavOcc	18.773

12.2 Logistic Regression Model

To perform a posthoc analysis of the logistic regression model, we analyzed the feature coefficients to determine their importance in predicting the outcome. The absolute magnitude of the coefficients was used to rank the features, identifying the top contributors to the model's predictions.

Figure 7 illustrates the top 10 features ranked by their importance, as determined by the logistic regression model. The most influential features include `AgeinYear`, `AgeInMonth`, and `AMS`, which align with key clinical indicators of risk.

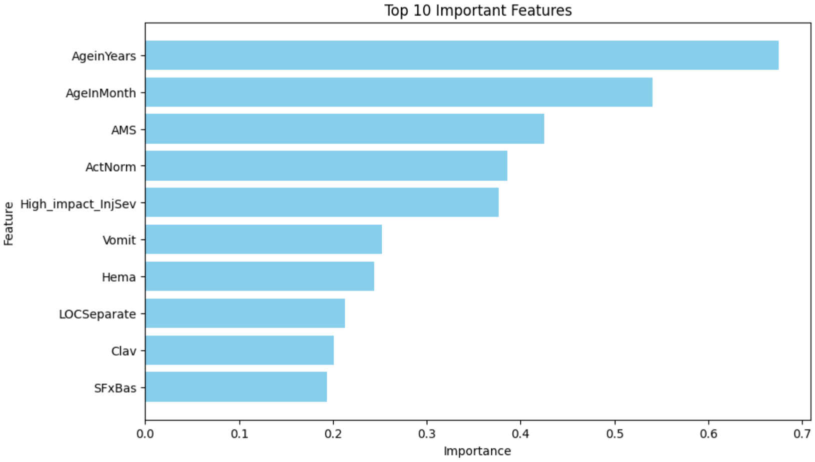

Figure 7: Top 10 Important Features Identified by Logistic Regression

Table 4 summarizes the top 10 features and their corresponding coefficients. The coefficient's sign indicates whether the feature positively or negatively contributes to the prediction, while the magnitude reflects its overall importance. Notably, `AgeinYear` has the largest positive contribution, while `AgeInMonth` contributes negatively.

Key Insights:

Table 4: Top 10 Features and Their Coefficients from Logistic Regression

Feature	**Coefficient**	**Importance (Magnitude)**
AgeinYear	0.6756	0.6756
AgeInMonth	-0.5402	0.5402
AMS	0.4254	0.4254
ActNorm	-0.3860	0.3860
High_impact_InjSev	0.3771	0.3771
Vomit	0.2523	0.2523
Hema	0.2442	0.2442
LOCSeparate	0.2128	0.2128
Clav	0.2007	0.2007
SFxBas	0.1938	0.1938

- **AgeinYear and AgeInMonth**: The age-related variables dominate the model, reflecting their importance in assessing traumatic brain injury risk.
- **AMS and ActNorm**: Altered mental status (`AMS`) and abnormal behavior (`ActNorm`) are strong predictors, highlighting neurological symptoms' critical role.
- **High_impact_InjSev and Vomit**: These features emphasize the role of trauma severity and physical symptoms in determining risk.

12.2.1 Additional Analyses and Extensions

To enhance interpretability, further analyses can include:

- **Subgroup Analysis**: Investigating feature importance across different patient subgroups (e.g., age, gender) to ensure model generalizability and fairness.
- **Feature Interaction Analysis**: Evaluating potential interactions between key features to identify synergistic effects that may impact predictions.
- **Clinical Integration**: Developing decision-support tools that integrate these findings, enabling clinicians to quickly assess patient risk using the top features.

These steps can bridge the gap between model insights and practical application, facilitating better outcomes in patient care and resource allocation.

13 Academic honesty

13.1 Statement

This report reflects our dedication to academic integrity and transparency. We affirm that all data analysis procedures described were independently conceived, designed, and executed by us. The text, figures, and results are entirely our original work. To promote reproducibility, we have carefully documented our thought processes, methodologies, decisions, and assumptions underlying the workflow.

We recognize the significance of acknowledging external contributions and confirm that all sources influencing our work have been properly cited. Our commitment to accountability and honesty ensures the authenticity and reliability of the research presented in this report.

13.2 Collaborators

No collaboration outside of the authors have been used in the present document.

13.3 LLM Usage

Coding

We utilized ChatGPT from OpenAI for consulting and troubleshooting related to the unbalanced sampling issues.

Writing

We utilized ChatGPT from OpenAI for polishing the language of the document.