Machine Learning for Predicting the Length of Stay in Hospitals

Valerie Chan, Chelsea Wang, and Terry Zhuang

I. Project Overview

Due to the issues exposed by the Covid-19 pandemic, Healthcare Management became crucial and it is important and beneficial for hospitals to have a sense of the length of patients' stay upon admission. The lack of accurate understanding of the capabilities of hospitals and healthcare services caused severe staff and supply chain shortages for hospitals, especially during Covid-19, and there need to be more efforts and advanced technology integrated for a better solution. Thus, we got this set of healthcare analytics data from Kaggle and start to explore different variables and prediction models for potentially more efficient operations and optimal resource allocations. These data are gathered from previous patients with personal data omitted. By utilizing data analytics and manipulation techniques, we aim to come up with models to predict the length of stay of patients with the highest possible accuracy scores. The main questions we sought to answer include the key variables that have the greatest impact on the length of stay as well as better machine learning models overall.

II. Descriptive Analysis

Our healthcare analytics data has 18 columns and 340 thousand rows with 14 categorical variables and 4 numerical variables. The length of stay (LOS) variable is divided into eleven classes ranging from 0-10 days to more than 100 days. Using LOS as our dependent variable, we would like to perform classification analyses to find out how different combinations of the bed grade, the age of the patients, the severity of the illness, the admission deposits, and some other variables can best predict the LOS. From there, we could further analyze how the model may draw helpful conclusions on the operation and resource allocation.

Some of the main independent variables are "Available Extra Rooms in Hospital", "Visitors with Patient", "Bed Grade", "Admission_Deposit", "Age", "Type of Admission", and "Severity of Illness". After our investigation, we realized that most of the variables are categorical, even for the ones we considered to be numerical through our initial observation, such as "Age". Therefore, we converted the categories of "Age" into the integer data type by taking the midpoints within the ranges.

After summarizing the data and visualizing the data distribution for our dependent variable "Stay", we can tell that there are peaks in the 11-20 and 21-30 days range. Also, in order to convert the categorical data to numeric for us to conduct more numerical analysis on the variable "Stay", we decided to take mid-points of each category as their values. At the same time, we used the value "172 days" for the 'more than 100 days" based on our research in the healthcare industry. Among the patients who stayed over 100 days, 74% of them were admitted to the gynecology department, so we made the assumption based on the gynecology inpatient LOS (Length of Stay) data. As the maximum LOS for this category is 243 days, we thus took the

midpoint “172 days.” To reduce the complexity coming from numerous categories in our categorical classification analysis, we combined the LOS of more than 60 days together to form the new category “Over 60 days.”

III. Conclusions

As briefly mentioned in the last section, we decided to compare the performances of the two models using the categorical and numerical versions of our dependent variable. To examine “Stay” as a numerical variable, we employed Linear Regression and the MLPRegressor to predict the “Stay” variable. After trying different combinations of the variables as shown in the second attempt at the Linear Regression, it was found that our current variables contributed to a slightly higher R^2 score of 35.3% on the test data. Using a similar tuning process, the Neural Networks model generated a score of 43.6% and 37.6% on train and test data respectively. Although the latter model performed better than the former, the fact that the train data had an overall higher score than the test was dissatisfying. We also applied GridSearchCV to the smaller dataset “df.Small” we worked with. After running for almost six hours, we got the best-hidden layer size of 100 with the best parameter score of 39.1%.

Regarding the classification models, we chose the MLPClassifier, Random Forests, and Logistic Regression to predict the variable “Stay_cat.” Since Logistic Regression required less time to generate the result, we started off by using Logistic Regression. It returned an accuracy score of 37.6% for train data and 37.7% for test data. Test data was slightly better than train data, but the model’s accuracy was low in general. To further improve the model's capability to predict LOS, we fit the data with the MLPClassifier using a depth of three layers: 84, 134, 212. This model generated accuracy scores of 46.8% and 35.7% for train and test data, respectively. Surprisingly, the Neural Networks model performed worse than the Logistic Regression model when predicting the test data, which was quite disappointing. As an attempt to figure out what variables lowered the accuracy score, we then proceeded to fit the data with the Random Forests. Due to its overfitting nature, we would not use this model for any prediction purpose. Instead, it was generated to observe influential variables regarding our dependent variable “Stay.” The result indicated that “Admission Deposit” played a significant role in predicting LOS for each patient.

IV. Future Improvements

Throughout our study, one of the most influential barricades that prevented us from making further improvements to our model is the enormous data size we were dealing with. Within the timeframe, it is quite difficult for us to conduct thorough investigations on all of the variables, including the categories they each possess. Later, although we used a smaller portion of the data for modeling, it was still tough to implement various tuning processes due to time

constraints. We made a trade-off of perfecting the tuning of our neural network classification model rather than polishing other models. Therefore, if we have more time, here are several improvements that our team would like to impose:

One way to potentially improve the predictability of our model is to further explore the correlation among independent variables. On the first glimpse of our correlation analysis, it seems like the correlation coefficients between each numeric variable are relatively weak, ranging from 9% to 15%. Even though, it is still essential to investigate their relationship in real contexts because an intrinsic correlation may lead to a lower predicting power. An example to show the relationship between columns is that better "bed grades" may be positively correlated to higher spending on the admission fee, contributing to a larger amount of "admission deposit". If we could do more outside research and more detailed correlation analysis, the output could be in a better position.

With a longer time span, we intended to apply our best models to a larger dataset for a higher accuracy score. The dataset we used to test and train the model was only a portion of the original one, as it was already split by the author prior to this project. We believe more data will provide a more comprehensive view of the data environment and an opportunity to tune the model based on new data, thus generating a better model. We would also like to have a deeper understanding of the variables we excluded from this analysis because of lack of information, such as "ward facility code" and "ward type". While no reference and interpretation come with the two variables mentioned above, we may be able to find their meaning if we conduct more research in the healthcare industry and hospitals in general. The use of this variable, which indicates the type of room patients live in, may plausibly increase the accuracy score as it seems like there is a relationship between a particular type of room and the length of stay.

Another way to improve the accuracy of our model is to apply more tuning techniques to the models. In our neural networks model, we intended to find the best parameter score and the value of the hyperparameter, depths, that can significantly improve the accuracy score. We tested this hypothesis by randomly selecting and sampling 60% of the data, and the result turned out to be 39.1%, which is the best result we generated so far. Our team also believed that we should apply more feature engineering techniques, such as spline and polynomial transformation, to deal with possible non-linear relationships within data. We tried to incorporate power transformation into our model, but the results weren't ideal. If we are able to figure out a combination of the feature transformations mentioned, we may receive a model with higher accuracy.