Improving Prediction of Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 9, 2024

Abstract

Accurate determination of tracheal tube depth is crucial for pediatric patients requiring mechanical ventilation. However, the current methods, such as chest X-ray analysis and formula-based models, have limitations in accurately determining optimal tracheal tube depth. To address this gap, we present a dataset of 969 pediatric patients aged 0-7 years who underwent surgery and post-operative mechanical ventilation. This dataset includes optimal tracheal tube depth determined by chest X-ray and patient electronic health record features. We employ a Random Forest model, a machine learning regression algorithm, to predict optimal tracheal tube depth. The results demonstrate that our model significantly outperforms formula-based models, reducing mean squared error from 3.19 to 1.45. This improvement indicates a considerable advancement in predicting optimal tracheal tube depth, which can enhance patient safety and outcomes. We acknowledge the need for further validation and exploration of additional predictor variables to strengthen the proposed approach. Our findings have valuable implications for improving the accuracy and efficiency of tracheal tube placement in pediatric patients, ultimately reducing complications and enhancing patient care.

Results

We begin by investigating whether height and age, stratified by sex, play notable roles in determining optimal tracheal tube depth (OTTD), as shown in Table 1. The dataset consisted of 969 pediatric patients, with the mean height and age for female and male patients calculated. These two variables were considered due to their potential influence on tracheal tube size and placement in pediatric patient care. Females have an average height of 65.4

cm (standard deviation: 18.7 cm) and age of 0.732 years (standard deviation: 1.4 years), while males have a slightly greater average height and age, 66.5 cm (standard deviation: 19.4 cm) and 0.781 years (standard deviation: 1.47 years) respectively. It is therefore evident that noticeable variation exists within our dataset for these patient characteristics, which may influence optimal tracheal tube placement.

Table 1: Descriptive statistics of height and age stratified by sex

	Height (cm)		Age (years)	
	mean	std	mean	std
female	65.4	18.7	0.732	1.4
male	66.5	19.4	0.781	1.47

All values are means and standard deviations. Sex is denoted as 'female' and 'male'. Age is in years, rounded to half years. Height is measured in cm.
Age (years): Patient age, years, rounded to half years
Height (cm): Patient height, cm

Subsequently, we conducted a comparative analysis between the Random Forest model and a height-based formula model to determine the most accurate predictor of OTTD. The principal motivation behind this comparison is to investigate whether a machine learning-based model can outperform a traditional height-based formula in predicting OTTD. The findings from Table 2 indicate that the Random Forest model significantly outperformed the formula-based model, achieving a mean squared error (MSE) of 1.45, compared to the latter's MSE of 3.19. The paired t-test performed on the squared errors demonstrated a statistically significant difference in the prediction accuracy between the two models, with a t-statistic of -5.08 and a p-value smaller than 1×10^{-6}, underscoring the superior performance of the Random Forest model.

The Random Forest model's performance was further quantified by the R-squared value of 0.4418, indicating that approximately 44% of the variability in OTTD is accounted for by our model. Notwithstanding the need for additional model refinement, this R-squared value suggests that the Random Forest model constitutes a considerable improvement to the current predictive capacity for determining OTTD.

In conclusion, our analysis confirms that employing a machine learning model, such as the Random Forest model, offers improved accuracy in predicting optimal tracheal tube depth in comparison to traditional formula-based models. By significantly reducing the mean squared error from 3.19

Table 2: Comparison of prediction errors from Random Forest model and a formula-based model

Model	Mean Squared Error	T-statistic	P-value
Random Forest	1.45	-5.08	$<10^{-6}$
Height Formula	3.19	-	-

Mean Squared Error: Mean Squared Error between the predicted and actual values
T-statistic: T-test statistic from pair t-test between the squared errors of the two models
P-value: P-value from paired t-test

to 1.45, this method may improve patient safety by ensuring more accurate tracheal tube placement. Consequently, these findings may have far-reaching implications for enhancing outcomes in pediatric patient care.

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
4  # Load in the csv file
5  df = pd.read_csv('tracheal_tube_insertion.csv')
6
7  # Open a file to write the exploration data to
8  with open('data_exploration.txt', 'w') as f:
9
10     # Data Size
11     f.write('# Data Size\n')
12     f.write('Number of rows: ' + str(df.shape[0]) + '\n')
13     f.write('Number of columns: ' + str(df.shape[1]) + '\n\n')
14
15     # Summary Statistics
16     f.write('# Summary Statistics\n')
17     f.write(str(df.describe()) + '\n\n')
18
19     # Categorical Variables
20     f.write('# Categorical Variables (if any)\n')
21     for cols in df.columns:
22         if df[cols].dtype == 'object':
23             f.write(f"{cols}'s distribution:\n")
```

```
            f.write(str(df[cols].value_counts().head()) +
            ↪  '\n\n')

    # Missing Values
    f.write('# Missing Values\n')
    f.write(str(df.isnull().sum()) + '\n\n')

    # Checking for special values
    # Special values are checked after missing value analysis
    ↪  to
    # avoid double counting of missing and special values
    f.write('# Special Numeric Values\n')
    special_count = (df==9999).sum()  # Note, replace "9999"
    ↪  with the special numeric value informed in your data
    ↪  description if any.
    f.write(str(special_count[special_count > 0]) + '\n\n')

    # Add any customizable summary of your preference here

```

B.2 Code Description

The Data Exploration code provided performs an analysis on the dataset to gain insights into its size, summary statistics, categorical variables, missing values, and special numeric values. It outputs the findings to a text file called "data_exploration.txt".

Specifically, the code performs the following analysis steps:

1. Data Size: It retrieves the number of rows and columns in the dataset using the "shape" attribute of the DataFrame. The code writes these values to the "data_exploration.txt" file.

2. Summary Statistics: It calculates descriptive statistics for each numerical column in the dataset using the "describe" function. The summary statistics include count, mean, standard deviation, minimum, quartiles, and maximum. The code writes these statistics to the "data_exploration.txt" file.

3. Categorical Variables: It identifies any categorical variables in the dataset by checking the data type of each column. If a column is of type "object", it is considered categorical. For each categorical variable, the code determines its distribution by using the "value_counts" function and writes the top values to the "data_exploration.txt" file.

4. Missing Values: It identifies missing values in the dataset by using the "isnull" function, which returns a boolean mask representing missing values. This mask is then summed along each column to obtain the count of missing values. The code writes these counts to the "data_exploration.txt" file.

5. Special Numeric Values: After analyzing missing values, the code checks for any special numeric values that might exist in the dataset. In this case, it looks for a value of 9999, but this can be customized based on the information provided in the data description. The code counts the occurrence of this special value for each column using the "sum" function and writes the counts to the "data_exploration.txt" file.

Additionally, the code allows for customizations by providing a section where further summary statistics or analysis steps can be added according to the specific requirements of the research study.

Overall, the code conducts a comprehensive exploratory analysis of the dataset and generates a report that helps researchers understand the data's characteristics, including its size, summary statistics, categorical variables, missing values, and special numeric values.

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 (if any)
# Missing Values
```

```
tube            0
sex             0
age_c           0
ht              0
wt              0
tube_depth_G    0
dtype: int64

# 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  from sklearn.model_selection import train_test_split
6  from sklearn.ensemble import RandomForestRegressor
7  from sklearn.metrics import mean_squared_error, r2_score
8  from scipy import stats
9  import pickle
10
11 # LOAD DATA
12 df = pd.read_csv("tracheal_tube_insertion.csv")
13
14 # DATASET PREPARATIONS
15 # No dataset preparations are needed.
16
17 # DESCRIPTIVE STATISTICS
18 ## Table 0: "Descriptive statistics of height and age
   ↪  stratified by sex"
19 df0 = df.groupby("sex").agg(["mean", "std"])[["ht", "age_c"]]
20 df0.index = ['female', 'male']
```

```
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# The variables are already in a format suitable for regression.

# ANALYSIS

# ML Model: Random Forest
# Split data into features and target
X = df[["sex", "age_c","ht","wt"]]
y = df["tube_depth_G"]

# Split dataset into train and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

# Create RF regressor object and train on the data
RF = RandomForestRegressor(n_estimators=500, random_state=1)
RF.fit(X_train, y_train)

# Predict on the test data and save predictions for comparison
rf_predictions = RF.predict(X_test)

# Calculation of errors for ML model
rf_errors = (y_test - rf_predictions) ** 2

# Formula-Based Model: height [cm] / 10 + 5 cm
formula_predictions = X_test["ht"] / 10 + 5

# Calculation of errors for Formula model
formula_errors = (y_test - formula_predictions) ** 2

# Perform paired t-test between the squared errors
ttest_results = stats.ttest_rel(rf_errors, formula_errors)

# Create dataframe for results
df1 = pd.DataFrame({
    'Model': ['Random Forest', 'Height Formula'],
```

```
    'Mean Squared Error': [np.mean(rf_errors),
    ↪  np.mean(formula_errors)],
    'T-Test Statistic': [ttest_results.statistic, '-'],
    'P-value': [ttest_results.pvalue, '-'],
    })
df1.set_index("Model", inplace=True)

# Save the dataframe for the scientific table
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': df.shape[0],
 'R_squared of Random Forest model': r2_score(y_test,
 ↪  rf_predictions),
}

with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)

```

C.2 Code Description

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

The code starts by loading the dataset, which contains information such as patient sex, age, height, weight, and the determined OTTD as determined by chest X-ray.

After loading the dataset, the code performs descriptive statistics to calculate the mean and standard deviation of height and age, stratified by sex. These statistics are stored in a pickle file for later reference.

Next, the code prepares the dataset for analysis, as no further preprocessing is required.

The analysis consists of two models: a Random Forest (RF) model and a formula-based model. The RF model is a machine learning model that uses the features (sex, age, height, and weight) to predict the OTTD. The dataset is split into training and test sets, and the RF model is trained on the training set. The trained model is then used to predict the OTTD for

the test set.

The formula-based model predicts the OTTD based on the patient's height. The formula used is height (in cm) divided by 10, plus 5.

After making the predictions, the code calculates the mean squared error (MSE) for both models. The MSE measures the average squared difference between the predicted and actual OTTD values.

To compare the performance of the two models, a paired t-test is conducted on the squared errors of the RF model and the formula-based model.

The code saves the results of the analysis in two pickle files. The first file contains a dataframe with the MSE, t-test statistic, and p-value for both models. The second file contains additional results, such as the total number of observations and the R-squared value of the RF model.

Overall, the code performs a data analysis to compare the performance of a machine learning model (RF) and a formula-based model in predicting the optimal tracheal tube depth for pediatric patients receiving mechanical ventilation.

C.3 Code Output

table_0.pkl

	ht		age_c	
	mean	std	mean	std
female	65.400447	18.701462	0.731544	1.402500
male	66.514368	19.403722	0.780651	1.472808

table_1.pkl

Model	Mean Squared Error	T-Test Statistic	P-value
Random Forest	1.447506	-5.078176	8.946e-07
Height Formula	3.186912	-	-

additional_results.pkl

```
{
    'Total number of observations': 969,
    'R_squared of Random Forest model': 0.4418             ,
}
```

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, AbbrToNameDef

# PREPARATION FOR ALL TABLES

shared_mapping: AbbrToNameDef = {
    'sex': ('Sex', 'Sex of patient (0=female, 1=male)'),
    'age_c': ('Age (years)', 'Patient age, years, rounded to half years'),
    'ht': ('Height (cm)', 'Patient height, cm'),
    'Mean Squared Error': ('Mean Squared Error', 'Mean Squared Error between the predicted and actual values')
}

# TABLE 0:
df_0 = pd.read_pickle('table_0.pkl')

# RENAME ROWS AND COLUMNS
mapping_0 = {k: v for k, v in shared_mapping.items() if is_str_in_df(df_0, k)}
abbrs_to_names_0, legend_0 = split_mapping(mapping_0)
df_0 = df_0.rename(columns=abbrs_to_names_0, index=abbrs_to_names_0)

# Save as latex
to_latex_with_note(
    df_0, 'table_0.tex',
    caption="Descriptive statistics of height and age stratified by sex",
    label='table:table_0',
```

```
    note="All values are means and standard deviations. Sex is
    ↪  denoted as 'female' and 'male'. Age is in years,
    ↪  rounded to half years. Height is measured in cm.",
    legend=legend_0
)

# TABLE 1:
df_1 = pd.read_pickle('table_1.pkl')

# FORMAT VALUES
df_1['P-value'] = df_1['P-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping_1 = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df_1, k)}
mapping_1 |= {
    'T-Test Statistic': ('T-statistic', 'T-test statistic from
    ↪  pair t-test between the squared errors of the two
    ↪  models'),
    'P-value': ('P-value', 'P-value from paired t-test'),
}
abbrs_to_names_1, legend_1 = split_mapping(mapping_1)
df_1 = df_1.rename(columns=abbrs_to_names_1)

# Save as latex
to_latex_with_note(
    df_1, 'table_1.tex',
    caption="Comparison of prediction errors from Random
    ↪  Forest model and a formula-based model",
    label='table:table_1',
    note=None,
    legend=legend_1
)
```

D.2 Provided Code

The code above is using the following provided functions:

```python
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 height and age stratified by sex}
\label{table:table_0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrr}
\toprule
 & \multicolumn{2}{r}{Height (cm)} & \multicolumn{2}{r}{Age (years)} \\
 & mean & std & mean & std \\
\midrule
\textbf{female} & 65.4 & 18.7 & 0.732 & 1.4 \\
\textbf{male} & 66.5 & 19.4 & 0.781 & 1.47 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item All values are means and standard deviations. Sex is denoted as 'female'
    and 'male'. Age is in years, rounded to half years. Height is measured in cm.
\item \textbf{Age (years)}: Patient age, years, rounded to half years
\item \textbf{Height (cm)}: Patient height, cm
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Comparison of prediction errors from Random Forest model and a formula-
    based model}
\label{table:table_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrll}
\toprule
```

```
 & Mean Squared Error & T-statistic & P-value \\
Model &  &  &  \\
\midrule
\textbf{Random Forest} & 1.45 & -5.08 & $<$1e-06 \\
\textbf{Height Formula} & 3.19 & - & - \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Mean Squared Error}: Mean Squared Error between the predicted and
    actual values
\item \textbf{T-statistic}: T-test statistic from pair t-test between the
    squared errors of the two models
\item \textbf{P-value}: P-value from paired t-test
\end{tablenotes}
\end{threeparttable}
\end{table}
```