Lab 1 - PECARN TBI Data, STAT 214, Spring 2025

February 22, 2025

1 Introduction

For Traumatic brain injury (TBI) diagnosis, clinically-important brain trauma (ciTBI) is critical in determining the need for medical care, especially using computed tomography (CT) scans. While CT is a powerful tool for identifying brain damage, it also carries potential risks including exposure to ionizing radiation. Therefore, there is a need to develop predictive models to help determine when CT scans are truly needed.

In this report, as part of the Pediatric Emergency Applied Research Network (PECARN) study, we performed data cleaning, exploratory data analysis, and rough modeling on this traumatic brain injury (TBI) dataset. The dataset contains patient demographic information, mechanisms of injury, clinical signs, symptoms, and CT scan results. Research on this dataset has significant meaning in academic research and clinical diagnosis. By identifying predictors of ciTBI, we can improve clinical decision-making, potentially reduce unnecessary CT scans, minimize radiation exposure, and optimize resource allocation in the emergency department.

The purpose of data cleaning and exploratory data analysis is to identify missing values, inconsistencies, and distribution patterns in the dataset. We will then do rough modeling part to predict response variable 'PosIntFinal' (identifying whether ciTBI exists for the patient) and whether we should suggest CT scan.

Following this introduction, the structure of the report is as follows:

- In section 2, we will provide an overview of this dataset, collection process, and mainly focus on data cleaning and data exploration.
- In section 3, we will provide three findings during data cleaning and the EDA process.
- In section 4, we will build the predictive model with Random Forest and Backward Logistic Regression.
- In section 5 and 6, 2e will discuss certain topics and summarize the conclusions.

2 Data

The dataset in this study came from the Pediatric Emergency Care Applied Research (PECARN). It includes clinical data on children under the age of 18 who presented to the emergency department with minor head trauma. Variables include patient demographics, injury details, symptoms, neurological assessment, and CT results.

The dataset contains 43,399 samples and 124 variables. The CSV file occupies about 41.4 MB of memory, and the feature data types in this file are float64 and int64. In the dataset, each sample represents information about one patient. These variables provide a comprehensive view of the factors that influence TBI and the decision to perform a CT scan. For each class of variables, there is one principal variable and several detail variables.

This dataset is highly relevant to solving the problem of predicting ciTBI and optimizing CT scan usage. By analyzing patient characteristics and injury-related variables, our final goal is to develop models that assist in determining which patients require CT scans and which can safely avoid unnecessary radiation exposure. This is important for improving emergency care efficiency and minimizing risks associated with the overuse of CT imaging.

2.1 Data Collection

The data was collected as part of a prospective observational cohort study conducted by the PECARN. Trained site investigators and other emergency department physicians recorded patient history, injury mechanism, and symptoms and signs on a standardized data form before knowing imaging results (if imaging was done).

Patients were admitted to the hospital at the emergency department physician's discretion, and their records were reviewed by research coordinators and site investigators to assess CT results and the presence of ciTBIs. For the followup process, to identify missed traumatic brain injuries, research coordinators did standardized telephone surveys of guardians of patients discharged from the emergency department between 7 and 90 days after the emergency department visit. Medical records and imaging results were obtained if a missed traumatic brain injury was suggested at follow-up. If a ciTBI was identified, the patient's outcome was classified accordingly. If unable to contact the patient's guardian, they reviewed the records to ensure that no discharged the patient was subsequently diagnosed with ciTBI.

The measurement of each variable in the data is volatile. Among 124 variables in the dataset, 121 are categorical variables. The category numbers range from 2 to more than 10. Some of them are unbalanced. It is worth noting that, for the variable *PosIntFinal*, it is also quite unbalanced that "1" (represents Yes for ciTBI) only occupies around 1 % of all records.

For the measurement and distribution of variables in the dataset, we will discuss more in the following parts.

2.2 Data Cleaning

This dataset consists of groups of highly related variables, missing data, non-applicable labeled variables and replicate information. After we import the dataset and learn about the data collection process, we go through the following steps to examine the data and create action items based on the data cleaning guidelines in Chapter 4 of Veridical Data Science.

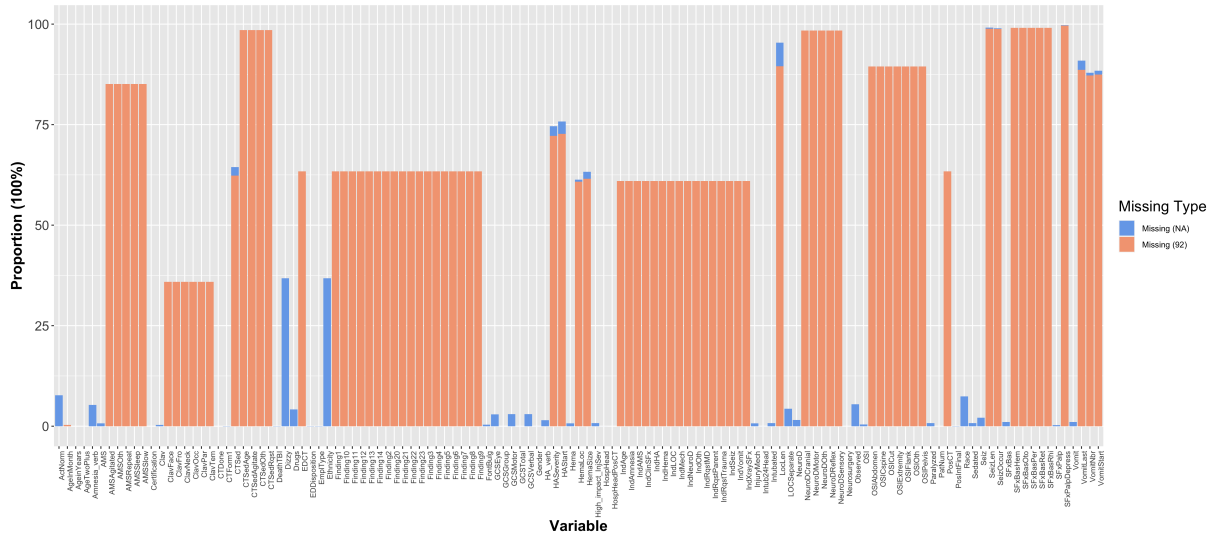
- **Invalid or inconsistent values**: Among 124 variables, 121 of them are categorical variables, so they represent a given meaning within the context setting. However, there are 84 variables that have the category: [90] other, [91] Pre-verbal/Non-verbal, and [92] Not applicable. And in some features, the [92] takes more than 50 % of records. Some of these situations are equivalent to NA, while some have a specific meaning and can not be transformed. (For example, the *VomitStart*, signifying when the patient starts to vomit, will be 92 if *Vomit* equals 0, meaning that the patient hasn't even vomited.) So we should check these [90], [91] and [92] case by case
- **Improperly formatted missing values**: this term is highly associated with the last term, as some invalid or inconsistent values actually refer to missing data. We will then organize the missing data and calculate the missing percentage of each column.
- **Nonstandard data format**: for a tidy dataset, each row corresponds to the data for a single observational unit, and each column corresponds to a unique type of measurement. After the simple check, we can find that this dataset satisfies this standard.
- **Messy column names**: after a simple check, we could find that the column names of this dataset are roughly tidy. All of the joint words are joined as the first upper letter, and the others are lower letters. Moreover, variables belonging to a certain group also share the same prefix, such as "Vomit" and "Gcs." This can help us better signify variables.
- **Improper variable types**: after checking, except for the four numeric variables we mentioned before, the other 121 variables are all categorical variables.
- **Incomplete data:**: data for which every observational unit should appear exactly once. We can check whether duplicate rows exist and drop them.

After a general check following this guideline, we can find that the major problem of this dataset is a great number of improperly formatted missing values and invalid or inconsistent values. Therefore, the data cleaning function we designed for this dataset can be roughly divided into three parts.

2.2.1 Missing, Inconsistent Data and Special Values (90,91,92)

For the 124 features, the missing data (NA and non-applicable) occupies a significant part. While NA is limited, [92] non-applicable occupies a larger part as another kind of missing data. So we will go through the missing data in the following steps.

Figure 1: Missing Values Proportion per Variable (Original)



For the features with category 92, it signifies whether the term is missing or this feature is not applicable since the *Prime Feature* of this group of features has a negative (usually represented by 0) category. The *Prime Feature* for signifying where features in this group are not applicable contains the list as below. We will either retain 92 (validating it's truly not applicable) or convert them into missing data NA based on these prime variables as follows:

- ***LOCSeparate, Seiz, HAverb, Vomit, AMS, SFxPalp, SFxBaws, Hema, Clav, NeuroD, OSI, CTForm1, CTDone***

Then, we go through the rows to check if inconsistent data exists. Based on our previous analysis, if a prime variable equals 0, there should not be any category for the corresponding sub-variables except 92 or NA. Therefore, we will delete rows with such cases. Simultaneously, we can also fill the NA sub-variables with 92 to represent the missing data due to their lack of applicability. In this way, we will both drop inconsistent data and fill in missing data correlated with the main variable.

Apart from the [92], the [90] represents 'Other' and the [91] represents 'Pre-verbal/Non-verbal.' After we go through the meaning of these variables and the domain knowledge, we can find that because of the existence of the [90] category, we can convert all of NA in the column into [90] if the variable has a [90] category. Moreover, since 'Pre-verbal/Non-verbal' usually represents a not clear-represented and unsure outcome, this situation can be considered similar to missing data.

2.2.2 Column and Row Filtering

After the general process with the missing data and inconsistent data, we will move forward to column and row filtering. We have gained the missing rate for each column and row in the last part. As the VDS Chapter 4 recommends, sometimes we can delete highly missing percentage variables to prepare for the modeling. Therefore, we can remove variables with more than a given threshold, like 50%.

Certain features contain totally duplicate information, so some of them can be removed, which will also help us avoid multilinearity. In groups like

- *GCSTotal, GCSGroup*
- *AgeInMonth, AgeinYears, AgeTwoPlus*

we will only keep one variable in each group. Moreover, a given list of features can also be deleted based on the needs of the researcher. For example, as our modeling goal is to predict whether we should recommend CT, the variables related to CT findings can be deleted. While our goal in this phase is only data cleaning, we will keep them for now.

At the initial step of the data cleaning process, we removed the rows with GCS scores that were out of range. We will further remove duplicate rows and remove rows with the missing rate above a given threshold.

2.2.3 Indicator Generation and Missing Data Imputation

To process missing data, the most simple way is to delete based on no matter columns or rows. However, simply deleting the data can easily introduce bias into the dataset. While we still set parameters in the last part to drop rows and columns with high missing rates, the default threshold I set in the function is 80%, which is comparatively high. And I have to supplement that, with the threshold as 80%, there are no columns and rows dropped, in fact. These two parts are only to enrich the applicability of the data cleaning function.

Two better methods are generating missing indicators and imputing missing data. We can set a range, say 10% to 50%, so that any variables with the missing rate in this range will generate a missing indicator (1 represents the variable that exists, and 0 represents that it's missing). Missing indicators will take effect in models that can not naturally deal with missing data, like classic logistic regression. The advantage of this method is that it will not introduce bias into the dataset. At the same time, it will further increase the number of variables.

Another way is to conduct missing data imputation. Generally, we can impute missing values with the mean value for numeric features, and mode or median for categorical features. However, this method should also be used carefully, as it can also introduce bias into dataset. Therefore in my function default seting, I leave the impute feature list blank. This part is to genralizd the use of data cleaning function, and prepare for the potential modeling in future steps.

The three parts in the *clean.py* are actually roughly divided into 10 steps in the function. If you have any questions about something unclear, please refer to the original code.

2.3 Data Exploration

After we conduct the data cleaning process, we can get the clean data file. The clean data occupies 23.2 MB. It contains 42430 rows and 132 features. Among the 132 features, 122 come from the original features after transformation, and 10 are missing indicators for columns with high missing rates. As the major use of missing indicators is for modeling, we will first focus on the 122 normal variables in most of the Data Exploration.

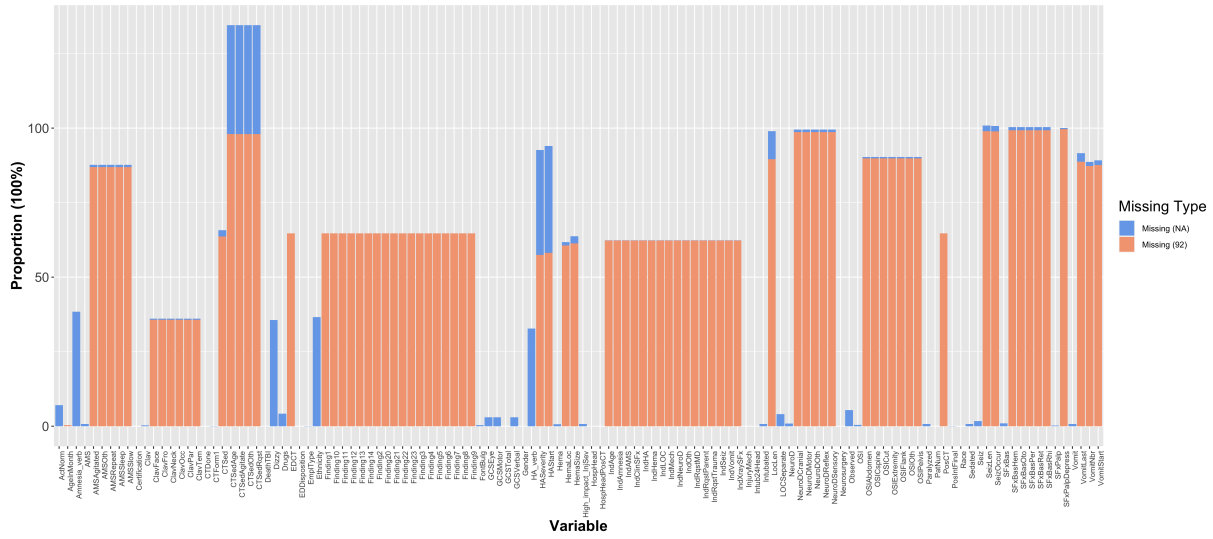
2.3.1 Missing Values Analysis

As we have pointed out, while this dataset has some NA missing values, there are many special values (the majority [92]). Therefore, we can explore the distribution of missing data and visualize the percentage of missing values in each variable.

From the picture, it's clear that most of the missing data are in the form of 92, while only part of them are the default NA. If we sum these two parts up, the missing proportion in each column could be high, sometimes over 90%. So, we should process them carefully when conducting forward analysis and modeling.

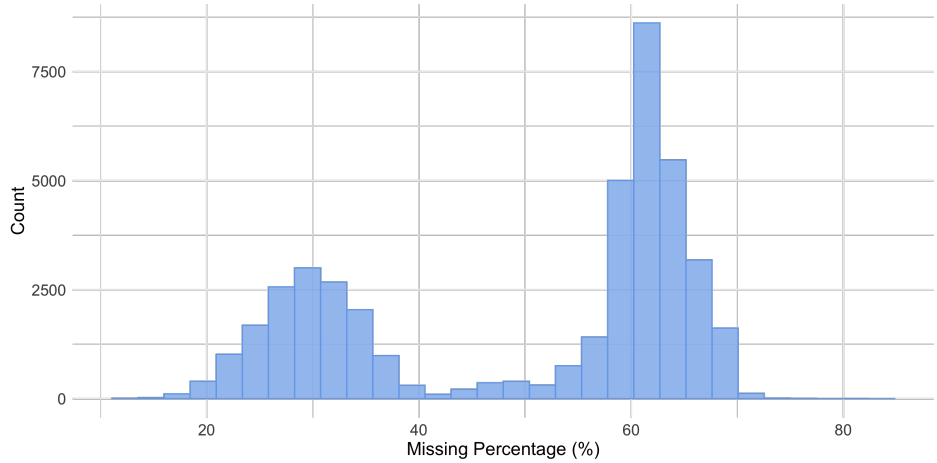
Also, we can see that the missing percentage between variables is not random. The missing percentage consists of some "chunks", as missing cases usually happen for variables in a group instead of missing

Figure 2: Missing Values Proportion per Variable (After Cleaning)



independently. This relation makes sense in the context of domain knowledge, as the data of the same group usually comes from the same source in the data collection process.

Figure 3: Missing Values Proportion per Row Histgram



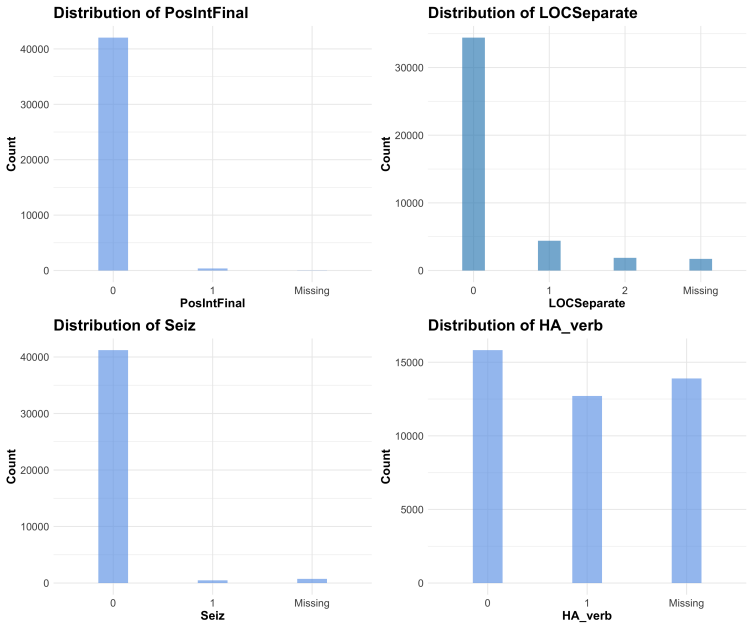
We have calculated the percentage of the missing value per column (for each variable). Likewise, we can calculate the percentage of the missing value per row. The figure shows a bimodal structure. Some of the missing data is concentrated in about 30 %, and the other part of the missing data is concentrated in about 60 %. This figure shows that NA and [92] also occupy a relatively high proportion of the row dimension.

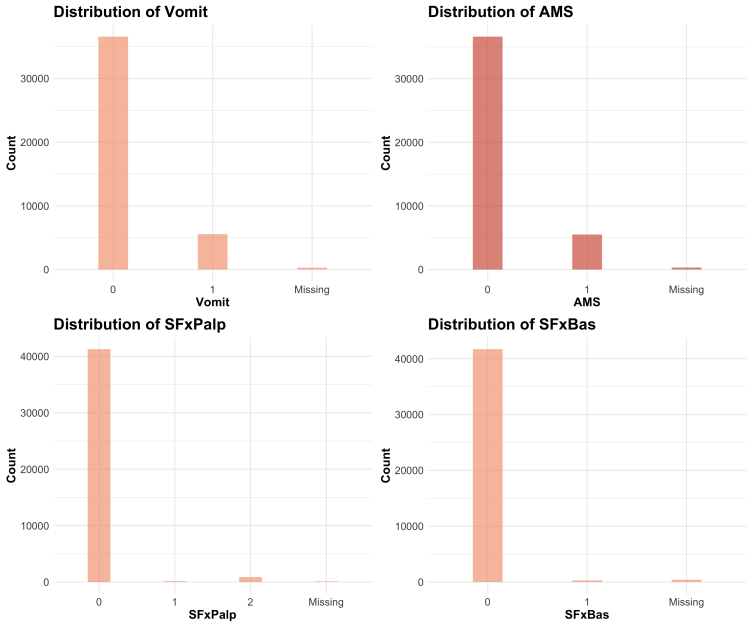
While the clean data only has a limited number of NA, it still has a larger amount of [92] non-applicable across most rows and columns. So, we have to address this missing structure carefully for further analysis.

2.3.2 Key Variables Distribution

Because there are more than 120 variables in the dataset, we can not display all the distribution of variables. Therefore, we choose some key variables and *PosIntFinal* is also included.

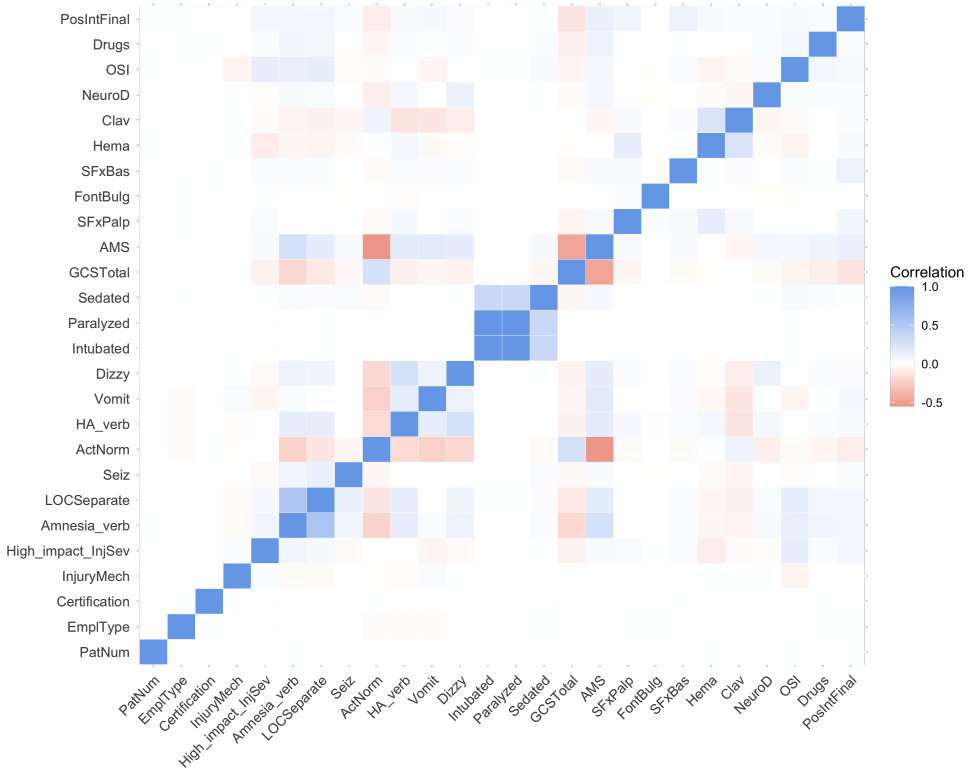
We include NA and [92] in the category of missing data. We can find that across most variables, they are


Distribution of PosIntFinal
Count
PosIntFinal
0
1
Missing
Distribution of LOCSeparate
Count
LOCSeparate
0
1
2
Missing
Distribution of Seiz
Count
Seiz
0
1
Missing
Distribution of HA_verb
Count
HA_verb
0
1
Missing


Distribution of Vomit
Count
Vomit
0
1
Missing
Distribution of AMS
Count
AMS
0
1
Missing
Distribution of SFxPalp
Count
SFxPalp
0
1
2
Missing
Distribution of SFxBas
Count
SFxBas
0
1
Missing

highly imbalanced. It is worth noting that the key variable *PosIntFinal*, which we will predict in modeling, is also highly unbalanced as 0 is far more than 1.

Figure 4: Key Variables Correlation Heatmap



2.3.3 Key Variables Correlation Analysis

As we explained before, missing cases usually happen for variables in a group instead of missing independently. Therefore, a large number of non-applicable [92] will disturb the analysis process and mask the true correlation. Therefore, we should filter out those variables with [92] and only analyze the key variables. The correlation heatmap is as follows.

From the heatmap, we can see that after variable filtering, the left variables show a comparatively weak correlation with each other.

Summary:

To conclude, the dataset, after cleaning, shows characteristics of a high missing values percentage, imbalanced category distribution, strong correlation within-group variables, and weak correlation across different groups.

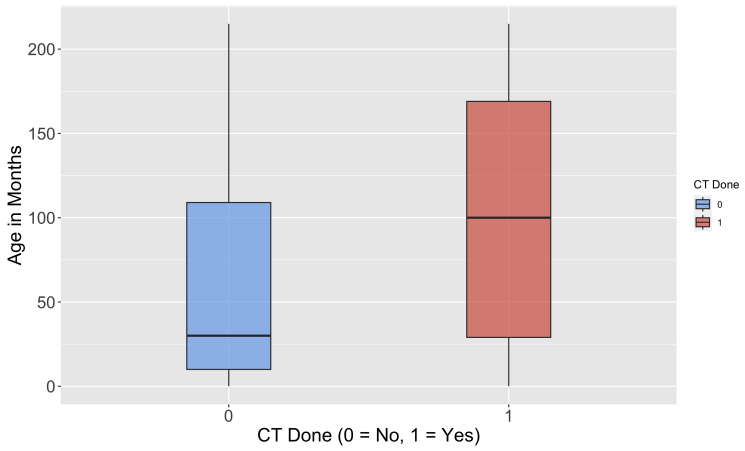
3 Findings

3.1 Find 1: Age Younger for CTDone=0

We can come up with such an intuitive assumption: generally, when the patient's age increases, doctors tend to have CT done because they will be more careful and reservative when having a young child do CT. We can look into the relation of variable *AgeInMonth* and *CTDone*. *CTDone* is a 0-1 variable signifying whether Head CT is performed in ED.

We can first check the CTDone vs Age boxplot.

Figure 5: CTDone vs Age Boxplot

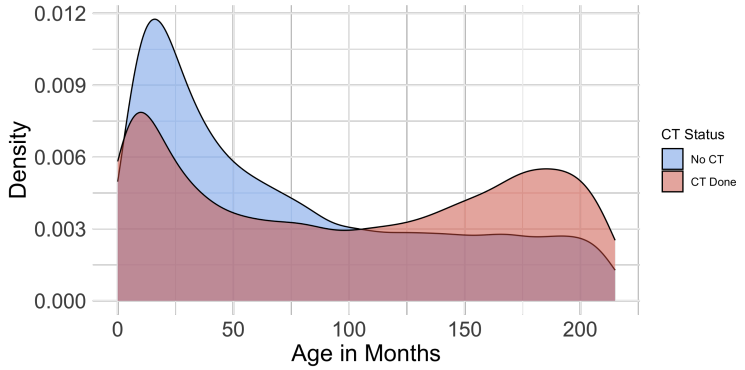


Apparently, we can see that the median for those who take CT is about 100 months, larger than the average for those who do not, around 30 months.

3.2 Finding 2: CTDone Percentage Change over Age

We can plot an Age vs CTDone Density Distribution figure to check the mechanism further.

Figure 6: Age vs CTDone Density Distribution

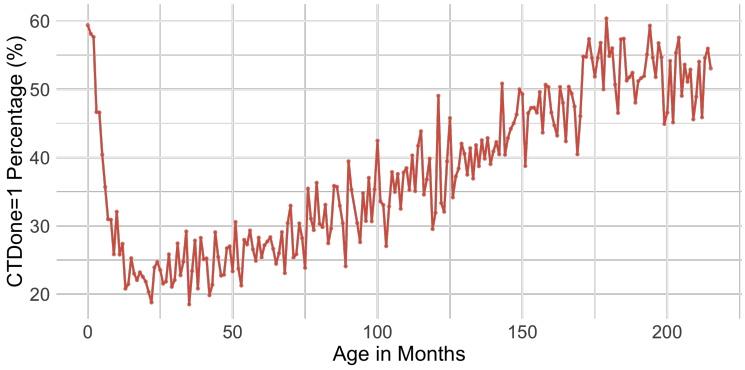


In the figure, we can see that about two years is a changing point. Before about five months, the young babies who take or do not take CT were around 50-50. No CT gained an advantage as the percentage of No CT kept increasing and exceeded CT. After around 25 months (2 years), however, the percentage of No CT began to fall. At about 100 months (8 years old), the CT percentage exceeded 50% and kept the majority.

The Age vs. CT Done Percentage shows this trend more intuitively: The percentage of CTDone=1 first start at 60%, the drops and reaches the bottom at around 2 years with percentage at 20% and rises again to around 50 %.

This trend is sensible considering the domain context. Doctors may tend not to do CT scans on babies younger than 2 years old to better protect their growing period. After 2 years old, children become less vulnerable to CT rays, so they are more likely to have CT scans.

Figure 7: Age vs CTDone=1 Percentage

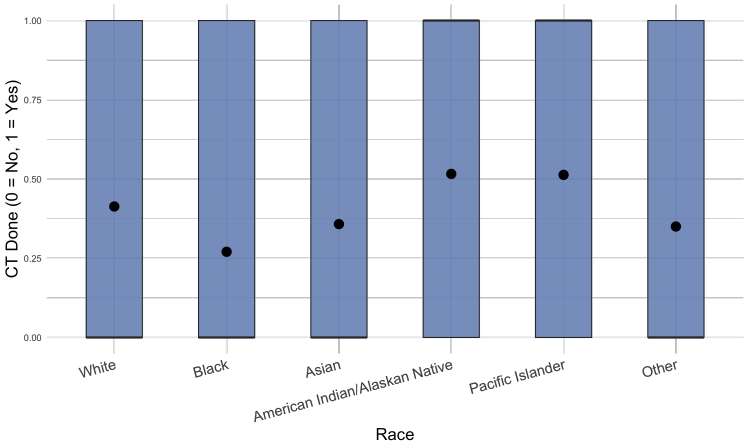


3.3 Finding 3: CTDone Differences across Races

Apart from ages, We can also check how the CTDone changes across different races group. Likewise, we can plot the CTDone vs Race boxplot. Among all racial groups, black patients have the lowest median CT scan rate (0.25), followed by Asians and others (0.38), and White patients (0.43). The highest median rates are observed among American Indian/Alaskan Native and Pacific Islander patients (¿0.5).

This outcome probably can be explained by the hysiological structure difference. However, it is worth noting that this differences requires further investigation into factors such as access to healthcare, socioeconomic status, and institutional policies that may influence CTDone rates.

Figure 8: CTDone vs Races Boxplot



3.4 Reality Check

To conduct a reality check for our findings, we shall compare our results with medical knowledge and datasets related to CT scans in cases of TBI for children.

1. Age and CT usage: In Finding 1, CT scan rates are lower in patients under 2 years old and increase after this age. This is consistent with many medical guidelines, which recommend doctors avoid unnecessary CT scans unless signs of serious injury.

2. Racial differences in CT use: In Finding 2, we observe differences in CT scan rates across ethnic groups. Racial disparities in medical imaging are often associated with differences in healthcare access, provider bias, and socioeconomic factors. Our data did not directly measure these external influences, but

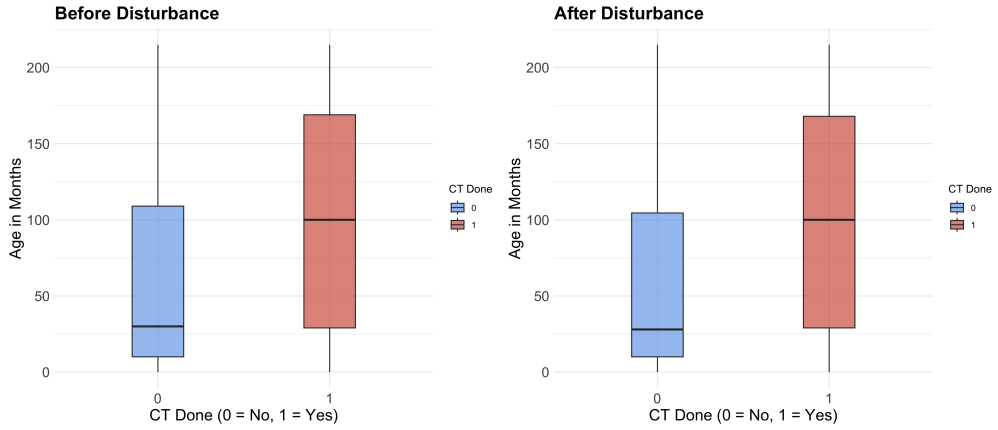
the observed differences can be further explored.

3. Overall data consistency: The trends in our data are generally consistent with medical domain knowledge. However, there might be potential biases in data collection process, missing variables, imbalanced variables, and differences in exogenous variables. We should conduct more analysis to validate our conclusions.

In conclusion, our findings are consistent with established medical guidelines but also require further validation of real-world datasets and further analysis.

3.5 Stability Check

Figure 9: CTDone vs Age Boxplot (Before/ After Disturbance)



For Finding 1, we can add a disturbance by randomly changing the 0-1 CTDone value by 10 % rows in the dataset. After that, we will conduct the CTDone vs. Age analysis. The boxplot we get shows a similar outcome as in the before vs. after disturbance figure.

After checking, we can find that the difference in the median of the CTDone=1 group is within 1 %. The difference in the median of the CTDone=0 group is around 2 % (though still not apparent in the figure). This may be because the 10% rows are a comparatively small disturbance due to the intrinsic relation of Age and CTDone.

4 Modeling

4.1 Implementation

Model 1: Random Forest

One huge advantage of random forest is that it can automatically model the dataset's missing values. At each split node, it will choose the best feature regardless of the missing data. Therefore, this method serves as a good choice for this dataset. Another advantage is its power in variable selection, as this dataset contains more than 120 variables.

Here, we use the R package *Ranger* to build the model (random forest in this package can process missing data). The modeling process is as follows:

- Data Split: 80% for train, 20% for test.
- Variables Selection: roughly select 70 main variables as predictors for the random forest.
- Hyperparameters Selection: use Cross Validation (K=5) to determine Tree numbers N and 0-1 prediction threshold p on train dataset.
- Model Evaluation: major indicators include accuracy, recall, precision, and AUC for train and test.

Generally, in a binary classification task, we will set the classification threshold p=0.5. However, this dataset is severely imbalanced, so we have to set the threshold p properly.

Model 2: Backward Logistic Regression

The logistic model has great advantages in its simple form and great interpretability. However, it requires a more subtle choice of variables. Therefore, we will consider backward regression to help us filter out variables with poor performance.

Here, we use the R package *MASS* for backward regression. Specifically, we will refer to the Feature Importance list in the random forest and choose the top 30 variables as a start. The modeling process is as follows:

- Data Split: 80% for train, 20% for test
- Variables Selection: select the top 30 in the importance list of RF.
- Hyperparameters Selection: use Cross Validation (K=5) to determine 0-1 prediction threshold p.
- Model Evaluation: major indicators include accuracy, recall, precision, and AUC for train and test.

4.2 Interpretability

Model 1: Random Forest

While the decision tree is highly interpretable, random forests improve the generalization ability at the risk of losing some interpretability. While the decision process for a tree is very intuitive, a random forest can be interpreted as a cluster of trees 'voted' for the prediction outcome.

However, variable importance for RF is a good indicator for us to explore the modeling process. Based on Gini Impurity, variable importance clearly shows how each variable performs in the prediction process.

We can use accuracy, recall, precision, and AUC to model the evaluation of this prediction task.

Table 1: RF: Train and Test

Indicator	Train	Test
Accuracy	0.99502	0.97913
Recall	0.99497	0.98643
Precision	1	0.99246
AUC	0.99995	0.87032

Table 2: Logistic Backwards: Train and Test

Indicator	Train	Test
Accuracy	0.98571	0.98502
Recall	0.99224	0.99308
Precision	0.99334	0.99181
AUC	0.79943	0.72010

Model 2: Backward Logistic Regression:

Logistic regression is of simple form and great interpretability, as it will directly predict the probability of $P(y = 1|x)$ via the linear regression form. The stepwise backward regression is also intuitive: Drop one variable of the best prediction performance at each step.

However, variable importance for RF is a good indicator for us to explore the modeling process. Based on Gini Impurity, variable importance clearly shows how each variable performs in the prediction process.

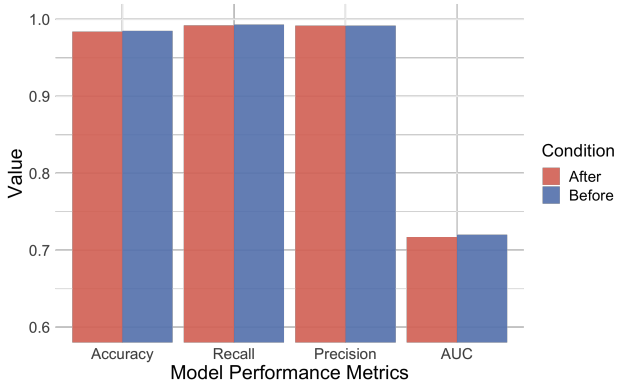
4.3 Stability

We do not use the CTDone variable in the modeling part. Likewise, we can add a disturbance by randomly changing the 0-1 PosInFinal value by 10 % rows in the dataset. The differences between them areare pretty limited. This may be because the dataset itself is so imbalanced that the disturbance of 10 % will not exert a significant impact on the model prediction

Table 3: Logistic Backward: After Disturbance

Indicator	Train	Test
Accuracy	0.98518	0.98413
Recall	0.99171	0.99219
Precision	0.99333	0.99180
AUC	0.80002	0.71721

Figure 10: Logistic Backward: Before/After Disturbance



5 Discussion

During the lab, the data size restriction was mainly in the variable amounts. The main challenge in this lab is was the high missing values (in the form of NA or non-applicable) across all variables. . Due to the complicated structures and highly correlated variables within variable groups, this added great difficulty to the work.

For the realm of data and reality, the dataset of this lab comes from the context of ciTBI diagnosis and recommends whether we should do a CT scan. In the realm of algorithms and models, we use machine learning algorithms, including random forest and logistic regression, to build prediction models. They are not the true reflection of the relation, but "All models are wrong. Some are useful." –The models lead us to the third realm, future data and reality. We are not exposed to future data, but the generalization of the models can lead us to predictions and help us make sensible judgments when the data arises. There is no one-to-one correspondence between the data and reality, nor reality and data visualization. However, data visualization is a powerful tool, just like algorithms, that explores an ideal situation in reality.

6 Conclusion

This PECARN clinical dataset contains observations of children with variables about their Demographics, Injury details, Symptoms, Neurological status, and CT findings. In the process of data cleaning and EDA, we find characteristics of high missing values percentages, imbalanced category distribution, and strong correlation within-group variables. We also conduct rough modeling for prediction tasks. However, we need further steps like subsampling to deal with problems like imbalance and improve our modeling.

7 Academic honesty statement

I make the academic integrity pledge here: All my work in this report is done independently. All sources I used are properly cited, whether from LLMs, papers, textbooks, or classmates.

Academic research honesty is necessary in the academy and also the foundation of our society. It guarantees fairness in research, stimulates research's activeness, and exerts a positive impact on the progress of science and technology. We should always adhere to the rules, which will create a more regulated and benign world.

8 Collaborators

I use GitHub Copilot for several parts of debugging in the data cleaning Python file, ChatGPT for several figure visualization recommendations, and Grammarly for correcting grammar for some sentences in this report.

9 Bibliography

[1] Kuppermann, Nathan et al. "Identification of children at very low risk of clinically-important brain injuries after head trauma: a prospective cohort study." Lancet (London, England) vol. 374,9696 (2009): 1160-70. doi:10.1016/S0140-6736(09)61558-0

[2] https://vdsbook.com/04-data_cleaning