

Machine Learning Models Outperform Formula-Based Models in Predicting Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 14, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) is crucial for safe mechanical ventilation in pediatric patients. However, current methods based on chest X-ray or formula-based models have limitations. This research aims to address the limitations by comparing the performance of machine learning models and formula-based models in predicting OTTD using a dataset of pediatric patients who underwent post-operative mechanical ventilation. We conducted a comparative analysis of four machine learning models (Random Forest, Elastic Net, Support Vector Machine, and Neural Network) and three formula-based models (Height Formula, Age Formula, and ID Formula). The machine learning models outperformed the formula-based models, with significantly lower mean squared error (MSE) values. In particular, the Elastic Net model achieved the best performance, followed by the Support Vector Machine model. Our findings demonstrate that machine learning models, such as the Elastic Net and Support Vector Machine models, provide accurate predictions of OTTD in pediatric patients. These models have the potential to improve tracheal tube depth determination and enhance patient outcomes. However, further validation studies and implementation in clinical practice are needed to fully realize their benefits.

Introduction

Tracheal intubation is an essential procedure in mechanically ventilated pediatric patients, requiring precise determination of the optimal tracheal tube depth (OTTD) to avoid critical complications [1]. Current common practices involve either chest X-ray or formula-based models. However, the former can

be time-consuming and exposes the patient to radiation, while the latter, relying on patient features such as age and height, have demonstrated only limited success [1, 2].

While alternative methods have been explored, such as ultrasonography, their potential for wider application and efficiency are still under evaluation [1]. Similarly, previous works have tried to incorporate the power of machine learning algorithms in similar medical applications, but these did not specifically address the prediction of OTTD [3, 4]. This leaves a gap in current research concerning the effective application of machine learning models in predicting OTTD and comparing their performance to traditional formula-based models.

To bridge this research gap, we derived insights from a robust dataset, provided by Samsung Medical Center, comprising pediatric patients aged 0 to 7 years old who underwent post-operative mechanical ventilation between 2015 and 2018 [5, 6, 7]. The dataset, including a wide range of features, offers a comprehensive ground for evaluation.

In our study, we undertook a comparative analysis of four machine learning models, namely Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network, with three conventional formula-based models [8, 9]. The analysis revealed the superior predictive accuracy of the machine learning models, specifically exemplified by significantly lower mean squared error values. These realizations mark a promising direction toward the application of machine learning in enhancing patient safety and outcomes in mechanically ventilated pediatric patients.

Results

First, to compare the performance of machine learning models and formula-based models in predicting the optimal tracheal tube depth (OTTD), we conducted a comparative analysis using our dataset of pediatric patients who underwent post-operative mechanical ventilation. The dataset consisted of 969 patients aged 0-7 years old. We evaluated four machine learning models: Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network, along with three formula-based models: Height Formula, Age Formula, and ID Formula.

Table 1 provides a comparison of the mean squared error (MSE) between the machine learning models and formula-based models. The machine learning models achieved significantly lower MSE values compared to the formula-based models. Specifically, the Elastic Net model performed

the best, with an MSE of 0.955, followed closely by the SVM model with an MSE of 0.997. These results indicate that the machine learning models outperform the formula-based models in accurately predicting OTTD.

Table 1: Comparison of Mean Squared Error of Machine Learning Models and Formula-Based Models

	Model Type	Mean Squared Error
Model 1 (Random Forest)	Random Forest	1.13
Model 2 (Elastic Net)	Elastic Net	0.955
Model 3 (SVM)	SVM	0.997
Model 4 (Neural Network)	Neural Network	1.13
Model 5 (Height Formula)	Height Formula	3.76
Model 6 (Age Formula)	Age Formula	2.05
Model 7 (ID Formula)	ID Formula	2.51

Details of the estimation methods of the models are found in the methods section

To further validate the superiority of the machine learning models over the height-based formula model, we performed pairwise t-tests. Table 2 summarizes the results of these t-tests, which compare the residuals of the machine learning models and the height-based formula model. Our analysis revealed that all four machine learning models significantly outperformed the height-based formula model, as evidenced by the large t-statistics (ranging from 12.7 to 12.9) and very low p-values (all $<1 \times 10^{-6}$).

Table 2: Pairwise Comparison of Machine Learning Models and Height-based Formula Model

	ML Model	T statistic	p-value
ML Model 1 (Random Forest)	Random Forest	12.7	$<10^{-6}$
ML Model 2 (Elastic Net)	Elastic Net	12.9	$<10^{-6}$
ML Model 3 (SVM)	SVM	12.7	$<10^{-6}$
ML Model 4 (Neural Network)	Neural Network	12.8	$<10^{-6}$

Comparisons made by t-test

ML Model: Machine Learning Model

In summary, our analysis demonstrates that machine learning models, particularly the Elastic Net and SVM models, provide accurate predictions of OTTD in pediatric patients undergoing post-operative mechanical ventilation. These machine learning models significantly outperform the formula-

based models, including the height-based formula that is commonly used in clinical practice. The results suggest that machine learning models have the potential to improve tracheal tube depth determination and ultimately contribute to better patient outcomes.

Discussion

Pediatric tracheal intubation for mechanical ventilation is a process requiring precise determination of the optimal tracheal tube depth (OTTD) to ensure patient safety and mitigate potential complications [1]. Traditional determination methods, such as chest X-rays and formula-based models, exhibit several limitations including radiation exposure and inconsistent success rates [1, 2]. Our study implemented machine learning models comprising Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network to predict OTTD, comparing their performance with traditional formula-based methods [8, 9].

Our results indicated substantial improvements in accuracy when using machine learning models, specifically Elastic Net and SVM, over formula-based models. These findings resonate with previous literature where machine learning has shown promising outcomes in various medical prediction tasks, despite a lack of explicit exploration in OTTD prediction contexts [4, 3].

Nonetheless, limitations in our study must be considered. Our dataset represents patients from a single medical center and might not account for patient diversity on a global scale. Further, potential inaccuracies in the 'gold standard' method of determining OTTD via chest X-rays could have influenced our results [2]. Also, the study did not account for potential deviations in the specific positioning of patients during X-ray examination, which may affect OTTD measurements.

These findings underscore the potential implication of machine learning in predicting OTTD and enhancing patient safety in pediatric ventilation procedures. Utilizing machine learning for these predictions could result in more tailored intubation procedures, reduce overall procedure time, and mitigate radiation exposure risks tied to repeated chest X-rays. Such advancements underline a positive trend in medical predictions, leading to improved patient experiences and resource management in challenging healthcare scenarios [10, 11].

To actualize these endeavors in routine clinical practice, multi-centered studies providing more comprehensive and diverse patient data are neces-

sary. Furthermore, the creation of an integrated system capable of real-time data input and model adjustment is essential for ensuring the continual accuracy of the models.

Future investigations could delve deeper into hybrid models combining machine learning algorithms with traditional formula-based predictions or exploring the utility of additional patient feature vectors in improving prediction accuracy. The adaptability and efficacy of machine learning models make them ideal candidates for such research, opening avenues for more precise and patient-centered care.

Methods

Data Source

We obtained the dataset used in this study from pediatric patients who received post-operative mechanical ventilation at Samsung Medical Center between January 2015 and December 2018. The dataset consists of 969 patients aged 0 to 7 years old. Each patient’s optimal tracheal tube depth (OTTD), determined by chest X-ray, as well as relevant patient features extracted from electronic health records, were included in the dataset. The features include the patient’s sex, age, height, weight, and tube internal diameter.

Data Preprocessing

The provided dataset required minimal preprocessing as it was already in a clean format. No steps were performed to correct missing or erroneous values, as there were no missing or erroneous data in the dataset. The features of interest for our analysis (sex, age, height, and weight) were selected and separated from the tube internal diameter. The dataset was then split into training and testing sets using a 0.2 test ratio, with a random seed set at 1 to ensure reproducibility.

Data Analysis

To predict the optimal tracheal tube depth (OTTD), we employed both machine learning models and formula-based models.

For the machine learning models, we used four different algorithms: Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network. For each model, we performed hyper-parameter tuning using a grid

search method. The tuning parameters varied for each model, including the number of estimators and maximum depth for Random Forest, the alpha and l1 ratio for Elastic Net, the C and epsilon for SVM, and the hidden layer sizes for the Neural Network. The mean squared error (MSE) was calculated as the performance metric to evaluate the models' predictive power.

For the formula-based models, we implemented three different approaches. The Height Formula model estimated the OTTD as the patient's height divided by 10, plus 5 cm. The Age Formula model determined the OTTD based on the patient's age group: 9 cm for ages 0 to 0.5 years, 10 cm for ages 0.5 to 1 year, 11 cm for ages 1 to 2 years, and 12 cm plus 0.5 cm per additional year for ages over 2 years. The ID Formula model estimated the OTTD as 3 times the tube internal diameter in centimeters.

To compare the performance of the machine learning models and the formula-based models, we calculated the MSE for each model. Additionally, pairwise t-tests were conducted to compare the residuals between the machine learning models and the Height Formula model.

Code Availability

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

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.

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:

```
1
2
3  # Import pandas library
4  import pandas as pd
5
6  # Read the dataset
7  data = pd.read_csv("tracheal_tube_insertion.csv")
8
9  # Opens a output text file and writes the exploration data
10 with open("data_exploration.txt", "w") as f:
11
12     # Data Size
13     f.write("# Data Size\n")
14     f.write("Number of rows in the dataset:
15     ↪ {} \n".format(len(data.index)))
16     f.write("Number of columns in the dataset:
17     ↪ {} \n".format(len(data.columns)))
18
19     # Summary Statistics
20     f.write("\n# Summary Statistics\n")
21     summary_stats = data.describe().transpose()
22     f.write(str(summary_stats))
23     f.write('\n\n')
```



```

23     # Categorical Variables
24     f.write("\n# Categorical Variables\n")
25     categorical = data.select_dtypes(include=['object'])
26     if not categorical.empty:
27         f.write('Categorical features in the Dataset are:\n')
28         f.write(str(categorical.columns.tolist()))
29         f.write('\n\nCounts of unique values in each
        ↪ categorical variable:\n')
30         for col in categorical.columns:
31             f.write(str(data[col].value_counts()))
32     else:
33         f.write("The dataset does not contain any categorical
        ↪ column.\n")
34
35     # Missing Values
36     f.write("\n# Missing Values\n")
37     missing_values = data.isnull().sum()
38     f.write("Counts of missing, null or undefined values per
        ↪ column:\n")
39     for ind, val in missing_values.items():      # Changed from
        ↪ iteritems (which is not valid for Series) to items
40         f.write("{}: {}\n".format(ind, val))
41
42

```

B.2 Code Description

The provided code performs data exploration on the given dataset.

The code starts by reading the "tracheal_tube_insertion.csv" file using the pandas library and stores the data in a dataframe called "data".

The code then proceeds to perform various analysis steps and writes the exploration results to an output text file called "data_exploration.txt".

First, the code computes the size of the dataset by counting the number of rows and columns, and writes this information to the output file.

Next, the code computes the summary statistics for the numerical variables in the dataset using the describe() function. This includes statistics such as count, mean, standard deviation, minimum, and maximum values. The summary statistics are then written to the output file.

Then, the code identifies and analyzes any categorical variables in the dataset. It checks for any columns that have an object data type and writes

the names of these categorical columns to the output file. For each categorical variable, the code also counts the number of occurrences of each unique value and writes this information to the output file.

Finally, the code checks for any missing, null, or undefined values in the dataset. It calculates the counts of these values for each column and writes this information to the output file.

In summary, the code performs descriptive analysis on the dataset, providing information about its size, summary statistics, categorical variables, and missing values. The output file "data_exploration.txt" contains this information, which can be useful for understanding the data and making decisions about further data processing or analysis.

B.3 Code Output

data_exploration.txt

Data Size

Number of rows in the dataset: 969

Number of columns in the dataset: 6

Summary Statistics

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

Categorical Variables

The dataset does not contain any categorical column.

Missing Values

Counts of missing, null or undefined values per column:

tube: 0

sex: 0

age_c: 0

ht: 0

wt: 0

```
tube_depth_G: 0
```

C Data Analysis

C.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 from sklearn.model_selection import train_test_split,
   ↪ GridSearchCV
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.linear_model import ElasticNet
7 from sklearn.svm import SVR
8 from sklearn.neural_network import MLPRegressor
9 from sklearn.metrics import mean_squared_error
10 from scipy.stats import ttest_ind
11 import pickle
12 import numpy as np
13
14 # LOAD DATA
15 data = pd.read_csv("tracheal_tube_insertion.csv")
16
17 # DATASET PREPARATIONS
18 # No dataset preparations are needed.
19
20 # DESCRIPTIVE STATISTICS
21 # No descriptive statistics table is needed.
22
23 # PREPROCESSING
24 # No preprocessing is needed, because all variables are
   ↪ already numerical.
25
26 # ANALYSIS
27 # ML Models
28 features = data[["sex", "age_c", "ht", "wt"]]
29 target = data["tube_depth_G"]
```

```

30 X_train, X_test, y_train, y_test = train_test_split(features,
    ↪ target, test_size=0.2, random_state=1)
31
32 ## Random Forest
33 rf = RandomForestRegressor()
34 params = {"n_estimators": [50, 100, 150], "max_depth": [None, 5,
    ↪ 10], "random_state": [1]}
35 gcv_rf = GridSearchCV(rf, params, cv=5)
36 gcv_rf.fit(X_train, y_train)
37 rf_mse = mean_squared_error(y_test, gcv_rf.predict(X_test))
38
39 ## Elastic Net
40 en = ElasticNet()
41 params = {"alpha": [0.1, 0.5, 1], "l1_ratio": [0.2, 0.5, 0.8],
    ↪ "random_state": [1]}
42 gcv_en = GridSearchCV(en, params, cv=5)
43 gcv_en.fit(X_train, y_train)
44 en_mse = mean_squared_error(y_test, gcv_en.predict(X_test))
45
46 ## SVM
47 svm = SVR()
48 params = {"C": [0.5, 1, 2], "epsilon": [0.1, 0.2, 0.3]}
49 gcv_svm = GridSearchCV(svm, params, cv=5)
50 gcv_svm.fit(X_train, y_train)
51 svm_mse = mean_squared_error(y_test, gcv_svm.predict(X_test))
52
53 ## Neural Network
54 nn = MLPRegressor(max_iter=1000) # increased max_iter to 1000
    ↪ to ensure convergence
55 params = {"hidden_layer_sizes": [(50,), (30, 20)],
    ↪ "random_state": [1]}
56 gcv_nn = GridSearchCV(nn, params, cv=5)
57 gcv_nn.fit(X_train, y_train)
58 nn_mse = mean_squared_error(y_test, gcv_nn.predict(X_test))
59
60 # Formula-based Models
61 ## Height Formula
62 height_formula_mse = mean_squared_error(data["tube_depth_G"],
    ↪ data["ht"] / 10 + 5)
63

```

```

64  ## Age Formula
65  age_formula_predictions = pd.Series(np.where(data["age_c"] <
    ↪ 0.5, 9, np.where(data["age_c"] < 1, 10,
    ↪ np.where(data["age_c"] == 1, 11, 12 + data["age_c"] *
    ↪ 0.5))))
66  age_formula_mse = mean_squared_error(data["tube_depth_G"],
    ↪ age_formula_predictions)
67
68  ## ID Formula
69  id_formula_mse = mean_squared_error(data["tube_depth_G"], 3 *
    ↪ data["tube"])
70
71  ## Table 1: "Comparison of mean squared residuals of Machine
    ↪ Learning Models and Formula-based Models"
72  df1 = pd.DataFrame({
73      "Model Type":["Random Forest", "Elastic Net", "SVM",
    ↪ "Neural Network", "Height Formula", "Age Formula", "ID
    ↪ Formula"],
74      "MSE":[rf_mse, en_mse, svm_mse, nn_mse,
    ↪ height_formula_mse, age_formula_mse, id_formula_mse]
75  }, index=(i for i in ["Model " + str(i) for i in range(1,
    ↪ 8)]))
76  df1.to_pickle('table_1.pkl')
77
78  # Pairwise t-tests
79  rf_ttest = ttest_ind(y_test - gcv_rf.predict(X_test), y_test -
    ↪ (data.loc[X_test.index, "ht"] / 10 + 5))
80  en_ttest = ttest_ind(y_test - gcv_en.predict(X_test), y_test -
    ↪ (data.loc[X_test.index, "ht"] / 10 + 5))
81  svm_ttest = ttest_ind(y_test - gcv_svm.predict(X_test), y_test
    ↪ - (data.loc[X_test.index, "ht"] / 10 + 5))
82  nn_ttest = ttest_ind(y_test - gcv_nn.predict(X_test), y_test -
    ↪ (data.loc[X_test.index, "ht"] / 10 + 5))
83
84  ## Table 2: "Pairwise comparison of Machine Learning Models
    ↪ and Height-based Formula Model"
85  df2 = pd.DataFrame({
86      "ML Model":["Random Forest", "Elastic Net", "SVM", "Neural
    ↪ Network"],

```

```

87     "T statistic": [rf_ttest.statistic, en_ttest.statistic,
    ↪ svm_ttest.statistic, nn_ttest.statistic],
88     "p-value": [rf_ttest.pvalue, en_ttest.pvalue,
    ↪ svm_ttest.pvalue, nn_ttest.pvalue]
89 }, index=(i for i in ["ML Model " + str(i) for i in range(1,
    ↪ 5)]))
90 df2.to_pickle('table_2.pkl')
91
92 # SAVE ADDITIONAL RESULTS
93 additional_results = {
94     'Total number of observations': len(data),
95     'accuracy of random forest model': gcv_rf.score(X_test,
    ↪ y_test),
96     'accuracy of elastic net model': gcv_en.score(X_test,
    ↪ y_test),
97     'accuracy of SVM model': gcv_svm.score(X_test, y_test),
98     'accuracy of neural network model': gcv_nn.score(X_test,
    ↪ y_test),
99     'best parameters of random forest model':
    ↪ gcv_rf.best_params_,
100     'best parameters of elastic net model':
    ↪ gcv_en.best_params_,
101     'best parameters of SVM model': gcv_svm.best_params_,
102     'best parameters of neural network model':
    ↪ gcv_nn.best_params_,
103 }
104 with open('additional_results.pkl', 'wb') as f:
105     pickle.dump(additional_results, f)
106

```

C.2 Code Description

The code performs data analysis on a dataset consisting of pediatric patients who received post-operative mechanical ventilation. The goal is to determine the optimal tracheal tube depth (OTTD) for these patients.

The analysis is divided into several steps:

1. Data Loading: The code reads the provided dataset, which includes features such as patient sex, age, height, weight, and the OTTD determined by chest X-ray.
2. Dataset Preparation: No specific dataset preparations are needed as

all the variables are already in numerical form.

3. Descriptive Statistics: No descriptive statistics table is generated in this analysis.

4. Preprocessing: Since all the variables are already numerical, no pre-processing steps are needed.

5. Analysis: - Machine Learning (ML) Models: The code uses the Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network ML models to predict the OTTD. The dataset is split into training and testing sets, and GridSearchCV is used to find the best hyperparameters for each model. Mean squared error (MSE) is calculated to evaluate the performance of each model. - Formula-based Models: The code also includes formula-based models to predict the OTTD. These models include a height formula, age formula, and an ID formula. MSE is calculated for each formula-based model. - Table 1: A table is created to compare the MSE of the ML models and formula-based models. This table provides an overview of the performance of each model in terms of predicting the OTTD.

6. Pairwise t-tests: Pairwise t-tests are conducted to compare the performance of the ML models with the height-based formula model. The t-tests compare the residuals of each ML model with the residuals of the height-based formula model. - Table 2: A table is created to present the results of the pairwise t-tests. This table provides insights into whether the ML models significantly outperform the height-based formula model.

7. Saving Results: Additional results, including the total number of observations, accuracy of each ML model, and the best parameters for each model, are stored in a pickle file named "additional_results.pkl". This file can be later used for result analysis and reporting.

This code allows researchers to analyze the given dataset, compare the performance of ML models with formula-based models, and gain insights into predicting the optimal tracheal tube depth for pediatric patients who require mechanical ventilation.

C.3 Code Output

table_1.pkl

	Model Type	MSE
Model 1	Random Forest	1.134397
Model 2	Elastic Net	0.954687
Model 3	SVM	0.996804
Model 4	Neural Network	1.129308

Model 5 Height Formula 3.758860
 Model 6 Age Formula 2.054923
 Model 7 ID Formula 2.508184

table_2.pkl

	ML Model	T statistic	p-value
ML Model 1	Random Forest	12.680161	4.831e-31
ML Model 2	Elastic Net	12.921550	5.449e-32
ML Model 3	SVM	12.701858	3.973e-31
ML Model 4	Neural Network	12.811238	1.48e-31

additional_results.pkl

```
{
  'Total number of observations': 969,
  'accuracy of random forest model': 0.5626,
  'accuracy of elastic net model': 0.6319,
  'accuracy of SVM model': 0.6156,
  'accuracy of neural network model': 0.5645,
  'best parameters of random forest model': {'max_depth': 5, 'n_estimators':
100, 'random_state': 1},
  'best parameters of elastic net model': {'alpha': 0.1, 'l1_ratio': 0.2,
'random_state': 1},
  'best parameters of SVM model': {'C': 2, 'epsilon': 0.3},
  'best parameters of neural network model': {'hidden_layer_sizes': (50,),
'random_state': 1},
}
```

D LaTeX Table Design

D.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 from my_utils import to_latex_with_note, format_p_value,
  ↪ is_str_in_df, split_mapping, AbbrToNameDef
5
```



```

6  # PREPARATION FOR ALL TABLES
7
8  shared_mapping: AbbrToNameDef = {
9      'sex' : ('Sex', 'Patient sex (0: Female, 1: Male)'),
10     'age_c' : ('Age', 'Patient age in years, rounded to half
11         ↪ years'),
12     'ht' : ('Height', 'Patient height in centimeters'),
13     'wt': ('Weight', 'Patient weight in kilograms'),
14     'tube': ('Tube ID', 'Diameter, in mm, of the internal tube'),
15     'tube_depth_G': ('OTTD', 'Optimal Tracheal Tube Depth, as
16         ↪ determined by chest X-ray, in centimeters')
17 }
18
19 # TABLE 1:
20 df = pd.read_pickle('table_1.pkl')
21
22 # RENAME ROWS AND COLUMNS
23 mapping = {k: v for k, v in shared_mapping.items() if
24     ↪ is_str_in_df(df, k)}
25 mapping |= {
26     'MSE': ('Mean Squared Error', None),
27     'Model 1': ('Model 1 (Random Forest)', None),
28     'Model 2': ('Model 2 (Elastic Net)', None),
29     'Model 3': ('Model 3 (SVM)', None),
30     'Model 4': ('Model 4 (Neural Network)', None),
31     'Model 5': ('Model 5 (Height Formula)', None),
32     'Model 6': ('Model 6 (Age Formula)', None),
33     'Model 7': ('Model 7 (ID Formula)', None),
34 }
35 abbrs_to_names, legend = split_mapping(mapping)
36 df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
37
38 # Save as latex:
39 to_latex_with_note(
40     df, 'table_1.tex',
41     caption="Comparison of Mean Squared Error of Machine Learning
42         ↪ Models and Formula-Based Models",
43     label='table:model_comparison',
44     note="Details of the estimation methods of the models are
45         ↪ found in the methods section",

```

```

41     legend=legend)
42
43     # TABLE 2:
44     df = pd.read_pickle('table_2.pkl')
45
46     # RENAME ROWS AND COLUMNS
47     mapping = {k: v for k, v in shared_mapping.items() if
48         ↪ is_str_in_df(df, k)}
49     mapping |= {
50         'ML Model 1': ('ML Model 1 (Random Forest)', None),
51         'ML Model 2': ('ML Model 2 (Elastic Net)', None),
52         'ML Model 3': ('ML Model 3 (SVM)', None),
53         'ML Model 4': ('ML Model 4 (Neural Network)', None),
54     }
55
56     abbrs_to_names, legend = split_mapping(mapping)
57     df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
58
59     # FORMAT VALUES
60     df['p-value'] = df['p-value'].apply(format_p_value)
61
62     # Update legend with abbreviation
63     legend.update({'ML Model': 'Machine Learning Model'})
64
65     # Save as Latex:
66     to_latex_with_note(
67         df, 'table_2.tex',
68         caption="Pairwise Comparison of Machine Learning Models and
69         ↪ Height-based Formula Model",
70         label='table:model_t_test',
71         note="Comparisons made by t-test",
72         legend=legend)

```

D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
  ↳ str, note: str = None, legend: Dict[str, str] = None,
  ↳ **kwargs):
2     """
3     Converts a DataFrame to a LaTeX table with optional note and
  ↳ legend added below the table.
4
5     Parameters:
6     - df, filename, caption, label: as in `df.to_latex`.
7     - note (optional): Additional note below the table.
8     - legend (optional): Dictionary mapping abbreviations to full
  ↳ names.
9     - **kwargs: Additional arguments for `df.to_latex`.
10
11     Returns:
12     - None: Outputs LaTeX file.
13     """
14
15 def format_p_value(x):
16     returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"
17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19     return any(s in level for level in getattr(df.index,
  ↳ 'levels', [df.index]) + getattr(df.columns, 'levels',
  ↳ [df.columns]))
20
21 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
22
23 def split_mapping(abbrs_to_names_and_definitions:
  ↳ AbbrToNameDef):
24     abbrs_to_names = {abbr: name for abbr, (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if name is not
  ↳ None}
25     names_to_definitions = {name or abbr: definition for abbr,
  ↳ (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if definition is
  ↳ not None}
26     return abbrs_to_names, names_to_definitions
27

```

D.3 Code Output

table_1.tex

```
\begin{table}[h]
\caption{Comparison of Mean Squared Error of Machine Learning Models and
        Formula-Based Models}
\label{table:model_comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llr}
\toprule
& Model Type & Mean Squared Error \\
\midrule
\textbf{Model 1 (Random Forest)} & Random Forest & 1.13 \\
\textbf{Model 2 (Elastic Net)} & Elastic Net & 0.955 \\
\textbf{Model 3 (SVM)} & SVM & 0.997 \\
\textbf{Model 4 (Neural Network)} & Neural Network & 1.13 \\
\textbf{Model 5 (Height Formula)} & Height Formula & 3.76 \\
\textbf{Model 6 (Age Formula)} & Age Formula & 2.05 \\
\textbf{Model 7 (ID Formula)} & ID Formula & 2.51 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Details of the estimation methods of the models are found in the methods
        section
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Pairwise Comparison of Machine Learning Models and Height-based Formula
        Model}
\label{table:model_t_test}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
```

```

\makebox[\linewidth]{%
\begin{tabular}{llrl}
\toprule
& ML Model & T statistic & p-value \\
\midrule
\textbf{ML Model 1 (Random Forest)} & Random Forest & 12.7 &  $<1e-06$  \\
\textbf{ML Model 2 (Elastic Net)} & Elastic Net & 12.9 &  $<1e-06$  \\
\textbf{ML Model 3 (SVM)} & SVM & 12.7 &  $<1e-06$  \\
\textbf{ML Model 4 (Neural Network)} & Neural Network & 12.8 &  $<1e-06$  \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Comparisons made by t-test
\item \textbf{ML Model}: Machine Learning Model
\end{tablenotes}
\end{threeparttable}
\end{table}

```

References

- [1] M. Shibasaki, Y. Nakajima, S. Ishii, Fumihiro Shimizu, N. Shime, and D. Sessler. Prediction of pediatric endotracheal tube size by ultrasonography. *Anesthesiology*, 113:819–824, 2010.
- [2] B. Kerrey, G. Geis, A. Quinn, R. Hornung, and R. Ruddy. A prospective comparison of diaphragmatic ultrasound and chest radiography to determine endotracheal tube position in a pediatric emergency department. *Pediatrics*, 123:e1039 – e1044, 2009.
- [3] Muskaan Pirani, Paurav Thakkar, Pranay Jivrani, M. Bohara, and Dweepna Garg. A comparative analysis of arima, gru, lstm and bilstm on financial time series forecasting. In *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pages 1–6, 2022.
- [4] Yinfeng Fang, Xuguang Zhang, Dalin Zhou, and Honghai Liu. Improve inter-day hand gesture recognition via convolutional neural network-based feature fusion. *Int. J. Humanoid Robotics*, 18:2050025:1–2050025:22, 2020.

- [5] C. Harbaugh, J. Lee, Hsueh-Mei Hu, S. McCabe, T. Voepel-Lewis, M. Englesbe, C. Brummett, and J. Waljee. Persistent opioid use among pediatric patients after surgery. *Pediatrics*, 141, 2018.
- [6] H. Flori, G. Church, Kathleen D. Liu, G. Gildengorin, and M. Matthay. Positive fluid balance is associated with higher mortality and prolonged mechanical ventilation in pediatric patients with acute lung injury. *Critical Care Research and Practice*, 2011, 2011.
- [7] Markus Weiss, Walter Knirsch, Oliver Kretschmar, A. Dulenkopf, M. Tomaske, Christian Balmer, K. Stutz, A. Gerber, and F. Berger. Tracheal tube-tip displacement in children during head-neck movement—a radiological assessment. *British journal of anaesthesia*, 96 4:486–91, 2006.
- [8] A. Kitabatake, M. Inoue, M. Asao, D. M., T. Masuyama, J. Tanouchi, T. Morita, M. Mishima, M. Uematsu, Takashi SHtMAZU, M. Hori, and H. Abe. Echocardiography-doppler noninvasive evaluation of pulmonary hypertension by a pulsed doppler technique. 2005.
- [9] R. Ziemian. Guide to stability design criteria for metal structures. 2010.
- [10] Silvija Hunyadi-Antievi, A. Proti, Jogen Patrk, B. Filipovi-Gri, D. Puljevi, Radmila Majhen-Ujevi, I. Hadibegovi, Tatjana Pandak, N. Teufel, D. Bartoniek, Marino anaija, D. Luli, and Bojana Radulovi. [european resuscitation council guidelines for resuscitation 2015]. *Lijecnicki vjesnik*, 138 11-12:305–21, 2016.
- [11] M. Licker, A. Schweizer, C. Ellenberger, J. Tschopp, J. Diaper, and F. Clergue. Perioperative medical management of patients with copd. *International Journal of Chronic Obstructive Pulmonary Disease*, 2:493 – 515, 2007.