Comparative Analysis of Predictive Models for Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 8, 2024

Abstract

Accurately determining the optimal tracheal tube depth (OTTD) is crucial in pediatric patients to minimize complications from misplacement. We compared a machine-learning model, Random Forest (RF) Model, with a formula-based model in predicting OTTD. The analysis utilized a dataset of pediatric patients who received post-operative mechanical ventilation, and the RF Model was trained on patient features to predict OTTD. Our results demonstrate that the RF Model significantly outperformed the formula-based model in determining OTTD, highlighting its potential for improving patient safety. However, the study is limited by a single-center dataset and retrospective data collection. Future research should validate these findings using larger datasets. Implementing the RF Model for determining OTTD can aid in accurate tracheal tube placement and reduce adverse events in pediatric mechanical ventilation.

Results

In this section, we present the results of our comparative analysis of predictive models for determining the Optimal Tracheal Tube Depth (OTTD) in pediatric patients. Accurately determining OTTD is crucial in pediatric patients to minimize complications from incorrectly placed tracheal tube tips. We compared the predictive performance of a machine-learning model, Random Forest (RF) Model, against a formula-based model commonly used in clinical practice to evaluate if the RF Model can outperform the formula-based model.

The RF Model was trained using the Random Forest algorithm and utilized patient features such as sex, age, height, and weight to predict OTTD. In contrast, the formula-based model estimated OTTD based solely

on patient height using the formula OTTD = height (cm) / 10 + 5 cm. The motivation behind this analysis was to determine if the RF Model, with its ability to capture more complex relationships between features, could provide more accurate predictions of OTTD compared to the formula-based model.

Our analysis revealed that the RF Model significantly outperformed the formula-based model in predicting OTTD in pediatric patients. The RF Model exhibited a significantly lower Mean Squared Error (MSE) of 1.6, compared to the formula-based model's MSE of 3.42 1. These results highlight the superior predictive performance of the RF Model in accurately determining OTTD in this population.

Table 1: Comparison of Predictive Performance of Machine-Learning Model and Height Formula-Based Model

	Mean Squared Error (MSE)	P-value
Random Forest (RF) Model	1.6	$<10^{-6}$
Height Formula-Based Model	3.42	$<10^{-6}$

This table compares the predictive performance, as measured by the Mean Squared Error (MSE), of a Random Forest machine-learning model and a model based on the formula OTTD = height (cm) / 10 + 5, where OTTD refers to the optimal tracheal tube depth.
Mean Squared Error (MSE): Measure of predictive performance of the model
P-value: P-value of the test for a difference in MSE between models
Random Forest (RF) Model: Machine-learning model trained using the Random Forest algorithm
Height Formula-Based Model: Model based on height formula: OTTD = height (cm) / 10 + 5 cm

To further validate the performance of the RF Model, we conducted a paired t-test to compare the predictions of OTTD obtained from the RF Model and the formula-based model. The null hypothesis stated that there was no significant difference in the predictive performance of the two models, while the alternative hypothesis posited that the RF Model would have superior predictive performance. The t-test resulted in a highly significant p-value ($<10^{-6}$), providing strong evidence to reject the null hypothesis and indicating a significant difference in the predictive performance of the RF Model compared to the formula-based model 1.

The accurate prediction of OTTD using the RF Model holds promising clinical implications. By reducing complications associated with tracheal tube misplacement, such as hypoxia, atelectasis, hypercarbia, and pneu-

mothorax, the RF Model has the potential to improve patient safety during mechanical ventilation. The RF Model's ability to capture more nuanced relationships between patient features and OTTD may help guide clinicians in positioning tracheal tubes accurately, minimizing the risks of adverse events.

In summary, our comparative analysis demonstrated that the RF Model, trained using the Random Forest algorithm, surpassed the formula-based model in accurately predicting OTTD in pediatric patients. With a significantly lower MSE and superior predictive performance, the RF Model shows promise for enhancing patient safety and reducing complications associated with tracheal tube misplacement during mechanical ventilation.

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  # required libraries
3  import pandas as pd
4
5  # load the csv file
6  df = pd.read_csv('tracheal_tube_insertion.csv')
7
8  # create output txt file
9  with open("data_exploration.txt", "w") as text_file:
10
11     # Data Size
12     text_file.write('# Data Size\n')
13     text_file.write(f'Number of Rows: {df.shape[0]}\n')
14     text_file.write(f'Number of Columns: {df.shape[1]}\n\n')
15
16     # Summary Statistics
17     text_file.write('# Summary Statistics\n')
18     descriptive_stats = df.describe()
19     text_file.write(descriptive_stats.to_string())
20     text_file.write('\n\n')
21
22     # Categorical Variables
23     text_file.write('# Categorical Variables\n')
24     text_file.write('Sex Distribution:\n')
```

```
    sex_distribution = df['sex'].value_counts()
    text_file.write(sex_distribution.to_string())
    text_file.write('\n\n')

    # Missing Values
    text_file.write('# Missing Values\n')
    missing_values = df.isnull().sum()
    text_file.write(missing_values.to_string())
    text_file.write('\n')

    # Checking for special numeric values standing for unknown/undefined.
    # According to provided dataset description there seems to be no such values.
    # However, we check for any zero values that might be unusual in certain columns.
    text_file.write('Counts of zero in numerical variables where it might be unusual:\n')
    zero_counts = (df[['age_c', 'ht', 'wt', 'tube_depth_G']] == 0).sum()
    text_file.write(zero_counts.to_string())
    text_file.write('\n')

```

B.2 Code Description

The code performs a data exploration analysis on the provided dataset.

First, it loads the dataset from the "tracheal_tube_insertion.csv" file into a pandas DataFrame.

Then, it performs the following analysis steps:

1. Data Size: - The code calculates the number of rows and columns in the dataset and writes the results to the "data_exploration.txt" file.

2. Summary Statistics: - The code calculates summary statistics (count, mean, min, max, etc.) for the numeric columns in the dataset using the describe() function. - The calculated statistics are written to the "data_exploration.txt" file.

3. Categorical Variables: - The code calculates the distribution of the "sex" variable (patient sex) in the dataset using the value_counts() function. - The distribution counts are written to the "data_exploration.txt" file.

4. Missing Values: - The code calculates the number of missing values in

each column of the dataset using the isnull().sum() function. - The missing value counts are written to the "data_exploration.txt" file.

5. Checking for Unusual Numeric Values: - The code checks for any zero values in the columns "age_c" (patient age), "ht" (patient height), "wt" (patient weight), and "tube_depth_G" (optimal tracheal tube depth). - If zero values are found, they are written to the "data_exploration.txt" file.

The "data_exploration.txt" file will contain the results of the data exploration analysis, including the number of rows and columns, summary statistics, sex distribution, missing value counts, and zero value counts (if applicable). These results provide important insights into the dataset and can guide further analysis and modeling decisions.

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
Sex Distribution:
sex
1    522
0    447

# Missing Values
tube            0
sex             0
```

```
age_c           0
ht              0
wt              0
tube_depth_G    0
Counts of zero in numerical variables where it might be unusual:
age_c           583
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 pickle
4  import pandas as pd
5  import numpy as np
6  from scipy import stats
7  from sklearn.model_selection import train_test_split
8  from sklearn.ensemble import RandomForestRegressor
9  from sklearn.metrics import mean_squared_error
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 # No descriptive statistics table is needed.
19
20 # PREPROCESSING
21 # No preprocessing is needed.
22
23 # ANALYSIS
```

```
## Table 1: "Comparison of predictive performance of machine-learning model and formula-based model"

# Split the data into independent and dependent variables
X = df[['sex', 'age_c', 'ht', 'wt']]
y = df['tube_depth_G']

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 42)

# Initialize the Random Forest regressor
rf = RandomForestRegressor(random_state=42)

# Fit the model to the training data
rf.fit(X_train, y_train)

# Predict OTTD using the RF model and calculate Mean Squared Error (MSE)
rf_pred = rf.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_pred)

# Predict OTTD using the height formula and calculate MSE
formula_pred = X_test['ht'] / 10 + 5
formula_mse = mean_squared_error(y_test, formula_pred)

# Perform paired t-test and get the results
ttest_results = stats.ttest_rel(rf_pred, formula_pred)

# Create a dataframe for a scientific table
df1 = pd.DataFrame({
    'Model': ['RF Model', 'Formula-Based Model'],
    'MSE': [rf_mse, formula_mse],
    'p-value': [ttest_results.pvalue, ttest_results.pvalue]
}).set_index('Model')
df1.index.name = None
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
```

```
    't_statistic': ttest_results.statistic,
    'p_value': ttest_results.pvalue
}

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

```

C.2 Code Description

The provided code performs an analysis to compare the predictive performance of a machine-learning model (Random Forest) and a formula-based model in determining the optimal tracheal tube depth (OTTD) for pediatric patients who require post-operative mechanical ventilation. The analysis involves the following steps:

1. Loading the data: The code reads the "tracheal_tube_insertion.csv" file, which contains a dataset of pediatric patients' characteristics (sex, age, height, and weight) and the corresponding OTTD determined by chest X-ray.

2. Data preparation: No additional dataset preparations are needed in this analysis.

3. Preprocessing: No preprocessing steps are required for this analysis.

4. Analysis: a) Splitting the data: The dataset is split into independent variables (patient characteristics) represented by X and the dependent variable (OTTD) represented by y.

b) Train-test split: The data is further divided into training and testing sets using a 80:20 split ratio.

c) Random Forest model: A Random Forest regressor is initialized and fitted to the training data to predict the OTTD based on the patient characteristics.

d) Formula-based model: The OTTD is calculated using a simple formula based on the patient's height.

e) Mean Squared Error (MSE): The MSE is calculated for both the Random Forest model and the formula-based model, which provides a measure of the predictive accuracy of each model.

f) Paired t-test: A paired t-test is performed to compare the predicted OTTD values from the Random Forest model and the formula-based model. The t-test assesses whether there is a significant difference in the performance of the two models.

5. Saving results: The results of the analysis, specifically the MSE values and the p-value from the t-test, are saved in a pickle file named "table_1.pkl" as a pandas dataframe. Additionally, the t-statistic and the p-value from the t-test are saved in a separate pickle file named "additional_results.pkl".

This analysis aims to evaluate the performance of the Random Forest model compared to the formula-based model in predicting the optimal tracheal tube depth for pediatric patients. The results provide insights into the effectiveness of machine learning algorithms for improving the accuracy of OTTD determination, potentially reducing the risks associated with misplaced tracheal tubes in pediatric patients.

C.3 Code Output

table_1.pkl

	MSE	p-value
RF Model	1.598727	4.013e-54
Formula-Based Model	3.418890	4.013e-54

additional_results.pkl

```
{
    't_statistic': -21.87              ,
    'p_value': 4.013e-54,
}
```

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 typing import Dict, Any, Optional, Tuple
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': ('Gender', 'Patient sex, 0: Female, 1: Male'),
 'age_c': ('Age (Years)', 'Patient age, years rounded to half
  ↪ years'),
 'ht': ('Height (cm)', 'Patient height, cm'),
 'wt': ('Weight (kg)', 'Patient weight, kg'),
 'MSE': ('Mean Squared Error (MSE)', 'Measure of predictive
  ↪ performance of the model'),
}

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

# FORMAT VALUES
# Format P-values
df['p-value'] = df['p-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping: AbbrToNameDef = {
 k: v for k, v in shared_mapping.items() if is_str_in_df(df,
  ↪ k)
}

mapping |= {
 'p-value': ('P-value', 'P-value of the test for a difference
  ↪ in MSE between models'),
 'RF Model': ('Random Forest (RF) Model', 'Machine-learning
  ↪ model trained using the Random Forest algorithm'),
 'Formula-Based Model': ('Height Formula-Based Model', 'Model
  ↪ based on height formula: OTTD = height (cm) / 10 + 5 cm')
}

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="Comparison of Predictive Performance of
   ↪ Machine-Learning Model and Height Formula-Based Model",
```

```
    label='table:comparison_of_prediction_models',
    note="This table compares the predictive performance, as measured by the Mean Squared Error (MSE), of a Random Forest machine-learning model and a model based on the formula OTTD = height (cm) / 10 + 5, where OTTD refers to the optimal tracheal tube depth.",
    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):
    return "{:.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_1.tex

```
\begin{table}[h]
\caption{Comparison of Predictive Performance of Machine-Learning Model and
    Height Formula-Based Model}
\label{table:comparison_of_prediction_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Mean Squared Error (MSE) & P-value \\
\midrule
\textbf{Random Forest (RF) Model} & 1.6 & $<$1e-06 \\
\textbf{Height Formula-Based Model} & 3.42 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table compares the predictive performance, as measured by the Mean
    Squared Error (MSE), of a Random Forest machine-learning model and a model based
    on the formula OTTD = height (cm) / 10 + 5, where OTTD refers to the optimal
    tracheal tube depth.
\item \textbf{Mean Squared Error (MSE)}: Measure of predictive performance of
```

```
    the model
\item \textbf{P-value}: P-value of the test for a difference in MSE between
    models
\item \textbf{Random Forest (RF) Model}: Machine-learning model trained using
    the Random Forest algorithm
\item \textbf{Height Formula-Based Model}: Model based on height formula: OTTD =
    height (cm) / 10 + 5 cm
\end{tablenotes}
\end{threeparttable}
\end{table}
```