

# Machine learning models for accurate determination of tracheal tube depth in pediatric patients

Data to Paper

February 20, 2024

## Abstract

Accurate determination of the optimal tracheal tube depth (OTTD) is crucial in pediatric patients undergoing mechanical ventilation to prevent serious complications. However, current methods for determining OTTD are either time-consuming or have limited success. This study addresses the need for a data-driven approach to determine OTTD in pediatric patients. We conducted a comprehensive analysis using a dataset of pediatric patients aged 0-7 years who received post-operative mechanical ventilation, with features extracted from their electronic health records. We applied random forest regression and elastic net regression models to predict the OTTD. Our results demonstrate the good performance of both models, with the elastic net regression model outperforming the random forest regression model in terms of accuracy. Importantly, these models utilize patient features such as age, sex, height, and weight as predictors, offering a faster and potentially more accurate alternative to the existing methods. However, validation of these models in a larger cohort is necessary before implementing them in clinical practice. Accurate determination of OTTD has the potential to greatly improve patient safety and outcomes in pediatric mechanical ventilation.

## Results

We first conducted a descriptive analysis of the patient characteristics in our dataset, yielding key insights about the patient demographics and the variation in OTTD values (Table 1). Our cohort included 969 pediatric patients aged 0-7 years. The mean and standard deviation (SD) of patient age was 0.758 years and 1.44 years respectively, while the mean height was 66 cm (SD=19.1 cm) and the mean weight was 7.13 kg (SD=4.77 kg). As

shown in Table 2, the mean OTTD, as determined by chest X-ray, was 10.2 cm (SD=1.77 cm).

Table 1: Descriptive Statistics of Age, Sex, Height, Weight, and OTTD

	Sex	Age	Height	Weight	OTTD
<b>mean</b>	0.539	0.758	66	7.13	10.2
<b>std</b>	0.499	1.44	19.1	4.77	1.77

**Sex:** Patient sex (0=female, 1=male)

**Age:** Patient age (years, rounded to half years)

**Height:** Patient height (cm)

**Weight:** Patient weight (kg)

**OTTD:** Optimal tracheal tube depth as determined by chest X-ray (in cm)

Table 2: Descriptive Statistics of Age, Sex, Height, Weight, and OTTD

	Sex	Age	Height	Weight	OTTD
<b>mean</b>	0.539	0.758	66	7.13	10.2
<b>std</b>	0.499	1.44	19.1	4.77	1.77

**Sex:** Patient sex (0=female, 1=male)

**Age:** Patient age (years, rounded to half years)

**Height:** Patient height (cm)

**Weight:** Patient weight (kg)

**OTTD:** Optimal tracheal tube depth as determined by chest X-ray (in cm)

Building on these patient features, we ventured to construct predictive models for OTTD using machine learning approaches. The performance of the random forest regression model is presented in Table 3, which shows a Mean Squared Error (MSE) of 1.95, demonstrating its predictive ability. However, the elastic net regression model outperformed the random forest regression model, achieving an MSE of 1.33 as detailed in Table 4.

Table 3: Random Forest Model Performance: Mean Square Error

	RF MSE
<b>Random Forest</b>	1.95

**RF MSE:** Random Forest model - Mean Square Error

In order to confirm the observed differences in the performance of the two models, we performed a paired-sample t-test on the squared prediction errors

Table 4: Elastic Net Model Performance: Mean Square Error

	EN MSE
<b>Elastic Net</b>	1.33

**EN MSE:** Elastic Net model - Mean Square Error

of both models. The t- statistic was found to be 3.752, with a corresponding p-value  $< 0.000212$ , thus establishing a statistically significant difference in the performance of the two models.

In summary, our results underscore the potential of machine learning models, particularly the elastic net model, in increasing the precision of OTTD prediction in pediatric patients receiving mechanical ventilation. Utilising features like sex, age, height, and weight, these models open up opportunities for safer and more effective OTTD determination in clinical practice.

## 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 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 # Open the summary file
9 with open('data_exploration.txt', 'w') as f:
10
11     # Data size
12     f.write('# Data Size\n')
13     f.write(f'Number of rows: {df.shape[0]}\n')
14     f.write(f'Number of columns: {df.shape[1]}\n\n')
15
16     # Summary statistics
17     f.write('# Summary Statistics\n')
18     summary_statistics = df.describe(include=[np.number])
19     summary_statistics.to_string(f)
20     f.write('\n')
21
22     # Categorical variables
23     categorical_variables = df.select_dtypes(include=['bool',
    ↪     'object'])
```

```

24
25     # If there is any categorical variable
26     if not categorical_variables.empty:
27         f.write('# Categorical Variables\n')
28         for var in categorical_variables.columns:
29             f.write(f'{var} value counts:\n')
30
31             ↪ f.write(f'{categorical_variables[var].value_counts()}\n')
32         f.write('\n')
33
34     # If no categorical variable is available
35     else:
36         f.write('# Categorical Variables\n')
37         f.write('There are no categorical variables in the
38         ↪ data.\n\n')
39
40     # Missing values
41     f.write('# Missing Values\n')
42     missing_values = df.isnull().sum()
43     missing_values.to_string(f)
44     f.write('\n')

```

## B.2 Code Description

The provided code performs a basic data exploration of the dataset contained in the "tracheal\_tube\_insertion.csv" file.

Firstly, it loads the dataset into a Pandas DataFrame.

Then, it writes a summary of the data size into the "data\_exploration.txt" file, including the number of rows and columns in the dataset.

Next, it calculates and writes summary statistics for the numerical variables in the dataset, such as count, mean, standard deviation, minimum, and maximum values. These statistics provide an overview of the data distribution and central tendency.

If the dataset contains any categorical variables, the code writes the value counts for each categorical variable into the "data\_exploration.txt" file. This provides information on the frequency of each category in the dataset.

Lastly, the code determines and writes the number of missing values for each variable in the dataset. This information is useful for identifying any data gaps or inconsistencies that may need to be addressed during analysis.

These exploratory analysis steps help researchers gain initial insights into the dataset and understand its general characteristics, such as its size, variable types, statistical properties, and missing data. These insights can guide further data cleaning, preprocessing, or feature selection steps in the research process.

### 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
There are no categorical variables in the data.

# Missing Values
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  import numpy as np
5  from sklearn.ensemble import RandomForestRegressor
6  from sklearn.linear_model import ElasticNetCV
7  from sklearn.model_selection import train_test_split
8  from sklearn.metrics import mean_squared_error
9  import scipy.stats as stats
10 import pickle
11
12 # LOAD DATA
13 data = pd.read_csv("tracheal_tube_insertion.csv")
14
15 # DATASET PREPARATIONS
16 # No dataset preparations are needed.
17
18 # DESCRIPTIVE STATISTICS
19 desc_stats = data[['sex', 'age_c', 'ht', 'wt',
20 ↪ 'tube_depth_G']].describe().loc[['mean', 'std']]
21 desc_stats.to_pickle('table_0.pkl')
22
23 # PREPROCESSING
24 # No preprocessing is needed, because all the data features
25 ↪ are already numeric.
26
27 # ANALYSIS
28 ## Table 1: "Descriptive statistics of age, sex, height,
29 ↪ weight, and OTTD"
30 df1 = desc_stats
31 df1.to_pickle('table_1.pkl')
32
33 # Model preparation
34 features = data[['sex', 'age_c', 'ht', 'wt']]
35 target = data['tube_depth_G']
36 X_train, X_test, y_train, y_test = train_test_split(features,
37 ↪ target, test_size=0.3, random_state=0)
38
39 # RF model
40 rf_model = RandomForestRegressor(random_state=1)

```



```

37 rf_model.fit(X_train, y_train)
38 rf_pred = rf_model.predict(X_test)
39
40 ## Table 2: "RF model performance: MSE"
41 df2 = pd.DataFrame({'RF_MSE': [mean_squared_error(y_test,
    ↪ rf_pred)]}, index=['Random Forest'])
42 df2.to_pickle('table_2.pkl')
43
44 # EN model
45 en_model = ElasticNetCV(cv=5, l1_ratio=[.1, .5, .7, .9, .95,
    ↪ .99, 1])
46 en_model.fit(X_train, y_train)
47 en_pred = en_model.predict(X_test)
48
49 ## Table 3: "EN model performance: MSE"
50 df3 = pd.DataFrame({'EN_MSE': [mean_squared_error(y_test,
    ↪ en_pred)]}, index=['Elastic Net'])
51 df3.to_pickle('table_3.pkl')
52
53 # Hypothesis test
54 t_test_result = stats.ttest_rel((rf_pred - y_test) ** 2,
    ↪ (en_pred - y_test) ** 2)
55
56 # SAVE ADDITIONAL RESULTS
57 additional_results = {
58     'Total number of observations': len(data),
59     't_stat for hypothesis test': t_test_result.statistic,
60     'p_val for hypothesis test': t_test_result.pvalue
61 }
62 with open('additional_results.pkl', 'wb') as f:
63     pickle.dump(additional_results, f)
64

```

## C.2 Code Description

The code performs a data analysis to determine the optimal tracheal tube depth (OTTD) for pediatric patients who require mechanical ventilation. The analysis is based on a dataset containing patient information such as sex, age, height, weight, and the OTTD determined by chest X-ray.

The code starts by loading the dataset and computing descriptive statis-

tics for the features and the target variable (OTTD). The statistics, including mean and standard deviation, are saved in a pickle file for further reference.

Next, the code prepares the data for analysis by splitting it into training and test sets. Two regression models, Random Forest (RF) and Elastic Net (EN), are then trained using the training set. The models are used to predict the OTTD for the test set.

The performance of each model is evaluated by calculating the mean squared error (MSE) between the predicted and actual values of OTTD. The RF and EN models' MSE values are saved in a pickle file for comparison.

To test the hypothesis that there is a difference in performance between the RF and EN models, a t-test is performed on the squared error values for the two models. The t-statistic and p-value of the test are computed and saved in the "additional\_results.pkl" file.

Overall, the code conducts a data analysis to determine the optimal tracheal tube depth for pediatric patients using regression models, evaluates the performance of the models, and performs a hypothesis test to compare the models' performance. The results are saved in pickle files for further analysis and reference.

### C.3 Code Output

#### table\_0.pkl

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

#### table\_1.pkl

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

#### table\_2.pkl

	RF_MSE
Random Forest	1.95202

table\_3.pkl

```
                EN_MSE
Elastic Net  1.329487
```

additional\_results.pkl

```
{
    'Total number of observations': 969,
    't_stat for hypothesis test': 3.752,
    'p_val for hypothesis test': 0.000212,
}
```

## 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
6  # import
7  from my_utils import to_latex_with_note, format_p_value,
   ↪ is_str_in_df, split_mapping, AbbrToNameDef
8
9  # PREPARATION FOR ALL TABLES
10 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
11
12 shared_mapping: AbbrToNameDef = {
13     'sex': ('Sex', 'Patient sex (0=female, 1=male)'),
14     'age_c': ('Age', 'Patient age (years, rounded to half
   ↪ years)'),
15     'ht': ('Height', 'Patient height (cm)'),
16     'wt': ('Weight', 'Patient weight (kg)'),
17     'tube_depth_G': ('OTTD', 'Optimal tracheal tube depth as
   ↪ determined by chest X-ray (in cm)'),
18     'RF_MSE': ('RF MSE', 'Random Forest model - Mean Square
   ↪ Error'),
```

```

19  'EN_MSE': ('EN MSE', 'Elastic Net model - Mean Square Error')
20  }
21
22  # TABLE 0:
23  df = pd.read_pickle('table_0.pkl')
24
25  # RENAME ROWS AND COLUMNS
26  mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
27  abbrs_to_names, legend = split_mapping(mapping)
28  df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
29
30  # Save as latex:
31  to_latex_with_note(
32  df, 'table_0.tex',
33  caption="Descriptive Statistics of Age, Sex, Height, Weight,
    ↪ and OTTD",
34  label='table:stat_desc_0',
35  note="",
36  legend=legend)
37
38  # TABLE 1:
39  df = pd.read_pickle('table_1.pkl')
40
41  # RENAME ROWS AND COLUMNS
42  mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
43  abbrs_to_names, legend = split_mapping(mapping)
44  df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
45
46  # Save as latex:
47  to_latex_with_note(
48  df, 'table_1.tex',
49  caption="Descriptive Statistics of Age, Sex, Height, Weight,
    ↪ and OTTD",
50  label='table:stat_desc_1',
51  note="",
52  legend=legend)
53
54  # TABLE 2:

```

```

55 df = pd.read_pickle('table_2.pkl')
56
57 # RENAME ROWS AND COLUMNS
58 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
59 abbrs_to_names, legend = split_mapping(mapping)
60 df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
61
62 # Save as latex:
63 to_latex_with_note(
64     df, 'table_2.tex',
65     caption="Random Forest Model Performance: Mean Square Error",
66     label='table:rf_model_performance',
67     note="",
68     legend=legend)
69
70 # TABLE 3:
71 df = pd.read_pickle('table_3.pkl')
72
73 # RENAME ROWS AND COLUMNS
74 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
75 abbrs_to_names, legend = split_mapping(mapping)
76 df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
77
78 # Save as latex:
79 to_latex_with_note(
80     df, 'table_3.tex',
81     caption="Elastic Net Model Performance: Mean Square Error",
82     label='table:en_model_performance',
83     note="",
84     legend=legend)
85
86

```

## 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\_0.tex

```
\begin{table}[h]
\caption{Descriptive Statistics of Age, Sex, Height, Weight, and OTTD}
\label{table:stat_desc_0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrr}
\toprule
& Sex & Age & Height & Weight & OTTD \\
\midrule
\textbf{mean} & 0.539 & 0.758 & 66 & 7.13 & 10.2 \\
\textbf{std} & 0.499 & 1.44 & 19.1 & 4.77 & 1.77 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\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{OTTD}: Optimal tracheal tube depth as determined by chest X-ray
(in cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

#### table\_1.tex

```
\begin{table}[h]
\caption{Descriptive Statistics of Age, Sex, Height, Weight, and OTTD}
\label{table:stat_desc_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrr}
\toprule
```

```

& Sex & Age & Height & Weight & OTTD \\
\midrule
\textbf{mean} & 0.539 & 0.758 & 66 & 7.13 & 10.2 \\
\textbf{std} & 0.499 & 1.44 & 19.1 & 4.77 & 1.77 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\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{OTTD}: Optimal tracheal tube depth as determined by chest X-ray
(in cm)
\end{tablenotes}
\end{threeparttable}
\end{table}

```

#### table\_2.tex

```

\begin{table}[h]
\caption{Random Forest Model Performance: Mean Square Error}
\label{table:rf_model_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
& RF MSE \\
\midrule
\textbf{Random Forest} & 1.95 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{RF MSE}: Random Forest model - Mean Square Error
\end{tablenotes}
\end{threeparttable}
\end{table}

```



table\_3.tex

```
\begin{table}[h]
\caption{Elastic Net Model Performance: Mean Square Error}
\label{table:en_model_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
& EN MSE \\\
\midrule
\textbf{Elastic Net} & 1.33 \\\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{EN MSE}: Elastic Net model - Mean Square Error
\end{tablenotes}
\end{threeparttable}
\end{table}
```