# Predicting Patient Length of Stay in Hospital

*Bhavya Arora, Elisha Shrestha, Gaurav Kumar, Komali Challa, Olushola Soyoye, and Sabrina Casas*

***Abstract*—Kaggle's Healthcare Analytics II presents the problem of predicting a patient's length of stay (LOS) in a hospital. The problem is both relevant and important to improve healthcare management efficiency in hospitals. To start off, the dataset was cleaned up by removing highly correlated variables and missing values which constituted less than 1.5% of the data. Notably, the dataset included variables with severe class imbalance. To remedy the less-than-ideal quality of the data, data augmentation was used to duplicate values in a minority class by using the SMOTE function. Two approaches were then taken to develop a model: 1) To use the SMOTE function on the clean dataset, and 2) To use the SMOTE function with a re-classified target variable, 'Stay'. 'Stay' was re-classified into three categories: low, medium, and high. Six different machine learning algorithms were then used on the two approaches to find the best model at predicting the patient's LOS by using 'accuracy' as the main performance indicator.**

## I. Introduction

It has been over three years since the World Health Organization declared the Covid-19 outbreak a global pandemic. Special attention has since been given to the healthcare system in the United States. However, according to Abraham Haileamlak, a professor of Pediatrics, the disruption to the healthcare system is not only due because of the pandemic, but also because of a system stretched past its capability [1]. For instance, in February 2023, Deidre McPhillips from CNN wrote an article titled 'The virus threat is easing, but US hospitals are still as full as ever'. The article reported that at the Massachusetts General Hospital, about 87% of all available hospital beds in the state were in use, and that the capacity strains had only slightly eased despite a decrease in the number of people being treated for a Covid-19 related illnesses [2].

Therefore, due to the ever-present threat of evolving microorganisms, such as other coronavirus variants, as well as a strained healthcare system, more work needs to be done to create a more resilient health care system. Within this article, we focus on Kaggle's Healthcare Analytics II problem of predicting a patient's length of stay (LOS) in a hospital. A patient's LOS is one critical parameter to observe and predict if one wants to improve the efficiency of hospitals' healthcare management. This helps hospitals identify patients with high LOS risk, patients who are more likely to stay hospitalized longer, at the time of admission. Once identified, patients with high LOS risk can have their treatment plan optimized to minimize LOS, which in turn also lowers the chance of both staff and visitor infection. Furthermore, predicting a patient's LOS can also help in the hospital logistics planning, such as in room and bed allocation.

Based off the data provided by Kaggle, we will create a model to predict a patient's LOS. To do this, we will first visualize and clean the dataset to gain understanding of the relationship between the target variable, 'Stay', and the predictor variables. We will then use six different machine learning algorithms to find the best model using 'accuracy' as the main performance indicator. Once a model has been developed, we will optimize the hyperparameters to further improve on the model's accuracy.

## II. Related Work

The Healthcare Analytics II dataset was previously used as part of an open competition hosted by Kaggle to explore and build models using data science and machine learning techniques. The participants submitted different algorithms to predict a patient's LOS in a hospital. Kaggle then scored and ranked their models based on accuracy. Overall, the competition saw 20,039 registered participants, with the top three submissions having less than 44% accuracy [3].

## III. Data

The dataset consists of one target variable and 17 predictor variables, and a total of 318,438 entries. The target variable 'Stay', contains the LOS of patients, and is classified into 11 different classes, ranging from 0-10 days to more than 100 days. To analyze the target variable, we first cleaned and visualized every feature in the dataset. The dataset was cleaned by using the dropna() function from Pandas to allow us to drop the rows with null values. The data visualization was done by plotting each predictor variable to gain insight with regards to its distribution, see Figure 1 – 5.

Figure 1 shows a graph, 'Age vs. Number of Patients', with data that appears to be normally distributed as it shows that the data near the mean are more frequent in occurrence than the data away from the mean. Similarly, Figure 2, 'Severity of Illness vs. Number of Patients' and Figure 4, 'Bed Grade vs. Number of Patients', show graphs with data that appears to be fairly normally distributed, albeit slightly biased. Conversely, Figures 3, 'Department vs. Number of Patients' and Figure 5, 'Stay vs. Number of Patients' show graphs with data that is obviously biased. For instance, in Figure 3 one can see that most patients come from the Gynecology Department. In fact, the Gynecology Department makes up over 78% of all the patients in the dataset. Figure 5 shows that the data is skewed to the left with regards to the patient's LOS as there are many more entries for patient's LOS that are less than 40 days.

After cleaning and visualizing the data, 5 predictor variables and 428 duplicate entries were discarded. Two predictor variable pairs, 1) 'Hospital Region Code' and 'City Code Hospital', and 2) 'Hospital Code' and 'Hospital Type Code', were found to be highly correlated. Refer to Figure 6 'Correlation Plot'. Hence, both 'Hospital Code' and 'City Code Hospital' variables were dropped from the dataset. Similarly, 'Case ID', Patient ID', and 'City Code Patient' were dropped. These variables were deemed irrelevant in predicting a patient's LOS. See Table 1 for the complete list of variables used to create our model after both data cleaning and visualization.

TABLE I
HEALTHCARE ANALYTICS II DATASET CONSISTING OF ONE TARGET VARIABLE, 'STAY', AND 17 PREDICTOR VARIABLES. THE VARIABLES HIGHLIGHTED IN RED WERE DROPPED FROM THE DATASET.

| **Variables** | **Description** |
|---|---|
| case_id | Case_ID registered in Hospital |
| Hospital_code | Unique code for the Hospital |
| Hospital_type_code | Unique code for the type of Hospital |
| City_Code_Hospital | City Code of the Hospital |
| Hospital_region_code | Region Code of the Hospital |
| Available Extra Rooms in Hospital | Number of Extra rooms available in the Hospital |
| Department | Department overlooking the case |
| Ward_Type | Code for the Ward type |
| Ward_Facility_Code | Code for the Ward Facility |
| Bed_Grade | Condition of Bed in the Ward |
| patientid | Unique Patient Id |
| City_Code_Patient | City Code for the patient |
| Type of Admission | Admission Type registered by the Hospital |
| Severity of Illness | Severity of the illness recorded at the time of admission |
| Visitors with Patient | Number of Visitors with the patient |
| Age | Age of the patient |
| Admission_Deposit | Deposit at the Admission Time |
| **Stay** | Stay Days by the patient |

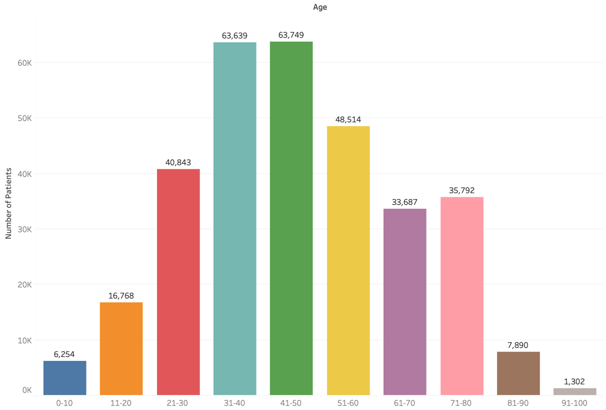

Fig. 1. Age Graph

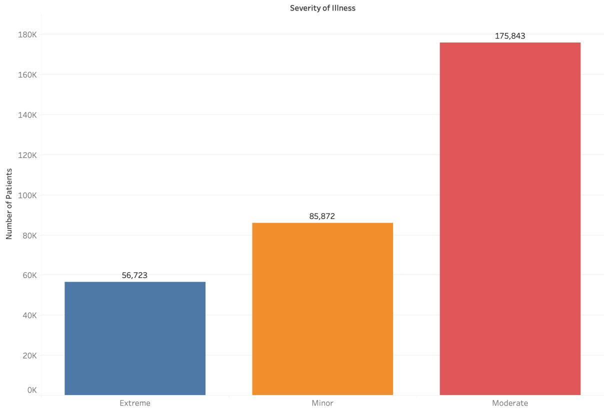

Fig. 2. Severity of Illness

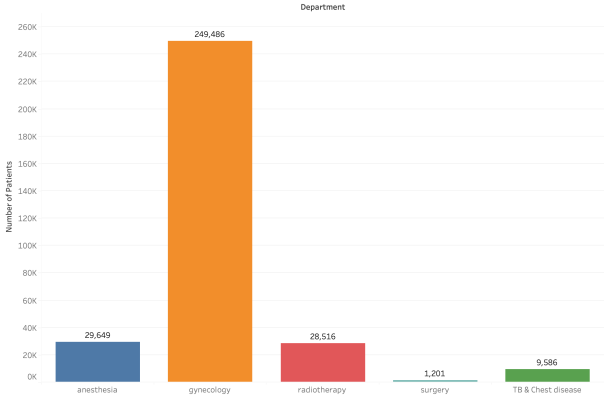

Fig. 3. Department

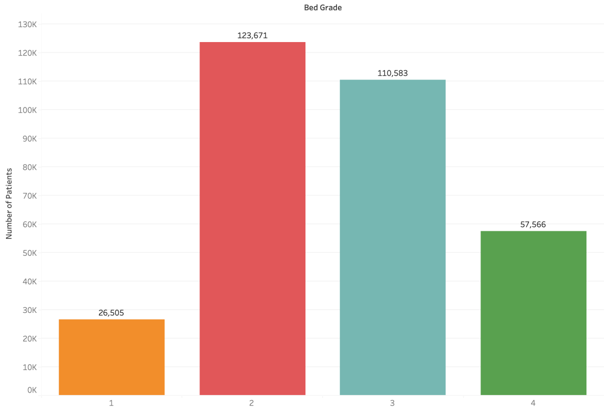

Fig. 4. Bed Grade Graph

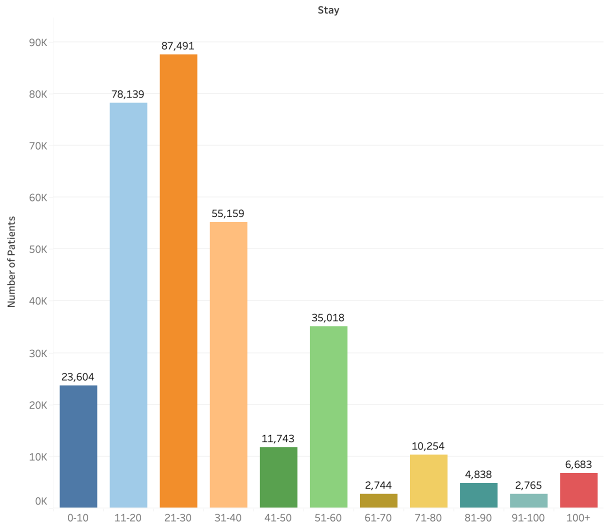

Fig. 5. Stay Bar Chart

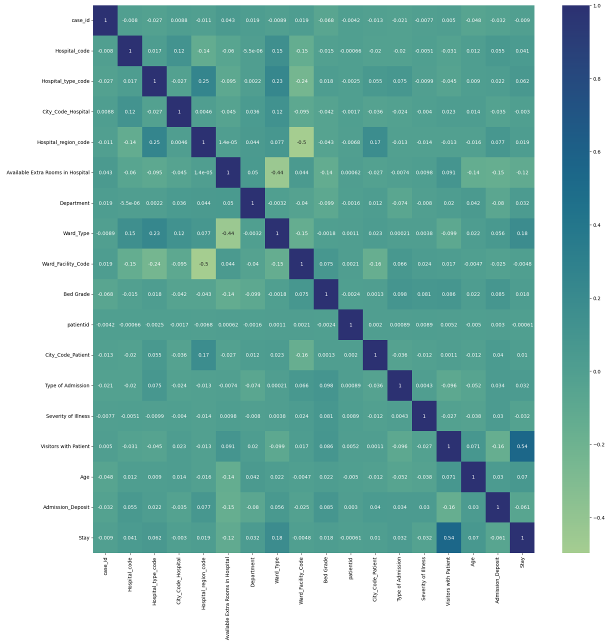

Fig. 6. Correlation Plot

## IV. Methods

Upon preprocessing the dataset by removing duplicate, missing, and irrelevant variables, the remaining predictor variables were either one-hot encoded or label encoded. The one-hot encoded variables are ‘Hospital Type Code’, ‘Hospital Region Code’, ‘Department’, ‘Ward Type’, and ‘Ward Facility Code’. The label encoded variables are ‘Bed Grade’, ‘Type of Admission’, ‘Severity of Illness’, ‘Age’, and ‘Stay’.

The ‘Train’ dataset was split into training and testing subsets since the ’Test’ data provided by Kaggle did not include the target variable, ’Stay’. The sklearn function ’Train Test Split’ was then used to split the data into an 80 - 20 ratio in which 80% of the data went into the training subset and the remaining 20% went into the testing subset. Once the training and testing subsets were established, sklearn’s ’Standard Scaler’ was used to standardize the numeric variables, ’Visitors with Patient’ and ’Admission Deposit’. This was done by removing their mean and scaling them to unit variance.

Six classification algorithms were then employed to predict the patient’s LOS, namely:

- RandomForestClassifier
- KNeighborsClassifier
- LogisticRegression
- DecisionTreeClassifier
- GaussianNB
- XGBoost

Table II shows the initial model’s accuracy score for each algorithm. It is clear that XGBoost provided the most accurate model, 41.32%, while the Decision Tree Classifier provided the least accurate model, 28.56%.

TABLE II
Initial Algorithm Accuracy Score

| Algorithm | Initial Model Accuracy (%) |
|---|---|
| Random Forest Classifier | 33.83 |
| Kneighbors Classifier | 29.53 |
| Logistic Regression | 36.08 |
| Decision Tree Classifier | 28.56 |
| Gaussian NB | 29.79 |
| XGBoost | 41.32 |

## V. Experiments

To improve on the accuracy of our aforementioned models, we tried to tuned the hyperparameters of the three best performing models, see Table II, using a trial and error strategy. No major performance improvement was observed by using this approach. Next, data augmentation was used to duplicate values in a minority class by using the SMOTE function. This was done to help with imbalanced variables, which led to a data increase of about three fold. Although using SMOTE increased the accuracy of most models, the accuracy of both GaussianNB and Logistic Regression dropped.

In parallel to the SMOTE approach, we re-categorized the target variable by splitting it into low, medium, and high values. Similarly to SMOTE, the target value re-categorization gave us mixed results. Overall, the highest accuracy obtained by using only SMOTE was 64.61% for the Random Forest Classifier, while the highest accuracy obtained by using only the target variable re-catetorization was 60.73% with the

XGBoost.

As a result of being able to improve on the accuracy of our models by implementing either SMOTE or the target variable re-categorization, we combined the two approaches to create one model. This in turn resulted in a large increase in accuracy for every algorithm. For instance, the highest accuracy score obtained for XGBoost, by combining both SMOTE and target variable re-categorization, was 68.44%.

To further improve on the accuracy of our models, we tried two hyperparameter tuning techniques: 1) GridSearchCV and 2) Bayesian Optimization. Nonetheless, this endeavor took hours of computation, since our dataset grew from 300k to 900k, only to result in multiple system crashes as our computers did not have sufficient memory. A similar outcome was experienced when we attempted to use Google Colab and Kaggle. Hence, we were unable to further improve our models by hyperparameter tuning.

TABLE III
ALGORITHM ACCURACY SCORES USING SMOTE, TARGET VARIABLE RE-CATEGORIZATION, AND A COMBINED APPROACH

| Algorithm | Model with SMOTE Accuracy (%) | Model with Recategorization Accuracy (%) | Model with Recategorization & SMOTE Accuracy (%) |
|---|---|---|---|
| Random Forest Classifier | 64.61 | 56.09 | 66.25 |
| Kneighbors Classifier | 51.81 | 54.84 | 66.29 |
| Logistic Regression | 34.29 | 56.85 | 58.87 |
| Decision Tree Classifier | 63.02 | 51.3 | 61.46 |
| Gaussian NB | 18.56 | 29.97 | 53.36 |
| XGBoost | 45.04 | 60.73 | 68.44 |

## VI. CONCLUSION

From our initial data visualization, we observed that one of the most influential variables in the dataset is ’Visitors with Patients’, see Figure 6 ’Correlation Plot’. For instance, a patient is less likely to have a long LOS, if they have a lot of visitors. This is because most of the data was comprised of patients with an 11 - 40 day stay. We also observed that most of the patients with a long LOS came from the Gynecology Department, see Figure 3 ’Department’. Other predictor variables, such as ’Bed Grade’, were difficult to interpret as they were already encoded and a proper variable definition was not provided, see Figure 4 ’Bed Grade Graph’.

Following the data visualization, data augmentation, and target variable re-categorization, we found that the most accurate model was achieved with the XGBoost algorithm, see Table III. The accuracy of the model with only re-categorization was 60.73%, while its accuracy using both re-categorization and SMOTE was 68%. Another notable mention is the model achieved using an ensemble algorithm. The accuracy of the model with the Random Forest Classifier and the data augmentation was 64.61%, see Table III.

Although we were unable to further improve on the accuracy of our model by hyperparameter tuning, we were able to make sizeable improvements to the model by both data augmentation and variable re-classification methods. Nonetheless, Kaggle’s Healthcare Analytics II dataset was challenging due to the imbalanced nature of some of the variables, particularly of the target variable.

## REFERENCES

[1] Haileamlak A. The impact of COVID-19 on health and health systems. Ethiop J Health Sci. 2021 Nov;31(6):1073-1074. doi: 10.4314/ejhs.v31i6.1. PMID: 35392335; PMCID: PMC8968362.

[2] D. McPhillips, “The virus threat is easing, but US hospitals are still as full as ever,” CNN, 6 Feb. 2023, https://www.cnn.com/2023/02/06/health/hospitals-full-not-covid/index.html

[3] Vetrivel, Healthcare Analytics II, kaggle, https://www.kaggle.com/datasets/vetrirah/av-healthcare2