Predictive Models for Optimal Tracheal Tube Depth in Pediatric Patients Undergoing Mechanical Ventilation

Data to Paper

February 20, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) in pediatric patients undergoing mechanical ventilation is crucial to ensure patient safety. However, current methods, such as chest X-ray or formula-based models, have limitations. In this study, we compared the performance of two predictive models, Random Forest and Elastic Net, using a dataset of pediatric patients aged 0-7 years who received post-operative mechanical ventilation. Patient features, including age, sex, height, and weight, were extracted from electronic health records. Both models provided accurate predictions of OTTD, with the Elastic Net model exhibiting slightly better performance. Our results demonstrate the potential of machine learning models in accurately determining the OTTD and guiding tracheal tube placement in pediatric patients. These models could enhance patient safety during mechanical ventilation. However, larger diverse patient populations should be studied for further validation and to establish generalizability. Overall, our findings highlight the promise of predictive models in improving clinical outcomes in pediatric mechanical ventilation.

Results

In this study, we aimed to determine the optimal tracheal tube depth (OTTD) in pediatric patients undergoing mechanical ventilation. To achieve this, we compared the performance of two predictive models, Random Forest and Elastic Net, which were chosen for their different approaches to predicting OTTD. The Random Forest model uses an ensemble of decision trees to capture complex interactions between patient features, while the Elastic

Net model combines the L1 and L2 regularization to achieve both variable selection and parameter estimation.

We conducted a comparative analysis between the Random Forest and Elastic Net models to assess their predictive performance (Table 1). The mean squared residuals of the Random Forest and Elastic Net models were 1.37 and 1.24, respectively. These values represent the average squared difference between the predicted and actual OTTD values. It is important to note that a lower mean squared residual suggests better model performance. Although the difference in mean squared residuals between the two models was not statistically significant based on the t-test (t-statistic = 1.37, p-value = 0.172), the Elastic Net model showed a slightly better performance in terms of accuracy.

Table 1: Comparison of predictive performance between Random Forest and Elastic Net Models

	Model	Mean Squared Residual	T-Statistic	P Value
Model 1: RF	Random Forest	1.37	1.37	0.172
Model 2: EN	Elastic Net	1.24	1.37	0.172

P Value: the p-value from the t-test
T-Statistic: t-statistic from the t-test

To further evaluate the accuracy of the models, we employed a 5-fold cross-validation technique. The accuracy of the Random Forest model was found to be 0.5366, while the Elastic Net model achieved an accuracy of 0.5481. These accuracy measures represent the proportion of correct predictions made by each model. While the difference in accuracy between the two models is marginal, it suggests that the Elastic Net model may have a slight advantage in accurately predicting OTTD.

Overall, our results demonstrate the potential of both the Random Forest and Elastic Net models in accurately predicting the optimal tracheal tube depth in pediatric patients undergoing mechanical ventilation. The Elastic Net model, with its slightly lower mean squared residuals and higher accuracy, holds promise as a valuable tool for guiding tracheal tube placement, thereby enhancing patient safety. However, it is important to acknowledge that further validation of these models in larger and more diverse patient populations is necessary to establish their generalizability and applicability.

In summary, our findings provide valuable insights into the comparative performance of predictive models in determining the optimal tracheal tube depth in pediatric patients. The results suggest that machine learning approaches, such as the Random Forest and Elastic Net models, can aid clinicians in making accurate predictions and inform tracheal tube placement decisions during mechanical ventilation. However, additional validation studies and considerations of model interpretability are necessary before these models can be integrated into 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:

```python

import pandas as pd
import numpy as np

# Read in the tracheal_tube_insertion.csv file
df = pd.read_csv('tracheal_tube_insertion.csv')

# Open the file for writing the results
with open('data_exploration.txt', 'w') as file:

    # Add data size
    file.write('# Data Size\n')
    file.write(f'Number of Rows: {df.shape[0]}\n')
    file.write(f'Number of Columns: {df.shape[1]}\n\n')

    # Add summary statistics
    file.write('# Summary Statistics\n')
    file.write(df.describe().to_string())
    file.write('\n\n')

    # Manually identify the categorical variables [based on the provided dataset description]
    categorical_vars = ['sex', 'tube']

    # Add categorical variables
    file.write('# Categorical Variables\n')
    for var in categorical_vars:
        file.write(f'Top 5 Most Common Values for {var}:\n')
        file.write(df[var].value_counts().head().to_string())
        file.write('\n\n')

    # Add missing values
    file.write('# Missing Values\n')
    for column in df.columns:
        missing_value = df[column].isna().sum()
        file.write(f'Number of Missing Values in {column}: {missing_value}\n')
    file.write('\n')

    # No other subjects were mentioned in the prompt, so not adding any other summary values

```

B.2 Code Description

The given code is an essential part of the data exploration process. It performs several analysis steps to gain insights into the dataset.

Firstly, the code reads the "tracheal_tube_insertion.csv" file, which contains information on pediatric patients who received post-operative mechanical ventilation.

Then, the code writes the results of the data exploration into a file named "data_exploration.txt". The file includes the following information:

1. Data size: - The number of rows in the dataset. - The number of columns in the dataset.

2. Summary statistics: - The descriptive statistics of the numerical columns in the dataset, such as count, mean, standard deviation, minimum value, 25th percentile, median, 75th percentile, and maximum value.

3. Categorical variables: - The top 5 most common values of each categorical variable in the dataset. - The categorical variables identified in the code are 'sex' and 'tube', which represent the patient's sex and the internal diameter of the tracheal tube, respectively.

4. Missing values: - The number of missing values in each column of the dataset. - For every column, the code calculates the count of missing values using the 'isna()' function and writes it to the file.

This code provides an overview of the dataset, including its size, summary statistics, common values of categorical variables, and the presence of missing values. The generated ”data_exploration.txt” file serves as a reference for further analysis and decision-making in subsequent research on optimal tracheal tube depth in pediatric patients.

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
Top 5 Most Common Values for sex:
sex
1    522
0    447

Top 5 Most Common Values for tube:
tube
3.5    392
4      211
3      205
4.5    112
5       30

# Missing Values
```

```
Number of Missing Values in tube: 0
Number of Missing Values in sex: 0
Number of Missing Values in age_c: 0
Number of Missing Values in ht: 0
Number of Missing Values in wt: 0
Number of Missing Values in tube_depth_G: 0
```

C Data Analysis

C.1 Code

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

```python

# IMPORT
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNet
from sklearn.metrics import mean_squared_error
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 in this case.

# DESCRIPTIVE STATISTICS
# No descriptive statistics table is needed in this case.

# PREPROCESSING
# Splitting the dataset into the predictors and target variable
x = df[['sex', 'age_c', 'ht', 'wt']]
y = df['tube_depth_G']
```

```

# Defining the train and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y,
↪  test_size=0.2, random_state=42)

# ANALYSIS
# Table 1: "Comparison of predictive performance between
↪  Random Forest and Elastic Net Models"
# Random Forest Model
rf = RandomForestRegressor()
parameters = {'n_estimators': range(20, 100, 10), 'max_depth':
↪  range(2, 8, 2)}
rf_optimal = GridSearchCV(rf, parameters, cv=5)
rf_optimal.fit(x_train, y_train)

# Apply the model to the test data
predictions_rf = rf_optimal.predict(x_test)
residuals_rf = (y_test - predictions_rf) ** 2

# Elastic Net Model
en = ElasticNet()
parameters = {'alpha': [0.0001, 0.001, 0.01, 0.1, 1, 10],
↪  'l1_ratio': [0.1, 0.3, 0.5, 0.7, 0.9]}
en_optimal = GridSearchCV(en, parameters, cv=5)
en_optimal.fit(x_train, y_train)

# Apply the model to the test data
predictions_en = en_optimal.predict(x_test)
residuals_en = (y_test - predictions_en) ** 2

# Perform a paired t-test
ttest_results = ttest_rel(residuals_rf, residuals_en)

# Record results in a DataFrame
df1 = pd.DataFrame({
    'Model': ['Random Forest', 'Elastic Net'],
    'Mean Squared Residual': [residuals_rf.mean(),
    ↪  residuals_en.mean()],
    't_statistic': [ttest_results.statistic,
    ↪  ttest_results.statistic],
```

```python
    'p_value': [ttest_results.pvalue, ttest_results.pvalue]
}, index=['Model 1: RF', 'Model 2: EN'])

df1.to_pickle('table_1.pkl')


# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(df),
 'accuracy of Random Forest model': rf_optimal.best_score_,
 'accuracy of Elastic Net model': en_optimal.best_score_
}

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 between the Random Forest model and the Elastic Net model for determining the optimal tracheal tube depth (OTTD) in pediatric patients.

First, the dataset is loaded from the ”tracheal_tube_insertion.csv” file. The dataset consists of 969 rows, with each row representing a patient. The predictors include the patient’s sex, age, height, and weight, while the target variable is the OTTD determined by chest X-ray.

Next, the dataset is split into training and test sets. The training set is used to train both the Random Forest model and the Elastic Net model, while the test set is used to evaluate their predictive performance.

The Random Forest model is trained using a grid search cross-validation approach to find the optimal combination of hyperparameters (n_estimators and max_depth). The model is then applied to the test data to make predictions of OTTD, and the mean squared residual values are calculated.

Similarly, the Elastic Net model is trained using another grid search cross-validation approach to find the optimal combination of hyperparameters (alpha and l1_ratio). The model is then applied to the test data to make predictions of OTTD, and the mean squared residual values are calculated.

After obtaining the mean squared residual values for both models, a paired t-test is performed to compare their performance. The t-test assesses whether there is a significant difference between the mean squared residuals of the two models.

The results of the analysis are recorded in a DataFrame, which includes the model names, mean squared residuals, t-statistic, and p-value. This DataFrame is then saved as "table_1.pkl".

Additionally, the code saves some additional results in the "additional_results.pkl" file. These results include the total number of observations in the dataset, as well as the accuracy of the Random Forest model and the Elastic Net model.

In summary, the code performs an analysis comparing the predictive performance of the Random Forest model and the Elastic Net model for determining the OTTD in pediatric patients. The analysis includes model training, prediction generation, mean squared residual calculation, paired t-test, and result recording.

C.3 Code Output

table_1.pkl

```
                     Model  Mean Squared Residual  t_statistic p_value
Model 1: RF  Random Forest               1.371273     1.371275  0.1719
Model 2: EN    Elastic Net               1.241301     1.371275  0.1719
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'accuracy of Random Forest model': 0.5366           ,
    'accuracy of Elastic Net model': 0.5481             ,
}
```

D LaTeX Table Design

D.1 Code

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

```

# IMPORT
import pandas as pd
```

```python
from my_utils import to_latex_with_note, format_p_value, is_str_in_df, split_mapping, AbbrToNameDef
from typing import Dict, Any, Tuple, Optional

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
 'sex': ('Sex', 'Patient Sex (0 = Female, 1 = Male)'),
 'age_c': ('Age', 'Patient Age (in years, rounded to half years)'),
 'ht': ('Height', 'Patient Height (in cm)'),
 'wt': ('Weight', 'Patient Weight (in kg)'),
}

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

# FORMAT VALUES
df['p_value'] = df['p_value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
column_mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
column_mapping.update({
    'p_value': ('P Value', 'the p-value from the t-test'),
    't_statistic': ('T-Statistic', 't-statistic from the t-test'),
})

df = df.rename(columns={k: v[0] for k, v in column_mapping.items() if v[0] is not None})

# Extract definitions to pass to legend
legend = {v[0]: v[1] for k, v in column_mapping.items() if v[1] is not None}

# Save as latex
to_latex_with_note(df, 'table_1.tex',
                   caption="Comparison of predictive performance between Random Forest and Elastic Net Models",
                   label='table:Table_1',
                   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_1.tex

```
\begin{table}[h]
\caption{Comparison of predictive performance between Random Forest and Elastic
    Net Models}
\label{table:Table_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llrrl}
\toprule
 & Model & Mean Squared Residual & T-Statistic & P Value \\
\midrule
\textbf{Model 1: RF} & Random Forest & 1.37 & 1.37 & 0.172 \\
\textbf{Model 2: EN} & Elastic Net & 1.24 & 1.37 & 0.172 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{P Value}: the p-value from the t-test
\item \textbf{T-Statistic}: t-statistic from the t-test
\end{tablenotes}
\end{threeparttable}
\end{table}
```