# MODELING OF MORTALITY PREDICTION AND LENGTH OF STAY

**The University of Hong Kong**
**STAT3612 Group Project**
**Group 9: Team LowAcc**

MARVELLA, Jennifer (3035425352), AKHRAS, Emad (3035662962),
RAY, Eehit (3035665122), SO, Ka Ho (3035690048)
LAM, Yiu Pong (3035691248)

## Abstract

This proposal will briefly introduce the methods we will be implementing for feature engineering as well as modeling on the given datasets: Mortality Prediction and Length-of-stay (LOS) Prediction. This report will lay down some data analysis, feature engineering and modeling applied to both datasets, and will evaluate their performances in doing predictions based on the results observed.

## 1 Introduction

The healthcare industry is undoubtedly one of the most researched fields, yet constantly demands time-to-time improvements through its operational efficiency, treatment decisions, and risk management. The Electronic Health Records (EHR) holds the potential to improve the healthcare industry in many approaches, including delivering better patient treatments, improving hospital operations and answering fundamental scientific questions. One of the most promising applications of EHRs is designing machine learning models for risk prediction tasks, such as mortality and long Length-of-stay (LOS) prediction. This report aims to evaluate the performance of predictions on mortality and long Length-of-stay (LOS) by applying feature engineering and modeling to the given dataset.

To achieve that, we have experimented with numerous machine learning approaches to build classification and regression models to evaluate the performance, including logistic regression (for mortality prediction only)/ linear regression (for LOS prediction only), gradient descent by XGBoost and convolutional neural networks by Inception Time. Evaluated by the AUROC/RMSE, the InceptionTime is shown to consistently outperform other classification algorithms on the test set. Logistic regression/ linear regression and XGBoost also produce adequate results with strong interpretation ability.

## 2 Related Work

Wang et al. (2020) present baseline scores as obtained through their MIMIC-Extract pipeline using logistic regression, random forests as well as gated recurrent units. Zhu et al. (2021) perform an analysis specific to mechanically ventilated patients in the same dataset, using a variety of machine learning methods, including KNNs, neural networks and XGBoost. Hou et al. (2020) present a similar analysis limited to patients suffering from sepsis-3, and Tsiklidis et al. (2022) focus an analysis on trauma patients.

While the majority of the aforementioned works primarily use traditional statistical machine learning techniques, a variety of works have also utilized more recent, deep-learning based methods. Nallabasannagari et al. (2020) predict mortality and bucketed length-of-stay using NLP techniques coupled with deep learning models, and Scherpf et al. (2019) provide an analysis of sepsis prediction using RNNs.

Since the field of medicine requires a high degree of interpretability and not just performant machine learning models, a great deal of care is also required in ensuring that the models are fair and transparent. Meng et al. (2022) present valuable research on the evaluation of fairness and interpretability of various deep learning models on the MIMIC-IV dataset. Liu et al. (2020) provide a similar analysis using XGBoost coupled with the SHAP method.

## 3 Dataset

### 3.1 Dataset description

In this project, the data given was extracted from the Medical Information Mart for Intensive Care (MIMIC-III) clinical database, and preprocessed using MIMIC-Extract. The input dataset contained a comprehensive clinical ICU data of around 19,000 patients preemptively split between training and validation sets. Both sets include patients' vital signs, medications, and data measured in labs (for example, hematology, chemistry and microbiology results). It is noted that there are a total of 104 input features, each with three sub-features measured in an hourly time series format. The dataset contains both categorical and quantitative features including the mask (1: the feature was measured in that hour; 0: otherwise), mean (the measured value of the feature for that hour) and time since measured (the number of hours since the last measurement).

### 3.2. Dataset preprocessing

Since the dataset contains a time series element (24 hours of data for mean, mask and time_since_measured for each variable), we must first preprocess the data in a way such that we do not lose the information in the original data, but also significantly reduce the number of features. Furthermore, we must also deal with the large number of hours where no measurement was made i.e. *mask* = 0. We thus establish a three-step preprocessing pipeline:

#### A. Interpolate values

For most of the variables, measurements do not exist (*mask*=0) for most of the hours (see *Section 3.2 Descriptive and Diagnostic Analysis* for further details). Thus, we linearly interpolate the value for these hours (where no measurement was made) from the nearest neighboring hours where a measurement was made (*mask*=1).

#### B. Create features by aggregating

In order to remove the time series element, we must apply some form of aggregation(s) to each of the variables, thus reducing the number of features from *24x3=72* per variable, to a smaller subset. We performed the following aggregations:

i. Variable mean: For each day, we calculate the mean value of the variable

ii. Variable change: For each day, we calculate the overall change of the variable during the day i.e. (last_value – first_value)/first_value

iii. Variable count: For each day, we calculate the number of times the variable was measured i.e. sum of *mask* for the day.

We also experimented with min/max rather than the variable mean, but we found that this produced inferior results compared to the variable mean.

#### C. Preliminary Feature Selections

After observing the correlation heatmap between the features, we saw that eight features were highly correlated with each other and also showed similar correlations with the other variables. Thus, we decided to remove these variables:

| | | | |
|---|---|---|---|
| albumin ascites | albumin urine | creatinine ascites | creatinine body fluid |
| creatinine pleural | lymphocytes atypical csl | lymphocytes percent | lymphocytes pleural |

Table 1: Removed variables due to intercorrelation

**Thus, at the end of our preprocessing, we reduce our columns from 104 x 3 x 24 = 7488 to 96 x 3 = 288 columns.**

### 3.2 Descriptive and Diagnostic Analysis

We start by looking at the distribution of counts of the various variables i.e. on average, how many times each variable was measured throughout the day.

First, we look at a histogram of the median counts for each variable in figure 1.

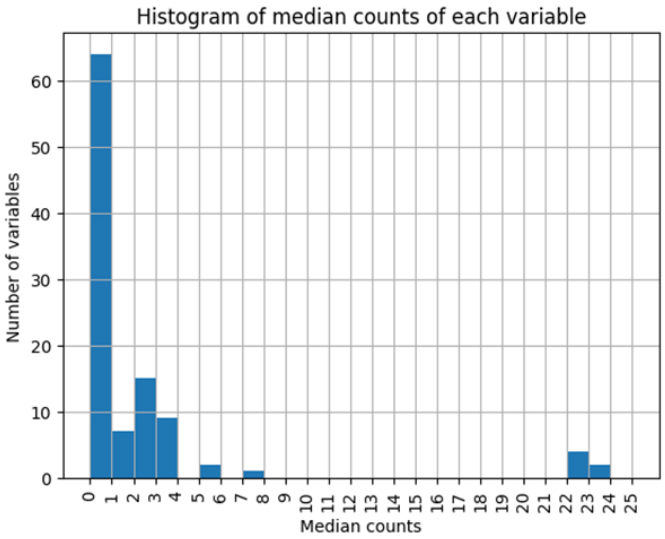


Figure 1: Histogram of median counts of each variable

This suggests to us that the majority of variables are measured infrequently throughout the day – over 80 out of 104 variables (first three bars) are measured less than 3 times. On the right side of the chart, we can see that there exist about 5 variables that are measured for almost every hour.

Some of these variables are lab tests that may be measured only for some patients. Furthermore, even when they are measured, they may be measured only once or twice. Thus, we look at the 3rd quartile of the data, as the counts may be right skewed (i.e. majorly zeroes).

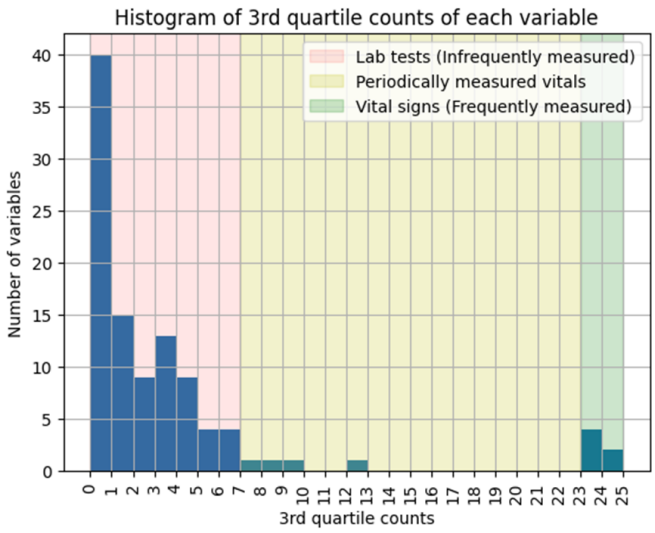


Figure 2: Histogram of 3rd quartile counts of each variable

This gives us a clearer view of the variables. In the red section, we have variables that are measured less than 7 times in the day (as per the 3rd quartile); these are primarily **lab tests**. Some of these variables include:

- "red blood cell count csf": a test to measure the amount of RBCs in cerebrospinal fluid (A.D.A.M Editorial Team & VeriMed Healthcare Network, 2021)
- "post void residual": a test to measure the amount of urine left in the bladder after urination (National Institute of Diabetes and Digestive and Kidney Diseases, 2019)
- "eosinophils": a test to measure a specific type of white blood cell (MedlinePlus, 2021)

In the yellow section, we have four variables that are measured somewhat frequently (twice – thrice a day); these are vitals that needn't be measured every hour:

- glascow coma scale total: objective assessment of brain injury level
- temperature
- glucose
- central venous pressure

Lastly, in the green section, we have six vital variables that are measured almost every hour. These are:

- diastolic blood pressure
- mean blood pressure
- oxygen saturation
- systolic blood pressure
- heart rate
- respiratory rate

Next, we observe which variables are highly correlated with the two output variables, *mort_icu* and *los_icu*.

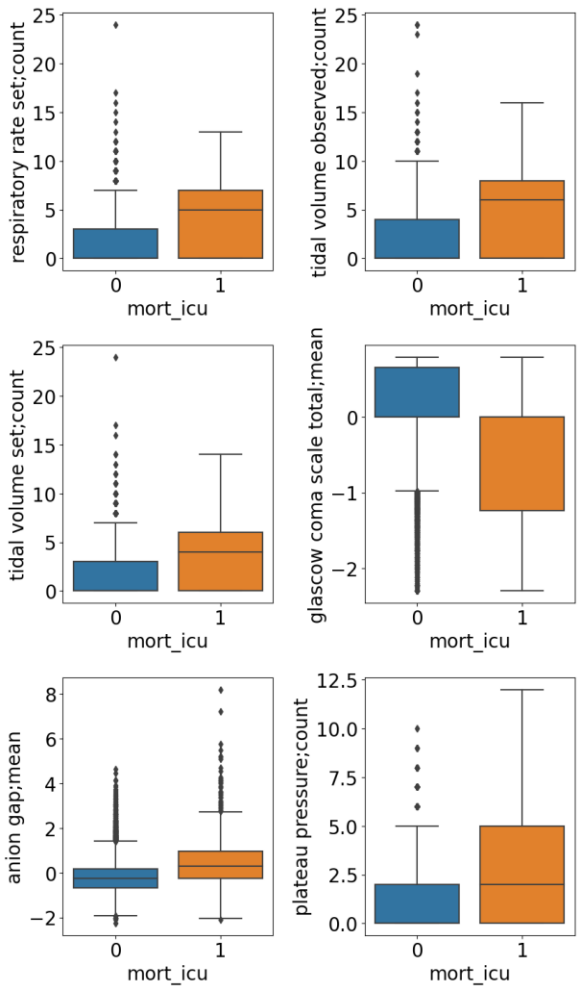


Figure 3: Grouped boxplots for the six most correlated variables with *mort_icu*

Four of the six variables in Figure 3 (*respiratory rate set; count, tidal volume observed; count, tidal volume set; count* and *plateau pressure; count*) are all variables that count the number of times a certain value on a **ventilator** was adjusted. Furthermore, higher values of these counts are associated with a higher chance of death. In other words, this tells us that **patients for whom a ventilator must be repeatedly reconfigured or readjusted (possibly because their respiratory system is failing) are at a higher chance of death**.

The variable *glascow coma scale total; mean* is a measure of severity of brain injury level (Jain & Iverson, 2022); lower values correspond to a more severe brain injury. This is corroborated by the data – for patients who die (*mort_icu = 1*), 75% of the values of this variable are below 0, while only 25% of the values are below 0 for those who survive.

Lastly, *anion gap; mean* is a measure of blood acidity (MedlinePlus, 2022). Here, our data tells us that higher values of anion gap are more associated with death. Typical reasons for high blood acidity include dehydration and diarrhea.

Thus, this study of correlations gives us an insight into some leading causes of death – respiratory failure, brain injury and dehydration.

Next, we analyze the highest correlations with *los_icu*.

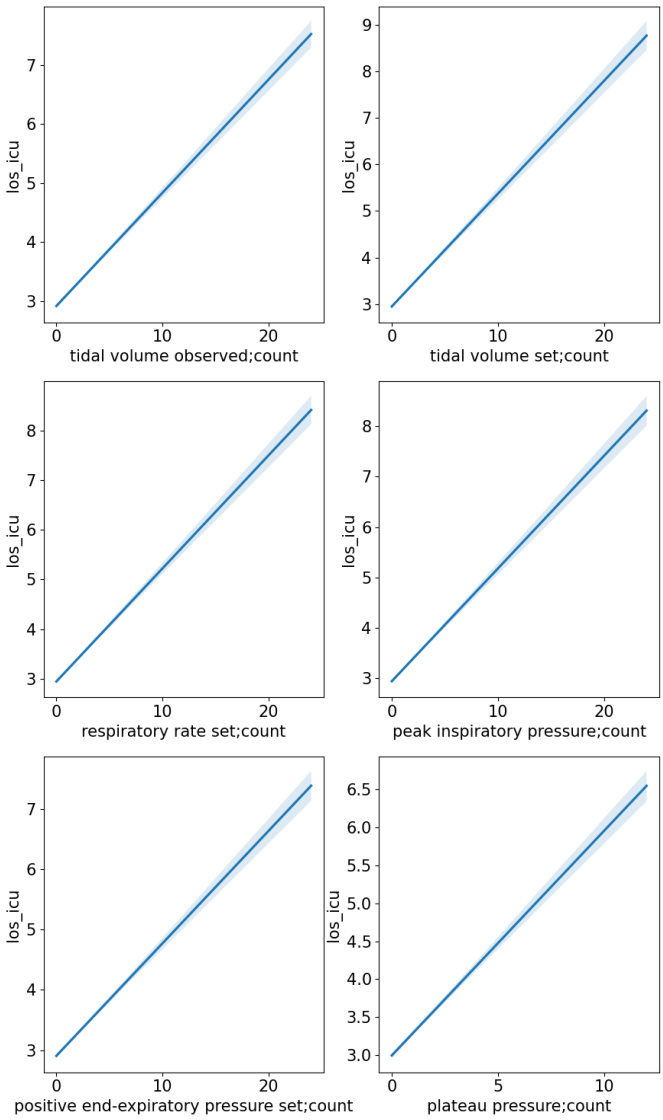


Figure 4: Regression lines for the 6 most correlated variables with *los_icu*

We observe that all six variables are actually counts of settings adjusted on a ventilator – the more often these settings are adjusted, the longer the patient typically has to stay in the ICU. This helps us conclude that the most significant medical condition for determining length-of-stay is respiratory system failure.

We also note that the 7th most correlated variable with *los_icu* is once again, *glascow coma scale total; mean*. This tells us that brain injury is also a significant factor in determining length of stay.

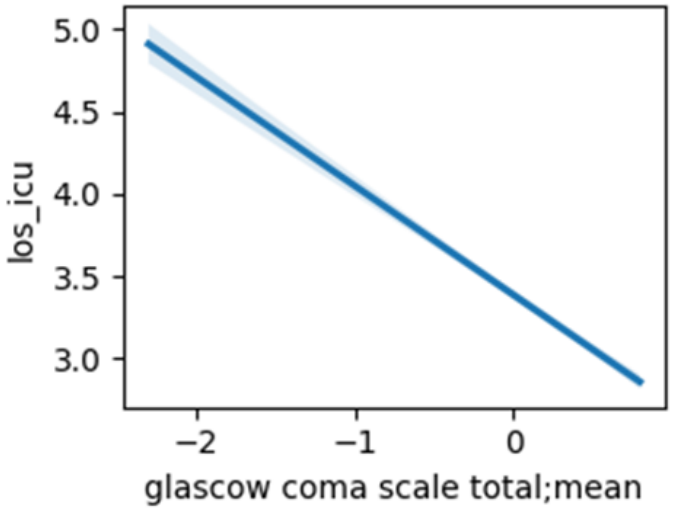


Figure 5: Regression lines for the 7th most correlated variable, glascow coma scale total; mean, with *los_icu*

## 4 Methodology

In attempting to tackle the above-mentioned tasks, we fit our curated and preprocessed input datasets into models that offer varying levels of interpretability and performance. For both tasks, we trained a baseline model at first. The rationale behind this choice is that typically a simple model offers high interpretability at the expense of low performance. We progressively increased the complexity of subsequent models trading-off interpretability for performance.

For instance, in the classification task, we initially trained a logistic regression model where a strict set of assumptions leads to lower performance, but the coefficients may provide direct insight into the predictions. The same argument holds for linear regression in the case of LoS predictions. Our second model of choice was Gradient Boosting, particularly XGBoost. With its classifier and regressor variants, we can analyze feature importance and draw useful conclusions. The last model we trained was a convolutional neural network called InceptionTime that retains and makes use of the time series element of our data.

For the baseline models and Gradient boosting, a list of different combinations of hyper-parameters was initialized

prior to fitting the data. In technical terms, for example, linear model classes imported from "scikit learn" library accept some predefined parameters, each comes with a finite set of values. The LogisticRegression class can be initialized with the norm of penalty, inverse of regularization strength, an algorithm to use in the optimization problem, etc. We wrote code that takes in a predefined set of values for a subset of parameters, and generates plausible combinations that can be used to initialize the model. In order to obtain the best combination, we performed grid search cross validation on the training data and recorded the optimal combination that was used later for validation and testing. See submitted code for reference.

### 4.1 Logistic Regression

Logistic Regression is the adopted baseline model for the mortality classification task. This classifier is built upon a linear regression modeled using a logit link that predicts the probabilities of possible outcomes. We used different penalty arguments ($l_1$ , $l_2$ ), solvers ('lbfgs', 'saga'), C [1e-3, 1e3]. Even though Logistic Regression doesn't assume the distributions of the classes in the feature space, its disadvantage is that it doesn't account for multicollinearity between independent variables.

### 4.2 Linear Regression

Linear Regression is the adopted baseline model for the LoS prediction task. An ordinary LR fits a linear model ( Pedregosa et. al, 2011) of the form: $y = X\beta + \epsilon$ , where $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ are the coefficients to be estimated and $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ are random errors that follow normal distribution. The fit minimizes the residual sum of squares between observed and predicted responses. A shortfall in Linear Regression is the assumption of independence of features. When the columns of the design matrix X are correlated, the least-square estimate is prone to fluctuate drastically in response to random errors in the observed responses.

### 4.3 Gradient Boosting - XGBoost

Gradient Boosting is a supervised learning class of algorithms that aims to predict response variables by fitting a set of simple and weak learners sequentially using the residuals of previous learners. XGBoost, which stands for Extreme Gradient Boosting, is an important gradient boosting algorithm that relies on decision trees and ensemble learning (*What Is XGBoost?*, n.d.).

In essence, XGBoost builds on top of a standard gradient boosting algorithm in that each input data point is linked through a regression tree to a leaf node with an objective score. The training is an iterative process in which the error prediction of regression trees are combined sequentially to produce the final prediction. Furthermore, XGBoost offers a modification to the convex loss function which optimizes for a regularized penalty norm function and penalizes for increased complexity.

In the classification task, we used the classifier variant of XGBoost for boosting coupled with Bayesian optimization to find the optimal hyper parameters (Optuna) (Warnes, 2021). On the other hand, in the prediction task, we used the regressor variant of XGBoost, in conjunction with grid search cross validation to find the optimal hyper parameters.

### 4.4. CNN - InceptionTime

Time series data has a "spatial" component i.e. values that are close to each other tend to be related to each other. As such, methods from image processing, such as image classification algorithms may show promise when applied to the raw time series present in our dataset.

Thus, we used a neural network architecture known as InceptionTime (Fawaz et al., 2020) on the raw time series data (i.e. *mean* and *mask* for each variable). InceptionTime takes the popular Inception architecture from computer vision and adapts it to time series classification and regression.

In essence, the InceptionTime architecture consists of multiple "Inception" modules, stacked on top of each other. Each Inception module consists of convolutions of varying lengths applied to the input time series. The modules are also connected using direct connections known as residual connections in order to ensure that gradient flows during backpropagation without vanishing. Finally, the output from the Inception modules is averaged together and run through a fully connected neural network, whose architecture depends on the task - a "classification head" i.e. linear layers + sigmoid output along with a cross entropy loss for classification, and a "regression head" i.e. linear layers + an MSE loss for regression.

For our experiments with InceptionTime, we utilized a far smaller version of InceptionTime, with just 4 InceptionModules and 4 convolution filters in order to prevent overfitting. We also employed dropout as well as weight decay for this purpose. **Note that we trained InceptionTime on time series data i.e. *mask* and *interpolated mean*** (i.e. Steps 2 and 3 of the preprocessing pipeline are not performed). Although InceptionTime offers superior performance as compared to the other two models (see *Section 5*), we note that this performance comes at the cost of interpretability, as neural networks are significantly harder to interpret owing to their complexity.

# 5 Experiments

## 5.1 Mortality prediction

### 5.1.1 Logistic regression

From the result in table 2, we can see that it highlights the imbalance in the dataset – while the model achieves good accuracy, the recall is substantially lower.

The coefficient of the logistic regression reveals that the best predictors are vital signs as expected. For example, having GCS_mean reduces the surviving odds ratio to exp(-0.6052) = 54.6%. Conversely, having a respiratory rate count, for example, increases the surviving odds ratio around 28%.

| **Metric** | **Value** |
|---|---|
| Precision | 0.686 |
| Recall | 0.270 |
| F1 score | 0.388 |
| Accuracy | 0.939 |
| **ROCAUC** | **0.897** |

Table 2: Metrics of logistic regression

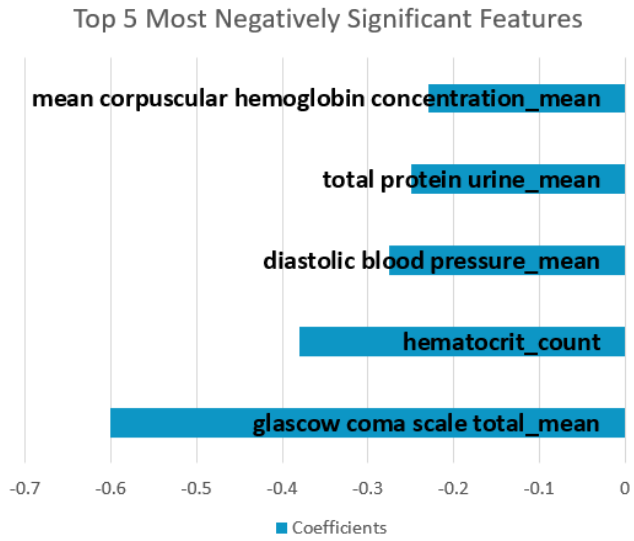


Figure 6: Most negatively significant features for logistic regression model

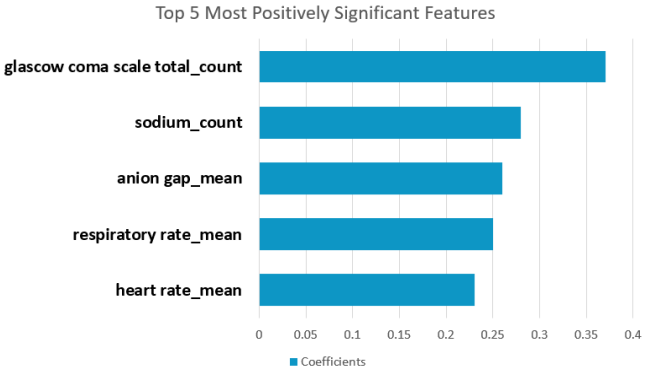


Figure 7: Most positively significant features for logistic regression model

### 5.1.2 Gradient Boosting (XGBoost)

By comparing the results in table 2 and 3, Boosting improves from LR on every metric, specially on precision – which means that the model is now less likely to achieve a good score by classifying everything as 0. Nevertheless, due to a potentially unlimited number of trees, they are highly prone to overfit.

The total number of times a variable is used to split nodes is used to evaluate the importance of the features. Glascow coma scale total is the most important feature followed by other important variables - vital signs, such as blood pressure, respiratory rate, heart rate etc.

| Metric | Value |
|---|---|
| Precision | 0.731 |
| Recall | 0.275 |
| F1 score | 0.399 |
| Accuracy | 0.941 |
| **ROCAUC** | **0.919** |

Table 3: Metrics of XGBoost

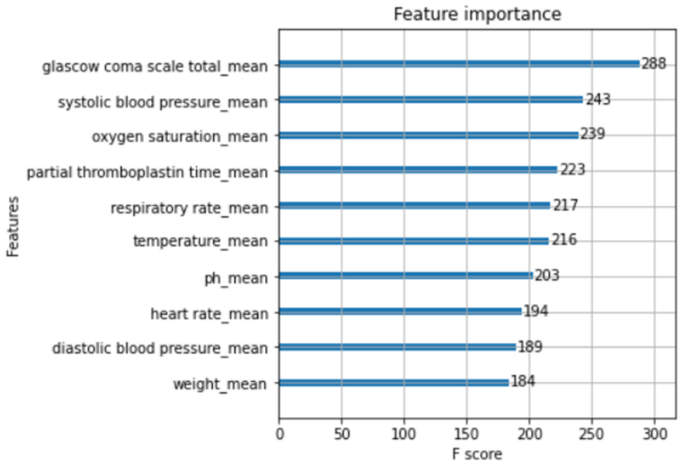


Figure 8: Feature importances for XGBClassifier model

### 5.1.3 Convolutional Neural Networks

To prevent overfitting, we used depth equal to 4 (i.e. 4 Inception modules) and 4 kernels and added dropout + weight decay (L2 regularization).

The result below shows that InceptionTime further improves upon XGBoost in every metric except AUC. However, one of the major drawbacks is that InceptionTime significantly trades away interpretability for performance (without additional tools like SHAP) and is very compute-intensive.

| Metric | Value |
|---|---|
| Precision | 0.75 |
| Recall | 0.304 |
| F1 score | 0.433 |
| Accuracy | 0.943 |
| **ROCAUC** | **0.904** |

Table 4: Metrics of InceptionTime classifier

## 5.2 Length-of-stay prediction

### 5.2.1 Linear Regression

We regressed LoS (Y), and then Log(Y) on the predictors (X), separately.

From the result in table 5, the most important predictors are vital signs. Holding all other predictors fixed, a unit increase in cholesterol hdl count is associated with a 0.671 increase in LoS. Similarly, a unit increase in red blood cell count ascites change is associated with a 8.674 decrease in LoS.

| Metric | Standard | Log Transformed |
|---|---|---|
| $R^2$ | 0.166 | 1.821 |
| RMSE | 0.172 | 1.859 |

Table 5: Metrics of linear regression

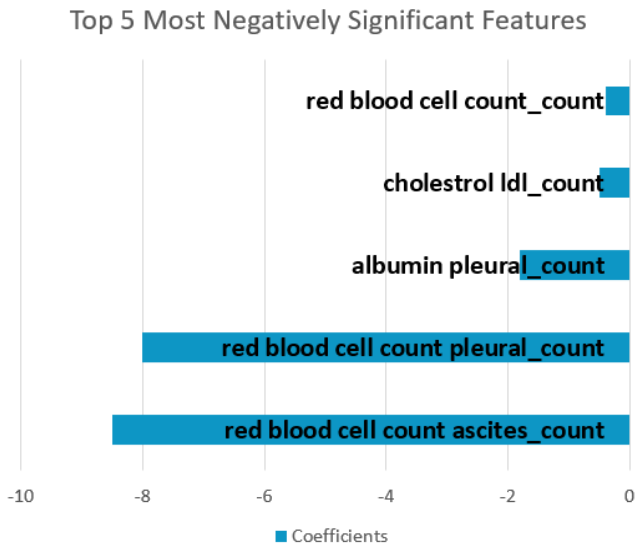


Figure 9: Most negatively significant features for linear regression model

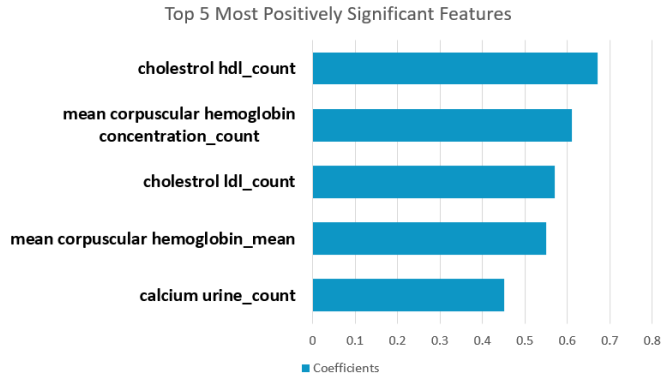


Figure 10: Most positively significant features for linear regression model

### 5.2.2 Gradient Boosting

we found that the most important variable is respiratory rate mean. Other important variables - vital signs, such as blood urea nitrogen, glascow coma scale, and heart rate also contribute significantly to the response variable.

| Metric | Value |
|---|---|
| $R^2$ | 0.211 |
| RMSE | 1.772 |

Table 6: Metrics of Gradient Boosting

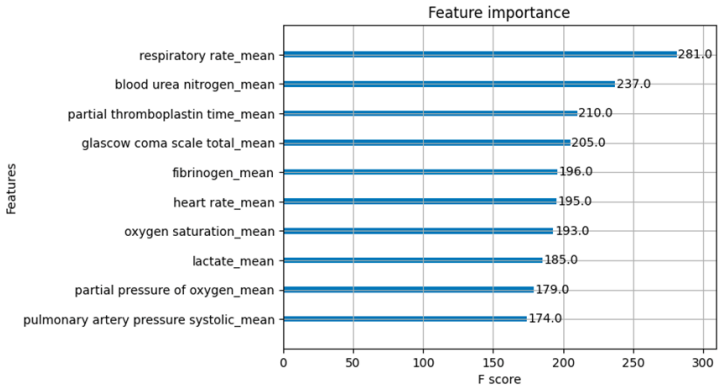


Figure 11: Feature importances for XGBRegressor model

### 5.2.3 Convolutional Neural Networks

We constructed the same network for the classification case, with no sigmoid activation at the end, and using MSE loss. And the model improves upon XGBoost in both R2 as well as RMSE.

| Metric | Value |
|---|---|
| $R^2$ | 0.211 |
| RMSE | 1.772 |

Table 7: Metrics of InceptionTime regression

### 5.3. Model Evaluation

In terms of the performance, InceptionTime outperforms the other models in almost every aspect. However, Logistic Regression and XGBoost may provide better interpretability and valuable insights through the important variables and given probabilities. XGBoost is far more computationally friendly than InceptionTime, as it requires about 24x less data (no time series component) + no GPU to train.

**Mortality Prediction: Classification**

| Metric | Logistic Regression | XGBoost | InceptionTime |
|---|---|---|---|
| Precision | 0.686 | 0.731 | **0.75** |
| Recall | 0.270 | 0.275 | **0.304** |
| F1 score | 0.388 | 0.399 | **0.433** |
| Accuracy | 0.939 | 0.941 | **0.943** |
| ROCAUC | 0.897 | **0.919** | 0.904 |

Table 8: Comparison of metrics between logistic regression, XGBoost and Inception time

**LoS Prediction: Regression**

| Metric | LR | LR(log transform) | XGBoost | InceptionTime |
|---|---|---|---|---|
| $R^2$ | 0.166 | 0.172 | 0.211 | **0.224** |
| RMSE | 1.821 | 1.859 | 1.772 | **1.756** |

Table 9: Comparison of R and RMSE between linear regression, linear regression with log transform, XGBoost and Inception time

## 6 Conclusion

After some experimentation, we concluded that the CNN model performed best in both tasks at the expense of high computational costs and interpretability. On the other hand, other models such as Logistic, Linear Regressions and XGBoost produced adequate results that are easy to interpret and reproduce.

For future work, we hope to try a different approach for curating the input data set, and perhaps attempt to fit more advanced neural network models.

## References


[1] A.D.A.M Editorial Team & VeriMed Healthcare Network. (2021, May 4). *Test - CSF cell count*. A.D.A.M. https://ssl.adam.com/content.aspx?productid=117&pid=1&gid=003625&site=makatimed.adam.com&login=MAKA1603

[2] Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., & Webb, G. I. (2020). InceptionTime: Finding AlexNet for Time Series Classification. *Data Mining and Knowledge Discovery*, *34*(1936–1962). Springer. https://doi.org/10.1007/s10618-020-00710-y

[3] Hou, N., Li, M., He, L., Xie, B., Wang, L., Zhang, R., Yu, Y., Sun, X., Pan, Z., & Wang, K. (2020, December 7). Predicting 30-days mortality for MIMIC-III patients with sepsis-3: a machine learning approach using XGboost. *Journal of Translational Medicine*, *18*(462). BMC. https://doi.org/10.1186/s12967-020-02620-5

[4] Jain, S., & Iverson, L. M. (2022, January). Glasgow Coma Scale. *StatPearls [Internet]*. NIH NCBI. https://www.ncbi.nlm.nih.gov/books/NBK513298/

[5] Liu, X., Hu, P., Mao, Z., Kuo, P.-C., Li, P., Liu, C., Hu, J., Li, D., Cao, D., Mark, R. G., Celi, L. A., Zhang, Z., & Zhou, F. (2020, January 28). Interpretable Machine Learning Model for Early Prediction of Mortality in Elderly Patients with Multiple Organ Dysfunction Syndrome (MODS): a Multicenter Retrospective Study and Cross Validation. *arXiv*. https://doi.org/10.48550/arXiv.2001.10977

[6] MedlinePlus. (2021, September 16). *White Blood Count (WBC)*. MedlinePlus. https://medlineplus.gov/lab-tests/white-blood-count-wbc/

[7] MedlinePlus. (2022, April 7). *Anion Gap Blood Test*. MedlinePlus. https://medlineplus.gov/lab-tests/anion-gap-blood-test/

[8] Meng, C., Trinh, L., Xu, N., Enouen, J., & Liu, Y. (2022, May 3). Interpretability and fairness evaluation of deep learning models on MIMIC-IV dataset. *Scientific Reports*, *12*(7166). Scientific Reports. https://doi.org/10.1038/s41598-022-11012-2

[9] Nallabasannagari, A. R., Reddiboina, M., Seltzer, R., Zeffiro, T., Sharma, A., & Bhandari, M. (2020, September 2). All Data Inclusive, Deep Learning Models to Predict Critical Events in the Medical Information Mart for Intensive Care III Database (MIMIC III). *arXiv*. https://doi.org/10.48550/arXiv.2009.01366

[10] National Institute of Diabetes and Digestive and Kidney Diseases. (2019, December). *Diagnosis of Urinary Retention*. National Institute of Diabetes and Digestive and Kidney Diseases. https://www.niddk.nih.gov/health-information/urologic-diseases/urinary-retention/diagnosis

[11] NVIDIA. (n.d.). *What is XGBoost?* NVIDIA. https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[12] Scherpf, M., Gräßer, F., Malberg, H., & Zaunseder, S. (2019, October). Predicting sepsis with a recurrent neural network using the MIMIC III database. *Computers in Biology and Medicine*, *113*. NCBI. https://doi.org/10.1016/j.compbiomed.2019.103395

[13] Tsiklidis, E. J., Sinno, T., & Diamond, S. L. (2022, January 19). Predicting risk for trauma patients using static and dynamic information from the MIMIC III database. *Plos One*. https://doi.org/10.1371/journal.pone.0262523

[14] Wang, S., McDermott, M. B. A., Chauhan, G., Hughes, M. C., Naumann, T., & Ghassemi, M. (2020). MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III. *Proceedings of the ACM Conference on Health, Inference, and Learning*. arXiv. https://doi.org/10.48550/arXiv.1907.08322

[15] Warnes, Z. (2021, October 28). *Hyper-Parameter Search in Optuna*. Towards Data Science. https://towardsdatascience.com/hyper-parameter-optimization-with-optuna-4920d5732edf

[16] Zhu, Y., Zhang, J., Wang, G., Yao, R., Ren, C., Chen, G., Jin, X., Guo, J., Liu, S., Zheng, H., Chen, Y., & Guo, Q. (2021, July 1). Machine Learning Prediction Models for Mechanically Ventilated Patients: Analyses of the MIMIC-III Database. *Clinical Application of Artificial Intelligence in Emergency and Critical Care Medicine*, *I*. Frontiers in Medicine. https://doi.org/10.3389/fmed.2021.662340