Optimal Tracheal Tube Depth Prediction in Pediatric Patients using Machine Learning

Data to Paper

January 7, 2024

Abstract

Accurate placement of the tracheal tube during mechanical ventilation is crucial for pediatric patients, but determining the optimal tracheal tube depth (OTTD) is challenging. Existing methods, such as chest X-rays and formula-based models, have limitations. In this study, we develop a machine learning approach to predict OTTD in pediatric patients aged 0-7 years. Our random forest regressor model is trained on a dataset of 969 patients, using features extracted from electronic health records. Compared to the formula-based model, our machine learning model achieves higher predictive accuracy, resulting in a mean predicted OTTD of 10.2 cm. Statistical analysis confirms the significant difference in accuracy ($p < 0.001$). Our model offers a non-invasive and efficient alternative to chest X-rays, potentially improving tracheal tube placement and patient safety. However, further validation through clinical trials is needed to assess the model's impact on patient outcomes and facilitate its adoption in pediatric healthcare settings.

Results

First, to understand the distribution of Optimal Tracheal Tube Depth (OTTD), we conducted a descriptive analysis stratified by the patient's sex (see Table 1). The average OTTD in male patients was 10.3 cm (sd=1.86), slightly larger than that of female patients, who had an average OTTD of 10.1 cm (sd=1.65). The difference in averages was negligible, indicating little variation in OTTD by sex in pediatric patients aged 0 to 7 years old.

Next, we wished to explore the performance of a machine learning model compared to the formula-based model in predicting OTTD. As explained in

Table 1: Descriptive statistics of Optimal Tracheal Tube Depth (OTTD) stratified by sex

	mean	std
female	10.1	1.65
male	10.3	1.86

Table 2, both models were trained on the training set and tested on a separate validation set. The results were promising for the machine learning model. The mean predicted OTTD by the machine learning model was 10.2 cm (sd=1.36), closer to the mean OTTD than the formula-based model's mean prediction of 11.6 cm (sd=1.98). In addition, the mean squared residual of the predicted OTTD by the machine learning model was 1.4 compared to 3.42 of the formula-based model, indicating a higher predictive accuracy of the machine learning model.

Table 2: Predictive performance of the machine-learning model vs the formula-based model

	Predicted ML	Predicted FBM	Residuals ML	Residuals FBM
mean	10.2	11.6	1.4	3.42
std	1.36	1.98	2.89	4.45

Residuals ML: Squared residuals of the machine learning model
Residuals FBM: Squared residuals of the formula-based model
Predicted ML: Predicted OTTD by ML model in cm
Predicted FBM: Predicted OTTD by the formula-based model in cm

Finally, we conducted a paired t-test to assess whether the difference in predictive performance between the machine learning and the formula-based model was statistically significant. The resulting T statistic was -6.227 with a very low p-value ($p < 2.901\ 10^{-9}$). The negative T statistic implies that our machine learning model has significantly smaller residuals, and hence higher accuracy, than the formula-based model.

Taken together, these results suggest that the accuracy of our machine learning model, in predicting OTTD, is significantly higher than the traditionally used formula-based 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
3  # Import Required Libraries
4  import pandas as pd
5
6  # Load Data
7  data = pd.read_csv('tracheal_tube_insertion.csv')
8
9  # Open File for Writing
10 with open('data_exploration.txt', 'w') as f:
11
12     # Data Size
13     f.write('# Data Size\n')
14     rows, columns = data.shape
15     f.write('Number of rows: {}\n'.format(rows))
16     f.write('Number of columns: {}\n'.format(columns))
17
18     # Summary Statistics
19     f.write('\n# Summary Statistics\n')
20     summary = data.describe()
21     f.write('{}\n'.format(summary))
22
23     # Categorical Variables
24     f.write('\n# Categorical Variables\n')
```

```
    f.write("Categorical Variable 'sex', most common value:
    ↪  '{}'\n".format(data['sex'].mode()[0]))

    # Missing Values
    f.write('\n# Missing Values\n')
    missing_values = data.isnull().sum()
    f.write('{}\n'.format(missing_values))
```

B.2 Code Description

The code performs data exploration on the provided dataset "tracheal_tube_insertion.csv".

First, the code loads the dataset into a pandas DataFrame.

Next, it performs the following analysis steps:

1. Data Size: It calculates the number of rows and columns in the dataset and writes this information in the "data_exploration.txt" file.

2. Summary Statistics: It computes descriptive statistics for each numerical column in the dataset using the describe() function in pandas. The summary statistics include count, mean, standard deviation, minimum, 25th percentile, 50th percentile (median), and maximum. These statistics provide insights into the distribution and range of values for each numerical feature.

3. Categorical Variables: It identifies the most common value in the 'sex' column, which represents the patient's sex. This information is written in the "data_exploration.txt" file.

4. Missing Values: It calculates the number of missing values for each column in the dataset using the isnull().sum() function in pandas. Missing values can impact the quality of the data and may require imputation or other handling strategies. The count of missing values for each column is written in the "data_exploration.txt" file.

The "data_exploration.txt" file serves as a documentation of the data exploration process and provides an overview of important details about the dataset, including its size, summary statistics, most common value of categorical variables, and the count of missing values. This information can help researchers gain initial insights into the dataset and make informed decisions for subsequent 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
Categorical Variable 'sex', most common value: '1'

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

```python
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_rel
import pickle

# LOAD DATA
data = pd.read_csv("tracheal_tube_insertion.csv")

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of Optimal Tracheal Tube Depth (OTTD) stratified by sex"
df0 = data.groupby('sex')['tube_depth_G'].agg(['mean', 'std'])
df0.index = ['female', 'male']  # Changing index to meaningful labels
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, because all our necessary features are numerical.
# Creating the formula-based column
data['predicted_formula'] = data['ht'] / 10 + 5

# ANALYSIS
## Table 1: "Predictive performance of the machine-learning model vs the formula-based model"
# Splitting the data
train, test = train_test_split(data, test_size=0.2, random_state=42)

# Define features and target
features = ['sex', 'age_c', 'ht', 'wt']
target = 'tube_depth_G'

# Training the model
rf_reg = RandomForestRegressor(max_depth=5, random_state=0)
```

```python
rf_reg.fit(train[features], train[target])

# Making predictions
test['predicted_ML'] = rf_reg.predict(test[features])

# Calculate squared residuals
test['residuals_ML'] = (test[target] - test['predicted_ML']) ** 2
test['residuals_formula'] = (test[target] - test['predicted_formula']) ** 2

# Table 1
table_1_cols = ['predicted_ML', 'predicted_formula', 'residuals_ML', 'residuals_formula']
df1 = test[table_1_cols].agg(['mean', 'std'])

df1.to_pickle('table_1.pkl')

## Hypothesis test
ttest_result = ttest_rel(test['residuals_ML'], test['residuals_formula'])

# Raise an error if our ML model does not have a significantly better predictive performance than the formula-based model
assert ttest_result.pvalue < 0.05, "The machine-learning model does not have a significantly better predictive performance than the formula-based model"

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(data),
 'T statistic': ttest_result.statistic,
 'P value': ttest_result.pvalue,
}
with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)

```

C.2 Code Description

The code performs an analysis on a dataset of pediatric patients who received post-operative mechanical ventilation. The dataset includes features extracted from patient electronic health records, such as patient sex, age, height, weight, and the optimal tracheal tube depth (OTTD) determined by chest X-ray.

The code first computes descriptive statistics of the OTTD stratified by sex and saves the results in a file called "table_0.pkl". This provides insights into the average and standard deviation of the OTTD for female and male patients.

Next, the code prepares the dataset for analysis by creating an additional column called "predicted_formula", which represents the OTTD calculated using a formula-based model. This column is based on the patient's height using a simple formula.

The code then splits the dataset into training and testing sets, with 80% of the data used for training a machine-learning model and 20% for testing the model's predictive performance. The machine-learning model used is a Random Forest Regressor.

After training the model, it makes predictions on the testing set. The predicted OTTD values are stored in the "predicted_ML" column. The code also calculates the squared residuals for both the machine-learning model (stored in the "residuals_ML" column) and the formula-based model (stored in the "residuals_formula" column).

To compare the predictive performance of the machine-learning model with the formula-based model, the code computes the mean and standard deviation of the predicted OTTD values and squared residuals for both models. These results are saved in a file called "table_1.pkl".

Finally, the code performs a hypothesis test (t-test) to determine if the machine-learning model has a significantly better predictive performance than the formula-based model. If the p-value of the test is less than 0.05, indicating statistical significance, the code saves additional results including the total number of observations, the t-statistic, and the p-value in a file called "additional_results.pkl".

The "additional_results.pkl" file provides information on the statistical comparison between the machine-learning model and the formula-based model, indicating whether the machine-learning model outperforms the formula-based model in predicting the optimal tracheal tube depth for pediatric patients on mechanical ventilation.

C.3 Code Output

table_0.pkl

	mean	std
female	10.062416	1.645478
male	10.298276	1.857778

table_1.pkl

	predicted_ML	predicted_formula	residuals_ML	residuals_formula
mean	10.179423	11.577835	1.395336	3.418890
std	1.358826	1.978500	2.893791	4.448652

additional_results.pkl

```
{
    'Total number of observations': 969,
    'T statistic': -6.227          ,
    'P value': 2.901e-09,
}
```

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 the patient (0: Female, 1: Male)'),
 'age_c': ('Age', 'Age of the patient in years'),
 'ht': ('Height', 'Height of the patient in cm'),
 'wt': ('Weight', 'Weight of the patient in kg'),
 'tube_depth_G': ('OTTD', 'Optimal Tracheal Tube Depth
 ↪  determined by chest X-ray in cm'),
```

```
 'residuals_ML': ('Residuals ML', 'Squared residuals of the machine learning model'),
 'residuals_formula': ('Residuals FBM', 'Squared residuals of the formula-based model'),
 'predicted_ML': ('Predicted ML', 'Predicted OTTD by ML model in cm'),
 'predicted_formula': ('Predicted FBM', 'Predicted OTTD by the formula-based model in cm'),
}

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as Latex:
to_latex_with_note(
 df, 'table_0.tex',
 caption="Descriptive statistics of Optimal Tracheal Tube Depth (OTTD) stratified by sex",
 label='table:OTTD_Stratified_by_Sex',
 legend=legend)

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as Latex:
to_latex_with_note(
 df, 'table_1.tex',
```

```
 caption="Predictive performance of the machine-learning model vs the formula-based model",
 label='table:Model_Comparison',
 legend=legend
)

```

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 Optimal Tracheal Tube Depth (OTTD) stratified
    by sex}
\label{table:OTTD_Stratified_by_Sex}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & mean & std \\
\midrule
\textbf{female} & 10.1 & 1.65 \\
\textbf{male} & 10.3 & 1.86 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Predictive performance of the machine-learning model vs the formula-
```

```
    based model}
\label{table:Model_Comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrr}
\toprule
 & Predicted ML & Predicted FBM & Residuals ML & Residuals FBM \\
\midrule
\textbf{mean} & 10.2 & 11.6 & 1.4 & 3.42 \\
\textbf{std} & 1.36 & 1.98 & 2.89 & 4.45 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Residuals ML}: Squared residuals of the machine learning model
\item \textbf{Residuals FBM}: Squared residuals of the formula-based model
\item \textbf{Predicted ML}: Predicted OTTD by ML model in cm
\item \textbf{Predicted FBM}: Predicted OTTD by the formula-based model in cm
\end{tablenotes}
\end{threeparttable}
\end{table}
```