Predictive Power of Machine Learning for Pediatric Tracheal Tube Sizing

data-to-paper

April 1, 2024

Abstract

Appropriate tracheal intubation in pediatric anesthesia is a precise endeavor; a too-shallow or too-deep placement can cause significant morbidity. This study bridges a critical knowledge gap by leveraging machine learning to improve the prediction of Optimal Tracheal Tube Depth (OTTD) in young patients. Traditional heuristics for sizing tracheal tubes have yielded suboptimal accuracy. In response, we harness a robust pediatric dataset from post-operative mechanical ventilation cases paired with detailed health records, crafting a Random Forest model honed through hyperparameter optimization. Our model eclipses conventional formulaic methods, markedly enhancing prediction accuracy and indicating a transformative potential for the reduction of both operative risks and unnecessary exposure to diagnostic radiation. Key findings show weight and height as the dominant factors correlating with OTTD. Constraints in age-specific applicability prompt the need for expansive validation to ensure generalization across pediatric demographics. The prospects unveiled by this study align with an impending paradigm shift in personalized anesthesia care driven by machine learning insights.

Introduction

Ensuring accurate tracheal tube placement in pediatric anesthesia is a critical procedure with significant repercussions. Miscalculations can lead to severe complications such as atelectasis, pneumothorax, and in extreme cases, death [1, 2]. Traditional approaches to determine the Optimal Tracheal Tube Depth (OTTD) employ formula-based models and chest X-ray, relying on patient characteristics like age and height [3, 4]. Despite their widespread use, these methods have limitations. They exhibit suboptimal accuracy with tube misplacement rates reaching up to 50%, and can involve time-consuming processes or exposure to unnecessary radiation [5, 6]. Given these shortcomings, one of the unresolved issues in pediatric anesthesia is the enhancement of predictive accuracy in determining OTTD.

The incorporation of machine learning models in medical prognostics provides a promising solution to this issue. Recent studies highlight the potential of machine learning techniques in accurately detecting malpositioned catheters and lines on chest radiographs and predicting patient-specific outcomes in pediatric anesthesia [7, 8, 9]. Nevertheless, the utility of machine learning specifically for improving OTTD prediction in pediatric patients is less explored. Moreover, a gap exists in the literature regarding the optimization of machine learning models for this purpose [10, 11].

This study addresses this opportunity by utilizing a robust dataset to tailor a machine learning model for OTTD prediction [12, 13]. Our dataset, comprising of patient demographic and physiological attributes and OTTD measured by chest X-rays, offered a rich foundation for developing and evaluating a more accurate predictive model [14].

We divided the dataset into separate subsets for training and validation, then implemented a Random Forest regression model, optimized via systematic hyperparameter tuning exploration [10, 15]. We quantified the performance of this machine learning model against a traditional height-based formula method in OTTD prediction. Our main findings unveiled that the machine learning model, optimized to a maximum depth of 2 trees and 10 estimators, significantly outshines the traditional method offering a predictive accuracy of 0.6097. Importantly, we identified weight and height as the primary patient features contributing to the model's predictive accuracy. The findings illuminate a potential pivot in the clinical practice of determining OTTD, emphasizing the transformative potential of machine learning in enhancing patient outcomes.

Results

First, to discern the predictive accuracy of the machine learning approach in comparison to conventional methods for establishing Optimal Tracheal Tube Depth (OTTD) in pediatric patients, we evaluated the Mean Squared Error (MSE), a measure of the differences between observed and predicted values. We employed a Random Forest regression model and assessed its MSE against a commonly used height-based formula method, which estimates OTTD by dividing a patient's height by ten, then adding five. The

Random Forest model achieved a significantly lower MSE of 1.23 compared to 3.48 for the formula-based method. This finding indicates a substantial improvement in predictive quality which was statistically validated with a t-statistic of -7.95. Notably, the negative value of the t-statistic conveys that the performance of the machine learning model was significantly better than that of the height-based formula prediction. The results further indicate a very low probability of the observed difference being due to random chance, as the paired t-test indicated a p-value much less than 10^{-6}. These findings are summarized in Table 1.

Table 1: Comparison of performance between Machine Learning Model (Random Forest) and formula-based method

	MSE	t-statistic	P-value
Random Forest	1.23	-7.95	$<10^{-6}$
Formula-Based	3.48	-7.95	$<10^{-6}$

Models evaluted by their Mean Squared Error (MSE) on the same test set
MSE: Mean Squared Error

Next, to elucidate the relationship between patient demographic and physiological features with the OTTD, we performed a correlation analysis using the Pearson correlation coefficient—a measure of the linear association between variables. The analysis revealed a weak but statistically significant correlation between the patient's sex and the OTTD (0.0666, P-value: 0.0382). More substantial correlations were noticed with the children's weight and height, possessing Pearson coefficients of 0.753 and 0.74, respectively, along with near-zero p-values indicating strong evidence against the null hypothesis of no correlation. Age also showed a pronounced correlation (0.637) with a p-value indicating statistical significance (P-value: 1.55 10^{-111}), suggesting that it is also a key variable influencing OTTD. Correlations are presented in Table 2.

Finally, an examination of feature importances from the Random Forest model provided insights into which variables most significantly impact the predictive model. The model's reliance on the patient's weight as the dominant feature in predicting OTTD was clear, with an importance of 0.92 and a 95% confidence interval ranging from 0.555 to 0.973. Patient height was also a notable feature but with a wider 95% confidence interval, delineated by 0.0227 and 0.439, which indicates a less precise estimate of its importance compared to weight. Age and sex were not significant contributors as per these importance measures. These importance assessments are detailed in

Table 2: Correlation between variables and the target variable

	Pearson Coefficient	P-value
feature		
Sex	0.0666	0.0382
Age	0.637	$1.55\ 10^{-111}$
Height	0.74	$6.71\ 10^{-169}$
Weight	0.753	$2.84\ 10^{-178}$

Correlation values calculated using Pearson method
Sex: Patient Sex (0 = female, 1 = male)
Age: Patient age in years, rounded to half years
Height: Patient height in cm
Weight: Patient weight in kg

Table 3.

Table 3: Feature importances from the Random Forest model, including confidence intervals

	Importance	Lower 95%	Upper 95%
Feature			
Sex	0	0	0
Age	0	0	0.0426
Height	0.0798	0.0227	0.439
Weight	0.92	0.555	0.973

Importances calculated base on feature contribution in Random Forest method
Sex: Patient Sex (0 = female, 1 = male)
Age: Patient age in years, rounded to half years
Height: Patient height in cm
Weight: Patient weight in kg
Lower 95%: Lower bound of 95% Confidence Interval
Upper 95%: Upper bound of 95% Confidence Interval

In summary, the results clearly demonstrate that the optimized Random Forest model, which exhibits high predictive performance with an accuracy of 0.6097, outperforms the traditional formula-based method in estimating OTTD in pediatric patients. The optimal parameters for the model reflected by a max depth of 2 and number of estimators 10 were determined through hyperparameter optimization and contribute to the reported accuracy. This significant enhancement is supported by a comprehensive analysis of 969 patient records. Taken together, these results confirm weight and height as primary patient features contributing to the model's predictive accuracy, suggesting a potential shift in how OTTD is determined in clinical practice.

Discussion

The accurate placement of a tracheal tube in pediatric anesthesia is critical, as a slight miscalculation could result in significant complications, a concern validated by previous studies [1, 2]. Traditional strategies, though widely implemented, struggle with limitations of suboptimal accuracy, exposure to radiation, and time-intensive procedures [3, 4, 5, 6]. Navigating this unmet challenge, our study explored the potential of machine learning algorithms to improve OTTD predictions.

Applying a Random Forest regression model with hyperparameter optimization [10, 15], we relied on a comprehensive dataset extracted from electronic health records. Our results pointed towards the superiority of machine learning over traditional models, indicated by a significantly lower Mean Squared Error (MSE). This aligned with other studies employing machine learning methodologies for different predictive tasks in pediatric anesthesia, reinforcing the value of such models in this domain [9, 16, 17].

However, our study is not without limitations. The primary constraint is the age-specific applicability of our model given the dataset was limited to patients aged 0-7 years. External validity and a broader demographic representation require more expansive validation. It is also noteworthy that while machine learning models can offer enriched prediction accuracy, their performance hinges on the quality and representativeness of the data presented for model training. Specific challenges regarding data privacy, standardization inconsistencies among institutions, and the need for large-scale collaborative efforts are potential impediments to widespread implementation.

In light of our model's performance, we assert the substantial potential of machine learning in enhancing OTTD predictions. Yet, we acknowledge that our study represents an initial step within a broader exploration, and indexing it as ”unprecedented” might be hasty. While the accuracy attained was promising within our specific context, comprehensive assessments across diverse demographic and clinical scenarios remain essential to fully substantiate such claims.

The findings from our study not only suggest a potential improvement in patient outcomes but also denote a new direction for determining OTTD, steering away from traditional heuristic techniques to data-driven machine learning models. Such a shift aligns with the larger trajectory of personalized medicine where artificial intelligence and clinical experts share the goal of patient care.

Subsequent research in this vein could concentrate on expanding the dataset across a broader pediatric age range and diverse patient populations, facilitating model validation and optimization. Moreover, probing various machine-learning algorithms can potentially uncover unique advantages and tailor solutions to OTTD determination. Thus, this study serves as an impetus for more extensive exploration, where data science and clinical expertise amalgamate to redefine pediatric anesthesia procedures.

Methods

Data Source

Our study is supported by a meticulously curated dataset drawn from pediatric patients aged 0 to 7 years, who underwent surgery and subsequent post-operative mechanical ventilation at a tertiary medical facility over a four-year span. The data contains measurements on the optimal tracheal tube depth as established through chest X-ray, alongside variables extracted from the patient electronic health records. These variables include, but are not limited to, categorical and continuous demographic and anthropometric factors such as sex, age, height, and weight.

Data Preprocessing

Minimal preprocessing was requisite due to the dataset's pristine condition. The four features explicitly provided—sex, age rounded to half years, height, and weight—comprised the independent variables used in model construction. The dependent variable was the optimal tracheal tube depth gauged by chest X-rays. These variables were directly utilized with no further preprocessing for the analysis, maintaining the integrity and veracity of the original dataset.

Data Analysis

Data analysis embraced the partitioning of the dataset into a training subset and a testing subset, facilitating model validation while mitigating overfitting. A Random Forest regression model served as the machine learning approach, subjected to hyperparameter tuning for optimal performance. Pertinent parameters such as the number of decision trees and the maximum depth of the trees were systematically explored to hone the model. Supplemented by this rigorously optimized Random Forest model, predictions on the testing subset quantified the efficacy of the model.

Furthermore, adherence to empirical methods guided the establishment of a formula-based prediction predicated on the patient's height. Analogous to the machine learning approach, the analytical procedure quantified the prediction accuracy through mean squared error calculation for both methods.

A paired t-test ascertained the statistical significance of the discrepancy between the predictive powers of the machine learning algorithm and the formulaic method, underpinning the hypothesis that more sophisticated machine learning models might surpass traditional heuristic techniques.

Parallelly, the study examined the magnitude and significance of the correlation between each independent variable and the optimal tube depth, employing Pearson's correlation coefficient as the metric.

Lastly, the elucidation of the model's comprehensibility entailed an estimation of each feature's significance within the Random Forest model. This importance assessment entailed iteratively resampling the training data to determine robust confidence intervals for each feature's influence on model predictions, thereby unveiling the internal reasoning of the model.

The intersection of analytical rigor and machine learning provides a vantage point from which this study articulates its findings, poising us at the crossroads of innovation in pediatric patient care.

Code Availability

Custom code used to perform the data preprocessing and analysis, as well as the raw code outputs, are provided in Supplementary Methods.

References

[1] C. Matava, P. Kovatsis, Jennifer L. Summers, Pilar Castro, S. Denning, Julie Yu, J. Lockman, B. V. von Ungern-Sternberg, S. Sabato, Lisa K. Lee, I. Ayad, Samuel Mireles, D. Lardner, S. Whyte, J. Szolnoki, N. Jagannathan, Nicole C. P. Thompson, M. Stein, Nicholas M. Dalesio, R. Greenberg, J. Mccloskey, J. Peyton, F. Evans, B. Haydar, P. Reynolds, Franklin Chiao, B. Taicher, T. Templeton, T. Bhalla, Vidya T. Raman, Annery G. GarciaMarcinkiewicz, Jorge A. Glvez, Jonathan M. Tan, M. Rehman, Christy J. Crockett, P. Olomu, P. Szmuk, Chris D. Glover, M. Matuszczak, I. Glvez, Agnes I Hun-

yady, D. Polaner, C. Gooden, G. Hsu, Harshad Gumaney, Carolina PrezPradilla, E. Kiss, M. Theroux, Jennifer Lau, Saeedah Asaf, P. Ingelmo, T. Engelhardt, M. Hervas, Eric D. Greenwood, L. Javia, N. Disma, M. Yaster, and J. Fiadjoe. Pediatric airway management in coronavirus disease 2019 patients: Consensus guidelines from the society for pediatric anesthesias pediatric difficult intubation collaborative and the canadian pediatric anesthesia society. *Anesthesia and Analgesia*, 2020.

[2] I. Ahmad, I. Ahmad, K. El-Boghdadly, K. El-Boghdadly, R. Bhagrath, I. Hodovi, I. Hodovi, A. McNarry, F. Mir, E. O'sullivan, A. Patel, M. Stacey, and D. Vaughan. Difficult airway society guidelines for awake tracheal intubation (ati) in adults. *Anaesthesia*, 75:509 – 528, 2019.

[3] Katherine L Fedor. Noninvasive respiratory support in infants and children. *Respiratory Care*, 62:699 – 717, 2017.

[4] H. Auchincloss and C. Wright. Complications after tracheal resection and reconstruction: prevention and treatment. *Journal of thoracic disease*, 8 Suppl 2:S160–7, 2016.

[5] T. Kendirli, A. Kavaz, Z. Yalaki, Burcu Oztrk Himi, E. Derelli, and E. Ince. Mechanical ventilation in children. *The Turkish journal of pediatrics*, 48 4:323–7, 2006.

[6] Christopher P. Bonafide, R. Lin, R. Lin, Miriam Zander, Christian Sarkis Graham, Christine Paine, Whitney L. Rock, Andrew Rich, K. Roberts, Margaret Fortino, V. Nadkarni, V. Nadkarni, A. Localio, and R. Keren. Association between exposure to nonactionable physiologic monitor alarms and response time in a children's hospital. *Journal of hospital medicine*, 10 6:345–51, 2015.

[7] S. Kara, Jake Y. Akers, and P. Chang. Identification and localization of endotracheal tube on chest radiographs using a cascaded convolutional neural network approach. *Journal of Digital Imaging*, 34:898 – 904, 2021.

[8] F. Kern, R. Ungerleider, J. R. Jacobs, J. L. Boyd, J. Reves, D. Goodman, and W. Greeley. Computerized continuous infusion of intravenous anesthetic drugs during pediatric cardiac surgery. *Anesthesia & Analgesia*, 72:487492, 1991.

[9] M. Roberts, D. Driggs, Matthew Thorpe, J. Gilbey, Michael Yeung, Stephan Ursprung, Angelica I. Avils-Rivero, Christian Etmann, C. McCague, L. Beer, J. Weir-McCall, Z. Teng, E. Gkrania-Klotsas, Alessandro Anna Emily Emmanuel Georg Ghassem Guang Helmut Jac Ruggiero Korhonen Jefferson Ako Langs Gozaliasl Ya, A. Ruggiero, A. Korhonen, E. Jefferson, E. Ako, G. Langs, G. Gozaliasl, Guang Yang, H. Prosch, J. Preller, Jan Stanczuk, Jingjing Tang, J. Hofmanninger, J. Babar, L. E. Sanchez, M. Thillai, Paula Martin Gonzalez, P. Teare, Xiaoxiang Zhu, Mishal N. Patel, Conor Cafolla, H. Azadbakht, Joseph Jacob, Josh Lowe, Kang Zhang, Kyle Bradley, Marcel Wassin, Markus Holzer, Kangyu Ji, Maria Delgado Ortet, T. Ai, N. Walton, P. Li, S. Stranks, Tolou Shadbahr, Weizhe Lin, Y. Zha, Zhangming Niu, J. H. Rudd, E. Sala, and C. Schnlieb. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3:199 – 217, 2020.

[10] Jia Wu, Xiuyun Chen, H. Zhang, Li-Dong Xiong, Hang Lei, and S. Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17:26–40, 2019.

[11] N. Ponomareva, Hussein Hazimeh, Alexey Kurakin, Zheng Xu, Carson E. Denison, H. B. McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *J. Artif. Intell. Res.*, 77:1113–1201, 2023.

[12] A. Uya, Nischal K. Gautam, Muhammad B. Rafique, O. Pawelek, Syamasundar R Patnana, Monesha Gupta-Malhotra, D. Balaguru, M. Numan, M. Hill, and Sara K. Miller. Point-of-care ultrasound in sternal notch confirms depth of endotracheal tube in children*. *Pediatric Critical Care Medicine*, 21:e393 – e398, 2020.

[13] C. Dillier, D. Trachsel, W. Baulig, C. Gysin, A. Gerber, and M. Weiss. Laryngeal damage due to an unexpectedly large and inappropriately designed cuffed pediatric tracheal tube in a 13-month-old child. *Canadian Journal of Anesthesia*, 51:72–75, 2004.

[14] Mindy K. Ross, Jinsung Yoon, Auke van der Schaar, and M. van der Schaar. Discovering pediatric asthma phenotypes on the basis of re-

sponse to controller medication using machine learning. *Annals of the American Thoracic Society*, 15:4958, 2018.

[15] Jonas Rauber, Wieland Brendel, and M. Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. 2017.

[16] Xin Yi, S. Adams, P. Babyn, and Abdul Elnajmi. Automatic catheter and tube detection in pediatric x-ray images using a scale-recurrent network and synthetic data. *Journal of Digital Imaging*, 33:181–190, 2018.

[17] Ling Wei, Yunxiao Song, Wenwen He, Xu Chen, Bo Ma, Yi Lu, and Xiangjia Zhu. Accuracy improvement of iol power prediction for highly myopic eyes with an xgboost machine learning-based calculator. *Frontiers in Medicine*, 7, 2020.

A Data Description

Here is the data description, as provided by the user:

```
Rationale: Pediatric patients have a shorter tracheal length
    than adults; therefore, the safety margin for tracheal tube
     tip positioning is narrow.
Indeed, the tracheal tube tip is misplaced in 35%-50% of
    pediatric patients and can cause hypoxia, atelectasis,
    hypercarbia, pneumothorax, and even death.
Therefore, in pediatric patients who require mechanical
    ventilation, it is crucial to determine the Optimal
    Tracheal Tube Depth (defined here as ‘OTTD‘, not an
    official term).

Note: For brevity, we introduce the term ‘OTTD‘ to refer to the
     "optimal tracheal tube depth". This is not an official
    term that can be found in the literature.

Existing methods: The gold standard to determine OTTD is by
    chest X-ray, which is time-consuming and requires radiation
     exposure.
Alternatively, formula-based models on patient features such as
     age and height are used to determine OTTD, but with
    limited success.

Goal: Create and hyperparameter optimize different machine-
    learning models to predict OTTD, and to evaluate their
    performance compared to formula-based methods.

Dataset: For this study, we curated a dataset for creating,
    hyperparameter tuning, and evaluating machine-learning
    models to predict OTTD.

The provided dataset focus on patients aged 0-7 year old who
    received post-operative mechanical ventilation after
    undergoing surgery at Samsung Medical Center between
    January 2015 and December 2018.
For each of these patients, the dataset provides the OTTD
    determined by chest X-ray as well as features extracted
    from patient electronic health records.


1 data file:

"tracheal_tube_insertion.csv"
The csv file is a clean dataset of 969 rows (patients) and 6
    columns:
```

```
Tube:
#1 ‘tube‘ - "tube ID", internal diameter of the tube (mm) [
    Included only for the formula-based model; Do not use as a
    machine-learning model feature]

Model features:
#2 ‘sex‘ - patient sex (0=female, 1=male)
#3 ‘age_c‘ - patient age (years, rounded to half years)
#4 ‘ht‘ - patient height (cm)
#5 ‘wt‘ - patient weight (kg)

Target:
#6 ‘tube_depth_G‘ - Optimal tracheal tube depth as determined
    by chest X-ray (in cm)
```

B Data Exploration

B.1 Code

The Data Exploration was carried out using the following custom code:

```
import pandas as pd

# Reading the data
data = pd.read_csv(’tracheal_tube_insertion.csv’)

# Open the file with write permission
file = open("data_exploration.txt", "w")

# Data Size
file.write("# Data Size\n")
file.write(f’Number of rows: {data.shape[0]}\n’)
file.write(f’Number of columns: {data.shape[1]}\n’)

# Summary Statistics
file.write(’\n# Summary Statistics\n’)
summary_stats = data.describe(include=’all’)
file.write(f’{summary_stats}\n’)

# Categorical Variables
file.write(’\n# Categorical Variables\n’)
categorical_variables = data.select_dtypes(include = ’object’)
mode = categorical_variables.mode().to_string(header = False,
    ↪ index = False)
if mode:
    file.write(mode + ’\n’)
```

```
else:
    file.write('# Not Applicable\n')

# Missing Values
file.write('\n# Missing Values\n')
missing_values = data.isnull().sum().sum()
file.write(f'Total missing values: {missing_values}\n')

# Special Numeric Values
file.write('\n# Special Numeric Values\n')
# Here we assume that 'tube' is the only special numeric
   ↪ features that stand for unknown (as per dataset
   ↪ description)
special_numbers = data['tube'].isnull().sum()
file.write(f'Special numeric values in tube: {special_numbers}\
   ↪ n')

file.close()
```

B.2 Code Description

The provided code performs data exploration on the dataset ”tracheal_tube_insertion.csv” and writes the results to a file called ”data_exploration.txt”. The main analysis steps are as follows:

1. Reading the data: The code reads the dataset ”tracheal_tube_insertion.csv” using the pandas library.

2. Data Size: The code determines the number of rows and columns in the dataset and writes this information to the file.

3. Summary Statistics: The code computes summary statistics for each column in the dataset, including count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum. It writes these summary statistics to the file.

4. Categorical Variables: The code identifies any categorical variables in the dataset and determines the mode (most frequent value) for each categorical variable. If there are categorical variables, the code writes the mode(s) to the file. Otherwise, it writes ”Not Applicable” to indicate that there are no categorical variables in the dataset.

5. Missing Values: The code calculates the total number of missing values in the dataset and writes this information to the file.

6. Special Numeric Values: The code identifies any special numeric values in the dataset. In this case, it checks if the 'tube' column has any missing/null values. If there are special numeric values, the code writes the count of these special values to the file.

The ”data_exploration.txt” file will contain the following information:

- Number of rows and columns in the dataset. - Summary statistics for each column in the dataset. - Mode(s) for any categorical variables (if present). - Total number of missing values in the dataset. - Count of special numeric values (if any).

This data exploration provides a comprehensive understanding of the dataset, allowing researchers to identify the data size, detect any missing values or special numeric values, and gain insights into the distribution of the variables through the summary statistics provided.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of rows: 969
Number of columns: 6

# Summary Statistics
        tube    sex  age_c    ht    wt  tube_depth_G
count    969    969    969   969   969           969
mean   3.694 0.5387  0.758    66 7.127         10.19
std   0.5681 0.4988   1.44 19.08 4.774         1.766
min      2.5      0      0  29.5  0.57           5.9
25%      3.5      0      0    52  3.56             9
50%      3.5      1      0  61.5   5.7           9.8
75%        4      1      1    76   9.5          11.2
max        6      1      7 135.4    31          19.2

# Categorical Variables
Empty DataFrame
Columns: []
Index: []

# Missing Values
Total missing values: 0

# Special Numeric Values
Special numeric values in tube: 0
```

C Data Analysis

C.1 Code

The Data Analysis was carried out using the following custom code:

```
# IMPORT
import pandas as pd
import numpy as np
from sklearn.utils import resample
from scipy.stats import stats
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
import pickle

# LOAD DATA
df = pd.read_csv("tracheal_tube_insertion.csv")

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# No descriptive statistics are needed.

# PREPROCESSING
features = df[['sex', 'age_c', 'ht', 'wt']]
target = df['tube_depth_G']

# ANALYSIS
## Table 1: Comparison of performance between Machine Learning
    ↪ Model (Random Forest) and formula-based method
X_train, X_test, y_train, y_test = train_test_split(features,
    ↪ target, test_size=0.3, random_state=42)

param_grid = {'n_estimators': [10, 50, 100, 200],'max_depth':[
    ↪ None, 2, 5, 10]}
grid_rf = GridSearchCV(RandomForestRegressor(random_state=42),
    ↪ param_grid=param_grid, cv=5)
grid_rf.fit(X_train, y_train)

# Compute predictions using the optimized Random Forest Model
rf_pred = grid_rf.predict(X_test)

# Compute predictions using formula-based method
formula_based_pred = X_test['ht'] / 10 + 5

# Compute Mean Squared Error (MSE) for both models
rf_errors = np.square(rf_pred - y_test)
formula_based_errors = np.square(formula_based_pred - y_test)

# Compare MSEs with a paired t-Test
```

```
t_test_results = stats.ttest_rel(rf_errors,
    ↪ formula_based_errors)

df1 = pd.DataFrame({
    'Mean Squared Error': [np.mean(rf_errors), np.mean(
        ↪ formula_based_errors)],
    't-statistic': [t_test_results.statistic, t_test_results.
        ↪ statistic],
    'p-value': [t_test_results.pvalue, t_test_results.pvalue]
}, index=['Random Forest', 'Formula-Based'])
df1.to_pickle('table_1.pkl')

## Table 2: Correlation between variables 'sex', 'age', 'height
    ↪ ', 'weight' and the target variable
correlations_list = []
for feature in features.columns:
    correlation_results = stats.pearsonr(df[feature], df['
        ↪ tube_depth_G'])
    correlations_list.append({
        'feature': feature,
        'pearson_correlation': correlation_results.statistic,
        'p_value': correlation_results.pvalue})

df2 = pd.DataFrame(correlations_list).set_index('feature')
df2.to_pickle('table_2.pkl')

## Table 3: "Feature importances from the Random Forest model,
    ↪ including confidence intervals"
rf_model = grid_rf.best_estimator_
feature_importances = rf_model.feature_importances_
n_iterations = 1000
bootstrap_feature_importances = np.zeros((len(X_train.columns),
    ↪  n_iterations))
for i in range(n_iterations):
    X_resample, y_resample = resample(X_train, y_train)
    rf_model.fit(X_resample, y_resample)
    bootstrap_feature_importances[:, i] = rf_model.
        ↪ feature_importances_
bootstrap_percentiles = np.percentile(
    ↪ bootstrap_feature_importances, [2.5, 97.5], axis=1)

df3 = pd.DataFrame({
    'Feature': X_train.columns,
    'Importance': feature_importances,
    'Lower 95% CI': bootstrap_percentiles[0, :],
    'Upper 95% CI': bootstrap_percentiles[1, :]
}).set_index('Feature')
df3.to_pickle('table_3.pkl')
```

```
# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': len(df),
    'Best parameters for Random Forest model': grid_rf.
       ↪ best_params_,
    'accuracy of Random Forest model': grid_rf.score(X_test,
       ↪ y_test)
}

with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
```

C.2 Code Description

The provided code performs analysis on a dataset to predict the optimal tracheal tube depth (OTTD) for pediatric patients who require mechanical ventilation. The code utilizes machine learning techniques and compares the performance of a Random Forest model with a formula-based method.

The analysis steps are as follows:

1. Loading the dataset: The code reads the dataset from a CSV file containing tube characteristics, patient features (sex, age, height, weight), and the target variable (tube depth determined by chest X-ray).

2. Preprocessing: The code selects the relevant features and separates them into features (sex, age, height, weight) and the target variable (tube depth).

3. Analysis - Table 1: Machine Learning vs. formula-based method comparison - Train-test split: The dataset is split into training and testing sets with a 70:30 ratio. - Hyperparameter optimization: The Random Forest model is optimized using GridSearchCV to find the best combination of hyperparameters (number of estimators and maximum depth) for the model. - Model prediction: The optimized Random Forest model is used to predict the tube depth on the test set. - Formula-based prediction: The tube depth is also predicted using a formula-based method based on the patient's height. - Mean Squared Error (MSE) calculation: The MSE is calculated for both models to evaluate their performance. - Paired t-test: The MSEs of the two models are compared using a paired t-test to determine if there is a significant difference.

4. Analysis - Table 2: Correlation between variables and target - Pearson correlation: The code calculates the Pearson correlation coefficient and p-value between each feature (sex, age, height, weight) and the target variable (tube depth). - Correlation table: The results are stored in a table showing the correlations and p-values.

5. Analysis - Table 3: Feature importances from the Random Forest model - Bootstrapping: The code performs bootstrapping on the training set to obtain confidence intervals for feature importances. - Feature importance calculation: The Random Forest model is trained multiple times on resampled training sets, and the feature importance values are averaged. - Confidence intervals: The code calculates the lower and upper confidence intervals (95%) for each feature's importance. - Feature importance table: The results are stored in a table showing the feature importances with confidence intervals.

6. Saving additional results: The code saves additional results, including the total number of observations in the dataset, the best parameters for the Random Forest model found during hyperparameter optimization, and the accuracy of the Random Forest model on the test set. These results are saved in a pickled file named "additional_results.pkl".

The code provides a comprehensive analysis of the dataset, comparing the performance of a machine learning model (Random Forest) with a formula-based method. The correlation between the features and the target variable is evaluated, and the importance of each feature in predicting the tube depth is determined. The code also saves additional results for reference and reporting purposes.

C.3 Code Output

table_1.pkl

```
               Mean Squared Error  t-statistic   p-value
Random Forest               1.228       -7.951  4.16e-14
Formula-Based               3.484       -7.951  4.16e-14
```

table_2.pkl

```
         pearson_correlation    p_value
feature
sex                  0.06661    0.03816
age_c                 0.6373 1.551e-111
ht                    0.7402 6.706e-169
wt                    0.7534 2.844e-178
```

table_3.pkl

```
         Importance  Lower 95% CI  Upper 95% CI
Feature
sex               0             0             0
age_c             0             0       0.04262
```

```
ht           0.07985       0.02267        0.4388
wt            0.9202         0.555        0.9735
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Best parameters for Random Forest model': {'max_depth': 2,
        'n_estimators': 10},
    'accuracy of Random Forest model': 0.6097                    ,
}
```

D LaTeX Table Design

D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```
# IMPORT
import pandas as pd
from my_utils import to_latex_with_note, is_str_in_df,
   ↪ split_mapping, AbbrToNameDef
from typing import Any, Dict, Optional, Tuple

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
    'sex': ('Sex', 'Patient Sex (0 = female, 1 = male)'),
    'age_c': ('Age', 'Patient age in years, rounded to half
       ↪ years'),
    'ht': ('Height', 'Patient height in cm'),
    'wt': ('Weight', 'Patient weight in kg'),
    'tube_depth_G': ('OTTD', 'Optimal tracheal tube depth as
       ↪ determined by chest X-ray in cm'),
}

# TABLE 1:
df1 = pd.read_pickle('table_1.pkl')

# FORMAT VALUES
# No formatting of values needed in this table

# RENAME ROWS AND COLUMNS
mapping1 = dict((k, v) for k, v in shared_mapping.items() if
   ↪ is_str_in_df(df1, k))
mapping1 |= {
    "Mean Squared Error": ("MSE", "Mean Squared Error"),
    't-statistic': ("t-statistic", None),
    'p-value': ("P-value", None),
}

abbrs_to_names1, legend1 = split_mapping(mapping1)
df1 = df1.rename(columns=abbrs_to_names1, index=abbrs_to_names1
    ↪ )

# SAVE AS LATEX
to_latex_with_note(
    df1, ’table_1.tex’,
    caption="Comparison of performance between Machine Learning
        ↪  Model (Random Forest) and formula-based method",
    label=’table:comparison_of_ML_and_formula_based’,
    note="Models evaluted by their Mean Squared Error (MSE) on
        ↪ the same test set",
    legend=legend1,
)


# TABLE 2:
df2 = pd.read_pickle(’table_2.pkl’)

# FORMAT VALUES
# No formatting of values needed in this table

# RENAME ROWS AND COLUMNS
mapping2 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df2, k))
mapping2 |= {
    ’pearson_correlation’: ("Pearson Coefficient", None),
    ’p_value’: ("P-value", None),
}

abbrs_to_names2, legend2 = split_mapping(mapping2)
df2 = df2.rename(columns=abbrs_to_names2, index=abbrs_to_names2
    ↪ )

# SAVE AS LATEX
to_latex_with_note(
    df2, ’table_2.tex’,
    caption="Correlation between variables and the target
        ↪ variable",
    label=’table:correlation_to_target’,
    note="Correlation values calculated using Pearson method",
    legend=legend2,
)


# TABLE 3:
df3 = pd.read_pickle(’table_3.pkl’)
```

```
# FORMAT VALUES
# No formatting of values needed in this table

# RENAME ROWS AND COLUMNS
mapping3 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df3, k))
mapping3 |= {
    'Importance': ("Importance", None),
    'Lower 95% CI': ('Lower 95%', 'Lower bound of 95%
        ↪ Confidence Interval'),
    'Upper 95% CI': ('Upper 95%', 'Upper bound of 95%
        ↪ Confidence Interval'),
}

abbrs_to_names3, legend3 = split_mapping(mapping3)
df3 = df3.rename(columns=abbrs_to_names3, index=abbrs_to_names3
    ↪ )

# SAVE AS LATEX
to_latex_with_note(
    df3, 'table_3.tex',
    caption="Feature importances from the Random Forest model,
        ↪ including confidence intervals",
    label='table:feature_importances',
    note="Importances calculated base on feature contribution
        ↪ in Random Forest method",
    legend=legend3,
)
```

D.2 Provided Code

The code above is using the following provided functions:

```
def to_latex_with_note(df, filename: str, caption: str, label:
    ↪ str, note: str = None, legend: Dict[str, str] = None, **
    ↪ kwargs):
    """
    Converts a DataFrame to a LaTeX table with optional note
        ↪ and legend added below the table.

    Parameters:
    - df, filename, caption, label: as in `df.to_latex`.
    - note (optional): Additional note below the table.
    - legend (optional): Dictionary mapping abbreviations to
        ↪ full names.
    - **kwargs: Additional arguments for `df.to_latex`.
    """
```

```
def is_str_in_df(df: pd.DataFrame, s: str):
    return any(s in level for level in getattr(df.index, '
       ↪ levels', [df.index]) + getattr(df.columns, 'levels',
       ↪ [df.columns]))

AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]

def split_mapping(abbrs_to_names_and_definitions: AbbrToNameDef
   ↪ ):
    abbrs_to_names = {abbr: name for abbr, (name, definition)
       ↪ in abbrs_to_names_and_definitions.items() if name is
       ↪ not None}
    names_to_definitions = {name or abbr: definition for abbr,
       ↪ (name, definition) in abbrs_to_names_and_definitions.
       ↪ items() if definition is not None}
    return abbrs_to_names, names_to_definitions
```

D.3 Code Output

table_1.tex

```
% This latex table was generated from: `table_1.pkl`
\begin{table}[h]
\caption{Comparison of performance between Machine Learning
    Model (Random Forest) and formula-based method}
\label{table:comparison_of_ML_and_formula_based}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrl}
\toprule
 & MSE & t-statistic & P-value \\
\midrule
\textbf{Random Forest} & 1.23 & -7.95 & $<$1e-06 \\
\textbf{Formula-Based} & 3.48 & -7.95 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Models evaluted by their Mean Squared Error (MSE) on the
    same test set
\item \textbf{MSE}: Mean Squared Error
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
% This latex table was generated from: ‘table_2.pkl‘
\begin{table}[h]
\caption{Correlation between variables and the target variable}
\label{table:correlation_to_target}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & Pearson Coefficient & P-value \\
feature &  &  \\
\midrule
\textbf{Sex} & 0.0666 & 0.0382 \\
\textbf{Age} & 0.637 & 1.55e-111 \\
\textbf{Height} & 0.74 & 6.71e-169 \\
\textbf{Weight} & 0.753 & 2.84e-178 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Correlation values calculated using Pearson method
\item \textbf{Sex}: Patient Sex (0 = female, 1 = male)
\item \textbf{Age}: Patient age in years, rounded to half years
\item \textbf{Height}: Patient height in cm
\item \textbf{Weight}: Patient weight in kg
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
% This latex table was generated from: ‘table_3.pkl‘
\begin{table}[h]
\caption{Feature importances from the Random Forest model,
    including confidence intervals}
\label{table:feature_importances}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrr}
\toprule
 & Importance & Lower 95\% & Upper 95\% \\
Feature &  &  &  \\
\midrule
\textbf{Sex} & 0 & 0 & 0 \\
\textbf{Age} & 0 & 0 & 0.0426 \\
\textbf{Height} & 0.0798 & 0.0227 & 0.439 \\
\textbf{Weight} & 0.92 & 0.555 & 0.973 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Importances calculated base on feature contribution in
   Random Forest method
\item \textbf{Sex}: Patient Sex (0 = female, 1 = male)
\item \textbf{Age}: Patient age in years, rounded to half years
\item \textbf{Height}: Patient height in cm
\item \textbf{Weight}: Patient weight in kg
\item \textbf{Lower 95\%}: Lower bound of 95\% Confidence
   Interval
\item \textbf{Upper 95\%}: Upper bound of 95\% Confidence
   Interval
\end{tablenotes}
\end{threeparttable}
\end{table}
```