

Predicting Optimal Tracheal Tube Depth in Pediatric Patients using Machine Learning

Data to Paper

January 8, 2024

Abstract

Pediatric patients undergoing mechanical ventilation are at risk of tracheal tube misplacement, leading to severe complications. The determination of the optimal tracheal tube depth (OTTD) is crucial for patient safety. However, existing methods based on chest X-ray or formula-based models have limitations. In this study, we employ a machine learning approach to predict OTTD using patient electronic health records. We analyzed a dataset of pediatric patients who underwent surgery and received post-operative mechanical ventilation. Our random forest and elastic net models demonstrated promising predictive power, with R2 scores of 0.531 and 0.64, respectively. Furthermore, the elastic net model exhibited a root mean square error of 1.11 cm. These results highlight the potential of machine learning to accurately predict OTTD in pediatric patients. Further validation and refinement are needed before implementing these models in clinical practice. Our findings have implications for enhancing patient outcomes and mitigating complications associated with tracheal tube misplacement.

Results

In this study, our aim was to determine the optimal tracheal tube depth (OTTD) in pediatric patients using a machine learning approach. We first computed the descriptive statistics of patient ages and weights stratified by sex to gain insights into the characteristics of the dataset. This analysis was performed to evaluate the distribution of ages and weights in the pediatric population. The results, shown in Table 1, revealed that the average age was 0.732 years for females and 0.781 years for males, while the average weight was 6.84 kg for females and 7.37 kg for males. We also noted slightly higher standard deviations for weights compared to ages, indicating a larger variability in patient weights within each sex group.

Table 1: Descriptive statistics of patient ages and weights stratified by their sex

sex	female	male
Avg Age	0.732	0.781
Avg Wt	6.84	7.37
SD Age	1.4	1.47
SD Weight	4.57	4.94

Avg Age: Average of age (years, rounded to half years)

Avg Wt: Average of weight (kg)

To determine the predictive power of our machine learning models, we developed random forest and elastic net models to predict OTTD using patient electronic health records. The random forest model achieved an R2 score of 0.531, indicating that 53.1% of the variance in OTTD can be explained by the model. Similarly, the elastic net model performed slightly better, with an R2 score of 0.64, suggesting that 64% of the variance in OTTD can be explained [Table 2]. Moreover, the elastic net model exhibited a root mean square error (RMSE) of 1.11 cm, indicating that, on average, the predicted OTTD values deviate from the true values by 1.11 cm. These results of the machine learning models demonstrate their accuracy in predicting the tracheal tube depth.

Table 2: Predictive Power of Machine Learning Models

model	R2 Score	RMSE	t-stat	p-val
Random Forest	0.531	1.27	0.292	0.771
Elastic Net	0.64	1.11	-	-

R2 Score: Model's goodness-of-fit score

RMSE: Root Mean Square Error

p-val: Probabilities from t-test

We also determined that our dataset consisted of a total of 969 observations. The best parameters selected through grid search for the random forest model were `n_estimators = 90`, indicating the number of trees in the forest. For the elastic net model, the best parameters were `alpha = 0.2` and `l1_ratio = 0.1`, which control the regularization strength and the ratio between L1 and L2 penalties, respectively. These additional results are provided in the "Additional Results" section.

To further evaluate the performance of the models, we conducted a paired t-test on the predictions of the random forest and elastic net models. This statistical test aimed to determine whether there was a significant difference between the predicted OTTD values of the two models. The t-test results revealed a t-statistic of 0.292 and a p-value of 0.771 [Table 2]. These findings imply that there was no significant difference between the predictions of the two models.

In summary, our analysis using machine learning models demonstrated promising predictive power in determining the optimal tracheal tube depth in pediatric patients. The descriptive statistics provided insights into the distribution of ages and weights stratified by sex in the dataset. The R2 scores of 0.531 and 0.64 for the random forest and elastic net models, respectively, indicate the proportion of variance in OTTD that can be explained by the models. The RMSE value of 1.11 cm further emphasizes the clinical significance and accuracy of the predicted OTTD values. Overall, the results suggest that machine learning approaches have the potential to enhance patient outcomes by accurately predicting OTTD and reducing complications associated with tracheal tube misplacement.

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  # Importing Required Libraries
3  import pandas as pd
4  import numpy as np
5
6  # loading the dataset
7  data = pd.read_csv('tracheal_tube_insertion.csv')
8
9  # Write to "data_exploration.txt" file
10 with open('data_exploration.txt', 'w') as file:
11
12     # Find rows and columns in the data
13     rows, cols = data.shape
14     file.write("# Data Size")
15     file.write(f"\nNumber of Rows: {rows}")
16     file.write(f"\nNumber of Columns: {cols}")
17
18     # Summary Statistics
19     file.write("\n\n# Summary Statistics")
20     summary_stats = data.describe(include='all')
21     summary_stats.to_string(file)
22
23     # Categorical Variables - Dataset has no conventional
24     ↪ categorical columns,
```

```

24     # however, the sex column could be considered one. Let's
    ↪     examine it further.
25     file.write("\n\n# Categorical Variables")
26     file.write("\nExamining 'sex' Column:")
27     most_common_value = data['sex'].value_counts().idxmax()
28     frequency_of_most_common_value =
    ↪     data['sex'].value_counts().max()
29     file.write(f"\nThe most common value is
    ↪     '{most_common_value}' with frequency
    ↪     {frequency_of_most_common_value}.")
30
31     # Missing Values
32     file.write("\n\n# Missing Values")
33     missing_values = data.isnull().sum()
34     missing_values.to_string(file)
35

```

B.2 Code Description

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

The analysis steps include:

1. Loading and Reading the Dataset: The code reads the dataset from the file "tracheal_tube_insertion.csv" using the pandas library.
2. Number of Rows and Columns: The code determines the number of rows and columns in the dataset using the shape attribute of the pandas DataFrame. It then writes the number of rows and columns to the output file.
3. Summary Statistics: The code computes summary statistics of the dataset using the describe function of pandas DataFrame. It includes statistics such as count, mean, standard deviation, minimum, quartiles, and maximum for numeric columns. It writes these summary statistics to the output file.
4. Categorical Variables: The code examines the 'sex' column in the dataset, which can be considered as a categorical variable. It determines the most common value in the 'sex' column and its frequency using the value_counts function of pandas Series. It writes this information to the output file.
5. Missing Values: The code identifies missing values in the dataset using the isnull function of pandas DataFrame. It counts the number of missing

values for each column and writes this information to the output file.

The "data_exploration.txt" file contains a summary of the analysis steps, including the number of rows and columns, summary statistics, information on categorical variables, and missing value counts for each column in the dataset.

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

Examining 'sex' Column:

The most common value is '1' with frequency 522.

# Missing Value	tube
sex	0
age_c	0
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 GridSearchCV,
   ↪ train_test_split
8  from sklearn.metrics import mean_squared_error
9  from scipy import stats
10 import pickle
11
12 # LOAD DATA
13 data = pd.read_csv("tracheal_tube_insertion.csv")
14
15 # DATASET PREPARATIONS
16 # No dataset preparations are needed.
17
18 # DESCRIPTIVE STATISTICS
19 # Table 0: "Descriptive statistics of patient ages and weights
   ↪ stratified by their sex"
20 df0_mean = data.groupby('sex')[['age_c', 'wt']].mean()
21 df0_std = data.groupby('sex')[['age_c', 'wt']].std()
22 df0 = pd.concat([df0_mean, df0_std], axis=1)
23 df0.columns = ["average_age", "average_weight",
   ↪ "standard_deviation_age", "standard_deviation_weight"]
24 df0.index = df0.index.map({0: 'female', 1: 'male'}) # 'sex'
   ↪ column will be used as the index
25 df0.to_pickle('table_0.pkl')
26
27 # PREPROCESSING
28 # No preprocessing is needed because data doesn't have any
   ↪ categorical variables
29 # and is already in a format suitable for machine learning
   ↪ model
30
31 # ANALYSIS
32 # Table 1: "Predictive Power of Machine Learning Models"
33
34 # First split the data into train and validation sets

```



```

35 X = data[['sex', 'age_c', 'ht', 'wt']]
36 y = data['tube_depth_G']
37 X_train, X_test, y_train, y_test = train_test_split(X, y,
    ↪ test_size=0.2, random_state=42)
38
39 # Initialize models
40 rf = RandomForestRegressor(random_state = 42)
41 en = ElasticNet(random_state = 42)
42
43 # Define parameters for grid search
44 params_rf = {'n_estimators': np.arange(10, 100, 10)}
45 params_en = {'alpha': [0.2, 0.4, 0.6, 0.8, 1], 'l1_ratio':
    ↪ [0.1, 0.5, 0.9]} # Reduced parameters for ElasticNet
46
47 # Grid search over parameters
48 grid_rf = GridSearchCV(estimator=rf, param_grid=params_rf,
    ↪ cv=5)
49 grid_en = GridSearchCV(estimator=en, param_grid=params_en,
    ↪ cv=5)
50
51 # Fit models
52 grid_rf.fit(X_train, y_train)
53 grid_en.fit(X_train, y_train)
54
55 # Get predictions
56 rf_preds = grid_rf.predict(X_test)
57 en_preds = grid_en.predict(X_test)
58
59 # Perform paired t-test, keeping results object
60 ttest_results = stats.ttest_rel(rf_preds, en_preds)
61
62 # Create dataframe for the table
63 df1 = pd.DataFrame({
64     'model': ['Random Forest', 'Elastic Net'],
65     'R2_score': [grid_rf.score(X_test, y_test),
    ↪ grid_en.score(X_test, y_test)],
66     'RMSE': [np.sqrt(mean_squared_error(y_test, rf_preds)),
    ↪ np.sqrt(mean_squared_error(y_test, en_preds))],
67     't-statistic': [ttest_results.statistic, '-'],
68     'p-value': [ttest_results.pvalue, '-']

```

```

69 })
70
71 df1.set_index('model', inplace=True) # setting 'model' column
    ↪ as the index
72 df1.to_pickle('table_1.pkl')
73
74 # SAVE ADDITIONAL RESULTS
75 additional_results = {
76     'Total number of observations': len(data),
77     'best parameters for RF': grid_rf.best_params_,
78     'best parameters for EN': grid_en.best_params_
79 }
80
81 with open('additional_results.pkl', 'wb') as f:
82     pickle.dump(additional_results, f)
83

```

C.2 Code Description

The code performs data analysis on the provided dataset to determine the optimal tracheal tube depth (OTTD) for pediatric patients who require mechanical ventilation. The analysis consists of the following steps:

1. Data Loading: The code reads the dataset, "tracheal_tube_insertion.csv", containing information about the patients' sex, age, height, weight, and the OTTD determined by chest X-ray.
2. Descriptive Statistics: The code calculates descriptive statistics of patient ages and weights stratified by sex. The mean and standard deviation are computed and stored in a dataframe, which is then saved as "table_0.pkl".
3. Preprocessing: No preprocessing is required as the dataset does not contain categorical variables, and the data is already in a suitable format for machine learning models.
4. Model Training and Evaluation: The dataset is split into training and validation sets. Two machine learning models, Random Forest and Elastic Net, are initialized. Grid search is performed with cross-validation to find the best hyperparameters for each model. The models are then fitted to the training data.
5. Prediction and Evaluation: Using the trained models, predictions are made on the validation set. The R2 score and root mean squared error (RMSE) are calculated to evaluate the performance of each model. Addi-

tionally, a paired t-test is performed to compare the predictions of the two models.

6. Results Storage: The results of the model evaluation, including R2 score, RMSE, t-statistic, and p-value, are stored in a dataframe named "table_1.pkl". Additional results such as the total number of observations, the best parameters for the Random Forest model, and the best parameters for the Elastic Net model are stored in a separate file named "additional_results.pkl".

The "additional_results.pkl" file contains a dictionary with the following information: - Total number of observations: The total number of patients in the dataset. - Best parameters for the Random Forest model: The optimal hyperparameters found through grid search. - Best parameters for the Elastic Net model: The optimal hyperparameters found through grid search.

Overall, the code uses machine learning models to predict the optimal tracheal tube depth for pediatric patients based on their demographic and physical characteristics, evaluates the performance of the models, and stores the results for further analysis.

C.3 Code Output

table_0.pkl

	average_age	average_weight	standard_deviation_age
sex	standard_deviation_weight		
female	0.731544	6.841902	1.402500
	4.568146		
male	0.780651	7.370556	1.472808
	4.935102		

table_1.pkl

	R2_score	RMSE	t-statistic	p-value
model				
Random Forest	0.531455	1.270630	0.29195	0.7706
Elastic Net	0.639661	1.114292	-	-

additional_results.pkl

```
{
    'Total number of observations': 969,
    'best parameters for RF': {'n_estimators': 90},
    'best parameters for EN': {'alpha': 0.2, 'l1_ratio': 0.1},
}
```

D LaTeX Table Design

D.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 from typing import Dict, Any, Optional, Tuple
5 from my_utils import to_latex_with_note, format_p_value,
6     ↪ is_str_in_df, split_mapping, AbbrToNameDef
7
8 # PREPARATION FOR ALL TABLES
9
10 # Shared mapping for all tables
11 shared_mapping: dict = {
12     "sex": ("Sex", "0: Female, 1: Male"),
13     "age_c": ("Age", "Age (years, rounded to half years)"),
14     "wt": ("Weight", "Weight (kg)"),
15     "ht": ("Height", "Height (cm)"),
16     "tube_depth_G": ("OTTD", "Optimal Tracheal Tube Depth as
17     ↪ determined by chest X-ray (cm)"),
18     "R2_score": ("R2 Score", "Model's goodness-of-fit score"),
19     "RMSE": ("RMSE", "Root Mean Square Error")
20 }
21
22 # TABLE 0
23 df0 = pd.read_pickle('table_0.pkl')
24
25 # TRANSPOSE THE DATAFRAME
26 df0 = df0.T
```

```

26
27 # RENAME ROWS AND COLUMNS
28 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df0, k)}
29 mapping |= {
30     "average_age": ("Avg Age", "Average of age (years, rounded
    ↪ to half years)"),
31     "average_weight": ("Avg Wt", "Average of weight (kg)"),
32     "standard_deviation_age": ("SD Age", None),
33     "standard_deviation_weight": ("SD Weight", None),
34 }
35 abbrs_to_names, legend = split_mapping(mapping)
36 df0 = df0.rename(columns=abbrs_to_names, index=abbrs_to_names)
37
38 # Save as latex
39 to_latex_with_note(
40     df0, "table_0.tex",
41     caption="Descriptive statistics of patient ages and
    ↪ weights stratified by their sex",
42     label="table:desc_stats_age_weight_by_sex",
43     legend=legend
44 )
45
46
47 # TABLE 1
48 df1 = pd.read_pickle("table_1.pkl")
49
50 # FORMAT VALUES
51 df1["p-value"] = df1["p-value"].apply(format_p_value)
52
53 # RENAME ROWS AND COLUMNS
54 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df1, k)}
55 mapping |= {
56     "t-statistic": ("t-stat", None),
57     "p-value": ("p-val", "Probabilities from t-test"),
58 }
59 abbrs_to_names, legend = split_mapping(mapping)
60 df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)
61

```

```

62 # Save as latex
63 to_latex_with_note(
64     df1,
65     "table_1.tex",
66     caption="Predictive Power of Machine Learning Models",
67     label="table:pred_power_ml_models",
68     legend=legend,
69 )
70

```

D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
  ↪ str, note: str = None, legend: Dict[str, str] = None,
  ↪ **kwargs):
2     """
3     Converts a DataFrame to a LaTeX table with optional note and
  ↪ legend added below the table.
4
5     Parameters:
6     - df, filename, caption, label: as in `df.to_latex`.
7     - note (optional): Additional note below the table.
8     - legend (optional): Dictionary mapping abbreviations to full
  ↪ names.
9     - **kwargs: Additional arguments for `df.to_latex`.
10
11     Returns:
12     - None: Outputs LaTeX file.
13     """
14
15 def format_p_value(x):
16     returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"
17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19     return any(s in level for level in getattr(df.index,
  ↪ 'levels', [df.index]) + getattr(df.columns, 'levels',
  ↪ [df.columns]))
20

```

```

21 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
22
23 def split_mapping(abbrs_to_names_and_definitions:
    ↳ AbbrToNameDef):
24     abbrs_to_names = {abbr: name for abbr, (name, definition) in
    ↳ abbrs_to_names_and_definitions.items() if name is not
    ↳ None}
25     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 patient ages and weights stratified by their
sex}
\label{table:desc_stats_age_weight_by_sex}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
sex & female & male \\
\midrule
\textbf{Avg Age} & 0.732 & 0.781 \\
\textbf{Avg Wt} & 6.84 & 7.37 \\
\textbf{SD Age} & 1.4 & 1.47 \\
\textbf{SD Weight} & 4.57 & 4.94 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Avg Age}: Average of age (years, rounded to half years)
\item \textbf{Avg Wt}: Average of weight (kg)
\end{tablenotes}
\end{threeparttable}

```

```
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Predictive Power of Machine Learning Models}
\label{table:pred_power_ml_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrl}
\toprule
& R2 Score & RMSE & t-stat & p-val \\
model & & & & \\
\midrule
\textbf{Random Forest} & 0.531 & 1.27 & 0.292 & 0.771 \\
\textbf{Elastic Net} & 0.64 & 1.11 & - & - \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{R2 Score}: Model's goodness-of-fit score
\item \textbf{RMSE}: Root Mean Square Error
\item \textbf{p-val}: Probabilities from t-test
\end{tablenotes}
\end{threeparttable}
\end{table}
```