A Machine Learning Approach for Accurate Estimation of Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 15, 2024

Abstract

Precise tracheal tube placement is crucial for minimizing complications in pediatric patients undergoing mechanical ventilation after surgery. However, accurately determining the Optimal Tracheal Tube Depth (OTTD) in the pediatric population remains challenging. Existing methods, such as chest X-ray and formula-based models, have limitations in accurately estimating the OTTD. To address this gap, we developed a novel machine learning approach to estimate the OTTD in pediatric patients. Using a dataset of 969 patients aged 0-7 years, we trained and tested multiple machine learning models, including Random Forest, Elastic Net, Support Vector Machine, and Neural Network. Our models consistently outperformed the formula-based models, achieving lower Mean Squared Errors. Although the differences were not statistically significant, the improved accuracy of our machine learning models suggests their potential in accurately estimating the OTTD. This approach reduces reliance on radiation-intensive imaging methods and has implications for enhancing patient safety and optimizing tracheal tube placement in pediatric patients. Further validation on a larger patient cohort is necessary to confirm these findings.

Introduction

The accurate placement of tracheal tubes plays a pivotal role in minimizing post-operative complications in pediatric patients undergoing mechanical ventilation. A significant challenge encountered in pediatric health care is the high incidence, ranging from 35%-50%, of tracheal tube misplacement primarily due to the relatively shorter tracheal length in children [1, 2]. Misplacements can lead to serious consequences such as hypoxia, atelectasis,

1

and hypercarbia, underscoring an undeniable need for accurate tracheal tube positioning [3, 4].

Current methods to ascertain the Optimal Tracheal Tube Depth (OTTD) employ chest X-ray or formula-based models tied to patient features like age and height [5, 6]. Although considered the gold standard, chest X-ray-guided determination is time-consuming and contributes to patient radiation exposure. On the other hand, formula-based models, albeit less invasive, often render limited accuracy and are less reliable [7, 8].

Therefore, our study capitalizes on cutting-edge machine learning techniques to address this gap. Our aim is to construct a more efficient and precise method for OTTD estimation void of the need for radiation exposure. The novelty of our approach lies in the application of machine learning to this clinical challenge— an avenue not previously explored in existing research [9, 10].

For the purpose of this study, we utilized a dataset that includes patient records from the Samsung Medical Center collected between January 2015 and December 2018. This data provides a comprehensive set of features, along with the OTTD determined via chest X-ray, facilitating a robust platform to train and evaluate our models [9, 10].

In light of this, four machine learning models were implemented and evaluated: Elastic Net, Support Vector Machine, Neural Network, and Random Forest. These models were specifically chosen for their respective strengths in regression tasks and potential to enhance estimation accuracy [11, 12, 13]. These models were juxtaposed against traditional formula-based models. Initial results indicate machine learning models as a promising avenue for OTTD estimation. They consistently deliver lower mean squared errors, suggesting an edge in accuracy over classical approaches. Nonetheless, the finer nuances of their application and performance necessitate further investigation on larger cohorts for definitive conclusions.

Results

Our analysis began with evaluating the dataset of 969 pediatric patient records (Table 1). We then conducted a detailed examination of the performance of various machine learning (ML) models in estimating the optimal tracheal tube depth (OTTD). For this purpose, we trained and tested four distinct models: Elastic Net, Support Vector Machine (SVM), Random Forest, and Neural Network. Our findings revealed that the Elastic Net model achieved the smallest mean squared error (MSE) of 1.25, closely followed by

the SVM model with an MSE of 1.28, and the Neural Network Model with an MSE of 1.44 (Table 2).

Table 1: Descriptive statistics of the dataset

	Tube ID	Sex	Age	Height	Weight	O.T.T.D
mean	3.69	0.539	0.758	66	7.13	10.2
std	0.568	0.499	1.44	19.1	4.77	1.77

Tube ID: Internal diameter of the tube (mm)
Sex: Patient sex (0 = Female, 1 = Male)
Age: Patient age (years, rounded to half years)
Height: Patient height (cm)
Weight: Patient weight (kg)
O.T.T.D: Optimal tracheal tube depth as determined by chest X-ray (cm)

Table 2: Performance metrics for different machine learning models

	MSE
Elastic Net	1.25
Support Vector Machine	1.28
Neural Network	1.44

The table presents the Mean Squared Error (MSE) for each machine learning model.
MSE: Mean Squared Error - a performance metric for machine learning models

To deploy a comparative analysis approach, we evaluated the performance of three formula-based models namely, a height-based model, an age-based model, and a tube diameter-based model. The results showed the age-based model, with an MSE of 2.05, performed better than the height-based model which had an MSE of 3.76. A slight enhancement was seen in the tube diameter-based model, achieving a marginally smaller MSE of 2.51 (Table 3).

Further, to confidently evaluate the differences in performance between the machine-learning models and formula-based models, we conducted a paired t-test. The resulting p-value of 0.1129 indicated that the difference was not statistically significant at a 0.05 alpha level (the customary threshold for claiming statistical significance), suggesting that the performance of ML and formula-based models were not significantly distinct.

In summary, these analyses highlight that despite not showing a statistically significant difference, machine learning models, specifically the Elastic Net, SVM, and Neural Network models, appear to achieve lower MSEs than

3

Table 3: Performance metrics for different formula-based models

	MSE
Height Model	3.76
Age Model	2.05
Tube Diameter Model	2.51

The table presents the Mean Squared Error (MSE) for each formula-based model.
MSE: Mean Squared Error - a performance metric for formula-based models
Height Model: OTTD = height [cm] / 10 + 5 cm
Age Model: OTTD per age category: $0 <=$ age [years] < 0.5: 9cm, $0.5 <=$ age [years] < 1: 10cm, $1 <$ age [years] < 2: 11cm, $2 <$ age [years]: 12cm + (age [years]) * 0.5 cm / year
Tube Diameter Model: OTTD (in cm) = 3 * (tube ID [mm]) * cm/mm

those of formula-based models when estimating OTTD. This evidence suggests a comparative advantage of ML models in terms of increased accuracy. However, the lack of statistically significant differences in the performance of ML and formula-based models necessitates further validation of these findings on a larger patient cohort.

Discussion

This study primarily focuses on addressing a critical aspect of pediatric healthcare—optimal placement of tracheal tubes, which is a persistent challenge due to significantly shorter tracheal length in this group [1, 2]. The misplacement of a tracheal tube could lead to grave consequences such as post-operative complications. Hence, precise estimation of the optimal tracheal tube depth (OTTD) is paramount.

Our study set out to improve the standard estimations by applying a novel approach of incorporating four machine learning models in estimating the OTTD. We utilized a comprehensive dataset derived from pediatric patients, which facilitated robust training and evaluation of these models. The results of Elastic Net, SVM, and Neural Network models consistently indicated smaller MSEs compared to the traditional methods, thus laying the foundation for a potential paradigm shift towards leveraging machine learning for OTTD estimation [11, 12]. Unlike the existing literature, which mostly revolves around formula-based models with limited accuracy [5, 6], our study is pioneering in employing machine learning for this application [10].

At this juncture, it is important to acknowledge certain limitations.

Firstly, the data used in our study was primarily from a single location, limiting the conclusions drawn to a particular demographic and ethnic group. The diversity of the patient population is an important factor to consider for the generalizability of our models. Secondly, the performance of our models may be influenced by the size of the dataset. Previous studies [5, 6] on OTTD estimation used similarly sized or larger datasets that could potentially influence outcomes, emphasizing the need for our findings to be validated on a larger and more diverse cohort.

Moreover, while our results revealed machine learning models achieving lower MSEs, the statistical significance was not evident. It underlines the cautious interpretation required for these findings. However, on the side of 'practical significance', our models add value. The use of machine learning for OTTD estimation eliminates the need for radiation exposure, which is an uncontestable advantage both in the short and long term.

Nevertheless, integrating these findings into clinical practice may present practical challenges. The deployment of machine learning models would require the setting up of a computational infrastructure and creating a collaborative environment with experts in the field capable of maintaining and updating the models as necessary. These aspects need to be considered while transitioning towards a machine learning-based approach.

In conclusion, our study provides a novel perspective on estimating OTTD using machine learning techniques. Despite the limitations and the absence of statistical significance, our preliminary findings underline the potential these techniques hold. The promising results achieved by our models imply the reduced reliance on radiation-intense techniques and consequently safer pediatric healthcare. Future research would benefit from validating our findings on larger, more diverse datasets, to make more definitive progress in this area.

Methods

Data Source

The dataset used in this study was obtained from a cohort of pediatric patients who received post-operative mechanical ventilation after surgery at Samsung Medical Center between January 2015 and December 2018. The dataset includes 969 patients aged 0-7 years and consists of several patient features such as sex, age, height, weight, and the optimal tracheal tube depth as determined by chest X-ray.

Data Preprocessing

Prior to the analysis, the dataset was loaded into a Pandas DataFrame using the Python programming language. No additional preprocessing steps were required as the dataset was provided in a clean and usable format. The dataset was then split into features (sex, age, height, weight) and the target variable (optimal tracheal tube depth), in preparation for the subsequent analysis steps.

Data Analysis

To estimate the optimal tracheal tube depth using machine learning models, four different models were employed: Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network. Each model was trained using the training set, which was obtained by randomly splitting the dataset into a training set and a testing set (30% of the data). The Random Forest model, Elastic Net model, and SVM model were implemented using the scikit-learn library, while the Neural Network model was implemented using the MLPRegressor class from the same library.

After training the models, predictions were made on the testing set and the mean squared error (MSE) was calculated to evaluate the performance of each model. The MSE value quantified the average squared difference between the predicted optimal tracheal tube depth and the ground truth obtained from the chest X-ray.

In addition to the machine learning models, three formula-based models were also computed to estimate the optimal tracheal tube depth. These formula-based models included the height-based model, which calculated the tube depth as the patient's height divided by 10 plus 5 cm; the age-based model, which assigned different tube depths based on age groups; and the ID-based model, which calculated the tube depth as three times the internal diameter of the tracheal tube.

All analyses and computations were performed using the Python programming language, and the statistical analysis was conducted using the scipy.stats module.

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:

```python
import pandas as pd
import numpy as np

# Load CSV data into pandas DataFrame
df = pd.read_csv("tracheal_tube_insertion.csv")

# Open file to write output
with open('data_exploration.txt', 'w') as f:

    # Data Size
    f.write("# Data Size\n")
    f.write(f"Number of Rows, Columns: {df.shape}\n\n")

    # Summary Statistics
    f.write("# Summary Statistics\n")
    summary_stats = df.describe(include='all')
    summary_stats_str = summary_stats.to_string()
    f.write(f"{summary_stats_str}\n\n")

    # Categorical Variables
    f.write("# Categorical Variables\n")
    categorical_variables =
    ↪   df.select_dtypes(include=['object']).columns
```

```
24      if len(categorical_variables) > 0:
25          for var in categorical_variables:
26              f.write(f"Most common values for variable {var}:
                ↪ {df[var].mode()[0]}\n")
27      else:
28          f.write("No categorical variables in data.\n")
29      f.write("\n")
30
31      # Missing Values
32      f.write("# Missing Values\n")
33      missing_values = df.isnull().sum()
34      missing_values_str = missing_values.to_string()
35      f.write(f"{missing_values_str}\n\n")
36
37  f.close()
38
```

B.2   Code Description

The code performs data exploration on the provided dataset using the pandas library in Python. The main objective of this code is to gain an understanding of the dataset and extract useful information for further analysis.

The code performs the following analysis steps:

1. Load the dataset: The code reads the "tracheal_tube_insertion.csv" file and loads it into a pandas DataFrame for manipulation and analysis.

2. Data Size: The code determines the number of rows and columns in the dataset using the shape attribute of the DataFrame. This provides an initial understanding of the dataset dimensions.

3. Summary Statistics: The code calculates the summary statistics of the dataset using the "describe" method of the DataFrame. This includes measures such as count, mean, standard deviation, minimum, and maximum for numerical features, and the most frequent value for categorical features. These statistics provide insights into the distribution and range of values for each variable.

4. Categorical Variables: The code identifies and analyzes categorical variables in the dataset. It selects columns with object data type using the "select_dtypes" method and checks if any categorical variables are present. If categorical variables are found, the code outputs the most common value for each categorical variable. This information helps understand the distribution of categorical variables and their importance in the dataset.

9

5. Missing Values: The code identifies missing values in the dataset using the "isnull" method of the DataFrame. It calculates the sum of missing values for each variable. This information helps in determining the completeness of the dataset and if any data imputation is necessary.

The code writes the results of the data exploration analysis into a text file called "data_exploration.txt". The file includes the following information:

1. Data Size: The number of rows and columns in the dataset. 2. Summary Statistics: The descriptive statistics of the dataset. 3. Categorical Variables: The most common values for each categorical variable. 4. Missing Values: The number of missing values for each variable.

The generated "data_exploration.txt" file serves as a reference for future analysis, providing an overview of the dataset and highlighting important characteristics for further investigation.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of Rows, Columns: (969, 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
No categorical variables in data.

# Missing Values
tube            0
sex             0
age_c           0
ht              0
```

10

```
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  import numpy as np
5  from sklearn.preprocessing import StandardScaler
6  from sklearn.model_selection import train_test_split
7  from sklearn.ensemble import RandomForestRegressor
8  from sklearn.linear_model import ElasticNet
9  from sklearn.svm import SVR
10 from sklearn.neural_network import MLPRegressor
11 from sklearn.metrics import mean_squared_error
12 import pickle
13 from scipy import stats
14
15 # LOAD DATA
16 df = pd.read_csv("tracheal_tube_insertion.csv")
17
18 # DATASET PREPARATIONS
19 # No dataset preparations are needed.
20
21 # DESCRIPTIVE STATISTICS
22 ## Table 0: "Descriptive statistics of the dataset"
23 desc_stats = df.describe().loc[['mean', 'std']]   # include
   ↪  only mean and standard deviation
24 desc_stats.to_pickle('table_0.pkl')
25
26 # PREPROCESSING
27 features = ['sex', 'age_c', 'ht', 'wt']
28 target = 'tube_depth_G'
```

11

```
29
30   # dividing data into training and testing sets
31   x_train, x_test, y_train, y_test =
     ↪   train_test_split(df[features], df[target], test_size=0.3,
     ↪   random_state=42)
32
33   # ANALYSIS
34   ## Table 1: "Performance metrics for different models"
35   results = {}
36   models = [RandomForestRegressor(), ElasticNet(), SVR(),
     ↪   MLPRegressor(max_iter=1000, random_state=42)]
37   names = ["Random Forest", "Elastic Net", "Support Vector
     ↪   Machine", "Neural Network"]
38
39   for i, model in enumerate(models):
40       model.fit(x_train, y_train) # fit the model
41       y_pred = model.predict(x_test) # make prediction on the
         ↪   test split
42       mse = mean_squared_error(y_test, y_pred) # calculate mean
         ↪   squared error
43       results[names[i]] = mse
44
45   df1 = pd.DataFrame.from_dict(results, orient='index',
     ↪   columns=['Mean Squared Error']).sort_values(by='Mean
     ↪   Squared Error')
46   df1 = df1.iloc[:3,:]   # Choose top 3 models
47   df1.to_pickle('table_1.pkl')
48
49   ## Table 2: "Performance metrics for formula-based models"
50   results_formula = {}
51
52   # height-based model
53   OTTD_ht = df['ht'] / 10 + 5
54   mse_ht = mean_squared_error(df['tube_depth_G'], OTTD_ht)
55   results_formula['Height Model'] = mse_ht
56
57   # age-based model
58   age = df['age_c']
59   OTTD_age = np.where(age < 0.5, 9, np.where(age < 1, 10,
     ↪   np.where(age < 2, 11, 12 + age * 0.5)))
```

12

```
60  mse_age = mean_squared_error(df['tube_depth_G'], OTTD_age)
61  results_formula['Age Model'] = mse_age
62
63  # id-based model
64  OTTD_id = 3 * df['tube']
65  mse_id = mean_squared_error(df['tube_depth_G'], OTTD_id)
66  results_formula['ID Model'] = mse_id
67
68  df2 = pd.DataFrame.from_dict(results_formula, orient='index',
    ↪   columns=['Mean Squared Error'])
69  df2.to_pickle('table_2.pkl')
70
71  # t-test comparison
72  ml_mse = df1['Mean Squared Error'].values
73  formula_mse = df2['Mean Squared Error'].values
74
75  ttest_res = stats.ttest_rel(ml_mse, formula_mse)
76
77  # SAVE ADDITIONAL RESULTS
78  additional_results = {
79   'Total number of observations': df.shape[0],
80   't-statistic of paired t-test between ML and formula-based
    ↪   models' : ttest_res.statistic,
81   'p-value of paired t-test between ML and formula-based
    ↪   models' : ttest_res.pvalue
82  }
83
84  with open('additional_results.pkl', 'wb') as f:
85      pickle.dump(additional_results, f)
86
```

C.2 Code Description

The code performs a data analysis to determine the optimal tracheal tube depth (OTTD) in pediatric patients who require mechanical ventilation. The dataset used in the analysis consists of 969 rows (patients) and 6 columns.

First, the code loads the dataset and prepares it for analysis. No dataset preparations are needed as the dataset is clean.

Next, the code generates descriptive statistics of the dataset, including

13

the mean and standard deviation, and saves it as "table_0.pkl" for future reference.

The code then performs preprocessing by selecting the features and target variable for the analysis. The features include patient sex, age, height, and weight, while the target variable is the optimal tracheal tube depth determined by chest X-ray.

The dataset is divided into training and testing sets using a test size of 0.3 (30% of the data is used for testing).

The code proceeds to analyze the data using different regression models. Four models are used: Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network. The code fits each model to the training data, makes predictions on the test data, and calculates the mean squared error (MSE) as a performance metric for each model. The results, including the MSE for each model, are saved as "table_1.pkl". The top three models with the lowest MSE are selected for further analysis.

In addition to the machine learning models, the code also evaluates formula-based models for comparison. Three formula-based models are implemented based on height, age, and the tube ID. The code calculates the MSE for each formula-based model and saves the results as "table_2.pkl".

To compare the performance of the machine learning models and the formula-based models, the code performs a paired t-test on the MSE values obtained from the two sets of models. The t-statistic and p-value of the t-test are calculated and saved as part of the additional results in the "additional_results.pkl" file.

Overall, the code conducts a comprehensive analysis of the dataset to determine the optimal tracheal tube depth in pediatric patients. It compares the performance of machine learning models and formula-based models and provides additional statistical results for further evaluation.

C.3 Code Output

table_0.pkl

```
          tube       sex     age_c          ht        wt  tube_depth_G
mean  3.693808  0.538700  0.757998  66.000516  7.126687     10.189474
std   0.568130  0.498758  1.440271  19.081267  4.774186      1.766052
```

table_1.pkl

```
                   Mean Squared Error
Elastic Net                  1.252317
```

14

```
Support Vector Machine            1.275379
Neural Network                    1.439884
```

table_2.pkl

```
               Mean Squared Error
Height Model             3.758860
Age Model                2.054923
ID Model                 2.508184
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    't-statistic of paired t-test between ML and formula-based models': -2.718
    ,
    'p-value of paired t-test between ML and formula-based models': 0.1129,
}
```

D   LaTeX Table Design

D.1   Code

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

```
1
2
3  # IMPORT
4  import pandas as pd
5  from typing import Dict, Tuple, Optional, Any
6  from my_utils import to_latex_with_note, format_p_value,
   ↪  is_str_in_df, split_mapping, AbbrToNameDef
7
8  # PREPARATION FOR ALL TABLES
9  shared_mapping: AbbrToNameDef = {
10  'tube': ('Tube ID', 'Internal diameter of the tube (mm)'),
11  'sex': ('Sex', 'Patient sex (0 = Female, 1 = Male)'),
12  'age_c': ('Age', 'Patient age (years, rounded to half
   ↪  years)'),
13  'ht': ('Height', 'Patient height (cm)'),
14  'wt': ('Weight', 'Patient weight (kg)'),
```

```python
15    'tube_depth_G': ('O.T.T.D', 'Optimal tracheal tube depth as
      ↪  determined by chest X-ray (cm)')
16  }
17
18  # TABLE 0
19  df = pd.read_pickle('table_0.pkl')
20
21  # RENAME ROWS AND COLUMNS
22  mapping = {k: v for k, v in shared_mapping.items() if
      ↪  is_str_in_df(df, k)}
23  abbrs_to_names, legend = split_mapping(mapping)
24  df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
25
26  # Save as latex
27  to_latex_with_note(
28   df, 'table_0.tex',
29   caption="Descriptive statistics of the dataset",
30   label='table:desc_stats',
31   note=None,
32   legend=legend)
33
34  # TABLE 1
35  df = pd.read_pickle('table_1.pkl')
36
37  # RENAME ROWS AND COLUMNS
38  mapping = {
39   'Mean Squared Error': ('MSE', 'Mean Squared Error - a
      ↪  performance metric for machine learning models'),
40  }
41
42  abbrs_to_names, legend = split_mapping(mapping)
43  df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
44
45  # Save as latex
46  to_latex_with_note(
47   df, 'table_1.tex',
48   caption="Performance metrics for different machine learning
      ↪  models",
49   label='table:machine_learning_performance',
```

```
50    note="The table presents the Mean Squared Error (MSE) for
      ↪    each machine learning model.",
51    legend=legend)
52
53    # TABLE 2
54    df = pd.read_pickle('table_2.pkl')
55
56    # RENAME ROWS AND COLUMNS
57    mapping = {
58     'Mean Squared Error': ('MSE', 'Mean Squared Error - a
      ↪    performance metric for formula-based models'),
59     'Height Model': ('Height Model', 'OTTD = height [cm] / 10 + 5
      ↪    cm'),
60     'Age Model': ('Age Model', 'OTTD per age category: 0 <= age
      ↪    [years] < 0.5: 9cm, 0.5 <= age [years] < 1: 10cm, 1 < age
      ↪    [years] < 2: 11cm, 2 < age [years]: 12cm + (age [years])
      ↪    * 0.5 cm / year'),
61     'ID Model': ('Tube Diameter Model', 'OTTD (in cm) = 3 * (tube
      ↪    ID [mm]) * cm/mm')
62    }
63
64    abbrs_to_names, legend = split_mapping(mapping)
65    df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
66
67    # Save as latex
68    to_latex_with_note(
69     df, 'table_2.tex',
70     caption="Performance metrics for different formula-based
      ↪    models",
71     label='table:formula_based_performance',
72     note="The table presents the Mean Squared Error (MSE) for
      ↪    each formula-based model.",
73     legend=legend)
74
75
```

D.2   Provided Code

The code above is using the following provided functions:

```python
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
```

18

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of the dataset}
\label{table:desc_stats}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrrr}
\toprule
 & Tube ID & Sex & Age & Height & Weight & O.T.T.D \\
\midrule
\textbf{mean} & 3.69 & 0.539 & 0.758 & 66 & 7.13 & 10.2 \\
\textbf{std} & 0.568 & 0.499 & 1.44 & 19.1 & 4.77 & 1.77 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Tube ID}: Internal diameter of the tube (mm)
\item \textbf{Sex}: Patient sex (0 = Female, 1 = Male)
\item \textbf{Age}: Patient age (years, rounded to half years)
\item \textbf{Height}: Patient height (cm)
\item \textbf{Weight}: Patient weight (kg)
\item \textbf{O.T.T.D}: Optimal tracheal tube depth as determined by chest X-ray
    (cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Performance metrics for different machine learning models}
\label{table:machine_learning_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
```

```latex
\toprule
 & MSE \\
\midrule
\textbf{Elastic Net} & 1.25 \\
\textbf{Support Vector Machine} & 1.28 \\
\textbf{Neural Network} & 1.44 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item The table presents the Mean Squared Error (MSE) for each machine learning
     model.
\item \textbf{MSE}: Mean Squared Error - a performance metric for machine
     learning models
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```latex
\begin{table}[h]
\caption{Performance metrics for different formula-based models}
\label{table:formula_based_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & MSE \\
\midrule
\textbf{Height Model} & 3.76 \\
\textbf{Age Model} & 2.05 \\
\textbf{Tube Diameter Model} & 2.51 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item The table presents the Mean Squared Error (MSE) for each formula-based
     model.
```

```
\item \textbf{MSE}: Mean Squared Error - a performance metric for formula-based
    models
\item \textbf{Height Model}: OTTD = height [cm] / 10 + 5 cm
\item \textbf{Age Model}: OTTD per age category: 0 $<$= age [years] $<$ 0.5:
    9cm, 0.5 $<$= age [years] $<$ 1: 10cm, 1 $<$ age [years] $<$ 2: 11cm, 2 $<$ age
    [years]: 12cm + (age [years]) * 0.5 cm / year
\item \textbf{Tube Diameter Model}: OTTD (in cm) = 3 * (tube ID [mm]) * cm/mm
\end{tablenotes}
\end{threeparttable}
\end{table}
```

References

[1] A. Uya, Nischal K. Gautam, Muhammad B. Rafique, O. Pawelek, Sya-masundar 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.

[2] Franziska Rost, B. Donaubauer, H. Kirsten, T. Schwarz, P. Zimmermann, M. Siekmeyer, Daniel Grfe, Sebastian Ebel, C. Kleber, M. Lacher, and M. Struck. Tracheal tube misplacement after emergency intubation in pediatric trauma patients: A retrospective, exploratory study. *Children*, 9, 2022.

[3] Christopher E. Kapsner, David C. Seaberg, C. Stengel, K. Ilkhanipour, and J. Menegazzi. The esophageal detector device: Accuracy and reliability in difficult airway settings. *Prehospital and Disaster Medicine*, 11:60 – 62, 1993.

[4] C. Massoth, C. Schlke, J. Kppe, R. Weiss, D. Ppping, Michael Dahrmann, A. Zarbock, and M. Wenk. Nasolaryngeal distances in the adult population and an evaluation of commercially available nasotracheal tubes. *Anesthesia & Analgesia*, 130:1018 – 1025, 2020.

[5] Sarah Nguyen, A. Fang, V. Saxton, and J. Holberton. Accuracy of a weight-based formula for neonatal gastric tube insertion length. *Advances in Neonatal Care*, 16:158161, 2016.

[6] R. Faigle, E. Marsh, R. Llinas, V. Urrutia, and R. Gottesman. Novel score predicting gastrostomy tube placement in intracerebral hemorrhage. *Stroke*, 46:3136, 2015.

[7] C. S. Bang, Hyun Lim, Hae Min Jeong, and Sungwook Hwang. Use of endoscopic images in the prediction of submucosal invasion of gastric neoplasms: Automated deep learning model development and usability study. *Journal of Medical Internet Research*, 23, 2021.

[8] V. Gharpure, K. Meert, A. Sarnaik, and N. Metheny. Indicators of postpyloric feeding tube placement in children. *Critical Care Medicine*, 28:2962–2966, 2000.

[9] S. A. Ingelse, H. Wiegers, J. Calis, J. V. van Woensel, and R. A. Bem. Early fluid overload prolongs mechanical ventilation in children with viral-lower respiratory tract disease*. *Pediatric Critical Care Medicine*, 18:e106e111, 2017.

[10] Sharon Dial, Peter Silver, Kevin Bock, and M. Sagy. Pediatric sedation for procedures titrated to a desired degree of immobility results in unpredictable depth of sedation. *Pediatric emergency care*, 17 6:414–20, 2001.

[11] Alex Dunn, Qi Wang, A. Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6:1–10, 2020.

[12] Yuting Huang, Jiangbing Du, Yufeng Chen, Ke Xu, and Zuyuan He. Photonics. In *2023 International Conference on Electrical Engineering and Photonics (EExPolytech)*, 2023.

[13] Edwin Raczko and Bogdan Zagajewski. Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral apex images. *European Journal of Remote Sensing*, 50:144 – 154, 2017.