

Research Article

Length of Stay Prediction Model of Indoor Patients Based on Light Gradient Boosting Machine

Xiangrui Zeng 

School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia

Correspondence should be addressed to Xiangrui Zeng; xaviertsang@student.usm.my

Received 31 May 2022; Revised 31 July 2022; Accepted 10 August 2022; Published 30 August 2022

Academic Editor: Najib Ben Aoun

Copyright © 2022 Xiangrui Zeng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The influx of hospital patients has become common in recent years. Hospital management departments need to redeploy healthcare resources to meet the massive medical needs of patients. In this process, the hospital length of stay (LOS) of different patients is a crucial reference to the management department. Therefore, building a model to predict LOS is of great significance. Five machine learning (ML) algorithms named Lasso regression (LR), ridge regression (RR), random forest regression (RFR), light gradient boosting machine (LightGBM), and extreme gradient boosting regression (XGBR) and six feature encoding methods named label encoding, count encoding, one-hot encoding, target encoding, leave-one-out encoding, and the proposed encoding method are used to construct the regression prediction model. The Scikit-Learn toolbox on the Python platform builds the prediction model. The input is the dataset named Hospital Inpatient Discharges (SPARCS De-Identified) 2017 with 2343569 instances provided by the New York State Department of Health verify the model after removing 2.2% of the missing data, and the model ultimately uses mean squared error (MSE) and coefficient of determination (R^2) as the performance measurement. The results show that the model with the LightGBM algorithm and the proposed encoding method has the best R^2 (96.0%) and MSE score (2.231).

1. Introduction

Globally, due to the pandemic and population changes, hospital inpatient departments are becoming more and more likely to face the influx and congestion of patients [1, 2] and hospitals in anticipation of the need to redeploy healthcare resources to meet the massive medical requirements of patients [3]. The LOS indicates the number of days between admission and discharge, and it can often affect the admission plan of emergency patients [4] or whether there is the possibility of transfer [5]. Moreover, when technical means can reduce the long duration of LOS, the consumption of healthcare resources would also be reduced to some extent [6]. However, the inpatient department does not know when existing patients will leave the hospital in most cases. If hospitals could accurately predict LOS, they could implement and improve healthcare resource management correctly [7, 8]. Therefore, this study tries to establish an ML model using the information about the

diagnosis, treatment, service, and cost of individual patients to predict LOS.

In the study, five ML algorithms (LR, RR, RFR, XGBR, and LightGBM) and six feature encoding methods (label encoding, count encoding, one-hot encoding, target encoding, leave-one-out encoding, and the proposed encoding method) were used and compared during the model building. The rest of the study is organized as follows: Section 2 reviews some related studies on LOS. Section 3 introduces the dataset used in this study and each step of the proposed framework in detail. Section 4 presents the experimental results and then discusses them. Section 5 draws the conclusions and direction of future work.

2. Related Works

Several researchers have conducted related studies on predicting hospital LOS. Some of them discrete the LOS value that transforms the regression problem into classification.

For example, Bacchi et al. [9] proposed an artificial neural network (ANN)-based prediction model for predicting the LOS in stroke patients. The objective is to predict whether the LOS was less than 8 days. And they finally achieved 0.62 and 0.66 area under curve (AUC) values on the inner and outer validation sets. Similarly, Daghistani et al. [10] converted the LOS values into three classes (<3 days, 3–5 days, and >5 days) and then used information gain (IG) to select features. They compared Random Forest (RF), Bayesian Network (BN), Support Vector Machine (SVM), and ANN technology for LOS prediction. The final RF model outperformed all other models (sensitivity (0.80), accuracy (0.80), and AUROC (0.94)). Furthermore, Zheng et al. [11] compared two discrete methods that are two (1–3 days and ≥ 4 days) and three (1–3 days, 4–8 days, and ≥ 9 days) classes. Six ML algorithms were applied to the model to make comparative predictions and finally obtained the best accuracy score (ACC) of 0.7689 and 0.6594 in the training and test sets, respectively. Furthermore, Ling et al. [12] used the RF algorithm and general medical characteristics to predict LOS in patients in the intensive care unit (ICU), and the AUC value of the optimal model is 0.86.

The limitation of classification-type studies is their generally poor performance and difficulty in guiding long-term LOS (e.g., LOS ≥ 10 days) prediction due to the small number of classes. Models of this discrete type are unrealistic to deploy and not recommended when hospitals hope to predict the LOS precisely (e.g., ± 1 day).

Data balancing techniques can improve model performance in predicting LOS. For example, Naemi et al. [13] proposed a multistage data processing method. The method first used k-nearest neighbors (KNN), decision tree (DT), gradient boosting (GB), Bayesian ridge (BR), Gaussian process (GP), and RF for missing value imputation and then used SMOTE to overcome data skewness. After these steps, the model used DT to predict the hours of stay. It ended up with an R^2 score of 0.729. Alsinglawi et al. [14] constructed a LOS prediction framework for lung cancer patients using RF and oversampling techniques (SMOTE and ADASYN). The framework gets an AUC score of 100% on the MIMIC-III dataset.

The datasets used in the above two studies have been artificially altered. Even though model performance is good on synthetic balanced data, it often does not perform well on unbalanced data. As a result, models using data balancing techniques are difficult to deploy because data tend to be biased in real life.

According to historical data, regression is the method that occupies the majority proportion of LOS prediction [15]. For example, Siddiqua et al. [16] used multiple linear regression (MLR), decision tree regression (DTR), LR, RR, XGBR, and RFR techniques to predict LOS. They found that RFR is the best model that achieved the 5 MSE and 0.92 R^2 scores. In another study, Abbas et al. [17] established a model based on multilayer perceptron (MLP) to predict LOS for total knee arthroplasty. The model uses medical data such as a patient's white blood cell count and type of anesthesia, and finally received 0.715 and 0.690 MSE scores on training and test sets, respectively. Zhong et al. [18] compared three

models based on backpropagation neural network (BPNN), support vector regression (SVR), and principal component regression (PCR). The best experimental result of the 1.5254 mean absolute error (MAE) score is on the PCR model. The study of Kolchun et al. [19] established a prediction model for passenger LOS after a motor vehicle collision. After comparing various ML methods, the MAE of LOS prediction by the neural network (NN) algorithm reaches 2.23.

Previous regression models have two limitations. First, some models are built on specific or posthospital physical examination data, so they lack generality. The other model built on datasets with high versatility is insufficient in performance ($R^2 < 0.95$). Based on the deficiencies of the three model types, this study attempts to propose a model that does not use artificially synthesized data and excels in both generality (e.g., using prehospital diagnosis results) and performance.

3. Materials and Methods

3.1. Data Description. The study used the Hospital Inpatient Discharges (SPARCS De-Identified) 2017 dataset provided by the New York State Department of Health [20]. This dataset uses the Open Database License (ODbL 1.0), which grants anyone to use the dataset for the duration of any applicable copyright and Database Rights. These rights explicitly include commercial use and do not exclude any field of endeavor [21]. The dataset contains 2343569 instances with 34 features that de-identify the detailed information of patient characteristics, diagnosis, treatment, services, and costs. The “Length of Stay” in the dataset is the target feature, while the purpose of the proposed model is to predict it by others. Table 1 shows the description of the features of the dataset.

3.2. The Proposed Framework. This study uses a few steps to build a complete application model. First, the raw data use visualization to analyze the internal relationship, and then the data are preprocessed for duplicates, missingness, and meaningless information. The third step determines whether each feature of the dataset positively affects the target, and the model only needs the positive partial. Then the six encodings make the information in the dataset unusable into a usable form.

The above steps make modifications to the raw data. Then the model divides data into a training set and a test set in a 99:1 ratio, where the training set uses a 10-fold cross-validation technique to improve model reliability. After the five ML algorithms have trained the model, MSE and R^2 will judge the model performance to support analysis.

Figure 1 presents the framework proposed in this study. And this framework fully expresses the methodology used to construct the model for this study. The following sections explain each step of the framework in detail.

3.3. Data Visualization Analysis. The study leverages visualization techniques to analyze datasets and find relationships between independent and dependent features. The

TABLE 1: Feature description of the dataset.

Feature name	Type	Description	
Hospital Service Area		Describe the location of the hospital	
Hospital County			
Permanent Facility ID		Hospital service information	
Facility Name			
Operating Certificate Number		Patient diagnostic information	
Type of Admission			
CCS Diagnosis Code			
CCS Diagnosis Description			
CCS Procedure Code			
CCS Procedure Description			
APR DRG Code			
APR DRG Description	Radom categorical		
APR MDC Code			
APR MDC Description			
APR Severity of Illness Code			
APR Severity of Illness Description			
Payment Typology 1		Patient cost information	
Payment Typology 2			
Payment Typology 3			
Zip Code - 3 digits			
Race		Patient personal information	
Ethnicity			
Patient Disposition			
Birth Weight		Patient diagnostic information	
Age Group	Ordered categorical		
APR Risk of Mortality			
APR Medical Surgical Description			
Gender	Three classes		Patient personal information
Discharge Year	One class		Patient treatment information
Abortion Edit Indicator			Patient service information
Emergency Department Indicator	Binary classes		Patient service information
Length of Stay			Target feature
Total Charges	Continuous		Patient cost information
Total Costs			

results produced by visualization methods are usually easily understandable by people who are not necessarily knowledgeable about ML [22]. Table 1 shows that the dataset has two or three class categorical, ordered categorical, random categorical, and continuous features, in which the target “Length of Stay” belongs to the continual type.

Figure 2 shows the density distribution of the target feature. It belongs to the long-tail distribution with an average value of 5.38. Therefore, all analysis methods that assume the normal distribution are less suitable for this study. Figure 3 shows the proportion of LOS in different categories of patients in the three features of “Gender,” “APR Medical Surgical Description,” and “Emergency Department Indicator.” The results showed that the LOS of the female patients was longer than that of the male patients but was uniform. In the middle of the figure, the LOS of medical inpatients accounts for about three-quarters of the total, of which the type is much longer than surgical inpatients. And the LOS of emergency patients is about twice that of non-emergency patients, showing that the condition of emergency patients is more ill and needs a longer recovery time. Figure 4 shows the density distribution of two continuous features, and the trend is similar to Figure 2. This figure

demonstrates that the two features correlate with the target. Finally, Figure 5 shows the LOS of two ordered categorical features, which shows that the younger the age and the higher the disease mortality rate, the shorter the LOS.

3.4. Data Preprocessing. Outliers and missing values during model building would affect the model performance [23], then data preprocessing is crucial. Among the 34 features, the missing value of “Payment Type 2” is missing completely at random (MCAR) and is missing at random (MAR) [24] in “Payment Typology 3” and “Birth Weight.” The proportion of their missing values is about 37.5%, 74.1%, and 90.3% [25]. Hence the process removed three features directly. The remaining dataset also needs to remove about 2% of the instances that still contain MCAR or MAR, as well as 20 samples with the value of “Unknown” in the “Gender” feature. It is worth mentioning that all eigenvalues “120+” were uniformly changed to “120” for the convenience of calculation in the feature “Length of Stay.” Finally, there are 2304296 instances with 31 features left in the dataset that the preprocessing process deleted 2.2% instances.

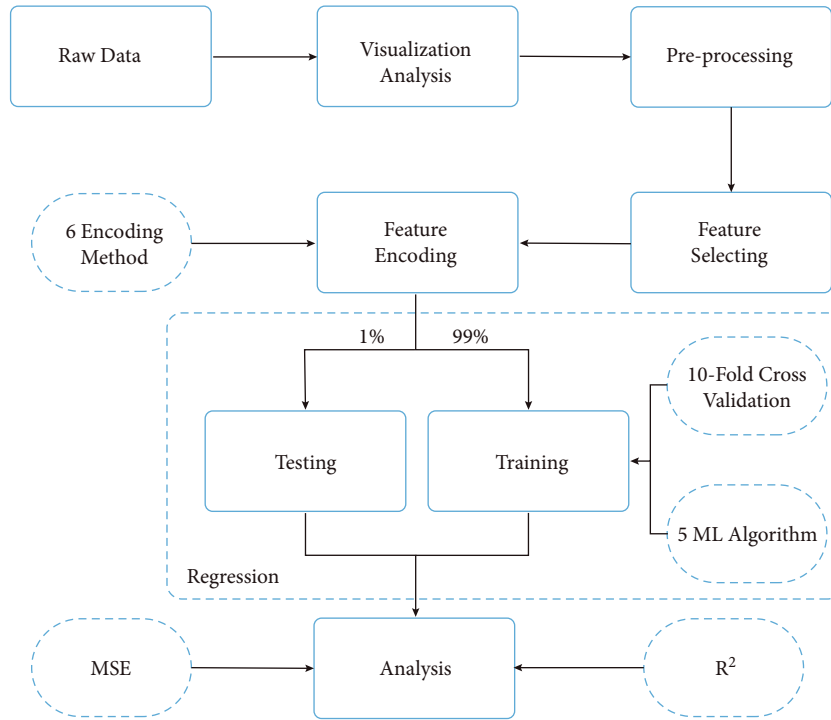


FIGURE 1: Visualization of the proposed framework.

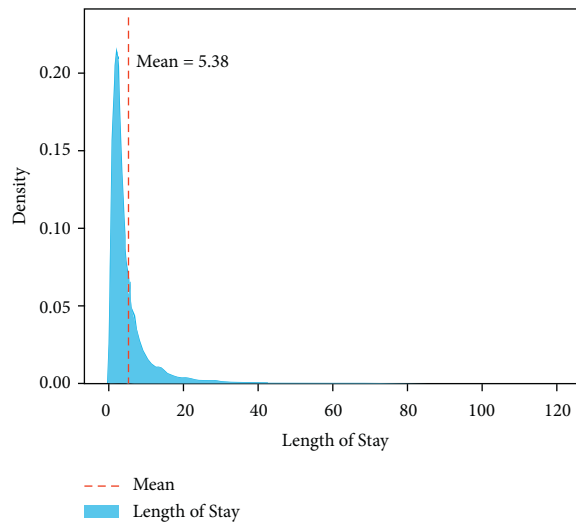


FIGURE 2: Density plot of length of stay.

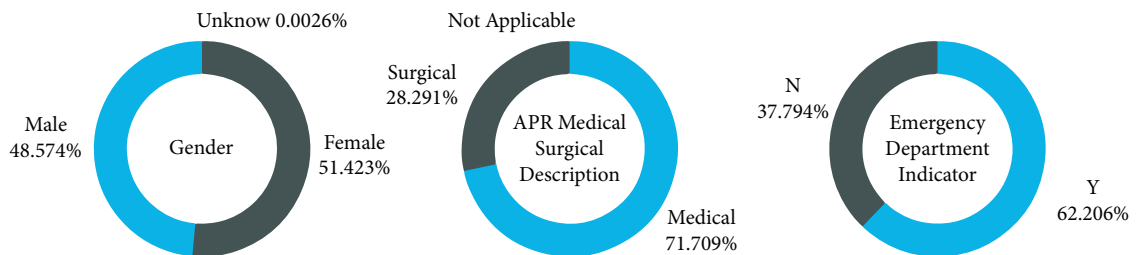


FIGURE 3: Length of stay distribution of two (three) class features.

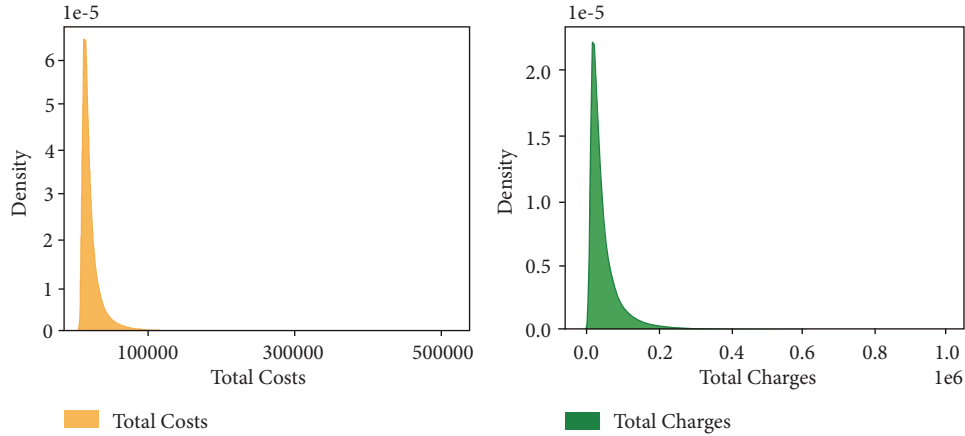


FIGURE 4: Density plot of Total Costs and Total Charges.

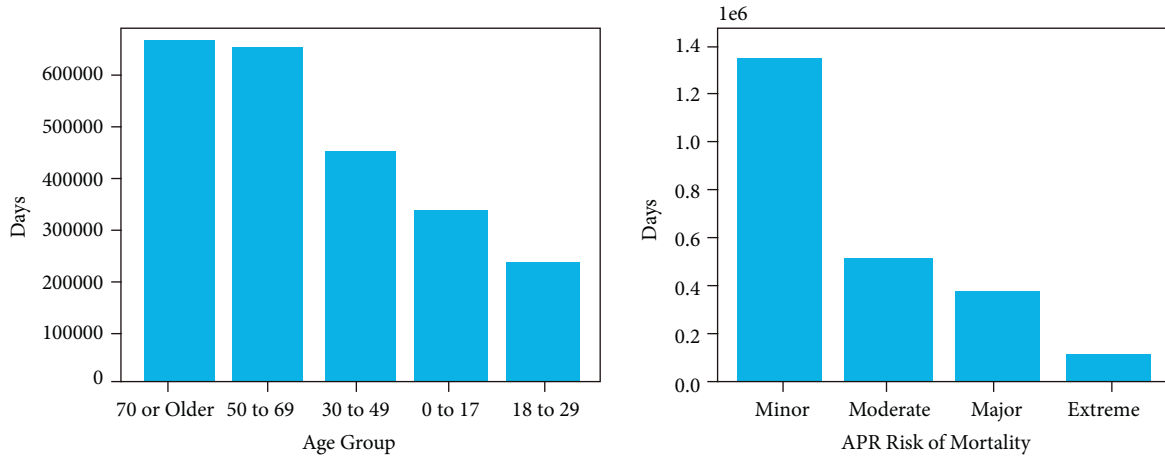


FIGURE 5: Length of stay in Age Group and APR Risk of Mortality.

3.5. Feature Selection. Among the remaining 31 features of the dataset after preprocessing, the five features of “CCS Diagnosis Description,” “CCS Procedure Description,” “APR DRG Description,” “APR MDC Description,” and “APR Severity of Illness Description” are different representations of the same information as the five features of “CCS Diagnosis Code,” “CCS Procedure Code,” “APR DRG Code,” “APR MDC Code,” and “APR Severity of Illness Code,” respectively, which are meaningless to the model and therefore deleted.

Among the 24 remaining features of the dataset, the “Length of Stay” is the continuous target feature. And others are divided into four types (Binary, Ordered Categorical, Random Categorical, and Continuous). Regarding the correlation between them and the target feature, it is necessary to use various techniques to investigate.

3.5.1. Binary Features. The point-biserial correlation is the value of Pearson’s product-moment correlation when one of the variables is dichotomous and the other variable is metric [26]. The calculation formula of the point-biserial correlation coefficient is as follows:

$$\gamma_{pb} = \frac{\bar{Y}_1 - \bar{Y}_0}{\bar{s}_n} \sqrt{\frac{n_1 n_0}{n(n-1)}}, \quad (1)$$

where n_1 and \bar{Y}_1 represent the frequency of the binary feature $X = 1$ and the mean of the corresponding target feature, respectively. n_0 and \bar{Y}_0 represent the frequency of the binary variable $X = 0$ and the mean of the corresponding target feature, respectively. And the \bar{s}_n in the denominator represents the standard deviation of the target feature [26]. Finally, the closer the absolute value of γ_{pb} is to 1, the higher the correlation between features.

3.5.2. Ordered Categorical Features. The correlation of ordered categorical features with continuous features requires first converting the latter to the former type. The two most popular measures of association for this feature type are Kendall’s tau and Spearman’s rho [27]. This study uses the Spearman coefficient for correlation analysis, and the general idea is as follows:

The method of Spearman first converts the string data $X^i = [X_1^i, X_2^i, \dots, X_n^i]$ in the ordered categorical features into numerical grade data $x^i = [x_1^i, x_2^i, \dots, x_n^i]$. Table 2 shows the

TABLE 2: Numerical conversion details of ordered categorical features.

Feature name	Original string data	Converted numerical data
Age Group	“0 to 17”	0
	“18 to 29”	1
	“30 to 49”	2
	“50 to 69”	3
	“70 or older”	4
APR Risk of Mortality	“Minor”	0
	“Moderate”	1
	“Major”	2
	“Extreme”	3

conversion detail. And the data $y = [y_1 = 1, y_2 = 2, \dots, y_n = 120]$ in the target feature are used directly without modification. The method then uses formula (2) [28, 29] to calculate the correlation between x and y . The absolute value of the result $\rho_{x,y}$ is between 0 and 1, and the closer to 1, the more correlated the features are.

$$\rho_{x,y} = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2} \sqrt{\sum(y - \bar{y})^2}} \quad (2)$$

3.5.3. Random Categorical Features. Since the target feature does not satisfy the normal distribution (Figure 2), it is suitable to use the Kruskal–Wallis test to calculate the correlation with the target feature. The Kruskal–Wallis test is a nonparametric statistical test that assesses the differences among three or more independently sampled groups on a single, nonnormally distributed continuous feature [30]. The basic idea is as follows:

The Kruskal–Wallis test first arranges the eigenvalues in ascending order, then finds their rank R_i , and examines whether there is a significant difference in the mean μ_i of the ranks of each eigenvalue. $H_0: \mu_1 = \mu_2 = \dots = \mu_k$ is the null hypothesis, and the alternative hypothesis H_1 is that at least two μ_i are not equal. The calculation formula [31] is as follows:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k n_i \bar{R}_i^2 - 3(N+1). \quad (3)$$

Through H in the above formula, the Kruskal–Wallis test can query the critical value table to get the corresponding P value. If the P value is below the significance level, there is a correlation between the features. And this study sets the threshold at 0.01.

3.5.4. Continuous Features. Since the target feature is not normally distributed (Figure 2), its correlation with continuous features needs to be judged by the Spearman correlation coefficient [32].

Finally, Table 3 summarizes the correlation between each feature and the target feature. And the results show the model could keep all features.

3.6. Feature Encoding. All categorical attributes of the dataset are represented by strings, while machine learning algorithms can only calculate numerical eigenvalues. Hence these features need to be rerecorded into numbers.

3.6.1. Label Encoding. In the label encoding method, the eigenvalues of each categorical feature are first sorted by frequency from small to large and then are assigned a value from 0 to $N-1$ in order (N indicates how many different eigenvalues the feature has). Even if there is no relationship between the eigenvalues before encoded, the algorithm would regard them according to the size of the values. Table 4 shows a sample of this method on one particular feature.

3.6.2. Count Encoding. Count encoding is a method that uses the frequency of eigenvalues as labels. In this method, the frequency of one feature will replace the value of this feature. And different eigenvalues may be encoded into the same number. When the frequency of categorical features correlates with the target feature, this method has positive significance for model training.

3.6.3. One-Hot Encoding. When a feature with M unique values, one-hot encoding will create M corresponding new features, where the new value (1 or 0) indicates whether the instance has the represented original categorical value. Table 5 demonstrates the principle of one-hot encoding. However, there are too many eigenvalues in discrete features in the dataset. If all features use one-hot encoding, more than 1500 new will be generated and will be too sparse. Hence, only features with a small number of unique values will use this method.

3.6.4. Target Encoding. Target encoding is a preprocessing scheme for high-cardinality categorical features based on a well-established statistical approach to models (empirical Bayes). It is a method based not only on the independent eigenvalues but also on the corresponding dependent feature [33]. This method depends on the distribution of dependent features, but the feature dimension remains unchanged after encoding, and its calculation formulas (4) and (5) [33] are as follows:

$$S_i = \lambda(n_i) \frac{\sum_{k \in L_i} Y_k}{n_i} + (1 - \lambda(n_i)) \frac{\sum_{k=1}^{N_{TR}} Y_k}{n_{TR}}, \quad (4)$$

$$\lambda(n_i) = \frac{1}{1 + e^{-(n-k/f)}}. \quad (5)$$

In formula (4), $\sum_{k \in L_i} Y_k$ represents the sum of the corresponding target feature’s values when the categorical eigenvalue is i . Its denominator n_i represents the frequency of categorical eigenvalue i . And the $\sum_{k=1}^{N_{TR}} Y_k$ on the right side of the formula represents the sum of the values of the target feature in the training set.

TABLE 3: Feature importance and selection results.

Feature name	Correlation or P value	Retain feature
Gender	0.053	
APR Medical Surgical Description	0.043	
Emergency Department Indicator	0.051	
Hospital County	$P = 2.2e - 16$	
Operating Certificate Number	$P = 2.2e - 16$	
Permanent Facility Id	$P = 2.2e - 16$	
Facility Name	$P = 2.2e - 16$	
Zip Code - 3 digits	$P = 2.2e - 16$	
CCS Diagnosis Code	$P = 2.2e - 16$	
CCS Procedure Code	$P = 2.2e - 16$	
APR DRG Code	$P = 2.2e - 16$	
APR MDC Code	$P = 2.2e - 16$	Yes
Patient Disposition	$P = 2.2e - 16$	
Hospital Service Area	$P = 2.2e - 16$	
Ethnicity	$P = 2.2e - 16$	
Type of Admission	$P = 2.2e - 16$	
Payment Typology 1	$P = 2.2e - 16$	
Race	$P = 2.2e - 16$	
APR Severity of Illness Code	$P = 2.2e - 16$	
APR Risk of Mortality	0.376	
Age Group	0.228	
Total Charges	0.602	
Total Costs	0.651	

TABLE 4: Label encoding example for the “Patient Disposition” feature.

Raw eigenvalues	Sorted eigenvalues	Numerical eigenvalues
Home or self-care	Short-term hospital	0
Skilled nursing home	Expired	1
Court/law enforcement	Hospice - medical facility	2
Skilled nursing home	Home or self-care	3
Court/law enforcement	Home or self-care	3
Short-term hospital	Skilled nursing home	4
Court/law enforcement	Skilled nursing home	4
Home or self-care	Court/law enforcement	5
Expired	Court/law enforcement	5
Hospice - medical facility	Court/law enforcement	5

TABLE 5: One-hot encoding example for the “Race” feature.

Raw feature	New features after numerical encoding		
	Race-White	Race-Black/African American	Race-other race
White	1	0	0
White	1	0	0
White	1	0	0
Black/African American	0	1	0
Black/African American	0	1	0
Black/African American	0	1	0
Black/African American	0	1	0
Other race	0	0	1
White	1	0	0

k in formula (5) represents the minimum times the eigenvalue must appear in the calculated feature. And λ represents the smoothing coefficient that the higher the value, the stronger the regularization of the formula.

3.6.5. Leave-One-Out Encoding. The leave-one-out encoding method uses the same principle and formula as target encoding. But to reduce the influence of outliers, when calculating the encoding value of an instance, the program will ignore the current and only use the remaining for target encoding.

3.6.6. Proposed Encoding Method. One-hot encoding method can obtain the information of categorical features well, but it will lead to sparse data. The other methods do not have the problem of sparsity but will lose a lot of data information. This study attempts to balance model performance and data dimensionality, thus combining two encodings to form a new method. Table 6 shows the encoding adopted for each feature.

3.7. Comparative Algorithms

3.7.1. Lasso Regression. LR is to fit the dataset $D = \{(x_1^1, x_2^1, \dots, x_m^1, y_1), (x_1^2, x_2^2, \dots, x_m^2, y_2), \dots, (x_1^n, x_2^n, \dots, x_m^n, y_n)\}$ (m represents the number of features and n indicates the number of instances) with a linear function (6) and minimize the cost function (7) [34], where $f(x)$ represents the predicted values and y_i is true values. The purpose of the operation is to find a solution (W, b) that minimizes $J(w)$. LR imposes constraints on the model parameters (i.e., adds a penalty $\lambda\|w_j\|$ to the loss function) that shrink the regression coefficients to zero [35]. For example, if a feature highly correlates with the target, LR will select it and then shrink others uncorrelated with zero and exclude them from the model. This approach reduces bias and improves the accuracy of linear regression models.

$$\begin{aligned} f(x) &= W^T \cdot X + b \\ &= \sum_{j=0}^n w_j x_j^{(i)}, \end{aligned} \quad (6)$$

$$J(w) = \sum_{i=1}^m \left(y_i - \sum_{j=0}^n w_j x_j^{(i)} \right)^2 + \lambda \|w_j\|. \quad (7)$$

By calculating the partial derivative concerning w of the residual on the left side and the penalty term on the right side of formula (7) could obtain formulas (8) and (9).

$$\frac{\partial}{\partial w_k} \text{RSS}(w) = -2 \sum_{i=1}^m \left(x_k^{(i)} y^{(i)} - x_k^{(i)} \sum_{j=0, j \neq k}^n w_j x_j^{(i)} - w_k x_k^{(i)^2} \right), \quad (8)$$

$$\frac{\partial}{\partial w_k} R(w) = \begin{cases} -\lambda, & w_k < 0, \\ [-\lambda, \lambda], & w_k = 0, \\ \lambda, & w_k > 0, \end{cases} \quad (9)$$

where $x_k^{(i)} \sum_{j=0, j \neq k}^n w_j x_j^{(i)} - w_k x_k^{(i)^2} = x_k^{(i)} \sum_{j=0}^n w_j x_j^{(i)}$. Set $P_k = \sum_{i=1}^m [x_k^{(i)} (y^{(i)} - \sum_{j=0, j \neq k}^n w_j x_j^{(i)})]$ and $Z_k = \sum_{i=1}^m x_k^{(i)^2}$, then combine (8) and (9) to obtain the partial derivative of (7) and solve it:

$$w_j = \begin{cases} \frac{(P_k + (\lambda/2))}{Z_k}, & P_k < -\left(\frac{\lambda}{2}\right), \\ 0, & -\left(\frac{\lambda}{2}\right) \leq P_k \leq \left(\frac{\lambda}{2}\right), \\ \frac{(P_k - (\lambda/2))}{Z_k}, & P_k > \left(\frac{\lambda}{2}\right). \end{cases} \quad (10)$$

3.7.2. Ridge Regression. RR is similar to LR and uses the linear formula (6). It obtains regression coefficients at the cost of losing some information and reducing accuracy by giving up unbiasedness. RR adds a penalty term to the loss function in standard linear regression to alleviate multicollinearity and overfitting problems [36]. Its estimates of regression coefficients tend to become too large in absolute values, and some may even have the wrong sign [37]. Formula (11) [38] is the loss function of RR, which is the penalty term added by $\lambda\|w_j\|^2$. And λ is a hyper-parameter used to control the strength of the penalty. The larger the λ , the simpler the generated model.

$$J_\beta(\beta) = \sum_{i=1}^n \left(y_i - \sum_j w_j x_j^{(i)} \right)^2 + \lambda \|w_j\|^2. \quad (11)$$

3.7.3. Random Forest Regression. RFR adopts the Bootstrap [39] technique to randomly divide the dataset D into n subsample sets $\{D_1, D_2, \dots, D_n\}$. The CART regression tree will build on these subsets and output the results, and the final RFR outputs the average of all predictions. There is no relationship between each regression tree, an increase in the number of trees does not cause the RFR to overfit the data [40]. Furthermore, RFR is insensitive to multicollinearity, and the results are robust to missing and unbalanced data [41].

The 31 features of each divided subsample set D_i are set to $A = \{A_1, A_2, \dots, A_{31}\}$. The CART algorithm first sorts the features A_j and then tries to use each interval between adjacent feature values as the segmentation point S . The set of eigenvalues on the left side of S is $R_1(A_j, S)$ and the right side is $R_2(A_j, S)$ (12). c_1 and c_2 are the mean values of the target feature corresponding to $R_1(A_j, S)$ and $R_2(A_j, S)$, respectively (13). The next step of the algorithm is to find which S can make the MSE of the feature minimum (14) and then use the segmentation point S together with the feature as the node of the tree. After the algorithm divides all features, the CART regression tree uses the average of all leaf nodes as the output (15) [42].

TABLE 6: The proposed encoding method.

Feature name	Encoding method
Gender	Label encoding
APR Medical Surgical Description	
Emergency Department Indicator	
Hospital County	
Operating Certificate Number	
Permanent Facility Id	
Facility Name	
Zip Code - 3 digits	
CCS Diagnosis Code	
CCS Procedure Code	
APR DRG Code	
APR MDC Code	
Patient Disposition	
Hospital Service Area	
Ethnicity	
Type of Admission	
Payment Typology 1	
Race	
APR Severity of Illness Code	
APR Risk of Mortality	Sort the feature values from low to high and then encode them from 0 to N-1.
Age Group	

$$R_1(A_i, S) = \{x | x^{A_i} \leq S\}, \quad (12)$$

$$R_2(A_i, S) = \{x | x^{A_i} \leq S\},$$

$$c_1 = \text{ave}(y_i | x_i \in R_1(A_i, S)), \quad (13)$$

$$c_2 = \text{ave}(y_i | x_i \in R_2(A_i, S)),$$

$$\min_{A_i, S} \left[\min_{c_1} \sum_{x_i \in R_1(A_i, S)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(A_i, S)} (y_i - c_2)^2 \right], \quad (14)$$

$$f(x) = \sum_{m=1}^{31} c_m I(x \in R_m). \quad (15)$$

3.7.4. Extreme Gradient Boosting Regression. Unlike RFR in the bagging form, XGBR is a boosting integrated ML algorithm based on the CART regression tree, which belongs to the regression implementation of extreme gradient boosting (XGBoost). It uses the second-order Taylor expansion and adds regularization to the objective function. And the algorithm adopts accurate greedy ideas in the tree generation [43]. Finally, XGBR uses the sum of the predictive values of all regression trees for the sample as the output of this sample in the system, and the definition function (16) [43] is as follows:

$$\hat{y}_i = \sum_{k=1}^K f_k(X_i), \quad f_k \in F, \quad (16)$$

where X_i is the sample feature and $f_k(X_i)$ is the prediction of the K th tree. The sum of values of all trees is the predicted value \hat{y}_i for the entire model. Since the algorithm belongs to the additive model, the predicted value of

the K th tree \hat{y}_i^k can be expressed by formula (17). Let the sum of the truth values be y_i . Formula (18) [43] summarizes the objective function.

$$\hat{y}_i^k = \hat{y}_i^{k-1} + f_k(X_i), \quad (17)$$

$$\min \left(L(\varphi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) = \sum_i l(f_k(X_i) + \hat{y}_i^{k-1}, y_i) + \sum_k \Omega(f_k) \right), \quad (18)$$

where $\sum_k \Omega(f_k) = \sum_{i=1}^{K-1} \Omega(f_i) + \Omega(f_K)$. $\sum_i l(\hat{y}_i, y_i)$ is the loss function between the predicted and true values that is MSE (11) in XGBR. Since the results of $K-1$ trees have been determined and remain unchanged when training the K th tree, $\sum_k \Omega(f_k)$ can convert to $\Omega(f_K)$. Then, the Taylor expansion can transform the objective function on the right side of formula (18) into (19).

$$\min \left(L(\varphi) = \sum_i \left[l(\hat{y}_i^{k-1}, y_i) + g_i \cdot f_k(X_i) + \frac{1}{2} h_i \cdot f_k^2(X_i) \right] + \Omega(f_K) \right), \quad (19)$$

where $\sum_i l(\hat{y}_i^{k-1}, y_i)$ is the sum of the prediction losses of the first $K-1$ trees. And it does not change when computing the K th tree and can therefore be ignored. $g_i = \partial_{\hat{y}_i^{k-1}} l(\hat{y}_i^{k-1}, y_i)$ and $h_i = \partial_{\hat{y}_i^{k-1}}^2 l(\hat{y}_i^{k-1}, y_i)$ can be treated as a constant too. $f_k(X_i)$ represents the prediction result of the K th tree, and it also indicates the leaf node position on the K th tree where the sample X_i . Here, the function $q(X_i)$ can be defined to represent the sample position in leaf nodes, and $w_{q(X_i)} = f_k(X_i)$ can express to solve the sample position. XGBoost defines $\Omega(f_K) = \gamma T + 1/2\lambda \sum_{t=1}^T (\omega_t)^2$ as the penalty function (where λ represents the penalty intensity and T is the number of leaf nodes) [43]. Formula (19) can convert to formula (20) by removing the constant term and substituting the penalty.

$$\min \left(L(\varphi) = \sum_i \left[g_i \cdot w_q(x_i) + \frac{1}{2} h_i \cdot w_q^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{t=1}^T (\omega_t)^2 = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) \cdot w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) \cdot w_j^2 \right] + \gamma T \right), \quad (20)$$

where only w_j is unknown, so the objective function becomes a typical quadratic type. XGBR adopts the CART regression tree that could fix the tree structure $q(X_i)$. At this time, the minimum solution of the function is $w_j^* = -((\sum_{i \in I_j} g_i) / (\sum_{i \in I_j} h_i + \lambda))$, substituting into formula (20) can get the objective function solution $-(1/2) \sum_{j=1}^T ((\sum_{i \in I_j} g_i)^2 / (\sum_{i \in I_j} h_i + \lambda)) + \gamma T$.

3.7.5. Light Gradient Boosting Machine. Microsoft launched an upgraded version of XGBoost named LightGBM in 2017. The LightGBM in this article uses the histogram algorithm to reduce the number of candidates' split points and the mutually Exclusive Feature Bundling (EFB) algorithm to reduce the number of features [44].

The histogram algorithm refers to discretizing continuous floating-point eigenvalues into k integers and constructing a histogram with a width of k . The algorithm counts the floating-point values within the range of the discretized values in the histogram according to the k values as an index. Then traverses the discretized values to find the optimal segmentation point. XGBoost travels all floating-point values, while LightGBM only travels k values by establishing histograms. EFB will compare and analyze the difference between features by sparse coding. When the difference between the two features is minor, it considers that there is a conflict. Otherwise, the two features will be one. EFB reduces the feature dimension through this method to speed up.

Hence LightGBM is more efficient run on the set in large-scale data. With the same performance as XGBR, LightGBM is 10x faster than train and consumes less memory [44].

3.8. Model Validation. Although the dataset has more than 2 million instances, the model is still at risk of overfitting. Secondly, the model training process is necessary to avoid information leakage caused by using the test set multiple times. Based on the above factors, the validation process divides the dataset into a training set and a test set in a 99:1 ratio. Then the training set is used for 10-fold cross-validation, and the test set checks the model performance. The entire validation process will use the training set ten times, but the test set only once.

The 10-fold cross-validation method could alleviate the overfitting and information leak [45]. The reason for choosing 10 is the estimate of prediction error is almost unbiased [46]. The 10-fold method will use different 90% training sets to train the model ten times, and the remaining measures the model performance.

3.9. Performance Measurement. The model in this study attempts to solve a regression problem, in which people usually achieve model performance measurement by

comparing the MSE and R^2 . The closer the MSE value is to 0, the smaller the gap between the predicted and the actual value. Formula (21) [47] calculates the MSE by subtracting each prediction from the truth, adding all the squared results, and dividing by the total number added.

$$MSE = \frac{1}{n} \sum (y_i - \hat{y})^2, \quad (21)$$

where y_i represents the actual value, \hat{y} represents the predicted value, and n represents the total number of squared values.

When the dimensions are different, MSE does not say much about the performance of the regression concerning the distribution of the ground truth elements. However, the R^2 score does not have the interpretability limitations of MSE and is more informative and truthful [48]. The value of the R^2 score is between $-\infty$ and 1. $R^2 = 1$ indicates the predicted values are the same as the actual values. Hence, the closer the score is to 1, the better the model performance. Formula (22) [48] defines the calculation method for R^2 .

$$\begin{aligned} R^2 &= 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 / n}{\sum_{i=1}^n (y_i - \bar{y})^2 / n} \\ &= 1 - \frac{MSE(\hat{y}, y)}{\text{Var}(y)}, \end{aligned} \quad (22)$$

where the numerator in the rightest is the MSE, and the denominator is the variance of the actual value.

4. Results and Discussion

4.1. Results

4.1.1. Model Processing. The dataset remains 2304296 instances with 53 features after preprocessing, feature selection, and feature encoding. This study builds the model using the Scikit-Learn ML toolkit on the Python platform with 8 cores and 16 GB RAM. To ensure reproducible results, all steps involving random processes set the random seed to 0.

The hyper-parameter λ in LR and RR models has the highest impact on performance. This study uses the default penalty coefficient $\lambda = 1$ in the toolkit. RFR, XGBR, and LightGBM are all tree-type models, and the hyper-parameter that most affects their performance is the number of CART regression trees (`n_estimators`). The more the number of trees, the higher the model performance may be, but the computing cost rises with it. The default `n_estimators = 100` for RFR, and to facilitate the horizontal comparison of the three models, XGBR and LightGBM refer to the same order of magnitude of fitting time (Table 7) to set `n_estimators` to 500 and 25000, respectively. In particular, the LightGBM algorithm can set the number of features discarded ratio at each iteration to prevent overfitting, which is 0.6 in this study.

TABLE 7: Model performance in this and related study.

Model	Model performance in this study					
	LR	RR	RR	RFR	XGBR	LightGBM
MSE	5.626	5.400	0.848	1.938	1.116	2.231
R^2	0.697	0.726	0.994	0.969	0.960	0.960
Hyper-parameters	Alpha (λ) = 1	Alpha (λ) = 1	n_estimators = 100	n_estimators = 500	n_estimators = 25000	feature_fraction = 0.6
One-fold fitting time	3.654s	1.653s	946.46s	900.799s	874.331s	
Model	Model performance in related study [16]					
	LR	RR	RFR	XGBR	MLP	DTR
MSE	42.58	39	0.76	5.30	39	0.002
R^2	0.31	0.37	0.987	0.914	0.37	0.999
	0.31	0.3711	0.92	0.908	0.371	0.903

4.1.2. Experimental Analysis. Table 7 shows the performance of models built with LR, RR, RFR, XGBR, and LightGBM algorithms, while the study results of Siddiqa et al. [16] are also listed side-by-side as a control. In the model of this study, the MSE (5.882) and R^2 (0.675) metrics of LR on the test set are the worst, and its training time (3.654s) is also longer than another linear algorithm RR (1.653s). The performance of the RR algorithm (MSE = 5.680 and $R^2 = 0.702$) outperforms the LR by a small margin, but the performance of both linear algorithms is far from satisfactory.

The RFR and XGBR-based models achieved MSE scores of 2.295 and 2.287, and the R^2 scores are both 0.958 on the test set, which is well behaved as ideal. Their single-fold fitting consumption is 946.465s and 900.799s, respectively. However, the LightGBM algorithm surpasses them in fitting time (874.331s), MSE (2.231), and R^2 (0.960), which performs best in the tree-type model.

Table 8 compares the performance variation of the best-performing LightGBM model in different encoding methods, where the hyper-parameter remains unchanged. In the results, label encoding (MSE = 2.248, $R^2 = 0.959$), target encoding (MSE = 2.252, $R^2 = 0.959$), and count encoding (MSE = 2.252, $R^2 = 0.959$) have similar performance, while leave-one-out (MSE = 7.777, $R^2 = 0.221$) performs the worst. And the proposed encoding (MSE = 2.231, $R^2 = 0.960$) is the best method.

4.2. Discussion. The LR ($R^2 = 0.675$) and RR ($R^2 = 0.702$) models based on linear algorithms are far from ideal, which means that the datasets used in this study tend to be nonlinear, and linear algorithms are difficult to apply in practice to the process of predicting LOS. However, the three tree-type models (RFR, XGBR, and LightGBM) performed pretty well, especially the LightGBM model. Its R^2 score of 0.960 is improved by 4.4% compared to the best-performing RFR model (5 MSE and 0.92 R^2) in the past study [16] as the control group, while the MSE score of 2.231 is a relative decrease of 55.4%. XGBR and RFR models in this study ranked second and third in performance, with 2.5% and 2.8% respective higher MSE and 0.2% lower R^2 scores relative to the best model.

Compared with the previous study [16], the encoding method in this study is a majority different. The models composed of LR, RR, RFR, and XGBR algorithms have significantly lower MSE scores (decreased by 86.3%, 85.2%, 54.1%, and 59.3% respectively) after using the proposed encoding method in this study, and R^2 scores are improved (117.7%, 89.2%, 4.1%, and 5.5% higher). The LightGBM model using the proposed encoding also reduces the MSE score by at least 0.76% compared to label encoding, count encoding, target encoding, and leave-one-out encoding and R^2 scores improved by at least 0.1%.

The model in this study can help the hospital to estimate the LOS of the patient, and the data to construct the model only need some prehospital diagnostic characteristics of the patient, thus reducing the threshold for the actual deployment and increasing the reality. In addition, the modeling

TABLE 8: The performance changes in different encoding methods.

Encoding Method	MSE		R^2	
	Training	Test	Training	Test
Label encoding	1.120	2.248	0.990	0.959
Count encoding	1.129	2.252	0.990	0.959
Target encoding	1.129	2.252	0.990	0.959
Leave-one-out encoding	0.023	7.777	0.999	0.221
Proposed encoding method	1.116	2.231	0.990	0.960

process balances the conflict between the curse of dimensionality and information retention. Even for millions of instances, the model can be trained and deployed quickly using a personal computer. However, the model performance highly correlated with the “Total Charges” and “Total Costs” feature. Where “Total Charges” can be obtained when the patient admitted to the hospital, but “Total Costs” need to be estimated from the doctor’s experience and other information about the patient. Uncertainty in the estimation results may affect model performance in reality.

5. Conclusions

The objective of this study was to construct a model to predict LOS in the hospital by exploring the prehospital diagnostic information of potential hospitalized patients. Many ML algorithms such as RFR, LR, RR, XGBR, MLP, and DTR are being investigated in recent studies for regression prediction of LOS. Eventually, the performance of the models constructed by these algorithms can hardly meet the requirements of actual deployment. Where the linear model is not suitable for predicting LOS and the tree model overfitting is obvious. This study proposed a model using one-hot encoding + label encoding combined with the LightGBM algorithm to investigate how to improve the accuracy of LOS prediction. The model is based on the 2017 dataset provided by the New York State Department of Health. The average LOS for patients in this dataset is 5.38 days, and most patients stay in the hospital for 1–5 days for minor illnesses, and more than 70% of illness types are medical. There was no significant difference in LOS between men and women, but over 50 spent more time in hospital. This study used a hybrid feature encoding approach to improve LOS prediction performance. And feature selection is also computed, which compared correlation scores to remove features that were not positive for prediction. The results of the correlation analysis showed that “Total Charges” and “Total Costs” were the features most associated with LOS. The proposed model ultimately successfully extends the results of related studies, with MSE and R^2 achieving the best scores of 2.231 and 0.960, respectively, which is much higher than the previous study. In the future, the problem of model overfitting still deserves more research to obtain higher accuracy for predicting LOS.

Data Availability

The data that support the findings of this study are publicly available at <https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/gaf8-ac33>.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

References

- [1] N. Kokudo and H. Sugiyama, "Hospital capacity during the COVID-19 pandemic," *Global Health & Medicine*, vol. 3, no. 2, pp. 56–59, 2021.
- [2] A. Chiarenza, M. Dauvrin, V. Chiesa, S. Baatout, and H. Verrept, "Supporting access to healthcare for refugees and migrants in European countries under particular migratory pressure," *BMC Health Services Research*, vol. 19, no. 1, p. 513, 2019.
- [3] V. N. Prachand, R. Milner, P. Angelos et al., "Medically necessary, time-sensitive procedures: scoring system to ethically and efficiently manage resource scarcity and provider risk during the COVID-19 pandemic," *Journal of the American College of Surgeons*, vol. 231, no. 2, pp. 281–288, 2020.
- [4] E. S. Powell, R. K. Khare, A. K. Venkatesh, B. D. Van Roo, J. G. Adams, and G. Reinhardt, "The relationship between inpatient discharge timing and emergency department boarding," *Journal of Emergency Medicine*, vol. 42, no. 2, pp. 186–196, 2012.
- [5] H. Baek, M. Cho, S. Kim, H. Hwang, M. Song, and S. Yoo, "Analysis of length of hospital stay using electronic health records: a statistical and data mining approach," *PLoS One*, vol. 13, no. 4, Article ID e0195901, 2018.
- [6] A. W. Arntstein, N. K. Rathlev, D. Neal et al., "Decreasing emergency department walkout rate and boarding hours by improving inpatient length of stay," *Western Journal of Emergency Medicine*, vol. 18, no. 6, pp. 982–992, 2017.
- [7] M.-T. Chuang, Y.-h. Hu, and C.-L. Lo, "Predicting the prolonged length of stay of general surgery patients: a supervised learning approach," *International Transactions in Operational Research*, vol. 25, no. 1, pp. 75–90, 2018.
- [8] A. T. Lamere, S. Nguyen, G. Niu, A. Olinsky, and J. Quinn, "Predicting the length of stay in hospital emergency rooms in Rhode Island," *Advances in Business and Management Forecasting*, vol. 14, pp. 35–48, 2021.
- [9] S. Bacchi, L. Oakden-Rayner, D. K. Menon et al., "Prospective and external validation of stroke discharge planning machine learning models," *Journal of Clinical Neuroscience*, vol. 96, pp. 80–84, 2022.
- [10] T. A. Daghistani, R. Elshawi, S. Sakr, A. M. Ahmed, A. Al-Thwayee, and M. H. Al-Mallah, "Predictors of in-hospital length of stay among cardiac patients: a machine learning approach," *International Journal of Cardiology*, vol. 288, pp. 140–147, 2019.
- [11] L. Zheng, J. Wang, A. Sheriff, and X. Chen, "Hospital length of stay prediction with ensemble methods in machine learning," in *Proceedings of the 2021 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pp. 1–5, Beijing, China, December 2021.
- [12] Y. Ling, Y. Chen, V. Chirikov et al., "A prediction model for length of stay in the icu among septic patients: a machine learning approach," *Value in Health*, vol. 21, no. S5, p. S5, 2018.
- [13] A. Naemi, T. Schmidt, M. Mansourvar, A. Ebrahimi, and U. K. Wiil, "Quantifying the impact of addressing data challenges in prediction of length of stay," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, p. 298, 2021.
- [14] B. Alsinglawi, O. Alshari, M. Alorjani et al., "An explainable machine learning framework for lung cancer hospital length of stay prediction," *Scientific Reports*, vol. 12, no. 1, p. 607, 2022.
- [15] V. Lequertier, T. Wang, J. Fondrevelle, V. Augusto, and A. Duclos, "Hospital length of stay prediction methods: a systematic review," *Medical Care*, vol. 59, no. 10, pp. 929–938, 2021.
- [16] A. Siddiq, S. Abbas Zilqurnain Naqvi, M. Ahsan, A. Ditta, H. Alquhayz, and M. Adnan Khan, "Robust length of stay prediction model for indoor patients," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5519–5536, 2022.
- [17] A. Abbas, J. Mosseri, J. R. Lex et al., "Machine learning using preoperative patient factors can predict duration of surgery and length of stay for total knee arthroplasty," *International Journal of Medical Informatics*, vol. 158, Article ID 104670, 2022.
- [18] H. Zhong, B. Wang, D. Wang et al., "The application of machine learning algorithms in predicting the length of stay following femoral neck fracture," *International Journal of Medical Informatics*, vol. 155, Article ID 104572, 2021.
- [19] J. P. G. Kolcun, B. Covello, J. E. Gernsback, I. Cajigas, and J. R. Jagid, "Machine learning to predict passenger mortality and hospital length of stay following motor vehicle collision," *Neurosurgical Focus*, vol. 52, no. 4, p. E12, 2022.
- [20] Health data, "Hospital Inpatient Discharges (SPARCS De-Identified) Downloadable File," 2017, <https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/gaf8-ac33>.
- [21] Open Database License, "Open Data Commons Open Database License (ODbL) v1.0," 2011, <https://opendatacommons.org/licenses/odbl/1-0/>.
- [22] A. Vellido, J. D. Martín, F. Rossi, and P. Lisboa, "Seeing is believing: the importance of visualization in real-world machine learning applications," in *Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2011*, pp. 219–226, Bruges, Belgium, April 2011.
- [23] M. L. Brown and J. F. Kros, "Data mining and the impact of missing data," *Industrial Management & Data Systems*, vol. 103, no. 8, pp. 611–621, 2003.
- [24] K. Bhaskaran and L. Smeeth, "What is the difference between missing completely at random and missing at random?" *International Journal of Epidemiology*, vol. 43, no. 4, pp. 1336–1339, 2014.
- [25] S. Bijlsma, I. Bobeldijk, E. R. Verheij et al., "Large-scale human metabolomics studies: a strategy for data (pre-) processing and validation," *Analytical Chemistry*, vol. 78, no. 2, pp. 567–574, 2006.
- [26] D. Kornbrot, *Point Biserial Correlation* Wiley StatsRef: Statistics Reference Online, Hoboken, NJ, USA, 2014.
- [27] W. Bergsma and A. Dassios, "A consistent test of independence based on a sign covariance related to Kendall's tau," *Bernoulli*, vol. 20, no. 2, pp. 1006–1028, 2014.
- [28] J. L. Myers, A. D. Well, and R. F. Lorch, *Research Design and Statistical Analysis*, Taylor & Francis, Oxfordshire, UK, 2013.
- [29] T. Cleff, *Exploratory data analysis in business and economics*, Springer International Publishing, New York, NY, USA, 2014.
- [30] P. E. McKight and J. Najab, "Kruskal-wallis test," in *The Corsini Encyclopedia of Psychology*, pp. 1–10, 2010.
- [31] W. H. Kruskal and W. A. Wallis, "Errata: use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 48, no. 264, pp. 907–911, 1953.

- [32] H. Akoglu, "User's guide to correlation coefficients," *Turkish Journal of Emergency Medicine*, vol. 18, no. 3, pp. 91–93, 2018.
- [33] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *SIGKDD Explor Newsl*, vol. 3, no. 1, pp. 27–32, 2001.
- [34] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [35] J. Ranstam and J. A. Cook, "LASSO regression," *British Journal of Surgery*, vol. 105, no. 10, p. 1348, 2018.
- [36] G. C. McDonald, "Ridge regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.
- [37] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [38] C. Saunders, A. Gammerman, and V. Vovk, "Ridge Regression Learning Algorithm in Dual Variables," *Proceedings of the Fifteenth International Conference on Machine Learning*, ACM, San Francisco, CA, USA, 1998.
- [39] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson, Eds., Springer, New York, NY, USA, 1992.
- [40] T. Hastie, R. Tibshirani, and J. Friedman, "Random forests," in *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2009.
- [41] L. Breiman, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [42] T. Hastie, R. Tibshirani, and J. Friedman, "Additive Models, Trees, and Related Methods," *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2009.
- [43] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, ACM, San Francisco, CA, USA, 2016.
- [44] G. Ke, M. Qi, F. Thomas et al., *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*, ACM, Long Beach, CA, USA, 2017.
- [45] D. Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, Amsterdam, Netherlands, 2019.
- [46] R. Simon, "Resampling strategies for model assessment and selection," in *Fundamentals of Data Mining in Genomics and Proteomics*, Springer, New York, NY, USA, 2007.
- [47] M. Hossin and S. Mn, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, 2015.
- [48] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021.