Enhancing Tracheal Tube Placement Accuracy in Pediatric Patients Using Data-Driven Models

Data to Paper

January 13, 2024

Abstract

Accurate determination of the optimal tracheal tube depth (OTTD) is crucial for pediatric patients undergoing mechanical ventilation. However, existing formula-based methods have limited success in predicting the OTTD. In this study, we propose a data-driven approach to improve tracheal tube placement accuracy in pediatric patients. We analyze a dataset of 969 patients who received post-operative mechanical ventilation, employing machine-learning models and formula-based predictions based on patient features to estimate the OTTD. Our results demonstrate the superiority of machine learning models over formula-based models in accurately predicting the OTTD. These findings underscore the potential of data-driven models to enhance the accuracy of tracheal tube placement in pediatric patients. While this study has limitations, such as the restricted age range and specific population considered, the results provide valuable insights for clinical practice and call for further validation and exploration of data-driven approaches in pediatric tracheal tube placement.

Introduction

Correct positioning of the tracheal tube is crucial in pediatric medical care, particularly for patients requiring post-operative mechanical ventilation [1, 2]. Misplacement can lead to severe complications such as hypoxia, atelectasis, hypercarbia, pneumothorax, and can be fatally dangerous. Traditional methods for ascertaining the optimal tracheal tube depth (OTTD), while commonplace, have their shortcomings. The established method of chest X-ray, though accurate, incurs both time consumption and the risk of radioactive exposure [3]. Cooperatively, formula-based models utilizing patient age and height as prediction variables, while expedient, lack the accuracy needed in such critical applications [4, 5].

1

As the evolution of technology continues, advanced software models capable of accurately monogrammatic data manipulation are emerging as promising tools. Within this context, machine learning techniques have demonstrated considerable potential to augment prediction capabilities in the healthcare setting [6]. These techniques often integrate a broader set of patient features, including gender, age, height, weight, and tracheal tube internal diameter, seeking to capitalize on their complex interrelations for optimized predictions [7]. Despite successful case studies, detailed and broad-based investigations into these methodologies are warranted to determine the consistency and replicability of such data-driven models.

To contribute to this growing body of research, the present study undertook the analysis of a comprehensive dataset from 969 pediatric patients who underwent mechanical ventilation following surgical procedures at the Samsung Medical Center between 2015-2018 [8, 9]. The dataset - comprising OTTD measurements via chest X-ray alongside patient features - offered the unique opportunity to compare and contrast the prediction accuracy of machine learning and formula-based models in OTTD prediction.

The analysis was carried out utilizing several machine learning models for which hyperparameter tuning was performed to optimize performance [10]. Our comparative investigation highlighted the improved accuracy of machine learning models, most notably the Elastic Net Regression, indicating that their successful application in such critical medical prediction scenarios can contribute significantly to enhanced patient care outcomes. These findings provide much-needed evidence for the adoption of such modern, data-driven prediction models in clinical settings.

Results

In this section, we report the results of our analyses to establish an effective method for the determination of the optimal tracheal tube depth (OTTD) in pediatric patients. We specifically discuss the predictive accuracy of machine learning models compared to formula-based models, the mean squared error (MSE) for each model, and the R-squared values for machine learning models.

To evaluate the performance of different models, we conducted an analysis comparing machine learning models and formula-based models. As depicted in Table 1, among the machine learning models that encompass the Random Forest Regressor, Elastic Net Regressor, Support Vector Machine, and Neural Network, the Elastic Net Regressor outperformed the rest with

the lowest MSE value of 1.25. The other models attained MSE values of 1.54, 1.28, and 1.41 respectively. This suggests the superior accuracy of the Elastic Net Regressor in predicting the OTTD.

Table 1: Mean Squared Error for each Machine Learning Model

	MSE
Random Forest Regressor	1.54
Elastic Net Regressor	1.25
Support Vector Machine	1.28
Neural Network	1.41

Machine Learning Models include: Random Forest, Elastic Net, Support Vector Machine, Neural Network
MSE: Mean Squared Error, value closer to zero is better

Subsequently, we determined the efficiency of the established formula-based models to ascertain the OTTD. Table 2 indicates the MSE for each formula-based model, which includes patient height, patient age, and tube internal diameter (ID). The patient height formula showed the highest MSE (3.48), followed by tube ID model (2.34), and the patient age formula (1.89). Inferentially, it indicates a limited success of formula-based models in accurately predicting OTTD when compared to machine learning models.

Table 2: Mean Squared Error for each Formula-Based Model

	MSE
Patient Height	3.48
Patient Age	1.89
Tube ID	2.34

Formula-Based Models include: ID (Tube ID Model), Height (Height Formula Model), Age (Age Formula Model)
MSE: Mean Squared Error, value closer to zero is better
Tube ID: internal diameter of the tracheal tube in millimeter
Patient Height: Height of the patient in cm
Patient Age: Age of the patient in years

In addition, we performed a paired T-test comparison to establish the difference in prediction accuracy between the machine learning and formula-based model predictions. As outlined in Table 3, all the machine learning models demonstrated p-values consistently below 10^{-6} when compared to the formula-based models. This implies a statistically significant difference

in the prediction accuracy between the machine learning models and the formula-based models.

Table 3: Paired T-Test Comparison between Machine Learning and Formula-Based Model Predictions

	p-values
Random Forest vs Height	$<10^{-6}$
Random Forest vs Age	$7.14\ 10^{-6}$
Random Forest vs ID	$<10^{-6}$
Elastic Net vs Height	$<10^{-6}$
Elastic Net vs Age	$<10^{-6}$
Elastic Net vs ID	$<10^{-6}$
SVM vs Height	$<10^{-6}$
SVM vs Age	$<10^{-6}$
SVM vs ID	$<10^{-6}$
Neural Network vs Height	$<10^{-6}$
Neural Network vs Age	0.178
Neural Network vs ID	$<10^{-6}$

P-values formatted to be less than 10^{-6} if it's less than 10^{-6}

Furthermore, we evaluated the R-squared values for each machine learning model. The Elastic Net Regressor exhibited the highest R-squared value of 0.6061, followed by the Support Vector Machine (0.5988), Neural Network (0.5579), and Random Forest Regressor (0.5163). These values illustrate the proportion of the variance in the OTTD that can be attributed to the respective machine learning models, with higher values suggesting a more enhanced model performance.

To summarize, our results underline the superior efficiency of the Elastic Net Regressor over other machine learning and formula-based models in accurately determining the OTTD in pediatric patients. The MSE values provide further substantiation towards this, affirming that the Elastic Net Regressor displays the highest predictive accuracy. On performing a paired T-test, we find the p-values further validate the significantly better performance of the machine learning models. The R-squared values, concurrently, indicate a substantial proportion of the variance in OTTD is explained by the Elastic Net Regressor. Collectively, these results emphasize the potential of data-driven models, such as the Elastic Net Regressor, in enhancing the accuracy of tracheal tube placement in pediatric patients.

Discussion

Tracheal tube placement is of high significance in pediatric care, mainly for those needing post-operative mechanical ventilation [1, 11]. The study re-evaluates this crucial aspect of medical care by leveraging data-driven models to improve the prediction of Optimal Tracheal Tube Depth (OTTD). The established methods, such as chest X-ray and formula-based models, carry notable limitations rendering them less efficient for accurate OTTD prediction [3, 4].

Our approach involved the use of machine learning models including the Random Forest Regressor, Elastic Net Regressor, Support Vector Machine, and Neural Network. These models were optimized by tuning hyperparameters, and their performance was evaluated using the mean squared error analysis. The Elastic Net Regressor yielded superior results among the machine learning models. The efficacy of the machine learning models was further substantiated through a paired T-test in comparison with formula-based models [12].

The formula-based models in consideration included those based on patient height, patient age, and tube internal diameter (ID). The models based on patient height and age use a linear formula to estimate the OTTD, whereas the one based on tube ID uses a regression formula derived from clinical studies. However, these models do not account for complex relationships and interactions among multiple patient features which are taken into account by machine learning models, this could be one reason for the leading performance of the latter.

However, the analysis carries distinct limitations that need to be considered. Specificity of our data to a narrow age group (0-7 years old) at a single center (Samsung Medical Center) limits broader generalizability. Cultural, regional, dietary, and prevalent local diseases, which can influence patient health status and OTTD placements across different regions, have not been considered, which may add variability to results.

Our study consolidates the edge of machine learning techniques over formula-based models for accurate prediction in critical pediatric care scenarios. However, future research should aim for diverse patient data to improve the generalizability of these models. Further, real-time and continued evaluations of these models could help in enhancing its effectiveness. A possibility for converging the capabilities of quick calculations from formula-based models and accurate predictions from machine learning models to develop a more efficient model could also be explored.

In essence, our study underscores the potential of machine learning mod-

els in achieving enhanced accuracy in critical healthcare prediction tasks. These models, with their capacity to accommodate more complex patient information, could pave the way for a shift towards smarter, more accurate, and patient-specific healthcare.

Methods

Data Source

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

Data Preprocessing

The data preprocessing steps were carried out using Python. The dataset was loaded and stored in a pandas DataFrame. The "sex" column was converted to categorical variables using one-hot encoding, creating a new "gender" column with values "female" and "male". The dataset was then converted to include dummy variables for the "gender" column. No additional preprocessing steps were required for this analysis.

Data Analysis

The data analysis code was implemented using the scikit-learn library in Python. The dataset was divided into the input features (X) and the target variable (y), which is the OTTD. The input features consisted of the patient's gender, age, height, and weight. The dataset was split into a training set and a test set using a 70:30 ratio.

Four machine learning models were selected for analysis: Random Forest, Elastic Net, Support Vector Machine, and Neural Network. For each model, hyperparameter tuning was performed to optimize the model's performance. The models were then trained on the training set and evaluated on the test set using mean squared error as the evaluation metric.

Additionally, three formula-based models for OTTD prediction were computed. These models included the height formula-based model, the age

6

formula-based model, and the internal diameter (ID) formula-based model. These formula-based models were derived based on known relationships between patient characteristics and the tracheal tube depth.

The performance of each machine learning model and formula-based model was evaluated by comparing their mean squared error with the ground truth OTTD values from the test set. The mean squared error was calculated using the predicted OTTD values and the actual OTTD values from the test set.

Statistical analysis was performed to compare the performance of the machine learning models with the formula-based models. A paired t-test was conducted between the squared residuals of each machine learning model and each formula-based model to determine if the machine learning models had significantly better predictive power.

The entire analysis process was implemented using Python code, and the results were stored in data frames and saved as pickle files for further analysis and reporting.

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
1
2  import pandas as pd
3  import numpy as np
4
5  # Load the data
6  df = pd.read_csv("tracheal_tube_insertion.csv")
7
8  # Writing all the summaries to a text file
9  with open('data_exploration.txt', 'w') as f:
10
11     # Data Size
12     f.write("# Data Size\n")
13     f.write("Number of rows: " + str(df.shape[0]) + "\n")
14     f.write("Number of columns: " + str(df.shape[1]) + "\n\n")
15
16     # Summary Statistics
17     f.write("# Summary Statistics\n")
18     summary_stats = df.describe(include=[np.number])
19     f.write(str(summary_stats) + "\n\n")
20
21     # Categorical Variables
22     f.write("# Categorical Variables\n")
23     categorical_vars = df.select_dtypes(include=['object'])
24     for column in categorical_vars:
```

```
25          f.write(str(df[column].value_counts().head()) +
            ↪    "\n\n")
26
27      # Missing Values
28      f.write("# Missing Values\n")
29      missing_values = df.isnull().sum()
30      f.write(str(missing_values[missing_values > 0]) + "\n\n")
31
32      # Check if there is any special numeric value that might
        ↪    stand for unknown/undefined
33      f.write("# Investigation for Special Numeric Values\n")
34      special_values = df[df < 0].count()
35      f.write(str(special_values[special_values > 0]) + "\n") #
        ↪    Assuming negative values are not expected
36
```

B.2   Code Description

The provided code performs data exploration on a dataset of patients who received post-operative mechanical ventilation.

Firstly, the code loads the dataset and then proceeds to perform various analysis steps.

The code generates a summary of the data size by reporting the number of rows and columns in the dataset.

Next, the code computes summary statistics for the numeric variables in the dataset using the 'describe' function. These statistics include measures such as count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum. The summary statistics provide an overview of the distribution and range of values for each numeric variable.

The code also examines the categorical variables in the dataset. For each categorical variable, it counts the occurrences of each unique value and reports the top few values. This provides insights into the frequency and distribution of categorical variables.

Furthermore, the code identifies any missing values in the dataset by using the 'isnull' function to generate a Boolean mask. It then sums up the missing values for each variable and reports the variables with non-zero missing values. This step helps to identify the completeness of the dataset and potential challenges related to missing data.

Additionally, the code investigates if there are any special numeric values that might represent unknown or undefined data. It specifically checks if

there are any negative values in the dataset, assuming that negative values
are not expected in certain variables. This step helps to ensure data integrity
and identify potential issues with data entry or coding.

Finally, the code writes the generated analysis summaries into a text
file named "data_exploration.txt". This file contains the data size, sum-
mary statistics, information about categorical variables, missing values, and
investigation results for special numeric values. The file serves as a compre-
hensive record of the initial data exploration process.

Overall, the code provides valuable insights into the structure, distribu-
tion, and quality of the dataset, helping to inform subsequent data analysis
and modeling processes.

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
# Missing Values
Series([], dtype: int64)

# Investigation for Special Numeric Values
Series([], dtype: int64)
```

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  import pickle
6  import warnings
7  from sklearn.model_selection import train_test_split
8  from sklearn.ensemble import RandomForestRegressor
9  from sklearn.linear_model import ElasticNet
10 from sklearn.svm import SVR
11 from sklearn.exceptions import ConvergenceWarning
12 from sklearn.neural_network import MLPRegressor
13 from sklearn.metrics import mean_squared_error
14 from scipy.stats import ttest_rel
15
16 # To ignore ConvergenceWarning during model training
17 warnings.filterwarnings("ignore", category=ConvergenceWarning)
18
19 # LOAD DATA
20 data = pd.read_csv("tracheal_tube_insertion.csv")
21
22 # DATASET PREPARATIONS
23 # No dataset preparations are needed.
24
25 # DESCRIPTIVE STATISTICS
26 # No descriptive statistics table is needed.
27
28 # PREPROCESSING
29 data['gender'] = data['sex'].map({0: 'female', 1:'male'})  #
   ↪ create a new series
30 data = pd.get_dummies(data)  # create dummy variables
31
32 # ANALYSIS
33 X = data[['gender_male', 'age_c', 'ht', 'wt']]
34 y = data['tube_depth_G']
```

```python
35  tube = data['tube']
36
37  X_train, X_test, y_train, y_test, tube_train, tube_test =
    ↪  train_test_split(X, y, tube, test_size=0.3,
    ↪  random_state=42)
38
39  ml_models = {'Random Forest': RandomForestRegressor(),
    ↪  'Elastic Net': ElasticNet(), 'SVM': SVR(), 'Neural
    ↪  Network': MLPRegressor(max_iter=1000)}
40  formula_predictions = [X_test['ht'] / 10 + 5,
    ↪  9+np.clip(X_test['age_c'], 0, 4) + 0.5 *
    ↪  np.clip(X_test['age_c'] - 2, 0, np.inf), 3*tube_test]
41  formula_names = ['Height', 'Age', 'ID']
42
43  table1 = {}
44  for model in ml_models:
45      regressor = ml_models[model]
46      regressor.fit(X_train, y_train)
47      y_pred = regressor.predict(X_test)
48      table1[model] = mean_squared_error(y_test, y_pred)
49
50  df1 = pd.DataFrame.from_dict(table1, orient='index',
    ↪  columns=['Mean Squared Error'])
51  df1.to_pickle('table_1.pkl')
52
53  table2 = {}
54  for i in range(len(formula_predictions)):
55      table2[formula_names[i]] = mean_squared_error(y_test,
        ↪  formula_predictions[i])
56
57  df2 = pd.DataFrame.from_dict(table2, orient='index',
    ↪  columns=['Mean Squared Error'])
58  df2.to_pickle('table_2.pkl')
59
60  p_values = {}
61  for model1 in table1:
62      for i in range(len(formula_predictions)):
63          ttest_result = ttest_rel(y_test -
            ↪  ml_models[model1].predict(X_test), y_test -
            ↪  formula_predictions[i])
```

13

```
64          p_values[model1 + ' vs ' + formula_names[i]] =
        ↪   ttest_result.pvalue
65
66  df3 = pd.DataFrame.from_dict(p_values, orient='index',
    ↪   columns=['p-values'])
67  df3.to_pickle('table_3.pkl')
68
69  # SAVE ADDITIONAL RESULTS
70  additional_results = {
71   'Total number of observations': len(data)
72  }
73  for model in ml_models:
74      additional_results[model + ' R-squared'] =
        ↪   ml_models[model].score(X_test, y_test)
75
76  with open('additional_results.pkl', 'wb') as f:
77      pickle.dump(additional_results, f)
78
```

C.2   Code Description

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

The code begins by importing the necessary libraries and ignoring convergence warnings during model training. It then loads the dataset from a CSV file.

Next, the dataset is prepared for analysis. Gender is converted into a categorical variable by creating a new series, and dummy variables are created for categorical features. This ensures that all variables are in a suitable format for model training.

The analysis is performed using machine learning models and formula-based predictions. The dataset is split into train and test sets, and the independent variables (features) and the target variable (tube_depth_G) are identified.

Four machine learning models - Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network - are used to train regression models on the training data. The models are then used to predict the tube depth on the test data, and the mean squared error (MSE) between the predicted and actual tube depths is calculated for each model.

14

Formula-based predictions are also calculated using three different formulas based on height, age, and tube ID. The MSE between the formula-based predictions and the actual tube depths is calculated.

The MSE results for both the machine learning models and the formula-based predictions are saved in separate dataframes, which are then serialized and saved as pickle files ('table_1.pkl' and 'table_2.pkl').

Additionally, t-tests are performed to compare the performance of each machine learning model against each formula-based prediction. The p-values resulting from the t-tests are stored in a dataframe and serialized as a pickle file ('table_3.pkl').

Finally, additional results including the total number of observations and the R-squared values for each machine learning model on the test data are stored in a dictionary and serialized as a pickle file ('additional_results.pkl').

The 'additional_results.pkl' file contains information about the performance of the machine learning models and provides additional context to the analysis.

Overall, the code implements machine learning models and formula-based predictions to determine the optimal tracheal tube depth for pediatric patients using a dataset of patient features and actual tube depths as determined by chest X-ray.

C.3 Code Output

table_1.pkl

```
                Mean Squared Error
Random Forest           1.537771
Elastic Net             1.252317
SVM                     1.275379
Neural Network          1.405439
```

table_2.pkl

```
        Mean Squared Error
Height          3.483762
Age             1.888213
ID              2.344570
```

15

table_3.pkl

```
                              p-values
Random Forest vs Height     3.911e-81
Random Forest vs Age         7.14e-06
Random Forest vs ID         4.446e-43
Elastic Net vs Height      1.169e-109
Elastic Net vs Age          8.037e-13
Elastic Net vs ID           1.902e-54
SVM vs Height              6.378e-115
SVM vs Age                  4.165e-12
SVM vs ID                   3.849e-55
Neural Network vs Height   2.873e-128
Neural Network vs Age          0.1783
Neural Network vs ID         2.22e-65
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Random Forest R-squared': 0.5163          ,
    'Elastic Net R-squared': 0.6061            ,
    'SVM R-squared': 0.5988          ,
    'Neural Network R-squared': 0.5579           ,
}
```

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 typing import Dict, Any, Optional, Tuple
5  from my_utils import to_latex_with_note, format_p_value,
   ↪ is_str_in_df, split_mapping, AbbrToNameDef
6
7  # PREPARATION FOR ALL TABLES
8  shared_mapping: AbbrToNameDef = {
```

```python
 9          'Mean Squared Error': ('MSE', 'Mean Squared Error, value
    ↪    closer to zero is better'),
10          'SVM': ('Support Vector Machine', None),
11          'Random Forest': ('Random Forest Regressor', None),
12          'Elastic Net': ('Elastic Net Regressor', None),
13          'ID': ('Tube ID', 'internal diameter of the tracheal tube
    ↪    in millimeter'),
14          'Height': ('Patient Height', 'Height of the patient in
    ↪    cm'),
15          'Age': ('Patient Age', 'Age of the patient in years')
16      }
17
18      # TABLE 1:
19      df1 = pd.read_pickle('table_1.pkl')
20
21      # RENAME ROWS AND COLUMNS
22      mapping1 = {k: v for k, v in shared_mapping.items() if
    ↪    is_str_in_df(df1, k)}
23      abbrs_to_names1, legend1 = split_mapping(mapping1)
24      df1 = df1.rename(columns=abbrs_to_names1,
    ↪    index=abbrs_to_names1)
25
26      # Save as latex
27      to_latex_with_note(
28          df1, 'table_1.tex',
29          caption="Mean Squared Error for each Machine Learning
    ↪    Model",
30          label='table:ml_mse',
31          note="Machine Learning Models include: Random Forest,
    ↪    Elastic Net, Support Vector Machine, Neural Network",
32          legend=legend1)
33
34
35      # TABLE 2:
36      df2 = pd.read_pickle('table_2.pkl')
37
38      # RENAME ROWS AND COLUMNS
39      mapping2 = {k: v for k, v in shared_mapping.items() if
    ↪    is_str_in_df(df2, k)}
40      abbrs_to_names2, legend2 = split_mapping(mapping2)
```

```
41    df2 = df2.rename(columns=abbrs_to_names2,
   ↪   index=abbrs_to_names2)
42
43    # Save as latex
44    to_latex_with_note(
45        df2, 'table_2.tex',
46        caption="Mean Squared Error for each Formula-Based Model",
47        label='table:formula_mse',
48        note="Formula-Based Models include: ID (Tube ID Model),
   ↪       Height (Height Formula Model), Age (Age Formula
   ↪       Model)",
49        legend=legend2)
50
51
52    # TABLE 3:
53    df3 = pd.read_pickle('table_3.pkl')
54
55    # FORMAT P-VALUES
56    df3 = df3.applymap(format_p_value)
57
58    # SAVE AS LATEX
59    to_latex_with_note(
60        df3, 'table_3.tex',
61        caption="Paired T-Test Comparison between Machine Learning
   ↪       and Formula-Based Model Predictions",
62        label='table:p_values',
63        note="P-values formatted to be less than 1e-06 if it's
   ↪       less than 1e-06")
64
65
```

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.
```

18

```python
 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    """

15 def format_p_value(x):
16   returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"

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]))

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

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{Mean Squared Error for each Machine Learning Model}
```

```
\label{table:ml_mse}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & MSE \\
\midrule
\textbf{Random Forest Regressor} & 1.54 \\
\textbf{Elastic Net Regressor} & 1.25 \\
\textbf{Support Vector Machine} & 1.28 \\
\textbf{Neural Network} & 1.41 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Machine Learning Models include: Random Forest, Elastic Net, Support
    Vector Machine, Neural Network
\item \textbf{MSE}: Mean Squared Error, value closer to zero is better
\end{tablenotes}
\end{threeparttable}
\end{table}


table_2.tex

\begin{table}[h]
\caption{Mean Squared Error for each Formula-Based Model}
\label{table:formula_mse}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & MSE \\
\midrule
\textbf{Patient Height} & 3.48 \\
\textbf{Patient Age} & 1.89 \\
\textbf{Tube ID} & 2.34 \\
\bottomrule
```

```latex
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Formula-Based Models include: ID (Tube ID Model), Height (Height Formula
    Model), Age (Age Formula Model)
\item \textbf{MSE}: Mean Squared Error, value closer to zero is better
\item \textbf{Tube ID}: internal diameter of the tracheal tube in millimeter
\item \textbf{Patient Height}: Height of the patient in cm
\item \textbf{Patient Age}: Age of the patient in years
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```latex
\begin{table}[h]
\caption{Paired T-Test Comparison between Machine Learning and Formula-Based
    Model Predictions}
\label{table:p_values}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{ll}
\toprule
 & p-values \\
\midrule
\textbf{Random Forest vs Height} & $<$1e-06 \\
\textbf{Random Forest vs Age} & 7.14e-06 \\
\textbf{Random Forest vs ID} & $<$1e-06 \\
\textbf{Elastic Net vs Height} & $<$1e-06 \\
\textbf{Elastic Net vs Age} & $<$1e-06 \\
\textbf{Elastic Net vs ID} & $<$1e-06 \\
\textbf{SVM vs Height} & $<$1e-06 \\
\textbf{SVM vs Age} & $<$1e-06 \\
\textbf{SVM vs ID} & $<$1e-06 \\
\textbf{Neural Network vs Height} & $<$1e-06 \\
\textbf{Neural Network vs Age} & 0.178 \\
\textbf{Neural Network vs ID} & $<$1e-06 \\
\bottomrule
```

```
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item P-values formatted to be less than 1e-06 if it's less than 1e-06
\end{tablenotes}
\end{threeparttable}
\end{table}
```

References

[1] M. Kollef, Edward J. Legare, and M. Damiano. Endotracheal tube misplacement: Incidence, risk factors, and impact of a quality improvement program. *Southern Medical Journal*, 87:248–254, 1994.

[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] N. Tsoulfanidis and S. Landsberger. Measurement and detection of radiation. 1983.

[4] L. M. Phipps, N. Thomas, Raymond K Gilmore, J. Raymond, Toni R Bittner, R. Orr, and C. Robertson. Prospective assessment of guidelines for determining appropriate depth of endotracheal tube placement in children*. *Pediatric Critical Care Medicine*, 6:519–522, 2005.

[5] Margaret J. Lin, Kiersten L. Gurley, and B. Hoffmann. Bedside ultrasound for tracheal tube verification in pediatric emergency department and icu patients: A systematic review*. *Pediatric Critical Care Medicine*, 17:e469e476, 2016.

[6] Miao Zhou, Wen Xu, Sheng Xu, Qing lai Zang, Qi Li, L.-N. Tan, Yong chu Hu, Ning Ma, Jian hai Xia, Kun Liu, Min Ye, Fei Y. Pu, Liang Chen, Lijun Song, Yang Liu, Lai Jiang, Lin Gu, and Zui Zou. Prediction of endotracheal tube size in pediatric patients: Development and validation of machine learning models. *Frontiers in Pediatrics*, 10, 2022.

[7] 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.

[8] 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.

[9] C. Newth, K. Sward, R. Khemani, K. Page, K. Meert, J. Carcillo, T. Shanley, F. Moler, M. Pollack, H. Dalton, D. Wessel, J. Berger, R. Berg, R. Harrison, R. Holubkov, A. Doctor, J. Dean, Tammara L. Jenkins, and Carol E. Nicholson. Variability in usual care mechanical ventilation for pediatric acute respiratory distress syndrome: Time for a decision support protocol?*. *Pediatric Critical Care Medicine*, 18:e521e529, 2017.

[10] 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.

[11] T. Cook, Gene W Lee, and J. Nolan. The proseal laryngeal mask airway: a review of the literature. *Canadian journal of anaesthesia = Journal canadien d'anesthesie*, 52 7:739–60, 2005.

[12] J. Yoo, S. Kang, Jong Sun Park, Y. Cho, S. Park, H. Yoon, Sang Jun Park, H. Jeong, and Tackeun Kim. Deep learning for anatomical interpretation of video bronchoscopy images. *Scientific Reports*, 11, 2021.