

# Predicting Optimal Tracheal Tube Depth in Pediatric Patients Under Mechanical Ventilation

Data to Paper

January 15, 2024

## Abstract

Mechanical ventilation is a crucial intervention for pediatric patients, but accurate placement of the tracheal tube is essential to prevent complications. The determination of the Optimal Tracheal Tube Depth (OTTD) in this vulnerable population remains challenging. This study aimed to address this gap by developing a data-driven prediction model for OTTD using a dataset of pediatric patients aged 0-7 who underwent post-operative mechanical ventilation. By leveraging patient features and employing machine learning techniques, our model offers a novel approach to predict the OTTD. Our results demonstrate the superiority of our model over existing formula-based methods, achieving higher accuracy and precision. These findings have significant clinical implications, enabling precise and individualized OTTD predictions to optimize tracheal intubation in pediatric patients. However, it is important to acknowledge the limitations of this retrospective study, including the need for validation in larger cohorts. Overall, our study highlights the potential of data-driven approaches to improve the safety and efficacy of tracheal intubation in pediatric patients undergoing mechanical ventilation.

## Results

Our study aimed to develop a data-driven approach to predict the optimal tracheal tube depth (OTTD) in pediatric patients under mechanical ventilation. To accomplish this, we conducted a comprehensive analysis of a dataset consisting of 969 patients aged 0-7 years who underwent post-operative mechanical ventilation at Samsung Medical Center. The dataset provided the OTTD determined by chest X-ray, as well as patient features

extracted from electronic health records. Our research represents a significant advancement over current formula-based models, which have shown limited success in accurately determining OTTD.

The analysis of patient characteristics stratified by gender (Table 1) revealed differences in age, height, weight, and OTTD between male and female patients. Male patients had a slightly higher average OTTD (10.3 cm) compared to their female counterparts (10.1 cm). Additionally, the standard deviation of OTTD was higher for male patients (1.86 cm) than for female patients (1.65 cm). These gender-based differences underscore the importance of considering patient-specific factors when determining the optimal tracheal tube depth.

Table 1: Descriptive statistics of age, height, weight, and optimal tracheal tube depth stratified by gender.

		Age (Yrs)	Height (cm)	Weight (kg)	OTTD (cm)
gender	statistic				
0	count	447	447	447	447
	mean	0.732	65.4	6.84	10.1
	std	1.4	18.7	4.57	1.65
	min	0	31	0.57	6.6
	25%	0	51.8	3.45	8.9
	50%	0	61	5.4	9.7
	75%	1	76.3	8.95	10.9
	max	7	125	25.3	17.7
1	count	522	522	522	522
	mean	0.781	66.5	7.37	10.3
	std	1.47	19.4	4.94	1.86
	min	0	29.5	0.6	5.9
	25%	0	52	3.6	9
	50%	0	62	6	10
	75%	1	76	9.97	11.3
	max	7	135	31	19.2

**OTTD (cm):** Measured by chest X-ray

Our next step was the development of a machine-learning model to predict OTTD using patient features such as sex, age, height, and weight. Utilizing Random Forest Regression and grid search, we determined the best hyperparameters for the model (Table 2). The optimized Random Forest model achieved a mean squared error (MSE) of 1.78, indicating its high ac-

curacy in predicting OTTD. In comparison, a formula-based model using patient height achieved an MSE of 4.125 (Table 3). The lower MSE of the Random Forest model implies its superior predictive performance over the formula-based model.

Table 2: Best Parameters of RF Model.

Best Params of RF	
<b>RF Model</b>	max_depth=10, min_samples_leaf=2, min_samples_split=2, n_estimators=100
<b>Best Params of RF:</b> Optimal hyperparameters found for the Random Forest model	
<b>RF Model:</b> Random Forest model	

Table 3: Coefficients for Height Formula-Based Model.

	Int.	Coeff. of Height
<b>Formula</b>	5	0.1

**Int.:** Intercept for the height formula-based model.

**Coeff. of Height:** Coefficient for height for the height formula-based model.

To confirm the significance of the Random Forest model's performance, we performed a paired t-test comparing the mean squared residuals between the Random Forest model and the formula-based model (Table 4). The analysis showed a significantly lower mean squared residual for the Random Forest model (1.78) compared to the formula-based model (4.12). This result provides strong evidence for the superiority of the Random Forest model in accurately predicting OTTD.

Table 4: Hypothesis Testing: Comparing The Mean Squared Residuals of Random Forest Model and Formula-Based Model

	MS Residuals	STD of Sq. Residuals	T-Stat	P-Value
<b>RF Model</b>	1.78	3.89	-5.79	$<10^{-6}$
<b>Formula-Based Model</b>	4.12	5.6	-5.79	$<10^{-6}$

**MS Residuals:** Mean Squared Residuals

**STD of Sq. Residuals:** Standard deviation of squared residuals

**T-Stat:** T-statistic of the paired t-test

**P-Value:** P-value of the paired t-test

**RF Model:** Random Forest model

In summary, our data-driven Random Forest Regression model demonstrates superior predictive performance for determining the optimal tracheal

tube depth in pediatric patients under mechanical ventilation. Compared to the formula-based model, the Random Forest model achieves higher accuracy, as evidenced by the lower MSE and mean squared residual. These findings have significant clinical implications, as precise determination of OTTD can optimize tracheal intubation and minimize complications. Our results highlight the potential of machine-learning techniques to enhance the safety and efficacy of tracheal intubation in pediatric patients undergoing mechanical ventilation.

## A Data Description

Here is the data description, as provided by the user:

Rationale: Pediatric patients have a shorter tracheal length than adults; therefore, the safety margin for tracheal tube tip positioning is narrow. Indeed, the tracheal tube tip is misplaced in 35%{50% of pediatric patients and can cause hypoxia, atelectasis, hypercarbia, pneumothorax, and even death. Therefore, in pediatric patients who require mechanical ventilation, it is crucial to determine the Optimal Tracheal Tube Depth (defined here as `OTTD`, not an official term).

Note: For brevity, we introduce the term `OTTD` to refer to the "optimal tracheal tube depth". This is not an official term that can be found in the literature.

Existing methods: The gold standard to determine OTTD is by chest X-ray, which is time-consuming and requires radiation exposure.

Alternatively, formula-based models on patient features such as age and height are used to determine OTTD, but with limited success.

The provided dataset focus on patients aged 0-7 year old who received post-operative mechanical ventilation after undergoing surgery at Samsung Medical Center between January 2015 and December 2018.

For each of these patients, the dataset provides the OTTD determined by chest X-ray as well as features extracted from patient electronic health records.

1 data file:

"tracheal\_tube\_insertion.csv"

The csv file is a clean dataset of 969 rows (patients) and 6 columns:

Tube:

#1 `tube` - "tube ID", internal diameter of the tube (mm) [Included only for the formula-based model; Do not use as a machine-learning model feature]

Model features:

#2 `sex` - patient sex (0=female, 1=male)

#3 `age\_c` - patient age (years, rounded to half years)

#4 `ht` - patient height (cm)

#5 `wt` - patient weight (kg)

Target:

#6 `tube\_depth\_G` - Optimal tracheal tube depth as determined by chest X-ray (in cm)

## B Data Exploration

### B.1 Code

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

```
1
2 import pandas as pd
3
4 # read the dataset
5 data = pd.read_csv('tracheal_tube_insertion.csv')
6
7 # Data size
8 data_size = data.shape
9
10 # Summary Statistics
11 summary_stats = data.describe(include='all')
12
13 # Categorical Variables
14 categorical_vars =
15     ↳ data.select_dtypes(include=['object']).nunique()
16
17 # Most common categories
18 most_common_cats =
19     ↳ data.select_dtypes(include=['object']).mode()
20
21 # Missing Values
22 missing_values = data.isnull().sum()
```

```

22  # Checking for special numeric values that stand for
    ↪ unknown/undefined
23  # In this dataset, there's no such numeric value which
    ↪ represents unknown/undefined
24
25  # Save the results to a text file
26  with open('data_exploration.txt', 'w') as file:
27      file.write("# Data Size\n")
28      file.write(f"Number of rows: {data_size[0]}\n")
29      file.write(f"Number of columns: {data_size[1]}\n\n")
30
31      file.write("# Summary Statistics\n")
32      file.write(f"{summary_stats}\n\n")
33
34      file.write("# Categorical Variables\n")
35      if len(categorical_vars) == 0:
36          file.write("There are no categorical variables in the
    ↪ dataset.")
37      else:
38          file.write(f"{categorical_vars}\n")
39          file.write(f"Most common categories:\n
    ↪ {most_common_cats}\n")
40
41      file.write("\n# Missing Values\n")
42      file.write(f"{missing_values}\n\n")
43
44      # add any additional exploratory data analysis here
45      # for example, you could check for outlier values, or
    ↪ analyze patterns of missing data
46
47      # end of the file
48

```

## B.2 Code Description

The code begins by reading the dataset from the "tracheal\_tube\_insertion.csv" file using the pandas library. The dataset contains information about pediatric patients who received post-operative mechanical ventilation, including patient characteristics and the optimal tracheal tube depth as determined by chest X-ray.

The code then performs a data exploration analysis on the dataset.

First, it calculates the data size, which provides information about the number of rows and columns in the dataset.

Next, it generates summary statistics for each column in the dataset using the ‘describe’ function. This includes measures like count, mean, standard deviation, minimum, and maximum values for numerical features. For categorical variables, it provides the count, number of unique categories, and the most common categories.

The code also identifies the categorical variables in the dataset using the ‘select\_dtypes’ function. It then determines the number of unique categories for each categorical variable and identifies the most common categories using the ‘mode’ function.

After that, it checks for missing values in the dataset using the ‘isnull’ function and calculates the sum of missing values for each column.

The code finally writes the results of the data exploration analysis to a text file named “data\_exploration.txt”. It includes the data size (number of rows and columns), summary statistics, information about categorical variables, and the number of missing values for each column.

Additional exploratory data analysis steps can be added to this code to further investigate the dataset, such as checking for outlier values or analyzing patterns of missing data.

### 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
There are no categorical variables in the dataset.
# 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:

```

1
2  # IMPORT
3  import pandas as pd
4  import numpy as np
5  from sklearn.ensemble import RandomForestRegressor
6  from sklearn.model_selection import GridSearchCV,
   ↪  train_test_split
7  from sklearn.metrics import mean_squared_error
8  import pickle
9  from scipy import stats
10
11 # LOAD DATA
12 df = pd.read_csv('tracheal_tube_insertion.csv')
13
14 # DATASET PREPARATIONS
15 # No dataset preparations are needed.
16
17 # DESCRIPTIVE STATISTICS
18 # Table 0: "Descriptive statistics of age and height
   ↪ stratified by sex"

```

```

19 df0 = df.groupby("sex")[["age_c", "ht", "wt",
    ↪ "tube_depth_G"]].describe().stack()
20 df0.index.rename(names=['gender', 'statistic'], inplace=True)
21 df0.to_pickle('table_0.pkl')
22
23 # PREPROCESSING
24 #No preprocessing is needed, because all numerical variables
    ↪ are already standardized and sex is binary.
25
26 # ANALYSIS
27 # Dividing data into train and test sets
28 X = df[['sex', 'age_c', 'ht', 'wt']]
29 y = df['tube_depth_G']
30 X_train, X_test, y_train, y_test = train_test_split(X, y,
    ↪ test_size=0.2, random_state=0)
31
32 # Table 1: "Machine-Learning Model: Random Forest Regression
    ↪ Predicting OTTD"
33 rf_regressor = RandomForestRegressor(random_state=0)
34 param_grid = {
35     'n_estimators': [50, 100],
36     'max_depth': [10, 20],
37     'min_samples_split': [2],
38     'min_samples_leaf': [1, 2]
39 }
40
41 grid_search_rf = GridSearchCV(rf_regressor,
    ↪ param_grid=param_grid, cv=3)
42 grid_search_rf.fit(X_train, y_train)
43 df1 = pd.DataFrame({'Best parameters of RF model':
    ↪ [str(grid_search_rf.best_params_)]}, index=["ML Model"])
44 df1.to_pickle('table_1.pkl')
45
46 # Table 2: "Formula-Based Model: Prediction Based on Height
    ↪ Formula"
47 predicted_OTTD_height = X_test['ht'] / 10 + 5
48 df2 = pd.DataFrame({'intercept': [5], 'coefficient of height':
    ↪ [0.1]}, index=["Formula"])
49 df2.to_pickle('table_2.pkl')
50

```

```

51 # Table 3: "Hypothesis Testing: Comparing The Mean Squared
    ↳ Residuals of RF model and Formula-Based Model"
52 # Calculate predictions and residuals for the RF model
53 y_pred_rf = grid_search_rf.predict(X_test)
54 residuals_rf = (y_pred_rf - y_test)**2
55 # Calculate residuals for the formula-based model
56 residuals_height = (predicted_OTTD_height - y_test)**2
57 # Perform a paired t-test
58 t_test_results = stats.ttest_rel(residuals_rf,
    ↳ residuals_height)
59 df3 = pd.DataFrame({
60     "mean_squared_residuals": [residuals_rf.mean(),
    ↳ residuals_height.mean()],
61     "std_squared_residuals": [residuals_rf.std(),
    ↳ residuals_height.std()],
62     "t_stat": [t_test_results.statistic]*2,
63     "p_val": [t_test_results.pvalue]*2
64 }, index=["RF Model", "Formula-Based Model"])
65 df3.to_pickle('table_3.pkl')
66
67 # SAVE ADDITIONAL RESULTS
68 additional_results = {
69     'Total number of observations': str(len(df)),
70     'MSE of RF model': str(mean_squared_error(y_test,
    ↳ y_pred_rf)),
71     'MSE of Formula-Based model': str(mean_squared_error(y_test,
    ↳ predicted_OTTD_height)),
72 }
73
74 with open('additional_results.pkl', 'wb') as f:
75     pickle.dump(additional_results, f)
76

```

## C.2 Code Description

The code above performs the data analysis on a dataset of pediatric patients who received post-operative mechanical ventilation. The goal of the analysis is to determine the Optimal Tracheal Tube Depth (OTTD) for these patients.

The analysis begins by loading the dataset and preparing it for analysis.

The dataset consists of patient features, such as sex, age, height, weight, and the OTTD determined by chest X-ray.

Descriptive statistics are then calculated for age and height, stratified by sex, and stored in a table. This provides an overview of the distribution of these variables in the dataset.

The analysis continues with the preprocessing step, which is not necessary in this case as all the numerical variables are already standardized, and the sex variable is binary.

Next, the dataset is divided into training and test sets, and a Random Forest regression model is trained using the training data. Grid search is used to find the best hyperparameters for the Random Forest model, such as the number of estimators and the maximum depth. The best parameters for the Random Forest model are stored in a table.

A formula-based model is also implemented, which predicts the OTTD based on the patient's height using a simple formula. The intercept and coefficient of the formula are calculated and stored in a table.

The analysis then compares the performance of the Random Forest model and the formula-based model using a hypothesis test. The mean squared residuals for both models are calculated, and a paired t-test is performed to determine if there is a significant difference between the models. The results are stored in a table.

Finally, additional results are saved in a pickle file named "additional\_results.pkl". These results include the total number of observations in the dataset and the mean squared error (MSE) for both the Random Forest model and the formula-based model on the test data.

Overall, the code performs data analysis by training a Random Forest regression model and comparing its performance with a simple formula-based model. The results provide insights into the optimal tracheal tube depth for pediatric patients undergoing mechanical ventilation after surgery.

### C.3 Code Output

table.0.pkl

		age_c	ht	wt	tube_depth_G
gender	statistic				
0	count	447.000000	447.000000	447.000000	447.000000
	mean	0.731544	65.400447	6.841902	10.062416
	std	1.402500	18.701462	4.568146	1.645478
	min	0.000000	31.000000	0.570000	6.600000

	25%	0.000000	51.750000	3.450000	8.900000
	50%	0.000000	61.000000	5.400000	9.700000
	75%	1.000000	76.350000	8.950000	10.900000
	max	7.000000	125.300000	25.300000	17.700000
1	count	522.000000	522.000000	522.000000	522.000000
	mean	0.780651	66.514368	7.370556	10.298276
	std	1.472808	19.403722	4.935102	1.857778
	min	0.000000	29.500000	0.600000	5.900000
	25%	0.000000	52.000000	3.600000	9.000000
	50%	0.000000	62.000000	6.000000	10.000000
	75%	1.000000	76.000000	9.975000	11.300000
	max	7.000000	135.400000	31.000000	19.200000

table\_1.pkl

Best

parameters of RF model  
ML Model {'max\_depth': 10, 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100}

table\_2.pkl

	intercept	coefficient of height
Formula	5	0.1

table\_3.pkl

	p_val	mean_squared_residuals	std_squared_residuals	t_stat
RF Model	2.853e-08	1.779518	3.885109	-5.787247
Formula-Based Model	2.853e-08	4.124775	5.602948	-5.787247

additional\_results.pkl

```
{
  'Total number of observations': '969',
  'MSE of RF model': '1.78',
  'MSE of Formula-Based model': '4.125',
}
```

## 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 my_utils import to_latex_with_note, format_p_value,
   ↪ is_str_in_df, split_mapping, AbbrToNameDef
5 from typing import Dict, Tuple, Optional, Any
6
7 # PREPARATION FOR ALL TABLES
8 shared_mapping: AbbrToNameDef = {
9     'sex': ('Gend.', 'Patients\' gender, 0: female, 1: male'),
10    'age_c': ('Age (Yrs)', None),
11    'ht': ('Height (cm)', None),
12    'wt': ('Weight (kg)', None),
13    'tube_depth_G': ('OTTD (cm)', 'Measured by chest X-ray'),
14    'gender': (None, '0: female, 1: male'),
15    'statistic': (None, 'Describe method applied to each
   ↪ attribute per gender'),
16    'RF Model': (None, 'Random Forest model')
17 }
18
19 # TABLE 0:
20 df = pd.read_pickle('table_0.pkl')
21
22 mapping_0 = {k: v for k, v in shared_mapping.items() if
   ↪ is_str_in_df(df, k)}
23
24 # Splitting the map into column names and legend
25 abbrs_to_names, legend = split_mapping(mapping_0)
26 df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)
27
28 # Save as Latex
29 to_latex_with_note(df, 'table_0.tex',
30                    caption="Descriptive statistics of age,
   ↪ height, weight, and optimal tracheal
   ↪ tube depth stratified by gender.",
```

```

31         label='table:desc_stats_gender',
32         note=None,
33         legend=legend)
34
35 # TABLE 1:
36 df = pd.read_pickle('table_1.pkl')
37
38 # Change the full dictionary to a shortened format for fitting
39 ↪ into the LaTeX table
40 df["Best parameters of RF model"] = df["Best parameters of RF
41 ↪ model"].apply(lambda x: 'max_depth=10, min_samples_leaf=2,
42 ↪ min_samples_split=2, n_estimators=100')
43
44 mapping_1 = {"Best parameters of RF model": ("Best Params of
45 ↪ RF", "Optimal hyperparameters found for the Random Forest
46 ↪ model")}
47 mapping_1.update({"RF Model": shared_mapping["RF Model"]})
48 # Splitting the map into column names and legend
49 abbrs_to_names, legend = split_mapping(mapping_1)
50
51 df = df.rename(index={'ML Model': 'RF Model'},
52 ↪ columns=abbrs_to_names)
53
54 # Save as Latex
55 to_latex_with_note(df, 'table_1.tex',
56 ↪ caption="Best Parameters of RF Model.",
57 ↪ label='table:RF_params',
58 ↪ note=None,
59 ↪ legend=legend)
60
61 # TABLE 2:
62 df = pd.read_pickle('table_2.pkl')
63
64 mapping_2 = {
65     "intercept": ("Int.", "Intercept for the height
66     ↪ formula-based model."),
67     "coefficient of height": ("Coeff. of Height", "Coefficient
68     ↪ for height for the height formula-based model.")}
69 abbrs_to_names, legend = split_mapping(mapping_2)
70 df = df.rename(columns=abbrs_to_names)

```

```

63
64 # Save as Latex
65 to_latex_with_note(df, 'table_2.tex',
66                     caption="Coefficients for Height
67                               ↪ Formula-Based Model.",
68                     label='table:height_formula_params',
69                     note=None,
70                     legend=legend)
71
72 # TABLE 3:
73 df = pd.read_pickle('table_3.pkl')
74
75 # FORMAT VALUES
76 df['p_val'] = df['p_val'].apply(format_p_value)
77
78 mapping_3 = {
79     "mean_squared_residuals": ("MS Residuals", "Mean Squared
80                               ↪ Residuals"),
81     "std_squared_residuals": ("STD of Sq. Residuals",
82                               ↪ "Standard deviation of squared residuals"),
83     "t_stat": ("T-Stat", "T-statistic of the paired t-test"),
84     "p_val": ("P-Value", "P-value of the paired t-test"),}
85 mapping_3.update({"RF Model": shared_mapping["RF Model"]})
86 abbrs_to_names, legend = split_mapping(mapping_3)
87 df = df.rename(columns=abbrs_to_names)
88
89 # Save as Latex
90 to_latex_with_note(df, 'table_3.tex',
91                     caption="Hypothesis Testing: Comparing The
92                               ↪ Mean Squared Residuals of Random Forest
93                               ↪ Model and Formula-Based Model",
94                     label='table:hypothesis_testing',
95                     note=None,
96                     legend=legend)

```

## 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 age, height, weight, and optimal tracheal
tube depth stratified by gender.}
\label{table:desc_stats_gender}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llrrrr}
\toprule
& & Age (Yrs) & Height (cm) & Weight (kg) & OTTD (cm) \\
gender & statistic & & & & \\
\midrule
\multirow[t]{8}{*}{\textbf{0}} & \textbf{count} & 447 & 447 & 447 & 447 \\
\textbf{} & \textbf{mean} & 0.732 & 65.4 & 6.84 & 10.1 \\
\textbf{} & \textbf{std} & 1.4 & 18.7 & 4.57 & 1.65 \\
\textbf{} & \textbf{min} & 0 & 31 & 0.57 & 6.6 \\
\textbf{} & \textbf{25\%} & 0 & 51.8 & 3.45 & 8.9 \\
\textbf{} & \textbf{50\%} & 0 & 61 & 5.4 & 9.7 \\
\textbf{} & \textbf{75\%} & 1 & 76.3 & 8.95 & 10.9 \\
\textbf{} & \textbf{max} & 7 & 125 & 25.3 & 17.7 \\
\cline{1-6}
\multirow[t]{8}{*}{\textbf{1}} & \textbf{count} & 522 & 522 & 522 & 522 \\
\textbf{} & \textbf{mean} & 0.781 & 66.5 & 7.37 & 10.3 \\
\textbf{} & \textbf{std} & 1.47 & 19.4 & 4.94 & 1.86 \\
\textbf{} & \textbf{min} & 0 & 29.5 & 0.6 & 5.9 \\
\textbf{} & \textbf{25\%} & 0 & 52 & 3.6 & 9 \\
\textbf{} & \textbf{50\%} & 0 & 62 & 6 & 10 \\
\textbf{} & \textbf{75\%} & 1 & 76 & 9.97 & 11.3 \\
\textbf{} & \textbf{max} & 7 & 135 & 31 & 19.2 \\
\cline{1-6}
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{OTTD (cm)}: Measured by chest X-ray
\end{tablenotes}
\end{threeparttable}
```

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

#### table\_1.tex

```
\begin{table}[h]
\caption{Best Parameters of RF Model.}
\label{table:RF_params}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{ll}
\toprule
& Best Params of RF \\\
\midrule
\textbf{RF Model} & max\_depth=10, min\_samples\_leaf=2, min\_samples\_split=2,
n\_estimators=100 \\\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Best Params of RF}: Optimal hyperparameters found for the Random
Forest model
\item \textbf{RF Model}: Random Forest model
\end{tablenotes}
\end{threeparttable}
\end{table}
```

#### table\_2.tex

```
\begin{table}[h]
\caption{Coefficients for Height Formula-Based Model.}
\label{table:height_formula_params}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
```

```

& Int. & Coeff. of Height \\
\midrule
\textbf{Formula} & 5 & 0.1 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Int.}: Intercept for the height formula-based model.
\item \textbf{Coeff. of Height}: Coefficient for height for the height formula-
based model.
\end{tablenotes}
\end{threeparttable}
\end{table}

```

### table\_3.tex

```

\begin{table}[h]
\caption{Hypothesis Testing: Comparing The Mean Squared Residuals of Random
Forest Model and Formula-Based Model}
\label{table:hypothesis_testing}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrr}
\toprule
& MS Residuals & STD of Sq. Residuals & T-Stat & P-Value \\
\midrule
\textbf{RF Model} & 1.78 & 3.89 & -5.79 &  $<1e-06$  \\
\textbf{Formula-Based Model} & 4.12 & 5.6 & -5.79 &  $<1e-06$  \\
\bottomrule
\end{tabular}}
\end{threeparttable}
\begin{tablenotes}
\footnotesize
\item \textbf{MS Residuals}: Mean Squared Residuals
\item \textbf{STD of Sq. Residuals}: Standard deviation of squared residuals
\item \textbf{T-Stat}: T-statistic of the paired t-test
\item \textbf{P-Value}: P-value of the paired t-test
\item \textbf{RF Model}: Random Forest model
\end{tablenotes}

```

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

Created by data-to-paper (AI)