Accurate Prediction of Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

February 20, 2024

Abstract

Pediatric patients undergoing mechanical ventilation require precise placement of the tracheal tube to mitigate the risk of complications. However, determining the optimal tracheal tube depth (OTTD) in this population remains challenging. Existing methods, such as chest X-ray and formula-based models, have limitations in accuracy or practicality. To address this gap, we developed and compared two predictive models, Random Forest and Elastic Net, using a dataset of pediatric patients who underwent post-operative mechanical ventilation. Our models achieved accurate predictions of OTTD, as demonstrated by mean squared error (MSE) values ranging from 1.239 to 1.405. We further evaluated the models' performance using R squared scores, which ranged from 0.592 to 0.64, indicating their ability to capture a substantial portion of the variability in OTTD. However, it is important to acknowledge the limitations of these models and exercise caution when applying them in clinical practice. Nonetheless, our findings offer insights into the potential of predictive models to improve tracheal tube placement in pediatric patients, enhancing patient safety and reducing complications.

Results

In this study, we aimed to develop predictive models to accurately determine the optimal tracheal tube depth (OTTD) in pediatric patients undergoing post-operative mechanical ventilation. Accurate determination of the OTTD is crucial in ensuring the safety and well-being of these patients. Misplacement of the tracheal tube can lead to complications such as hypoxia, atelectasis, hypercarbia, pneumothorax, and even death. By developing predictive models, we can reduce the reliance on time-consuming chest X-rays, which expose patients to radiation. This can improve efficiency in determining the OTTD and enhance patient safety.

We compared the performance of two predictive models, Random Forest and Elastic Net, in predicting the OTTD. The Random Forest model was trained using hyperparameter tuning, with n_estimators ranging from 50 to 200 and max_depth set to None, 5, or 10. The Elastic Net model was optimized by adjusting the alpha and l1_ratio parameters, with alpha values of 0.1, 0.5, and 1.0, and l1_ratio values of 0.1, 0.5, and 1.0. Both models achieved accurate predictions, as indicated by the mean squared error (MSE) values. The MSE for the Random Forest model was 1.405, while the MSE for the Elastic Net model was 1.239. These values, reported in Table 1, reflect the average squared difference between the predicted and actual OTTD values and are consistent with those in the "Additional Results (additional_results.pkl)".

Table 1: Comparison of Random Forest and Elastic Net model performance in predicting OTTD after hyperparameter tuning.

Models	MSE	T-test P-value
Random Forest	1.41	0.0533
Elastic Net	1.24	0.0533

MSE: Mean Squared Error of the model predictions
T-test P-value: P-value of the paired t-test comparing the squared residuals of the 2 models

To evaluate the goodness of fit of the models, we calculated the R squared scores. The R squared score represents the proportion of the variance in the OTTD that can be explained by the models. The Random Forest model achieved an R squared score of 0.592, and the Elastic Net model achieved an R squared score of 0.64. These scores, reported in Table 2, are consistent with those obtained from the "Additional Results (additional_results.pkl)" and demonstrate the models' ability to capture a significant portion of the variability in the OTTD.

We further assessed the accuracy of the models by calculating the root mean squared error (RMSE). The RMSE measures the average deviation of the predicted OTTD values from the true values. The Random Forest model had an RMSE of 1.19, while the Elastic Net model had an RMSE of 1.11. These values, reported in Table 3, indicate the model's precision in estimating the tracheal tube depth and align with those provided in the

Table 2: R squared score for Random Forest and Elastic Net models in predicting OTTD.

Models	R Squared Score
Random Forest	0.592
Elastic Net	0.64

R Squared Score: Coefficient of determination of the model predictions

"Additional Results (additional_results.pkl)".

Table 3: Root Mean Squared Error for Random Forest and Elastic Net models in predicting OTTD.

Models	RMSE
Random Forest	1.19
Elastic Net	1.11

RMSE: Root Mean Squared Error of the model predictions

In summary, our analysis demonstrates that both the Random Forest and Elastic Net models can accurately predict the OTTD in pediatric patients undergoing post-operative mechanical ventilation. The models' performance, as measured by MSE, R squared score, and RMSE, highlights their ability to provide reliable estimates of the tracheal tube depth. These findings have important implications for improving tracheal tube placement in pediatric patients, ultimately enhancing patient safety and reducing complications.

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:

```

# necessary libraries
import pandas as pd

# loading the CSV dataset
dataset = pd.read_csv('tracheal_tube_insertion.csv')

# Creating the output text file
with open('data_exploration.txt', 'w') as file:

    # Data Size
    file.write('# Data Size\n')
    file.write('Number of Rows: ' + str(dataset.shape[0]) +
     ↪ '\n')
    file.write('Number of Columns: ' + str(dataset.shape[1]) +
     ↪ '\n\n')

    # Summary Statistics
    file.write('# Summary Statistics\n')
    file.write(str(dataset.describe()) + '\n\n')

    # Categorical Variables
    file.write('# Categorical Variables\n')
    # Assuming 'sex' is the only categorical variable
    sex_groupby =
    ↪  dataset.groupby("sex").size().reset_index(name='Counts')
    file.write(str(sex_groupby) + '\n\n')

    # Missing Values
    file.write('# Missing Values\n')
    file.write(str(dataset.isnull().sum()) + '\n\n')

file.close()

```

B.2 Code Description

The provided code performs data exploration on the given dataset by analyzing its size, summary statistics, categorical variables, and missing values.

First, the code loads the dataset from the provided CSV file using the pandas library.

Next, the code creates an output text file, "data_exploration.txt", where the results of the data exploration will be recorded.

The code then proceeds to perform the following analysis steps:

1. Data Size: The code determines and writes the number of rows and columns in the dataset, providing an overview of the dataset's size.

2. Summary Statistics: The code calculates and writes the summary statistics of the dataset, which includes measures such as count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum for each numerical column in the dataset. This provides a quantitative description of the distribution and central tendencies of the dataset.

3. Categorical Variables: Assuming the "sex" column is the only categorical variable, the code groups the dataset by sex and calculates the counts for each category. It then writes the resulting counts into the output file. This analysis provides insight into the distribution of the categorical variable in the dataset.

4. Missing Values: The code checks for missing values in the dataset and writes the number of missing values for each column into the output file. This analysis helps identify any missing data that may need to be addressed before further analysis or modeling.

The output text file, "data_exploration.txt", includes the results of the data exploration analysis mentioned above. This file can be used for reference and documentation purposes, providing a summary of the dataset's key characteristics and properties.

Overall, this code provides a systematic analysis of the dataset, allowing researchers to understand the data's structure, distributions, and potential data quality issues, thus enabling informed decision-making for subsequent steps in data analysis and modeling.

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  Counts
0    0     447
1    1     522

# 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:

```python


# IMPORT
import pandas as pd
import numpy as np
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, r2_score
from scipy.stats import ttest_rel
import pickle
from math import sqrt

# LOAD DATA
df = pd.read_csv('tracheal_tube_insertion.csv')

# DATASET PREPARATIONS
## No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
## No descriptive statistics table is needed.

# PREPROCESSING
features = df[['sex', 'age_c', 'ht', 'wt']]
target = df['tube_depth_G']

# Splitting the dataset
X_train, X_test, y_train, y_test = train_test_split(features,
↪ target, test_size=0.2, random_state=42)

# ANALYSIS

```

```python
## Table 1: "Comparison of Random Forest and Elastic Net model
↪  performance in predicting OTTD after hyperparameter
↪  tuning."

# Random Forest model with hyperparameter tuning
rf = RandomForestRegressor()
param_grid_rf = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 5, 10]
}
gs_rf = GridSearchCV(rf, param_grid_rf)
gs_rf.fit(X_train, y_train)
rf_predictions = gs_rf.predict(X_test)
rf_squared_errors = (y_test - rf_predictions) ** 2

# Elastic Net model with hyperparameter tuning
enet = ElasticNet()
param_grid_enet = {
    'alpha': [0.1, 0.5, 1.0],
    'l1_ratio': [0.1, 0.5, 1.0]
}
gs_enet = GridSearchCV(enet, param_grid_enet)
gs_enet.fit(X_train, y_train)
enet_predictions = gs_enet.predict(X_test)
enet_squared_errors = (y_test - enet_predictions) ** 2

# T-Test
ttest_results = ttest_rel(rf_squared_errors,
↪  enet_squared_errors)

# Results dataframe
results = {
    "Models": ["Random Forest", "Elastic Net"],
    "Mean Squared Error": [mean_squared_error(y_test,
    ↪  rf_predictions), mean_squared_error(y_test,
    ↪  enet_predictions)],
    "T-test P-value": [ttest_results.pvalue,
    ↪  ttest_results.pvalue]
}

```

```python
df1 = pd.DataFrame(results)
df1.set_index('Models', inplace=True)  # to set "Models" as
↪  the index
df1.to_pickle('table_1.pkl')

## Table 2: "R squared score for Random Forest and Elastic Net
↪  models"

# DataFrame for R squared scores
df2 = pd.DataFrame({
    'Models': ['Random Forest', 'Elastic Net'],
    'R Squared Score': [r2_score(y_test, rf_predictions),
    ↪  r2_score(y_test, enet_predictions)]
})
df2.set_index('Models', inplace=True)
df2.to_pickle('table_2.pkl')

## Table 3: "Root Mean Squared Error for Random Forest and
↪  Elastic Net models"

# DataFrame for Root Mean Squared Error (RMSE)
df3 = pd.DataFrame({
    'Models': ['Random Forest', 'Elastic Net'],
    'RMSE': [sqrt(mean_squared_error(y_test, rf_predictions)),
    ↪  sqrt(mean_squared_error(y_test, enet_predictions))]
})
df3.set_index('Models', inplace=True)
df3.to_pickle('table_3.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': df.shape[0],
 'Random Forest Mean Squared Error':
 ↪  mean_squared_error(y_test, rf_predictions),
 'Elastic Net Mean Squared Error': mean_squared_error(y_test,
 ↪  enet_predictions)
}
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 performance of two models, Random Forest and Elastic Net, in predicting the Optimal Tracheal Tube Depth (OTTD) for pediatric patients who require mechanical ventilation. The analysis is based on a dataset containing patient information such as sex, age, height, and weight, as well as the OTTD determined by chest X-ray.

The code begins by loading the dataset and preparing the data for analysis by separating the features (sex, age, height, and weight) and the target variable (OTTD). The dataset is then split into training and testing sets for model evaluation.

The analysis proceeds in three major steps: 1. Random Forest and Elastic Net model performance comparison: - The code tunes the hyperparameters of the Random Forest and Elastic Net models using GridSearchCV, optimizing the number of estimators and maximum depth for Random Forest, and the alpha and l1_ratio for Elastic Net. - The tuned models are used to make predictions on the testing set, and the mean squared error (MSE) is calculated for each model. These MSE values are used for model comparison. - A T-test is performed on the squared errors of the Random Forest and Elastic Net models to assess if there is a significant difference in their performance.

2. R-squared score comparison: - The code calculates the R-squared score, which indicates the proportion of the variance in the target variable that is predictable by the models. The R-squared scores are computed for both the Random Forest and Elastic Net models on the testing set.

3. Root Mean Squared Error (RMSE) comparison: - The code calculates the RMSE, which provides an easily interpretable estimate of the average prediction error of the models. The RMSE values are computed for both the Random Forest and Elastic Net models on the testing set.

Additionally, the code saves the results of the analysis in three separate tables, stored as pickled files: - "table_1.pkl" contains the comparison of Random Forest and Elastic Net model performance, including mean squared error values and the T-test p-value. - "table_2.pkl" contains the R-squared scores for the Random Forest and Elastic Net models. - "table_3.pkl" contains the RMSE values for the Random Forest and Elastic Net models.

Furthermore, the code saves additional results in the "additional_results.pkl" file, including the total number of observations in the dataset, as well as the mean squared error values for both the Random Forest and Elastic Net models.

This analysis would help evaluate the effectiveness of the Random Forest and Elastic Net models in predicting the OTTD for pediatric patients, providing insights into which model may be more suitable for clinical use.

C.3 Code Output

table_1.pkl

```
               Mean Squared Error T-test P-value
Models
Random Forest            1.405343         0.0533
Elastic Net              1.239276         0.0533
```

table_2.pkl

```
               R Squared Score
Models
Random Forest         0.592155
Elastic Net           0.640349
```

table_3.pkl

```
                   RMSE
Models
Random Forest  1.185472
Elastic Net    1.113228
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Random Forest Mean Squared Error': 1.405                 ,
    'Elastic Net Mean Squared Error': 1.239              ,
}
```

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': ('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'),
  'tube_depth_G': ('OTTD', 'Optimal tracheal tube depth as
   ↪  determined by chest X-ray (in cm)'),
  'Mean Squared Error': ('MSE', 'Mean Squared Error of the
   ↪  model predictions'),
  'T-test P-value': ('T-test P-value', 'P-value of the paired
   ↪  t-test comparing the squared residuals of the 2 models'),
  'R Squared Score': ('R Squared Score', 'Coefficient of
   ↪  determination of the model predictions'),
  'RMSE': ('RMSE', 'Root Mean Squared Error of the model
   ↪  predictions')
}

# TABLE 1:
df1 = pd.read_pickle('table_1.pkl')

# FORMAT VALUES
df1['T-test P-value'] = df1['T-test
 ↪  P-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
 ↪  is_str_in_df(df1, k)}
abbrs_to_names, legend = split_mapping(mapping)
df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex:
```

```
to_latex_with_note(
    df1, 'table_1.tex',
    caption = "Comparison of Random Forest and Elastic Net
    ↪  model performance in predicting OTTD after
    ↪  hyperparameter tuning.",
    label = 'table:comparison_models',
    note = None,
    legend = legend
)

# TABLE 2:
df2 = pd.read_pickle('table_2.pkl')

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df2, k)}
abbrs_to_names, legend = split_mapping(mapping)
df2 = df2.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as Latex:
to_latex_with_note(
    df2, 'table_2.tex',
    caption = "R squared score for Random Forest and Elastic
    ↪  Net models in predicting OTTD.",
    label = 'table:r_squared',
    note = None,
    legend = legend
)

# TABLE 3:
df3 = pd.read_pickle('table_3.pkl')

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df3, k)}
abbrs_to_names, legend = split_mapping(mapping)
df3 = df3.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as Latex:
to_latex_with_note(
    df3, 'table_3.tex',
    caption = "Root Mean Squared Error for Random Forest and
    ↪ Elastic Net models in predicting OTTD.",
    label = 'table:root_mse',
    note = None,
    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]]]
```

```python

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

```latex
\begin{table}[h]
\caption{Comparison of Random Forest and Elastic Net model performance in
    predicting OTTD after hyperparameter tuning.}
\label{table:comparison_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & MSE & T-test P-value \\
Models &  &  \\
\midrule
\textbf{Random Forest} & 1.41 & 0.0533 \\
\textbf{Elastic Net} & 1.24 & 0.0533 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{MSE}: Mean Squared Error of the model predictions
\item \textbf{T-test P-value}: P-value of the paired t-test comparing the
    squared residuals of the 2 models
\end{tablenotes}
\end{threeparttable}
```

```
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{R squared score for Random Forest and Elastic Net models in predicting
    OTTD.}
\label{table:r_squared}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & R Squared Score \\
Models &  \\
\midrule
\textbf{Random Forest} & 0.592 \\
\textbf{Elastic Net} & 0.64 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{R Squared Score}: Coefficient of determination of the model
    predictions
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
\begin{table}[h]
\caption{Root Mean Squared Error for Random Forest and Elastic Net models in
    predicting OTTD.}
\label{table:root_mse}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & RMSE \\
Models &  \\
\midrule
\textbf{Random Forest} & 1.19 \\
\textbf{Elastic Net} & 1.11 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{RMSE}: Root Mean Squared Error of the model predictions
\end{tablenotes}
\end{threeparttable}
\end{table}
```