Improving Tracheal Tube Depth Prediction in Pediatric Patients

Data to Paper

February 20, 2024

Abstract

Accurate positioning of the tracheal tube tip is crucial for optimal mechanical ventilation in pediatric patients. However, current methods for determining the optimal tracheal tube depth (OTTD) have limitations, leading to frequent misplacements with potential complications. To address this challenge, we developed a data-driven approach to predict tracheal tube depth in pediatric patients using machine learning algorithms. We utilized a dataset of 969 patients, aged 0-7 years, and employed Random Forest and Elastic Net models. Our study fills the research gap in accurate OTTD prediction and demonstrates the potential of data-driven approaches. The analysis revealed significant variation in OTTD and identified the Elastic Net model as having a marginally lower mean squared residual compared to the Random Forest model. These results highlight the potential of our approach to improve tracheal tube positioning and enhance patient safety. However, our study acknowledges limitations in the reliance on a single-center dataset. Additional research is needed to validate our findings and explore alternative methods.

Results

Initially, we undertook a comprehensive exploratory data analysis to comprehend the distribution and correlation among the variables in a dataset consisting of 969 pediatric patient observations. Our primary focus was on the Optimal Tracheal Tube Depth (OTTD), defined as the optimal position of the tracheal tube tip in pediatric patients for mechanical ventilation, and patient characteristics such as sex, age, height and weight. Table 1 provides a descriptive statistical overview of these variables. We observed a substantial variation in the OTTD measurement, with an average OTTD of 10.2

cm (SD = 1.77 cm). The sample contained almost an equal distribution of sexes (mean = 0.539, SD = 0.499) and ages ranging from 0 to 7 years (mean = 0.758 years, SD = 1.44 years).

Table 1: Descriptive Statistics of Patient Features and OTTD

	Sex	Age	Height	Weight	OTTD
Mean	0.539	0.758	66	7.13	10.2
Standard Deviation	0.499	1.44	19.1	4.77	1.77

This table provides the mean and standard deviation for each variable.
Sex: 0: Female, 1: Male
Age: Age in years, rounded to the nearest half-year
Height: Height in centimeters (cm)
Weight: Weight in kilograms (kg)
OTTD: Optimal Tracheal Tube Depth as determined by Chest X-ray in cm

The aim of this study was to predict OTTD using machine learning models, specifically Random Forest and Elastic Net. The models were trained on a selected training set of 775 observations and evaluated on a testing set of 194 observations. The summary of the performance of these models, as measured by the Mean Squared Residuals, is provided in Table 2. It demonstrates that the Elastic Net model had a marginally lower mean squared residual of 1.24, compared to the Random Forest's 1.55.

Table 2: Mean Squared Residuals for Random Forest and Elastic Net

Model	Mean Squared Residual
Random Forest	1.55
Elastic Net	1.24

Finally, we carried out a paired t-test to statistically compare the mean squared residuals of the Random Forest and Elastic Net models. As suggested by the results shared in Table 3, a statistically significant difference was observed between these models, substantiated by a t-statistic value of 2.34 and a p-value less than 0.05. However, this result merely indicates a statistically significant difference in the mean squared residuals of the two models, and shouldn't be interpreted as a comprehensive measure of model superiority.

Collectively, these results evidence a wide variation in pediatric patient

Table 3: Paired T-Test Results for Mean Squared Residuals

Model Comparison	t-statistic	p-value
RF vs EN Mean Squared Residuals	2.34	0.0203

features and OTTD. They also reflect the potential of the Elastic Net model, as demonstrated by its marginally lower mean squared residual and the statistically significant difference in the t-test comparison with the Random Forest model.

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 a file to write the output
9  with open('data_exploration.txt','w') as outfile:
10
11     # Data Size
12     outfile.write("# Data Size\n")
13     outfile.write("Number of Rows: " + str(df.shape[0]) +
       ↪ "\n")
14     outfile.write("Number of Columns: " + str(df.shape[1]) +
       ↪ "\n\n")
15
16     # Summary Statistics
17     outfile.write("# Summary Statistics\n")
18     outfile.write(str(df.describe()) + "\n\n")
19
20     # Categorical Variables
21     outfile.write("# Categorical Variables\n")
```

```
    categorical_columns =
    ↪  df.select_dtypes(include=['object']).columns
    for col in categorical_columns:
        outfile.write("Most Common Value for "+ col +": " +
        ↪  str(df[col].mode().values[0]) + "\n")
    outfile.write("\n")

    # Missing Values
    outfile.write("# Missing Values\n")
    outfile.write("Missing or undefined values for each
    ↪  column:\n" + str(df.isnull().sum()) + "\n\n")
    outfile.write("Special numeric values that stand for
    ↪  unknown/undefined if any:\n")
    for col in df.columns:
        if df[col].dtype in ['int16', 'int32', 'int64',
        ↪  'float16', 'float32', 'float64']:
            outfile.write("Column "+ col +" contains "+
            ↪  str(df[df[col] < 0].shape[0]) +" negative
            ↪  values.\n")

# Close the file
outfile.close()
```

B.2 Code Description

The provided code performs data exploration on the "tracheal_tube_insertion.csv" dataset to gain insights and summarize its key characteristics.

The analysis steps performed by the code can be summarized as follows:

1. Loading the Data: - The code reads the data from the "tracheal_tube_insertion.csv" file and stores it in a pandas DataFrame called "df".

2. Data Size: - The code calculates the number of rows and columns in the dataset using the "shape" function of the DataFrame, and writes the results to the output file.

3. Summary Statistics: - The code calculates summary statistics, including count, mean, standard deviation, minimum, quartiles, and maximum, for all numerical columns in the dataset using the "describe" function of the DataFrame, and writes the results to the output file.

4. Categorical Variables: - The code identifies categorical variables in the dataset by selecting columns with the "object" data type, and for each

categorical column, it determines the most common value (mode) using the "mode" function of the DataFrame, and writes the results to the output file.

5. Missing Values: - The code checks for missing or undefined values in each column of the dataset using the "isnull" function of the DataFrame, calculates the count of missing values per column, and writes the results to the output file. - Additionally, for each column with numeric data type, the code checks if there are any negative values, indicating special numeric values that stand for unknown or undefined, and writes the count of negative values per column to the output file.

The output file "data_exploration.txt" contains the following information:

- Data Size: Number of rows and columns in the dataset. - Summary Statistics: Descriptive statistics for all numerical columns in the dataset. - Categorical Variables: The most common value for each categorical column in the dataset. - Missing Values: The count of missing values per column, and if applicable, the count of negative values per numeric column.

This data exploration process helps in understanding the structure and characteristics of the dataset and provides initial insights into the distributions and quality of the data for further analysis.

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
Missing or undefined values for each column:
tube            0
sex             0
age_c           0
ht              0
wt              0
tube_depth_G    0
dtype: int64

Special numeric values that stand for unknown/undefined if any:
Column tube contains 0 negative values.
Column sex contains 0 negative values.
Column age_c contains 0 negative values.
Column ht contains 0 negative values.
Column wt contains 0 negative values.
Column tube_depth_G contains 0 negative values.
```

C Data Analysis

C.1 Code

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

```

# IMPORT
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNetCV
from scipy.stats import ttest_rel
import pickle

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

# DATASET PREPARATIONS
```

```
# No dataset preparations are needed.


# DESCRIPTIVE STATISTICS
# Table 0: Descriptive statistics of patient features and OTTD
desc_stats_df = df[['sex', 'age_c', 'ht', 'wt',
↪  'tube_depth_G']].agg(['mean', 'std'])
desc_stats_df.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, because the dataset does not
↪  contain any non-numeric variables.

# Splitting the dataset into train and test sets
features = df[['sex', 'age_c', 'ht', 'wt']]
target = df['tube_depth_G']
X_train, X_test, y_train, y_test = train_test_split(features,
↪  target, test_size=0.2, random_state=42)

# ANALYSIS

# Table 1: Mean squared residuals for Random Forest and
↪  Elastic Net
rf_model = RandomForestRegressor()
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)
rf_residuals = np.square(rf_predictions - y_test)

en_model = ElasticNetCV()
en_model.fit(X_train, y_train)
en_predictions = en_model.predict(X_test)
en_residuals = np.square(en_predictions - y_test)

# Revised df1 to include model names as the index
df1 = pd.DataFrame({
  'Model': ['Random Forest', 'Elastic Net'],
  'Mean Squared Residual': [rf_residuals.mean(),
  ↪  en_residuals.mean()]
}).set_index('Model')
df1.to_pickle('table_1.pkl')
```

```

# Table 2: Paired t-test results for the mean squared residuals of Random Forest and Elastic Net
ttest_res = ttest_rel(rf_residuals, en_residuals)

# Revised df2 to include comparison descriptor as the index
df2 = pd.DataFrame({
    'Model Comparison': ['RF vs EN Mean Squared Residuals'],
    't-statistic': [ttest_res.statistic],
    'p-value': [ttest_res.pvalue]
}).set_index('Model Comparison')

df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': len(df),
    'Training set size': len(X_train),
    'Testing set size': len(X_test)
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)

```

C.2 Code Description

The purpose of the provided code is to analyze the dataset containing information about pediatric patients who received post-operative mechanical ventilation. The dataset includes patient features such as sex, age, height, weight, and the optimal tracheal tube depth (OTTD) determined by chest X-ray.

The analysis steps performed in the code are as follows:

1. Loading the dataset: The code reads the "tracheal_tube_insertion.csv" file, which contains the clean dataset, into a pandas DataFrame.

2. Descriptive statistics: The code calculates the mean and standard deviation of the patient features (sex, age, height, weight) and the OTTD. These statistics are stored in the "table_0.pkl" file.

3. Preprocessing: Since the dataset only contains numeric variables, no preprocessing is required.

4. Splitting the dataset: The code splits the dataset into training and testing sets using the train_test_split function from scikit-learn. The features (sex, age, height, weight) are stored in the 'features' DataFrame, and the target variable (OTTD) is stored in the 'target' Series.

5. Model Training and Evaluation: - Random Forest: The code trains a Random Forest Regressor on the training data and then evaluates its performance on the testing data. The mean squared residuals between the predicted and actual OTTD values are calculated and stored. - Elastic Net: The code trains an Elastic Net Regressor on the training data and evaluates its performance on the testing data. Again, the mean squared residuals are calculated and stored.

6. Results: - Table 1: The code creates a DataFrame ('df1') that includes the mean squared residuals for both the Random Forest and Elastic Net models. The model names are used as the index, and the DataFrame is stored in the "table_1.pkl" file. - Table 2: The code performs a paired t-test to compare the mean squared residuals of the Random Forest and Elastic Net models. The t-statistic and p-value are stored in the 'df2' DataFrame, which is then saved in the "table_2.pkl" file.

7. Additional Results: The code saves additional results, such as the total number of observations, training set size, and testing set size, in the "additional_results.pkl" file. These results provide supplementary information about the dataset and the analysis.

Overall, the code performs descriptive statistics, splits the dataset, trains and evaluates machine learning models, and generates tables with the model evaluation results. The additional_results.pkl file contains extra information about the dataset and the analysis.

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

Model	Mean Squared Residual
Random Forest	1.549838
Elastic Net	1.238410

table_2.pkl

	t-statistic	p-value
Model Comparison		
RF vs EN Mean Squared Residuals	2.339968	0.02031

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Training set size': 775,
    'Testing set size': 194,
}
```

D LaTeX Table Design

D.1 Code

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

```

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

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

# PREPARATION FOR ALL TABLES

shared_mapping: AbbrToNameDef = {
    'sex': ('Sex', '0: Female, 1: Male'),
    'age_c': ('Age', 'Age in years, rounded to the nearest
    ↪  half-year'),
    'ht': ('Height', 'Height in centimeters (cm)'),
    'wt': ('Weight', 'Weight in kilograms (kg)'),
    'mean': ('Mean', None),
    'std': ('Standard Deviation', None),
}

```

```python
special_mapping: AbbrToNameDef = {
    'tube_depth_G': ('OTTD', 'Optimal Tracheal Tube Depth as determined by Chest X-ray in cm')
}

# TABLE 0

df0 = pd.read_pickle('table_0.pkl')

# RENAMING ROWS AND COLUMNS

mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df0, k)}
mapping |= {k: v for k, v in special_mapping.items() if is_str_in_df(df0, k)}

abbrs_to_names, legend = split_mapping(mapping)
df0 = df0.rename(columns=abbrs_to_names, index=abbrs_to_names)

# SAVE AS LATEX

to_latex_with_note(
    df0, 'table_0.tex',
    caption="Descriptive Statistics of Patient Features and OTTD",
    label='table:descriptive-statistics',
    note="This table provides the mean and standard deviation for each variable.",
    legend=legend
)


# TABLE 1

df1 = pd.read_pickle('table_1.pkl')

# SAVE AS LATEX

to_latex_with_note(
    df1, 'table_1.tex',
```

```
    caption="Mean Squared Residuals for Random Forest and Elastic Net",
    label='table:squared-residuals',
    note=None,
    legend=None
)


# TABLE 2

df2 = pd.read_pickle('table_2.pkl')

# FORMAT P-VALUE

df2['p-value'] = df2['p-value'].apply(format_p_value)

# SAVE AS LATEX

to_latex_with_note(
    df2, 'table_2.tex',
    caption="Paired T-Test Results for Mean Squared Residuals",
    label='table:t-test',
    note=None,
    legend=None
)


```

D.2 Provided Code

The code above is using the following provided functions:

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

```

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

    Returns:
    - None: Outputs LaTeX file.
    """

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

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

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

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

```

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive Statistics of Patient Features and OTTD}
\label{table:descriptive-statistics}
```

```
\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{Standard Deviation} & 0.499 & 1.44 & 19.1 & 4.77 & 1.77 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table provides the mean and standard deviation for each variable.
\item \textbf{Sex}: 0: Female, 1: Male
\item \textbf{Age}: Age in years, rounded to the nearest half-year
\item \textbf{Height}: Height in centimeters (cm)
\item \textbf{Weight}: Weight in kilograms (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{Mean Squared Residuals for Random Forest and Elastic Net}
\label{table:squared-residuals}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Mean Squared Residual \\
Model &  \\
\midrule
\textbf{Random Forest} & 1.55 \\
\textbf{Elastic Net} & 1.24 \\
```

```
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Paired T-Test Results for Mean Squared Residuals}
\label{table:t-test}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & t-statistic & p-value \\
Model Comparison &  &  \\
\midrule
\textbf{RF vs EN Mean Squared Residuals} & 2.34 & 0.0203 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```