Predicting Optimal Tracheal Tube Depth in Pediatric Patients using Machine Learning

Data to Paper

February 20, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) in pediatric patients is crucial for safe mechanical ventilation after surgery. However, current methods, such as chest X-ray and formula-based models, have limitations in accurately estimating OTTD. To address this challenge, we applied machine learning models to predict OTTD using patient features extracted from electronic health records. Our study focused on a dataset of 969 pediatric patients aged 0-7 years who underwent surgery at Samsung Medical Center. We trained Random Forest and Elastic Net models using this dataset and evaluated their accuracy on a separate test set. Both models demonstrated promising accuracy in predicting OTTD, with the Elastic Net model achieving 64.03% accuracy. Furthermore, analysis of residuals showed that both models produced accurate and consistent estimates of OTTD. Our findings highlight the potential of machine learning in optimizing tracheal tube depth and improving post-operative care and patient outcomes in pediatric populations. Further research is needed to validate our results and explore the wider application of machine learning in clinical settings.

Results

To assess the performance of our machine learning models in predicting the Optimal Tracheal Tube Depth (OTTD) in pediatric patients, we conducted several analyses using a dataset of 969 patients who underwent surgery at Samsung Medical Center.

First, we randomly split the dataset into a training set (80% of the data) and a test set (20% of the data). The Random Forest (RF) and Elastic Net (EN) models were trained using the training set, and their accuracy was evaluated on the test set.

The RF model achieved an accuracy of 59.65% on the test set, while the EN model achieved an accuracy of 64.03%. These results indicate that both models have predictive capabilities in estimating the OTTD in pediatric patients.

To assess the quality of the predictions further, we analyzed the residuals of the models based on the test set. The mean residuals for the RF and EN models were 0.037 cm and 0.0454 cm, respectively, indicating that, on average, the models were able to estimate the OTTD accurately (Table 1). The standard deviations of the residuals were 1.18 cm for the RF model and 1.11 cm for the EN model, suggesting that the predictions were consistent and had low variability.

Table 1: Summary of residuals from the Random Forest and Elastic Net models (Transposed)

	Residuals
RF Mean	0.037
RF Std.	1.18
EN Mean	0.0454
EN Std.	1.11

RF Mean: Mean of residuals from Random Forest model
RF Std.: Standard Deviation of residuals from Random Forest model
EN Mean: Mean of residuals from Elastic Net model
EN Std.: Standard Deviation of residuals from Elastic Net model

Additionally, we performed a paired t-test on the squared residuals to investigate whether there was a significant difference between the RF and EN models. The analysis showed no significant difference in the squared residuals between the two models (t-statistic = 1.58, p-value = 0.116) (Table 2). This indicates that both models had comparable performance in terms of overall prediction accuracy.

Table 2: T-test results for Random Forest and Elastic Net models

	T-Stat	P-value
Paired t-test	1.58	0.116

T-Stat: The calculated T-statistic from the paired T-test
P-value: The significance of the paired T-test

In summary, our machine learning models, the RF and EN models, demonstrated promising accuracy in predicting the OTTD in pediatric patients. The analysis of residuals showed that both models produced accurate and consistent estimates of the OTTD. Furthermore, the paired t-test revealed no significant difference between the models, indicating that they performed similarly. These findings suggest that machine learning approaches can be valuable tools for estimating the OTTD and improving patient outcomes.

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

# Required Libraries
import pandas as pd

# Reading Data
df = pd.read_csv('tracheal_tube_insertion.csv')

# Data Size
data_size = df.shape

# Summary Statistics
summary_stats = df.describe(include='all')

# Categorical Variables
cat_vars = df.select_dtypes(include=['object']).columns
cat_vals = {}
for col in cat_vars:
    cat_vals[col] = df[col].value_counts().idxmax()

# Missing Values
missing_vals = df.isnull().sum()

# Writing to text file
with open('data_exploration.txt', 'w') as file:
    file.write("# Data Size\n")
    file.write(f'Number of Rows: {data_size[0]}\n')
    file.write(f'Number of Columns: {data_size[1]}\n\n')

    file.write("# Summary Statistics\n")
    file.write(f'{summary_stats}\n\n')

    file.write("# Categorical Variables\n")
    if cat_vars.empty:
        file.write('No categorical variables\n\n')
    else:
        for col, val in cat_vals.items():
            file.write(f'{col} has maximum values of
            ↪  {val}\n\n')

    file.write("# Missing Values\n")
    num_missing = missing_vals.sum()
    if num_missing == 0:
        file.write('No missing values\n')
    else:
        for col, val in missing_vals.items():
            if val > 0:
                file.write(f'{col} has {val} missing
                ↪  values\n')

```

B.2 Code Description

The provided code performs data exploration on the dataset "tracheal_tube_insertion.csv". This dataset contains information about pediatric patients who received post-operative mechanical ventilation. The goal of the data exploration is to gain insights into the dataset and understand its characteristics.

The code starts by reading the dataset into a pandas DataFrame, which allows for easy manipulation and analysis of the data.

Next, the code calculates the size of the dataset, providing the number of rows and columns. This information gives an overview of the dataset's volume.

The code then computes summary statistics for each column in the dataset. This includes measures such as count, mean, standard deviation, and quartiles for numerical variables, and frequency of occurrence for categorical variables. The summary statistics provide a quick overview of the distribution and central tendency of the dataset.

The code also identifies categorical variables in the dataset and determines the most common value for each categorical variable. This information helps in understanding the dominant categories within the dataset.

Furthermore, the code checks for missing values in the dataset and determines the number of missing values for each column. This helps identify any gaps or inconsistencies in the dataset that may need to be addressed during data preprocessing.

Finally, the code writes the results of the data exploration to a text file named "data_exploration.txt". The file includes the data size (number of rows and columns), summary statistics, information about categorical variables (including the most common values), and details of any missing values present in the dataset.

Overall, the code provides a comprehensive overview of the dataset's characteristics, allowing researchers and analysts to make informed decisions during subsequent data preprocessing and analysis stages.

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
No categorical variables
```

```
# Missing Values
No missing values
```

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
import numpy as np

# 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']].values
target = df['tube_depth_G'].values
tr_X, test_X, tr_y, test_y = train_test_split(features, target, train_size=0.8, random_state=42)

# ANALYSIS
## Table 1: "Average Residuals and Their Standard Deviations for Random Forest and Elastic Net Models"
```

```python
# Initialize models with default parameters
rf = RandomForestRegressor(random_state=42)
en = ElasticNet(random_state=42)

# GridSearchCV for hyperparameter tuning
parameter_grid_rf = {'n_estimators': [100, 200], 'max_depth': [None, 5, 10]}
parameter_grid_en = {'alpha': [0.1, 0.5, 1], 'l1_ratio': [0.1, 0.5, 1]}

gs_rf = GridSearchCV(rf, parameter_grid_rf, cv = 5)
gs_rf.fit(tr_X, tr_y)
gs_en = GridSearchCV(en, parameter_grid_en, cv = 5)
gs_en.fit(tr_X, tr_y)

rf_best = gs_rf.best_estimator_
en_best = gs_en.best_estimator_

# Now, apply these models to the test set and calculate residuals
residuals_rf = test_y - rf_best.predict(test_X)
residuals_en = test_y - en_best.predict(test_X)

# Now, create a dataframe with the mean and std of these residuals and save the dataframe
df1 = pd.DataFrame({
    'rf_residuals_mean': [np.mean(residuals_rf)],
    'rf_residuals_std': [np.std(residuals_rf)],
    'en_residuals_mean': [np.mean(residuals_en)],
    'en_residuals_std': [np.std(residuals_en)]},
    index=['Residuals'])
df1.to_pickle('table_1.pkl')

## Table 2: "Paired T-test Results for Squared Residuals of the Two Models"
# Now, we perform the paired t-test in order to see whether the squared residuals are significantly different
t_results = ttest_rel(residuals_rf**2, residuals_en**2)

```

```
df2 = pd.DataFrame({'t_statistic': [t_results.statistic], 'p_value': [t_results.pvalue]},
                   index=['Paired t-test'])
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': df.shape[0],
    'Accuracy of RF model': rf_best.score(test_X, test_y),
    'Accuracy of EN model': en_best.score(test_X, test_y)
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)

```

C.2 Code Description

The provided code performs an analysis to determine the optimal tracheal tube depth (OTTD) for pediatric patients who require mechanical ventilation.

First, the dataset is loaded, which contains patient information such as sex, age, height, weight, and the OTTD as determined by chest X-ray.

Next, the dataset is split into training and test sets. The training set is used to train two regression models: Random Forest and Elastic Net. Hyperparameter tuning is performed using GridSearchCV to find the best parameters for these models.

The models are then applied to the test set, and the residuals (difference between the predicted OTTD and actual OTTD) are calculated for both models. These residuals are used to create Table 1, which shows the average residuals and their standard deviations for both models.

Table 2 is created by performing a paired t-test on the squared residuals of the two models. This test determines whether the squared residuals are significantly different, providing insights into the performance of the models.

Finally, additional results are saved in the ”additional_results.pkl” file. These include the total number of observations, and the accuracy of the Random Forest and Elastic Net models.

Overall, this code provides an analysis of different regression models to determine the optimal tracheal tube depth for pediatric patients, and evaluates their performance using statistical tests and additional results.

C.3 Code Output

table_1.pkl

```
            rf_residuals_mean  rf_residuals_std  en_residuals_mean
    en_residuals_std
Residuals            0.036963          1.178505           0.045375
    1.112302
```

table_2.pkl

```
                t_statistic p_value
Paired t-test      1.578506  0.1161
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Accuracy of RF model': 0.5965            ,
    'Accuracy of EN model': 0.6403            ,
}
```

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 = {
 'rf_residuals_mean': ('RF Mean', 'Mean of residuals from
  ↪  Random Forest model'),
 'rf_residuals_std': ('RF Std.', 'Standard Deviation of
  ↪  residuals from Random Forest model'),
 'en_residuals_mean': ('EN Mean', 'Mean of residuals from
  ↪  Elastic Net model'),
 'en_residuals_std': ('EN Std.', 'Standard Deviation of residuals from Elastic Net model'),
}

# TABLE 1
df = pd.read_pickle('table_1.pkl')
df = df.T # transposing the dataframe

# 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="Summary of residuals from the Random Forest and Elastic Net models (Transposed)",
 label='table:summary_of_residuals',
 legend=legend
)

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

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

# RENAME ROWS AND COLUMNS
mapping = {
 't_statistic': ('T-Stat', 'The calculated T-statistic from the paired T-test'),
 'p_value': ('P-value', 'The significance of the paired T-test'),
}

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_2.tex',
 caption="T-test results for Random Forest and Elastic Net models",
 label='table:ttest',
 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]))

```

```python
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

```latex
\begin{table}[h]
\caption{Summary of residuals from the Random Forest and Elastic Net models (Transposed)}
\label{table:summary_of_residuals}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Residuals \\
\midrule
\textbf{RF Mean} & 0.037 \\
\textbf{RF Std.} & 1.18 \\
\textbf{EN Mean} & 0.0454 \\
\textbf{EN Std.} & 1.11 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{RF Mean}: Mean of residuals from Random Forest model
\item \textbf{RF Std.}: Standard Deviation of residuals from Random Forest model
\item \textbf{EN Mean}: Mean of residuals from Elastic Net model
```

```
\item \textbf{EN Std.}: Standard Deviation of residuals from Elastic Net model
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{T-test results for Random Forest and Elastic Net models}
\label{table:ttest}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & T-Stat & P-value \\
\midrule
\textbf{Paired t-test} & 1.58 & 0.116 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{T-Stat}: The calculated T-statistic from the paired T-test
\item \textbf{P-value}: The significance of the paired T-test
\end{tablenotes}
\end{threeparttable}
\end{table}
```