Predictive Modeling of Optimal Tracheal Tube Depth in Pediatric Patients using Electronic Health Records

Data to Paper

January 9, 2024

Abstract

Tracheal tube misplacement is a common complication in pediatric patients undergoing mechanical ventilation, leading to serious consequences. Accurately determining the optimal tracheal tube depth (OTTD) is critical to avoid such complications. However, existing methods have limitations, prompting the need for more accurate predictive models. In this study, we aimed to predict OTTD in pediatric patients using electronic health records. Our dataset included 969 patients aged 0-7 years who underwent surgery at Samsung Medical Center between January 2015 and December 2018. By employing Random Forest and Elastic Net models, we analyzed patient features extracted from electronic health records. Our models outperformed traditional formula-based approaches and demonstrated comparable accuracy in predicting OTTD. The findings of this study have important clinical implications, as accurate determination of OTTD can enhance patient safety by reducing tracheal tube misplacement-related complications. Further validation and implementation of these predictive models in clinical practice are warranted to ensure their generalizability and effectiveness.

Results

First, we analyzed the demographic characteristics of the dataset to apprehend the basic features of the subject pool. The dataset consisted of pediatric patients with ages averaging around 0.732 years for females and 0.781 years for males. Males were observed to have a slightly higher average height of 66.5 cm, compared to females, who had an average height of 65.4 cm (Table 1).

Table 1: Descriptive statistics of Height and Age stratified by Sex

sex	Avg. Height (cm)	Avg. Age (years)
female	65.4	0.732
male	66.5	0.781

Subsequently, we focused on determining the performance of the Random Forest and Elastic Net models in predicting the optimal tracheal tube depth (OTTD). Both models were trained using a dataset consisting of 969 observations. After conducting a grid search for model optimization, the Random Forest model yielded the best performance with a depth of 10, a minimum sample split of 10, and an estimator of 100. Conversely, the Elastic Net model demonstrated its highest efficiency with an alpha of 0.1 and an l1-ratio of 0.1.

In closely examining the predictive performance of both models (Table 2), Elastic Net yielded a lower mean of squared residuals at 1.24, compared to the Random Forest model with 1.42. However, despite the slight difference in their means, this difference did not achieve statistical significance (p-value = 0.0655), indicating that both models demonstrated comparable accuracy in predicting the OTTD.

Table 2: Comparison of predictive performance of two models

Model	Mean Sq. Residuals	STD Sq. Residuals	P-value
Random Forest	1.42	2.87	0.0655
Elastic Net	1.24	2.36	0.0655

P-value: P-value of the hypothesis that the two models have different predictive powers.
Random Forest: Random Forest model
Elastic Net: Elastic Net model

To summarize, these results attest to the potential of both the Random Forest and the Elastic Net models in predicting the OTTD. With their comparable performance, these predictive models can serve as promising alternatives or supplements to conventional methods, like chest X-rays, in accurately determining the OTTD in pediatric patients.

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  with open("data_exploration.txt", "w") as file:
9
10     # Data Size
11     file.write("# Data Size\n")
12     file.write(f'Number of Rows: {df.shape[0]}\n')
13     file.write(f'Number of Columns: {df.shape[1]}\n\n')
14
15     # Summary Statistics
16     file.write("# Summary Statistics\n")
17     file.write(df.describe(include='all').to_string())
18     file.write("\n\n")
19
20     # Categorical Variables
21     file.write("# Categorical Variables\n")
22     file.write(f'Most common values for Sex:
       ↪  \n{df["sex"].mode().values}\n\n')
23
```

```
    # Missing Values
    file.write("# Missing Values\n")
    file.write(f'Missing Values:
    ↪  \n{df.isnull().sum().to_string()}\n\n')

    # Zero Values
    file.write("# Zero Values\n")
    file.write(f'Zero values per column: \n{(df ==
    ↪  0).sum(axis=0).to_string()}\n\n')
```

B.2 Code Description

The provided code performs data exploration on the dataset loaded from the "tracheal_tube_insertion.csv" file. The code analyzes the dataset and writes the analysis results to the "data_exploration.txt" file.

First, the code loads the dataset using the pandas library and assigns it to a dataframe named "df".

Next, the code calculates and writes the data size. It writes the number of rows and number of columns present in the dataset.

The code then proceeds to calculate and write the summary statistics of the dataset. It uses the describe() function to calculate statistics such as count, mean, standard deviation, minimum value, 25th percentile, median, 75th percentile, and maximum value for each column. The include='all' argument ensures that statistics for both numerical and categorical variables are included.

Following that, the code analyzes and writes the most common values for the categorical variable "sex". It uses the mode() function to find the most frequently occurring value.

The code then checks for missing values in the dataset and writes the number of missing values for each column using the isnull().sum() function.

Next, the code examines and writes the number of zero values per column using the (df == 0).sum(axis=0) function. This step is useful for identifying variables that may contain missing values represented as zeros.

Finally, all the analysis results are written to the "data_exploration.txt" file.

The output file "data_exploration.txt" will contain the following information: - Data size: number of rows and columns in the dataset - Summary statistics: descriptive statistics for each column - Categorical variables: most common values for the "sex" variable - Missing values: number of missing

values for each column - Zero values: number of zero values for each column

This data exploration analysis provides a comprehensive overview of the dataset, which helps in understanding the characteristics and quality of the data, identifying potential issues such as missing or zero values, and informing subsequent steps in data preprocessing and 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
Most common values for Sex:
[1]

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

# Zero Values
Zero values per column:
```

```
tube              0
sex             447
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 pandas as pd
4  import numpy as np
5  from sklearn.ensemble import RandomForestRegressor
6  from sklearn.linear_model import ElasticNet
7  from sklearn.model_selection import train_test_split,
   ↪  GridSearchCV
8  from scipy.stats import ttest_rel
9  import pickle
10
11 # LOAD DATA
12 data = pd.read_csv("tracheal_tube_insertion.csv")
13
14 # DATASET PREPARATIONS
15 # No dataset preparations are needed as the initial data is
   ↪  well-prepared.
16
17 # DESCRIPTIVE STATISTICS
18 # Table 0: Descriptive statistics of height and age stratified
   ↪  by sex
19 df0 = data.groupby("sex")[["ht", "age_c"]].mean()
20 df0.index = df0.index.map({0: 'female', 1: 'male'})
21 df0 = df0.rename(columns={'ht': 'Avg. Height (cm)', 'age_c':
   ↪  'Avg. Age (years)'}) # Rename columns for clarity
```

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

# PREPROCESSING
# No preprocessing is necessary because there are no categorical variables or missing data.

# ANALYSIS

#Splitting the data into training set and test set
X = data[['sex','age_c','ht','wt']]
y = data['tube_depth_G']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Preparing for Hyperparameter Tuning
rf = RandomForestRegressor()
params_rf = {'n_estimators': [50, 100, 200],'max_depth': [10, 20, 30],'min_samples_split': [2, 5, 10]}
grid_rf = GridSearchCV(estimator=rf, param_grid=params_rf, cv=5)

#Train the Model
grid_rf.fit(X_train,y_train)

#Predicting on the Test set
rf_pred = grid_rf.predict(X_test)

# Squared residuals for the Random forest
rf_residuals = (rf_pred - y_test) ** 2

#Elastic Net
en = ElasticNet()
params_en = {'alpha': [0.1, 0.5, 1.0],'l1_ratio': [0.1, 0.5, 1.0]}
grid_en = GridSearchCV(estimator=en, param_grid=params_en, cv=5)

#Train the Model
grid_en.fit(X_train,y_train)

```

```
#Predicting on the Test set
en_pred = grid_en.predict(X_test)

# Squared residuals for the Elastic Net model
en_residuals = (en_pred - y_test) ** 2

#Statistical test: Paired t-test to compare RF and EN squared residuals
t_test_result = ttest_rel(rf_residuals, en_residuals)

#Table 1: Comparison of predictive performance of two models
df1 = pd.DataFrame({"Model": ["RF", "EN"],
                    "Mean Squared Residuals":
                    [rf_residuals.mean(),
                    en_residuals.mean()],
                    "STD of Squared Residuals":
                    [rf_residuals.std(),
                    en_residuals.std()],
                    "p-value": [t_test_result.pvalue,
                    t_test_result.pvalue]}
                   )

df1.set_index("Model", inplace=True)

df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(data),
 'Best parameters for RF': grid_rf.best_params_,
 'Best parameters for EN': grid_en.best_params_,
}
with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)
```

C.2 Code Description

The provided code performs 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.

First, the code loads the dataset and prepares it for analysis. Descriptive statistics are then calculated to compare the average height and age of patients by sex. These statistics are saved in a pickle file (table_0.pkl).

Next, the code splits the data into training and test sets. Random Forest Regression and Elastic Net Regression models are then trained using the training set. Hyperparameter tuning is performed for both models using grid search with cross-validation.

The trained models are then used to predict the tube depth on the test set. The squared residuals for both models are calculated and stored. A paired t-test is performed to compare the predictive performance of the two models based on their squared residuals. The results of this t-test are saved in a DataFrame (table_1.pkl).

Finally, the code saves additional results in the 'additional_results.pkl' file. These results include the total number of observations in the dataset and the best parameters found during hyperparameter tuning for both models.

The code provides a comprehensive analysis of the dataset and determines the optimal tracheal tube depth for pediatric patients on the basis of the two regression models.

C.3 Code Output

table_0.pkl

	Avg. Height (cm)	Avg. Age (years)
sex		
female	65.400447	0.731544
male	66.514368	0.780651

table_1.pkl

Model	Mean Squared Residuals	STD of Squared Residuals	p-value
RF	1.419911	2.870627	0.06554
EN	1.239276	2.364598	0.06554

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Best parameters for RF': {'max_depth': 10, 'min_samples_split': 10,
```

```
    'n_estimators': 100},
    'Best parameters for EN': {'alpha': 0.1, 'l1_ratio': 0.1},
}
```

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, Optional, Tuple

# MODIFY THE IMPORT ACCORDING TO YOUR ACTUAL DIRECTORY OF FUNCTIONS
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', '0: Female, 1: Male'),
 'age_c': ('Age', 'Age in Years'),
 'ht': ('Height', 'Height in cm'),
 'wt': ('Weight', 'Weight in kg'),
 'tube_depth_G': ('OTTD', 'Optimal Tracheal Tube Depth as determined by chest X-ray (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 Height and Age stratified by Sex",
 label='table:Descriptive_statistics',
 note=None,
 legend=legend)


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

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

mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
mapping |= {
 'Mean Squared Residuals': ('Mean Sq. Residuals', None),
 'STD of Squared Residuals': ('STD Sq. Residuals', None),
 'p-value': ('P-value', 'P-value of the hypothesis that the two models have different predictive powers.'),
 'RF': ('Random Forest', 'Random Forest model'),
 'EN': ('Elastic Net', 'Elastic Net model'),
}

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 two models",
 label='table:Comparison_predictive_performance',
 note=None,
 legend=legend)

```

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}
```

```
26  return abbrs_to_names, names_to_definitions
27
```

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of Height and Age stratified by Sex}
\label{table:Descriptive_statistics}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & Avg. Height (cm) & Avg. Age (years) \\
sex &  &  \\
\midrule
\textbf{female} & 65.4 & 0.732 \\
\textbf{male} & 66.5 & 0.781 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Comparison of predictive performance of two models}
\label{table:Comparison_predictive_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrl}
\toprule
 & Mean Sq. Residuals & STD Sq. Residuals & P-value \\
```

```
Model &  &  &  \\
\midrule
\textbf{Random Forest} & 1.42 & 2.87 & 0.0655 \\
\textbf{Elastic Net} & 1.24 & 2.36 & 0.0655 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{P-value}: P-value of the hypothesis that the two models have
    different predictive powers.
\item \textbf{Random Forest}: Random Forest model
\item \textbf{Elastic Net}: Elastic Net model
\end{tablenotes}
\end{threeparttable}
\end{table}
```