Machine Learning Outperforms Traditional Methods in Pediatric Tracheal Tube Positioning

data-to-paper

April 25, 2024

Abstract

Accurate tracheal tube placement is critical for pediatric patient safety during mechanical ventilation, with current standard methods showing a high rate of misplacement. Recognizing the need to improve patient outcomes, this study explores the capability of machine learning (ML) models to predict Optimal Tracheal Tube Depth (OTTD) as an alternative to existing age- and height-based formulas. Utilizing a dataset comprising demographic and physiological data for 969 pediatric patients from a single center, a range of ML models including RandomForest, Support Vector Machine, Neural Networks, and ElasticNet was employed. These models successfully demonstrated superior performance in OTTD prediction, overtaking traditional formulaic approaches with statistically significant reductions in mean squared errors and enhanced predictive accuracy. Despite its retrospective design and limitation to data from one center, our findings reveal that ML has the potential to transform pediatric tracheal tube placement, enhancing safety and possibly reducing reliance on radiographic confirmation. Future work is needed to validate these models prospectively and across diverse clinical settings to solidify their role in critical care.

Introduction

Effective mechanical ventilation, a vital element of pediatric patient care, hinges upon accurate positioning of the tracheal tube. The shorter tracheal length in pediatric patients necessitates greater precision, as endotracheal tube misplacement often risks serious complications such as hypoxia, atelectasis, and hypercarbia, potentially resulting in fatal consequences [1, 2]. To ascertain the optimal tracheal tube depth (OTTD), healthcare providers traditionally rely on chest X-rays as the gold standard for positional determination [3, 4]. However, this method imposes the burden of radiation

exposure and protracted durations, leading to the exploration of alternative formula-based approaches.

Orthodox formula-based models engage features such as age and height in their calculations [5, 6]. Despite their widespread use, these existing models have shown limited success and a significant margin of error in estimating the OTTD, signifying a clear research gap in this crucial area of patient care [5, 6].

Filling the aforementioned gap, our study introduces the use of machine learning (ML) models to predict the OTTD. Utilizing a comprehensive dataset derived from Samsung Medical Center's pediatric patients aged 0 to 7 years, we compare ML models to conventional formula-based methods to ascertain the most effective means of determining the OTTD [7, 8]. With the dataset comprising critical demographic and physiological parameters, we optimize various ML models, comprising random forests, support-vector machines, and others, as seen in similar healthcare scenarios [9].

Our results demonstrate the superior performance of machine learning models, particularly Support Vector Machine and ElasticNet, compared to traditional formula-based methods. These ML models significantly reduced mean squared errors and improved the accuracy of OTTD prediction. The introduction of these ML models heralds a potential transformation in pediatric tracheal tube placement, with the prospect of enhancing patient safety and reducing reliance on radiographic methods for positional confirmation [9].

Results

First, to establish a comprehensive profile of the patient population for accurate tracheal tube placement, we analyzed the descriptive statistics stratified by sex. From the Samsung Medical Center dataset comprising 969 pediatric patients between 2015 and 2018, we found an average age of 0.758 years with a standard deviation of 1.44. Average height and weight were 66 cm (SD = 19.1) and 7.13 kg (SD = 4.77), respectively. The gender ratio was balanced with 53.9% males and 0.461% females, as detailed in Table 1.

Subsequently, we compared the performance of machine learning (ML) models to formula-based methods in estimating the Optimal Tracheal Tube Depth (OTTD). After hyperparameter optimization, the models' mean squared errors (MSEs) were evaluated. The Support Vector Machine (SVM) produced the most promising results with the lowest MSEs—0.937, 0.937, and 0.937—when compared to age-, height-, and tube-based formulaic predic-

Table 1: Descriptive statistics of sex, age, height, and weight stratified by sex.

	Sex	Age (Years)	Height (cm)	Weight (kg)
mean	0.539	0.758	66	7.13
std	0.499	1.44	19.1	4.77
count	969	969	969	969

Sex: 0: Female, 1: Male

tions, which had MSEs of 1.61, 3.19, and 1.84, respectively. These represent average accuracy gains of the ML model over the formula-based methods. The statistical significance of these differences is evident with low p-values (SVM vs. age-based formula: 1.76 10^{-5}, SVM vs. height-based formula: P $< 10^{-6}$, SVM vs. tube-based formula: P $< 10^{-6}$), as shown in Table 2.

Table 2: Comparison of Mean Squared Error (MSE) and p-value for ML models and formulas.

ML Model vs Formula	ML MSE	F. MSE	p-val.
RandomForest Vs F depth formula age	1.12	1.61	0.00311
RandomForest Vs F depth formula height	1.12	3.19	$<10^{-6}$
RandomForest Vs F depth formula tube	1.12	1.84	0.000152
SVM Vs F depth formula age	0.937	1.61	1.76 10^{-5}
SVM Vs F depth formula height	0.937	3.19	$<10^{-6}$
SVM Vs F depth formula tube	0.937	1.84	$<10^{-6}$
NN Vs F depth formula age	1.21	1.61	0.0279
NN Vs F depth formula height	1.21	3.19	$<10^{-6}$
NN Vs F depth formula tube	1.21	1.84	0.00138
ElasticNet Vs F depth formula age	0.95	1.61	3.46 10^{-5}
ElasticNet Vs F depth formula height	0.95	3.19	$<10^{-6}$
ElasticNet Vs F depth formula tube	0.95	1.84	$<10^{-6}$

ML MSE: Mean Squared Error of Machine Learning Models
F. MSE: Mean Squared Error of Formula-based Methods
p-val.: p-value for hypothesis testing

To further verify the statistical strength of machine learning models over formulas, we reviewed the p-values across all comparisons. Each ML model tested—namely RandomForest, Support Vector Machine (SVM), Neural

Networks (NN), and ElasticNet—demonstrated statistically significant improvements over traditional formula-based methods. This is particularly noteworthy for the ElasticNet model, which also showed enhanced accuracy with MSEs markedly lower than those obtained by formula-based predictions, confirming the robust numerical advancements offered by these ML models.

In summary, our results demonstrate that machine learning models, especially SVM and ElasticNet, surpass the traditional formula-based methods in estimating the OTTD for pediatric patients, as indicated by statistically significant lower MSE levels. These findings underscore the machine learning models' potential to better guide clinical care by improving tracheal tube placement accuracy in pediatric patients.

Discussion

The accurate placement of the endotracheal tube in pediatric patients during mechanical ventilation is paramount to ensure patient safety and prevent serious complications due to the unique physiological characteristics of this population, such as shorter tracheal length [1]. Traditional methods, such as chest X-rays and formula-based calculations according to age and height, have been routinely used. However, the radiographic methods bear the burden of time consumption and radiation exposure, and both methods are hampered by limited precision, contributing to significant tube misplacement rates [2, 3]. This highlights a critical need for improved methods for OTTD determination, which was the impetus for this study.

Our study utilized refined machine-learning (ML) models to predict the OTTD and compared these models' performance for OTTD prediction with traditional, formula-based approaches. After the careful optimization of our models' hyperparameters, we discovered that the ML models significantly reduced the mean squared errors and improved the predictive accuracy notably over that of the traditional models, with SVM and ElasticNet models showing particular promise. This aligns with a trend of ML applications in healthcare enhancing the precision and efficacy in patient care, consistent with the results of similar studies [9].

However, our study was not without limitations. It utilized a single-center, retrospective design, limiting the ability to generalize our findings widely. Ideally, future works are needed to examine these models' utility across different populations and clinical settings to prove their efficacy further and confirm these initial findings. Although our models have been

optimized, their performance could vary with different data streams, which may necessitate the development of methods that can improvise model retuning in an automated manner.

Looking to the future, our study indicates the potential transformation in routine clinical practice that these ML models could herald. The possibility of integrating such ML models directly into the protocols of patient care, particularly in critical care settings where precision is crucial, could lead to significant improvements not only in patient outcomes but also in cost-effectiveness by reducing the dependence on radiographic confirmation and the associated logistical burden.

In summary, while previous OTTD determination methods continue to provide valuable service, this study suggests that a new era of ML-enhanced clinical care could be approaching. Our findings are particularly significant in the context of the prior literature that expressed the need for improved methods [9]. Machine learning models, specifically SVM and ElasticNet, have shown superior performance, which underscores the potential for ML in enhancing critical care by increasing the accuracy of tracheal tube placement. However, prospective and multi-center studies are warranted to confirm these findings, optimally tailor these ML models for disparate clinical scenarios, and evaluate their incorporation into routine clinical practice.

Methods

Data Source

The dataset utilized for this study comprised patient-level data, focusing on pediatric patients aged 0 to 7 years old who required mechanical ventilation following surgery. The dataset included demographic and physiological measures essential for estimating the optimal tracheal tube depth (OTTD). The target variable in this dataset was the OTTD as determined by the gold standard chest X-ray, a process currently burdened by time consumption and the need for radiation exposure.

Data Preprocessing

A formula-based approach was initially applied to the data as part of preprocessing to calculate expected OTTD values. This involved the assignment of OTTD values according to defined age categories and through a ledger of calculations based on patients' height and the internal diameter of the tracheal tube. The data were then prepared for analysis by extracting rel-

evant features, namely sex, age, height, and weight, excluding the internal tube diameter from the machine learning model features in compliance with scientific parameters. The cohort was randomly splitted into a training set and a test set, maintaining a test proportion to enable model validation while avoiding data leakage.

Data Analysis

A suite of machine-learning algorithms was leveraged for the prediction of OTTD. The machine-learning models selected for the study encompassed a range that covered ensemble methods, support vector regression, neural networks, and regularization-based methods to capture non-linear and complex relationships within data. An essential phase of model training involved hyperparameter tuning, where a grid search approach was employed to identify the optimal model specifications with the highest predictive power. The predictive accuracy of the machine-learning models was statistically evaluated and compared to that of the formula-based methods using the mean squared error, a measure of prediction accuracy. Moreover, the differences in predictive errors between machine-learning predictions and formula-based assertions were rigorously analyzed through paired sample t-tests to validate the superiority of machine-learning techniques over traditional methods of OTTD estimation. The study yielded quantifiable measures of performance, documenting both the predictive accuracies and the statistical significance of the observed differences.

Code Availability

Custom code used to perform the data preprocessing and analysis, as well as the raw code outputs, are provided in Supplementary Methods.

References

[1] M. Kollef, Edward J. Legare, and M. Damiano. Endotracheal tube misplacement: Incidence, risk factors, and impact of a quality improvement program. *Southern Medical Journal*, 87:248–254, 1994.

[2] T. Cook, Gene W Lee, and J. Nolan. The proseal laryngeal mask airway: a review of the literature. *Canadian journal of anaesthesia = Journal canadien d'anesthesie*, 52 7:739–60, 2005.

[3] B. Kerrey, G. Geis, A. Quinn, R. Hornung, and R. Ruddy. A prospective comparison of diaphragmatic ultrasound and chest radiography to determine endotracheal tube position in a pediatric emergency department. *Pediatrics*, 123:e1039 – e1044, 2009.

[4] V. Rajajee, J. Fletcher, L. Rochlen, and T. Jacobs. Real-time ultrasound-guided percutaneous dilatational tracheostomy: a feasibility study. *Critical Care*, 15:R67 – R67, 2011.

[5] E. Mariano, C. Ramamoorthy, L. Chu, Michael I. Chen, and G. Hammer. A comparison of three methods for estimating appropriate tracheal tube depth in children. *Pediatric Anesthesia*, 15, 2005.

[6] M. Weiss, A. Gerber, and A. Dullenkopf. Appropriate placement of intubation depth marks in a new cuffed paediatric tracheal tube. *British journal of anaesthesia*, 94 1:80–7, 2005.

[7] S. A. Ingelse, H. Wiegers, J. Calis, J. V. van Woensel, and R. A. Bem. Early fluid overload prolongs mechanical ventilation in children with viral-lower respiratory tract disease*. *Pediatric Critical Care Medicine*, 18:e106e111, 2017.

[8] P. Gupta, M. Robertson, B. Beam, J. Gossett, M. Schmitz, C. Carroll, J. Edwards, J. Fortenberry, and W. Butt. Relationship of ecmo duration with outcomes after pediatric cardiac surgery: a multi-institutional analysis. *Minerva anestesiologica*, 81 6:619–27, 2015.

[9] Ming Xia, Chenyu Jin, Shuang Cao, Bei Pei, Jie Wang, Tianyi Xu, and Hong Jiang. Development and validation of a machine-learning model for prediction of hypoxemia after extubation in intensive care units. *Annals of Translational Medicine*, 10, 2022.

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.

Goal: Create and hyperparameter optimize different machine-
    learning models to predict OTTD, and to evaluate their
    performance compared to formula-based methods.

Dataset: For this study, we curated a dataset for creating,
    hyperparameter tuning, and evaluating machine-learning
    models to predict OTTD.

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:

```
import pandas as pd

# Load the dataset
data = pd.read_csv("tracheal_tube_insertion.csv")

# Define functions which will generate various parts of the
    ↪ summary text
def data_size(df):
    return "# Data Size\n" + f"Number of Rows: {df.shape[0]} \
        ↪ nNumber of Columns: {df.shape[1]}\n"

def summary_stats(df):
    desc = df.describe(include='all').transpose()
    desc.index.name = 'Column'
    return "# Summary Statistics\n" + desc.to_string() + "\n"

def categorical_vars(df):
    cat_vars = df.select_dtypes(include=['object', 'bool', '
        ↪ category'])
    if cat_vars.empty:
        return "# Categorical Variables\nNot Applicable\n"
    else:
        text = "# Categorical Variables\n"
        for col in cat_vars:
```

```
            text += df[col].value_counts().head().to_frame().
                ↪ rename({col: 'value_counts'}, axis=1).
                ↪ to_string(header=False)
            text += '\n'
        return text

def missing_values(df):
    num_missing = df.isnull().sum()
    nonzero_missing = num_missing[num_missing != 0].rename('
        ↪ num_missing')
    if nonzero_missing.empty:
        return "# Missing Values\nNot Applicable\n"
    else:
        return "# Missing Values\n" + nonzero_missing.to_string
            ↪ () + "\n"

# Other custom summary functions can be added here...

# Now put it all together
with open('data_exploration.txt', 'w') as f:
    f.write(data_size(data))
    f.write(summary_stats(data))
    f.write(categorical_vars(data))
    f.write(missing_values(data))
    # If any other summaries were defined, their functions
        ↪ could be called here
```

B.2 Code Description

The code performs data exploration on a given dataset. It starts by loading the dataset from a CSV file.

The first function, `data_size`, calculates the number of rows and columns in the dataset and returns it as part of the summary text.

The next function, `summary_stats`, generates summary statistics for each column in the dataset using the `describe` function. It includes statistics such as count, mean, standard deviation, minimum, quartiles, and maximum. These statistics provide insights into the distribution and basic characteristics of the data.

The `categorical_vars` function identifies any categorical variables in the dataset and generates frequency counts of the top values for each categorical variable. This allows for a quick overview of the categorical data in terms of their distribution.

The `missing_values` function calculates the number of missing values for each column in the dataset and returns the columns with missing values

along with the respective counts. This helps to identify any missing data that may require further investigation or imputation.

The code then writes the generated summary text to a "data_exploration.txt" file. This file will contain the following sections of information: - Data Size: the number of rows and columns in the dataset. - Summary Statistics: descriptive statistics for each column in the dataset. - Categorical Variables: summary of the top values for each categorical variable. - Missing Values: columns with missing values and the respective counts.

The output file serves as a documentation of the initial exploration of the dataset, providing a concise summary of its characteristics. This information is valuable for understanding the data and making informed decisions regarding data preprocessing and further analysis.

B.3 Code Output

data_exploration.txt

```
\# Data Size
Number of Rows: 969
Number of Columns: 6
\# Summary Statistics
              count    mean   std  min  25\%  50\%  75\%   max
Column
tube            969   3.694 0.5681  2.5  3.5  3.5    4     6
sex             969  0.5387 0.4988    0    0    1    1     1
age\_c          969   0.758   1.44    0    0    0    1     7
ht              969      66  19.08 29.5   52 61.5   76 135.4
wt              969   7.127  4.774 0.57 3.56  5.7  9.5    31
tube\_depth\_G   969   10.19  1.766  5.9    9  9.8 11.2  19.2
\# Categorical Variables
Not Applicable
\# Missing Values
Not Applicable
```

C Data Analysis

C.1 Code

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

```
# IMPORT
import pandas as pd
import numpy as np
from scipy.stats import ttest_rel
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import train_test_split
import pickle

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

# DATASET PREPARATIONS
df.loc[df['age_c'] < 0.5, 'depth_formula_age'] = 9
df.loc[(df['age_c'] >= 0.5) & (df['age_c'] < 1),'
    ↪ depth_formula_age'] = 10
df.loc[(df['age_c'] >= 1) & (df['age_c'] < 2),'
    ↪ depth_formula_age'] = 11
df.loc[df['age_c'] >= 2,'depth_formula_age'] = 12 + (df['age_c'
    ↪ ] * 0.5)
df['depth_formula_height'] = (df['ht'] / 10) + 5
df['depth_formula_tube'] = df['tube'] * 3

# DESCRIPTIVE STATISTICS
## Table 0: "Descriptive statistics of sex, age, height, weight
    ↪  stratified by sex"
df0 = df[['sex','age_c','ht','wt']].agg(['mean','std','count'])
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# Split data into features and target
X = df[['sex', 'age_c', 'ht', 'wt']]
Y = df['tube_depth_G']

# Split data into training and test sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    ↪ test_size=0.2, random_state=1)

# ANALYSIS
models = [RandomForestRegressor(random_state=1),
          SVR(kernel='linear'),
          MLPRegressor(max_iter=500),
          ElasticNet()]

params_list = [{"max_depth": range(1, 4), "n_estimators": [10,
    ↪ 30, 50]},
               {'C': [0.1, 1, 10], 'epsilon': [0.1, 0.2,
                   ↪ 0.3]},
               {'alpha': [0.0001, 0.001, 0.01], 'learning_rate
                   ↪ ': ['constant']},
```

```
                {'alpha': [0.1, 1, 2], 'l1_ratio': [0.1, 0.5,
                    ↪ 0.7]}]

model_names = ['RandomForest', 'SVM', 'NN', 'ElasticNet']
formulae = ['depth_formula_age', 'depth_formula_height', '
    ↪ depth_formula_tube']

## Table 1: "Mean Squared Error and p-value for each Machine
    ↪ Learning Model compared with each formula-based method"
df1 = pd.DataFrame(columns=['MSE ML Model', 'MSE Formula', 'p-
    ↪ value'])

for idx, model in enumerate(models):
    grid = GridSearchCV(model, params_list[idx], cv=5, verbose
        ↪ =0)
    grid.fit(X_train, Y_train)
    best_model = grid.best_estimator_
    predictions = best_model.predict(X_test)
    mse_ml = ((Y_test - predictions) ** 2).mean()

    for formula in formulae:
        formula_predictions = df.loc[X_test.index, formula]
        if formula_predictions.isnull().values.any():
            formula_predictions.fillna(formula_predictions.mean
                ↪ (), inplace=True)
        mse_formula = ((Y_test - formula_predictions) ** 2).
            ↪ mean()
        p_value = ttest_rel((Y_test - predictions) ** 2, (
            ↪ Y_test - formula_predictions) ** 2).pvalue
        row_index = model_names[idx] + ' vs ' + formula
        df1.loc[row_index] = [mse_ml, mse_formula, p_value]

df1 = df1.rename_axis('ML Model vs Formula')
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': df.shape[0],
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
```

C.2 Code Description

The provided code performs the following steps to analyze the dataset and evaluate machine-learning models for predicting the Optimal Tracheal Tube Depth (OTTD).

1. Dataset Preparations: - The code prepares the dataset by creating additional columns for calculating the OTTD using formula-based methods based on patient age, height, and tube diameter.

2. Descriptive Statistics: - Table 0: Descriptive statistics of patient sex, age, height, and weight stratified by sex are calculated and saved as a pickle file (table_0.pkl).

3. Preprocessing: - The data is split into feature matrix X and target variable Y. - The data is further split into training and test sets using a 80-20 split.

4. Model Analysis: - Different machine-learning models (Random Forest, SVM, Neural Network, ElasticNet) are instantiated. - Hyperparameters for each model are defined in a params_list. - GridSearchCV is used to find the best hyperparameters for each model using 5-fold cross-validation on the training set. - The best model is selected for each model type based on the lowest mean squared error (MSE) on the test set. - Predictions are made using the best models for each model type. - For each model and formula-based OTTD method, MSE and p-value are calculated by comparing the model predictions with the formula-based predictions. - Table 1 is created, which shows the MSE and p-value for each machine-learning model compared with each formula-based method. This table is saved as a pickle file (table_1.pkl).

5. Additional Results: - The code calculates and saves additional results, including the total number of observations, in the file additional_results.pkl using pickle.

The resulting tables (table_0.pkl and table_1.pkl) and the additional results are important in evaluating the performance of different machine-learning models compared to formula-based methods for predicting OTTD.

C.3 Code Output

table_0.pkl

	sex	age_c	ht	wt
mean	0.5387	0.758	66	7.127
std	0.4988	1.44	19.08	4.774
count	969	969	969	969

table_1.pkl

	MSE ML Model	MSE Formula	p-value
ML Model vs Formula			

```
RandomForest vs depth\_formula\_age                1.12
    1.611    0.00311
RandomForest vs depth\_formula\_height             1.12
    3.187   9.26e-10
RandomForest vs depth\_formula\_tube               1.12
    1.836   0.000152
SVM vs depth\_formula\_age                        0.9373
    1.611   1.76e-05
SVM vs depth\_formula\_height                     0.9373
    3.187   4.77e-12
SVM vs depth\_formula\_tube                       0.9373
    1.836   2.27e-07
NN vs depth\_formula\_age                          1.212
    1.611     0.0279
NN vs depth\_formula\_height                       1.212
    3.187   3.62e-13
NN vs depth\_formula\_tube                         1.212
    1.836    0.00138
ElasticNet vs depth\_formula\_age                 0.9503
    1.611   3.46e-05
ElasticNet vs depth\_formula\_height              0.9503
    3.187   2.36e-11
ElasticNet vs depth\_formula\_tube                0.9503
    1.836   8.95e-07
```

additional_results.pkl

```
{
    'Total number of observations': 969,
}
```

D LaTeX Table Design

D.1 Code

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

```python
# IMPORT
import pandas as pd
from my_utils import to_latex_with_note, is_str_in_df,
    ↪ split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
    'sex': ('Sex', '0: Female, 1: Male'),
    'age_c': ('Age (Years)', None),
    'ht': ('Height (cm)', None),
    'wt': ('Weight (kg)', None),
```

```
    'depth_formula_age': ('Age-based OTTD (cm)', 'Optimal
        ↪ tracheal tube depth based on age'),
    'depth_formula_height': ('Height-based OTTD (cm)', 'Optimal
        ↪  tracheal tube depth based on height'),
    'depth_formula_tube': ('Tube-based OTTD (cm)', 'Optimal
        ↪ tracheal tube depth based on tube ID size'),
}

# TABLE 0:
df0 = pd.read_pickle('table_0.pkl')

# RENAME ROWS AND COLUMNS
mapping0 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df0, k))
abbrs_to_names0, legend0 = split_mapping(mapping0)
df0 = df0.rename(columns=abbrs_to_names0, index=abbrs_to_names0
    ↪ )

# SAVE AS LATEX:
to_latex_with_note(
    df0, 'table_0.tex',
    caption="Descriptive statistics of sex, age, height, and
        ↪ weight stratified by sex.",
    label='table:DescriptiveStatistics',
    legend=legend0
)


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

# RENAME ROWS AND COLUMNS
mapping1 = {
    'MSE ML Model': ('ML MSE', 'Mean Squared Error of Machine
        ↪ Learning Models'),
    'MSE Formula': ('F. MSE', 'Mean Squared Error of Formula-
        ↪ based Methods'),
    'p-value': ('p-val.', 'p-value for hypothesis testing'),
}

# Modify df1 Index
new_index = [i.replace('vs', 'Vs F').replace('_', ' ') for i in
    ↪  df1.index] # Shorten comparison labels and replace "_"
    ↪ with " "
df1.rename(index=dict(zip(df1.index, new_index)), inplace=True)
abbrs_to_names1, legend1 = split_mapping(mapping1)
df1.rename(columns=abbrs_to_names1, inplace=True)

# SAVE AS LATEX:
```

```
to_latex_with_note(
    df1, 'table_1.tex',
    caption="Comparison of Mean Squared Error (MSE) and p-value
        ↪  for ML models and formulas.",
    label='table:MLEComparison',
    legend=legend1
)
```

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`.
    """

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_0.tex

```
\% This latex table was generated from: `table\_0.pkl`
\begin{table}[h]
\caption{Descriptive statistics of sex, age, height, and weight
    stratified by sex.}
\label{table:DescriptiveStatistics}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lrrrr}
\toprule
 \& Sex \& Age (Years) \& Height (cm) \& Weight (kg) \\
\midrule
\textbf{mean} \& 0.539 \& 0.758 \& 66 \& 7.13 \\
\textbf{std} \& 0.499 \& 1.44 \& 19.1 \& 4.77 \\
\textbf{count} \& 969 \& 969 \& 969 \& 969 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Sex}: 0: Female, 1: Male
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\% This latex table was generated from: `table\_1.pkl`
\begin{table}[h]
\caption{Comparison of Mean Squared Error (MSE) and p-value for
    ML models and formulas.}
\label{table:MLEComparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lrrl}
\toprule
 \& ML MSE \& F. MSE \& p-val. \\
ML Model vs Formula \&  \&  \&  \\
\midrule
\textbf{RandomForest Vs F depth formula age} \& 1.12 \& 1.61 \&
    0.00311 \\
\textbf{RandomForest Vs F depth formula height} \& 1.12 \& 3.19
    \& \$$<$\$1e-06 \\
\textbf{RandomForest Vs F depth formula tube} \& 1.12 \& 1.84
    \& 0.000152 \\
\textbf{SVM Vs F depth formula age} \& 0.937 \& 1.61 \& 1.76e
    -05 \\
\textbf{SVM Vs F depth formula height} \& 0.937 \& 3.19 \& \$$<
    $\$1e-06 \\
```

```
\textbf{SVM Vs F depth formula tube} \& 0.937 \& 1.84 \& \$$<$\
    $1e-06 \\
\textbf{NN Vs F depth formula age} \& 1.21 \& 1.61 \& 0.0279 \\
\textbf{NN Vs F depth formula height} \& 1.21 \& 3.19 \& \$$<$\
    $1e-06 \\
\textbf{NN Vs F depth formula tube} \& 1.21 \& 1.84 \& 0.00138
    \\
\textbf{ElasticNet Vs F depth formula age} \& 0.95 \& 1.61 \&
    3.46e-05 \\
\textbf{ElasticNet Vs F depth formula height} \& 0.95 \& 3.19
    \& \$$<$\$1e-06 \\
\textbf{ElasticNet Vs F depth formula tube} \& 0.95 \& 1.84 \&
    \$$<$\$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{ML MSE}: Mean Squared Error of Machine Learning
    Models
\item \textbf{F. MSE}: Mean Squared Error of Formula-based
    Methods
\item \textbf{p-val.}: p-value for hypothesis testing
\end{tablenotes}
\end{threeparttable}
\end{table}
```

E Calculation Notes

- (1 - 0.539) = 0.461

  Percentage calculation for female patients