REPORT
PROJECT HOSPITAL ICU

Course

Stochastic Processes

Professor

Mickael Gaury

Authors

Cristian Medina

Jhonnatan Betancourt

Master of Data Analytics for Business

KEDGE Business School – Bordeaux 2024

1. Introduction

The project of Hospital ICU objective is to generate two models that aim to improve the operational efficiency and patient outcome predictions within the Intensive Care Unit (ICU) environment. To go through the critical challenges faced by healthcare professionals in critical care settings, we utilized simulation and predictive modeling techniques.

For the first part of the project, we developed a machine learning model that predicts patient outcomes in the ICU. The objective of this model is to determine whether a patient requires further care or is ready to be discharged, based on clinical assessments and treatment responses. We integrated variables such as the Simplified Acute Physiology Score (SAPS), Glasgow Coma Scale, Therapeutic Intervention Scoring System (TISS), and vital signs to predict the probability of negative outcomes, designated as 'bad'. We built and validated the predictive model through setting “bad” column as the target in the data analysis and threshold studies along with the feature ensuring its robustness and accuracy.

For the second part of the project, we leveraged AnyLogic software to create a simulation model representing patient flow within a 15-bed ICU. The simulation model focuses on operational aspects such as patient admission processes, bed occupancy, nurse-patient assignments, and patient recirculation within the system. We analyzed a provided dataset to extract essential parameters such as arrival rates, length of stay, and return rates. These parameters form the basis of our simulation to ensure its accuracy and relevance. The key performance indicators (KPIs) identified for this study include bed utilization rate, patient rejection rate, and nurse workload. Additionally, we created a 3D visualization of the model to provide intuitive insights into the operational dynamics of the ICU.

Both components of our report are interconnected, offering a comprehensive view of real-time operations and future predictions within the ICU. By utilizing current available libraries, analytical techniques, and software interfaces we aim to bring about tangible improvements in critical care management.

2. Python Code

In the starting point of the project, the team dived into the "DataForClass.csv" file by first reading the csv, which was provided as semicolon separated values. Thus, analyzing and preprocessing the dataset to generate and assure a high-quality data frame. As viewed in the following table, we saved the data frame in "data" variable and showed it the first five rows.

Table 1: Data frame from variable "data"

	id	codes	age	gender	dates	bad	last	SAPS	Glasgow	TISS	PA	FC	Temp	dayIn
0	8309	20017421	60	1	10/10/2019	0	in	NaN	6.0	NaN	73.0	130.0	37.30	1
1	8309	20017421	60	1	11/10/2019	0	in	85.0	5.0	37.0	71.0	100.0	37.69	2
2	8309	20017421	60	1	12/10/2019	0	in	66.0	10.0	35.0	94.0	95.0	37.69	3
3	8309	20017421	60	1	13/10/2019	1	last	57.0	14.0	29.0	101.0	93.0	37.88	4
4	6112	40026409	60	1	28/04/2018	0	in	NaN	11.0	29.0	36.0	124.0	36.32	1

In the process of setting up the data frame, we identified that some of the information was missing from the selected features the team considered that those features are highly correlated: 'SAPS', 'Glasgow', 'TISS', 'PA', 'FC', 'Temp', and drop the features like 'id', 'codes', and 'last' because are likely non-predictive.

After that the next step was to import the library "KNNImputer". The principal reason for this strategy is that this module imputes the missing values of the features using a specific number of close neighbors. In the case of the code, we used seven as selected neighbors. After that, we designated "bad" as label attribute to predict for the matter of solving the ICU case. Then, for testing the model, we proceeded to split the data with test size of 0.2. However, we noticed that the data was unbalanced, and for that reason we reviewed the possibility to evaluate the model using SMOTE library.

The reason that we used SMOTE is to provide a balance in the classes in the classification problem, since the idea of predicting classified data such as 0 and 1 in the target "bad", it's necessary to have samples that are representatives in a balanced way because the contrary will lead to biased models with poor perform. In fact, imbalance was the first challenge the group encountered at creating the model. Thus, is was necessary to use the synthetic minority over-sampling technique (SMOTE) method to balance the class distribution. Once we the counted number of classified 0 and 1 was 4263 values for each class.

Following the algorithm to generate a machine learning method, the team decided to perform multiple models to get the best possible classification report score, specifically for precision. The best model to obtain a reliable precision in the classification report for binary classification was by importing the library GridSearchCV and using GradientBoostingClassifier.

For the latter, it was necessary to use a set of parameters for the algorithm to allow the optimization. For the parameters setup it was necessary to do a review of the important parameters: learning rate, number of estimators, max depth, and number of iterations to change.

Image 1: Parameters of GradientBoostingClassifier model

```
param_dict = {
    'learning_rate': [0.1, 0.2],
    'n_estimators': [20, 50, 75],
    'max_depth': [1, 3, 5, None],
    'n_iter_no_change': [None, 1, 5, 10]
}
```

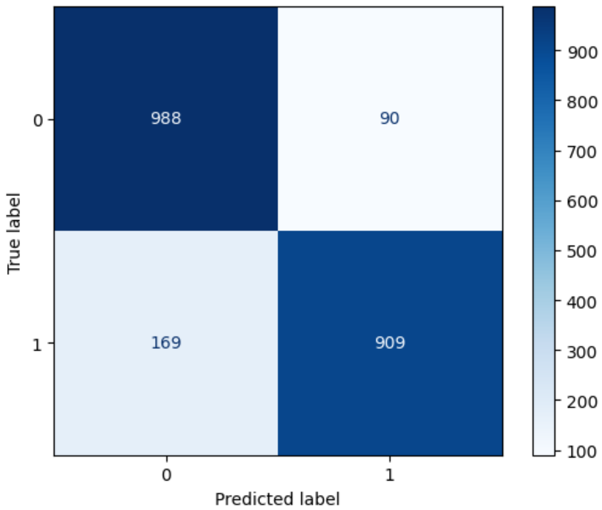
The model is generated, and we use the balanced sample that the model should test with the best parameters of the model. Results of the classification report indicates that the classification balanced even for 0 and 1, and that the precision scores were above 0.8.

Image 2: Model's classification report

	precision	recall	f1-score	support
0	0.85	0.92	0.88	1078
1	0.91	0.84	0.88	1078
accuracy			0.88	2156
macro avg	0.88	0.88	0.88	2156
weighted avg	0.88	0.88	0.88	2156

The evaluation of the model was done for a balanced dataset in the binary classification. Precision scored more than 0.8, which is a good indicator of the accuracy of the model. We may see in the confusion matrix displayed below that True Positives (TP) and True Negatives (TN) are substantially high for a 2156 data evenly divided for 1078 each for values of 0 and 1.

Image 3: Model's Confusion matrix



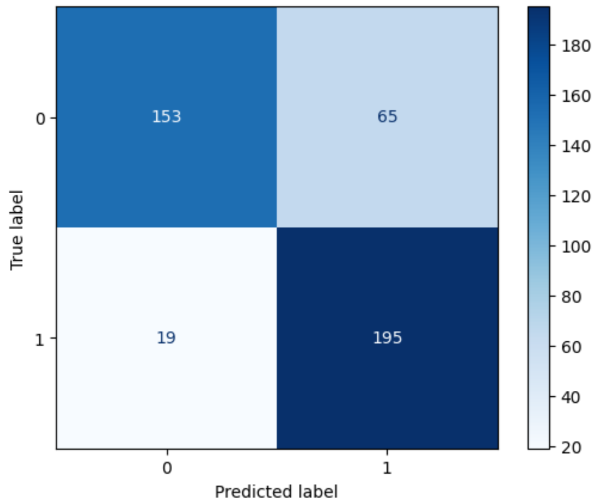
After obtaining the results, we attempted the Deep Learning model to identify if it was possible to optimize the classification report with a binary classification neural network. The idea is using Multi-layer ANN. The parameters that we wanted to modify of the model was the batch size, number of layers and neurons, as well as, learning rate, weight decay, and the number of epochs.

Image 4: Multi-layer ANN classification report

	precision	recall	f1-score	support
0	0.89	0.70	0.78	218
1	0.75	0.91	0.82	214
accuracy			0.81	432
macro avg	0.82	0.81	0.80	432
weighted avg	0.82	0.81	0.80	432

Further analysis in the variation of those parameters for the model should be made, however, multiple simulations were done to compare any improvement of the model always having an optimized threshold for the classification of the predicted class.

Image 5: ANN Model's Confusion matrix



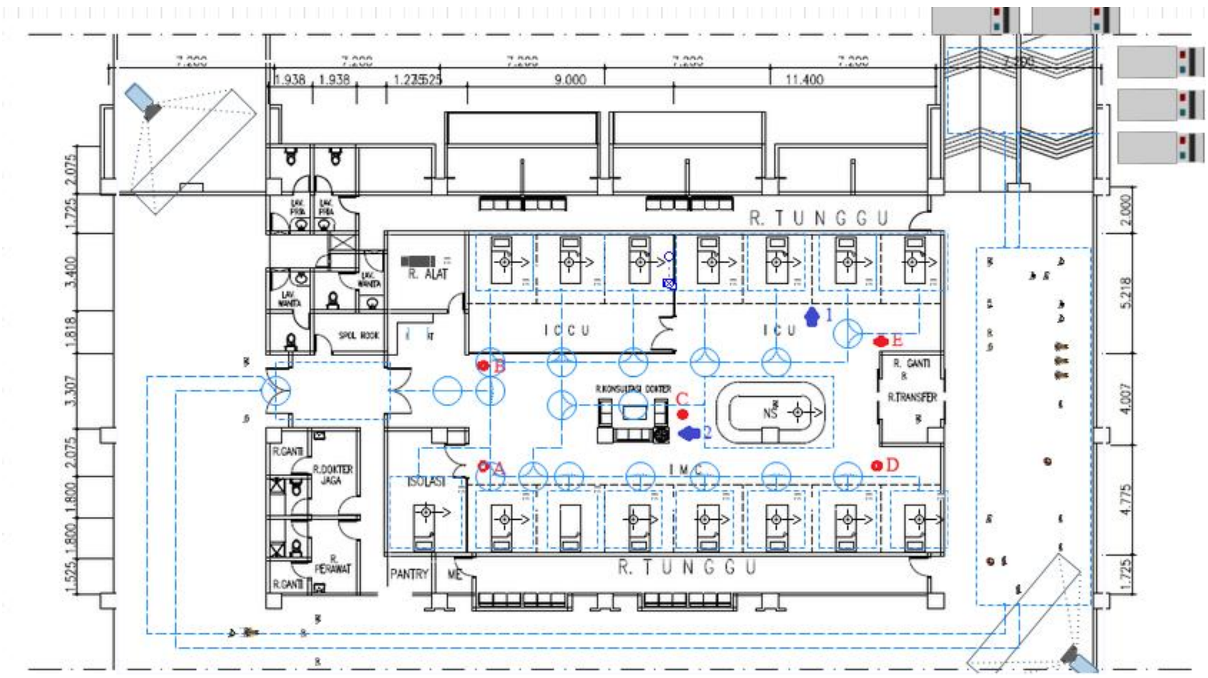
Finally, we opted to use the results for the first model as it got higher accuracy. For further examination of the code and determining the input information for Anylogic, we proceeded to arrange the main data frame to append the predicted results with the case constraints. Thus, we obtained that the average daily rate of patients staying or incorrectly staying is 7.37 patients per day, meanwhile the average of patients that leave or incorrectly staying is 1.16 patients per day. In the next step of the project, we proceed to use Anylogic using Machine Learning results as input.

3. Anylogic Implementation

In the second phase of our project, we utilized the capabilities of AnyLogic software to construct a dynamic simulation model that provides a detailed illustration of patient flow and resource allocation within a 15-bed Intensive Care Unit (ICU).

The purpose of this simulation is to address operational management complexities and optimize the dynamics between patient care and resource utilization. In image 6 it's possible to see the arrangement of the ICU for 15 beds, which have the following description: rooms and beds, central nursing station, equipment and facilities, access points, support areas, and circulation space. The lines in blue are the setup flow of patients when entering to the ICU area.

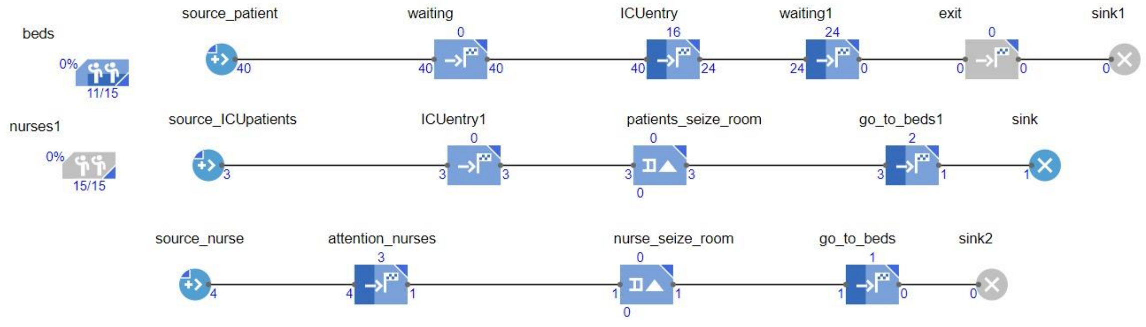
Image 6: ICU Layout design



The model developed with AnyLogic is designed to represent several operational aspects, such as patient admissions, the allocation of beds, nurse-to-patient assignments, and the regular pattern of patient recirculation within the system. To achieve this, the model is built on a foundation of parameters carefully derived from the provided data, which includes arrival rates, duration of stay, and frequency of return visits. These parameters serve as the basis of our simulation, ensuring its accuracy and relevance.

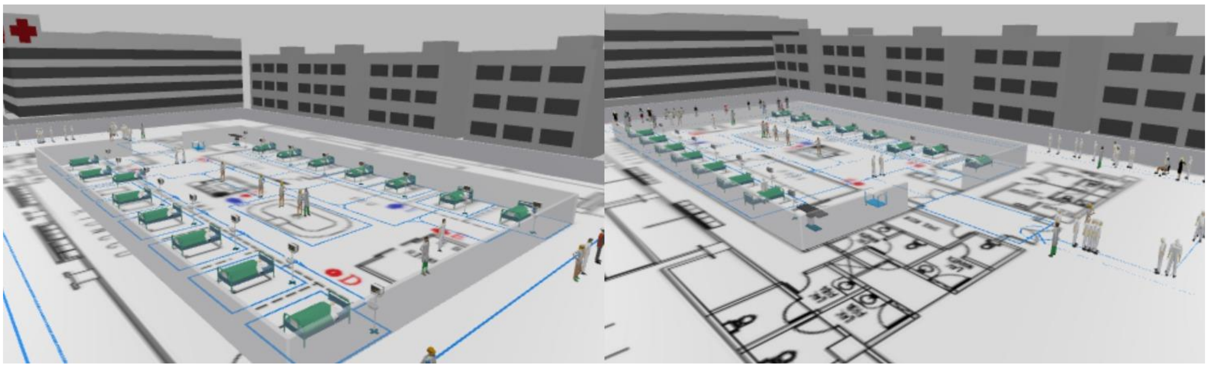
Image 7 reveals the detailed layout of the ICU within AnyLogic. This layout is a veritable blueprint that mirrors the actual floor plan of the ICU, encompassing patient rooms, nursing stations, and other critical areas. The meticulousness of the layout facilitates an understanding of spatial dynamics and resource distribution within the ICU setting.

Image 7: Simulation process blueprint



In simulated image 8, you can get a better sense of the environment. There are two perspectives shown in the image - one focused on the bed area, while the other offers a broader view of the ICU. These perspectives provide a comprehensive understanding of the patient flow, utilization of space, and personnel.

Image 8: Simulation process blueprint



4. Conclusion

The report emphasizes that ongoing refinement and validation are essential to increase the efficacy of the AnyLogic simulation model and the machine learning predictive model, despite their promising results. The multi-layer ANN, a machine learning model, needs further exploration into the parametric nuances that govern its learning capabilities. Such iterative improvements are vital for advancing the operational efficacy of the ICU, ultimately hoping to reduce preventable patient casualties.

This project demonstrates the potential of integrating machine learning and simulation modeling in healthcare. It shows how data analytics can be used to foster decision-making processes that are both data-driven and centered on the patient. Our endeavor establishes an interesting approach for future studies to build upon continually evolving analytical models to derive with the evolving landscapes of healthcare demands.

This integrative approach combining predictive analytics with dynamic simulation paves the way for a revolutionary shift in ICU management. It offers an innovative lens through which patient outcomes can be forecasted, and operational efficiencies can be realized, transcending traditional methodologies. Thus, this project stands not as an endpoint but as a conduit to progressive advancements in critical care.