ADS1002

Mortality in ICU Patients

Prepared by

You Wei Lim
(34069518)

Regine Chong
(34901167)

Dayvin Rushil Athirathan
(33589070)

Table of contents

Executive Summary

Problem

Severe patients who are not doing well medically in a hospital are normally admitted into an Intensive Care Unit (ICU) ward. This is where there is specialized equipment in order to help patients. In addition, patients are given more care and clinical measurements are continuously being recorded. In this project, we were required to build a successful predictive model by using the provided preprocessed data set. The data set contained clinical measurements of patients and by using this, the goal is to develop a model which can predict a patient's mortality rate. This means whether a patient survives or dies in an ICU ward.

The Data

The preprocessed data given to us was taken from PhysioNet Computing in Cardiology Challenge in 2012. The raw data of 12000 ICU patients recorded was preprocessed to only about a little over 1400 ICU patients data. From this, we had to apply data wrangling which included the removal of Min. and Max. prefixes of some continuous data because we used the Mean. of the continuous variable as it gives an overall better understanding of the data instead of the extreme values. Moreover, we were given two sets of data where one ended with .x and another with .y . We decided to use the data containing .y because the overall correlation of the data ending with .y is higher compared to data ending with .x . Lastly, we checked to see for any outliers and if there were any missing values in the data that will need to be imputed. After doing the last steps, we were then ready to move to the next step.

Findings

By using correlation analysis in our exploration data analysis, we found that the top 3 correlating features with our target variable which is In.hospital_death is Mean_GCS.y , SOFA and Mean_Lactate.y . By using these features, we first decided to develop a K-Nearest Neighbor model. On run 1 the model achieved an accuracy score of 0.72. However, after fine tuning the k value, the model then had an accuracy of 0.73. Furthermore, we developed a Logistic Regression model which had an accuracy of 0.71. Next, we also developed a Decision Tree model which had an accuracy of 0.70. Lastly, we developed a Random Forest model which had the lowest accuracy score of 0.62. In conclusion, the K-Nearest Neighbors model of Run 2 should be used in predicting the mortality rate of ICU patients as it had the highest accuracy score.

Introduction

What is an ICU?

An ICU ward is found in hospitals where it stands for Intensive care unit. This is a place where medically serious cases of people are admitted into. The potential serious medical cases have certain requirements before getting classified into an ICU. These potential organ failures and also potential death stemming from a condition. ICU wards are necessary in hospitals because it provides more thorough care for patients. This includes constant collection of patients vitals, oxygen levels, temperature and many more. ICU wards typically have specialized equipment in order for the collection and in assistance such as ventilators for breathing. ICU wards are important as it provides close monitoring on patients and treatment. This allows patients to recover from their critical illness back into a normal hospital ward. However, some people admitted into an ICU ward do not survive as their condition is too critical. As close monitoring of a patient's health is done in an ICU ward, data of the patient must be thoroughly noted down. (Smith, 2022).

Supplied Data

For our project, we were supplied with a comma separated values (CSV) file which contained the data of ICU patients. The data from the CSV file were given by PhysioNet Computing in Cardiology Challenge in 2012. The data contains about 12000 ICU ward admissions being recorded. From these admissions, various medical conditions of a patient were recorded instead of just a single medical condition. This data was recorded for the first 48 hours of being admitted to the ICU ward.

The CSV file included the various features of a patient's medical record. These features includes 'RecordID', 'Age', 'Gender', 'Height', 'ICUType2', 'ICUType3', 'ICUType4', 'Weight', 'GCS', 'HR', 'NIDiasABP', 'NIMAP', 'NISysABP', 'RespRate', 'Temp', 'Urine', 'BUN', 'Creatinine', 'Glucose', 'HCO3', 'GCT', 'Mg', 'Platelets', 'K', 'Na', 'WBC', 'pH', 'PaCO2', 'PaO2', 'DiasABP', 'FiO2', 'MAP', 'MechVent', 'SysABP', 'SaO2', 'Albumin', 'ALP', 'ALT', 'AST', 'Bilirubin', 'Lactate', 'Cholesterol', 'TroponinI', 'TroponinT', 'Albumax', 'SAPS.I', 'SOFA' and 'In.hospital_death'.

The above includes demographic information such as a person's age and gender. There were clinical measurements such as a person's heart rate and blood pressure which were continuous data. There were outcome features called "In-hospital death" which were stored as boolean variables in 1's (which means that a patient died) and 0's (which means that a patient survived) for the feature of In-hospital death. Many sequential time series variables were also present in the CSV file but the "RecordID" feature stood out as it showed the order of patients which were admitted in the ICU ward.

Objective

The objective of this project is to create a predictive model for the mortality of ICU patients. This should be done by using existing data from the PhysioNet Challenge 2012 data given to us. The target feature in this project is "In-hospital death" as this is the feature which shows the mortality of a patient. Therefore, supervised learning models should be used in creating a successful model which can be used to forecast the likelihood of an ICU patient surviving based on their medical measurements recorded.

Furthermore, this project encourages the use of algorithms in order to create the predictive model. This includes algorithms such as K Nearest Neighbors, Logistic Regression and Decision Trees which will be further explained in the latter part of the report.

The integral objective of this project is to develop an accurate predictive model. This means that in whichever supervised learning technique used to create the final predictive model, it should have a good score in accuracy which includes the precision and recall score. Therefore, enhancing the accuracy of the model is crucial as it has the ability in predicting correctly or falsely.

Preprocessing of Data

The data given unto us were already preprocessed from the raw original data given by PhysioNet Challenge in 2012. Some preprocessing was done by PhysioNet and also from Monash. This can be seen from the difference of the raw data from the original website and also the data given to us for the project by Monash which we were supposed to use. Firstly, from the 12000 ICU patients recorded, ICU admissions of less than 48 hours were removed. This is because the patient did not hit the criteria of being admitted for 48 hours.

Secondly, extraneous age of patients where it could be mistakenly recorded were removed from the dataset given to us. Next, the features of 3 different "ICUType" (ICUType2, ICUType3 & ICUType4) were included in the data to us indicating the seriousness of a patient where if all of it were 0's then it shows that the "ICUType" is at the lowest level of "ICUType1". This was done by data engineering.

Therefore, the preprocessing of the raw data included the removal of erroneous data and data engineering. This makes the final data set given to us to work with to come up with a predictive model to contain 1474 rows and 232 columns as shown in Figure 1 below.

	RecordID	Age	Gender	Height	ICUType2	ICUType3	ICUType4	Mean_Weight.x	Mean_GCS.x	Mean_HR.x	...	Max_ALT.y	Max_AST.y	Max_Bilirub
0	132543	68	1.0	180.300000	0	1	0	84.600000	14.888889	72.971429	...	12.000000	15.000000	0.200
1	132545	88	0.0	169.787227	0	1	0	83.054136	15.000000	79.520000	...	189.169186	260.323666	1.979
2	132547	64	1.0	180.300000	0	0	0	114.000000	8.333333	81.318182	...	60.000000	162.000000	0.400
3	132551	78	0.0	162.600000	0	1	0	48.400000	13.250000	78.125000	...	46.000000	82.000000	0.300
4	132554	64	0.0	169.787227	0	1	0	60.700000	15.000000	129.363636	...	189.169186	260.323666	1.979
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1469	142661	89	1.0	177.800000	0	0	1	64.000000	11.214286	86.260870	...	189.169186	260.323666	1.979
1470	142662	86	1.0	162.600000	0	1	0	53.000000	13.000000	85.517241	...	28.000000	35.000000	0.900
1471	142664	51	0.0	169.787227	0	0	1	75.000000	9.909091	91.147059	...	189.169186	260.323666	1.979
1472	142665	70	0.0	169.787227	0	0	1	87.000000	10.833333	101.083333	...	189.169186	260.323666	1.979
1473	142671	37	1.0	169.787227	0	1	0	87.400000	6.400000	91.785714	...	1513.000000	1277.000000	0.600

1474 rows × 232 columns

Figure 1: Dataframe of Preprocessed Data

Data Wrangling

Before we could get into Exploratory Data Analysis, we first needed to examine the data frame for ourselves. This is so that we can see if there is any cleaning which can be done to the data. After cleaning, the data frame would then allow for clearer analysis to be done on it. Firstly, as we can see from Figure 1 above there are variables containing prefixes of Mean and Max. There were also prefixes containing Min for variables. The lowest, highest and average results of each clinical continuous measurement were all recorded in the data set. For this project we decided to remove the variables with prefixes containing Min and Max. This is because the Min and Max values are the more extreme set of values of the features. The Min and Max values could potentially cause overfitting and underfitting when creating a predictive model. Therefore in this project, continuous features with prefixes of Mean were used as it takes into account all data points. This will ensure a better predictive model is created when using the data for training a model.

Secondly, an issue we faced was there were two sets of features where one ends with (.x) and (.y) respectively. These features can be seen in Figure 1 where it has Mean_Weight.x and also Max_ALT.y. This means that there is also the opposite of it containing Mean_Weight.y and also Max_ALT.x. After getting confirmation, we were told that we can choose either set of data to build our predictive model. We chose to get features ending with .y and remove all features ending with .x. We came to this decision by getting the absolute correlation values of the features ending with .x and .y respectively with the target feature which is the In.hospital_death. Figure 2 below shows the top 5 absolute correlation features. Therefore, we can see that the top 5 features ending with .y has a higher overall correlation value than the features ending with .x.

	Feature	ICU_y_Correlation
0	Mean_GCS.y	0.386052
1	SOFA	0.225708
2	Mean_Lactate.y	0.177576
3	SAPS.I	0.167370
4	Mean_HCO3.y	0.160508

	Feature	ICU_x_Correlation
0	Mean_GCS.x	0.242397
1	SOFA	0.225708
2	SAPS.I	0.167370
3	Mean_Lactate.x	0.161595
4	Mean_Bilirubin.x	0.140048

Figure 2: Top 5 absolute correlation of features in dataframe ending with .x and .y respectively

Lastly, we checked for outliers in the dataset by using a robust technique where we tried to remove any values not in the lower and upper bounds of data. This was done by getting the interquartile range and using the formula to get the lower bound (Q1 - 1.5 x IQR) and the upper bound (Q3 + 1.5 x IQR) as seen in the code below. However, after filtering out the data from dataframe, we realized that there were only 17 rows of data present.

```
In [47]: # Calculate Q1 (25th percentile) and Q3 (75th percentile)
         Q1 = ICU_filtered.quantile(0.25)
         Q3 = ICU_filtered.quantile(0.75)
         IQR = Q3 - Q1  # Interquartile Range

         # Define lower and upper bounds for outliers
         lower_bound = Q1 - 1.5 * IQR
         upper_bound = Q3 + 1.5 * IQR

         # Filter out rows with outlier values
         ICU_cleaned = ICU_filtered[~((ICU_filtered < lower_bound) | (ICU_filtered > upper_bound)).any(axis=1)]

         print("Original DataFrame shape:", ICU_filtered.shape)
         print("Cleaned DataFrame shape:", ICU_cleaned.shape)

         Original DataFrame shape: (1474, 47)
         Cleaned DataFrame shape: (17, 47)
```

Therefore, we decided in the end to not remove the outliers based on the Interquartile technique as this would mean that we do not have sufficient data to proceed with creating a predictive model. Hence, after our data wrangling steps, our final data set that we were working to create the predictive model contains 1474 rows and 47 columns as seen in Figure 3 below.

	RecordID	Age	Gender	Height	ICUType2	ICUType3	ICUType4	Mean_Weight.y	Mean_GCS.y	Mean_HR.y	...	Mean_ALT.y	Mean_AST.y	Mean_Bilirul
0	132543	68	1.0	180.300000	0	1	0	84.600000	15.000000	68.200000	...	12.000000	15.000000	0.2
1	132545	88	0.0	169.787227	0	1	0	83.308061	15.000000	70.000000	...	178.687645	236.484503	1.9
2	132547	64	1.0	180.300000	0	0	0	114.000000	8.888889	94.880000	...	52.500000	104.500000	0.4
3	132551	78	0.0	162.600000	0	1	0	48.400000	9.600000	65.341463	...	46.000000	82.000000	0.3
4	132554	64	0.0	169.787227	0	1	0	60.700000	15.000000	125.291667	...	178.687645	236.484503	1.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1469	142661	89	1.0	177.800000	0	0	1	64.000000	10.777778	85.833333	...	178.687645	236.484503	1.9
1470	142662	86	1.0	162.600000	0	1	0	53.000000	14.666667	77.869565	...	28.000000	35.000000	0.9
1471	142664	51	0.0	169.787227	0	0	1	75.000000	10.000000	102.458333	...	178.687645	236.484503	1.9
1472	142665	70	0.0	169.787227	0	0	1	87.000000	14.750000	97.642857	...	178.687645	236.484503	1.9
1473	142671	37	1.0	169.787227	0	1	0	87.400000	4.333333	84.583333	...	1513.000000	1277.000000	0.6

1474 rows × 47 columns

Figure 3: Data frame after Data Wrangling

Exploratory Data Analysis

As a part of our exploratory data analysis, we decided to get the correlations of different features with the target feature which is the In-hospital_death. Correlation is an important term which helps in understanding the relative strength one feature has on another feature (Taylor, 2023). Therefore, understanding which features play a role in determining the target feature is important as it will help in the development of the predictive model. Firstly, we decided to get the top 10 absolute correlation values relative to the target feature and Figure 4 below shows the data frame which was created for it.

	Feature	Absolute Correlation
0	Mean_GCS.y	0.386052
1	SOFA	0.225708
2	Mean_Lactate.y	0.177576
3	SAPS.I	0.167370
4	Mean_HCO3.y	0.160508
5	Mean_Bilirubin.y	0.152299
6	Mean_BUN.y	0.137934
7	Mean_HR.y	0.135712
8	ICUType2	0.131581
9	Mean_WBC.y	0.130429

Figure 4: Top 10 Absolute correlation of features to the target feature

After getting this, we saw that the correlations were relatively quite low. This made us narrow down the numbers to the top 6 absolute correlation and in order to visualize this, we decided to create a bar plot for it as seen below in Figure 5.

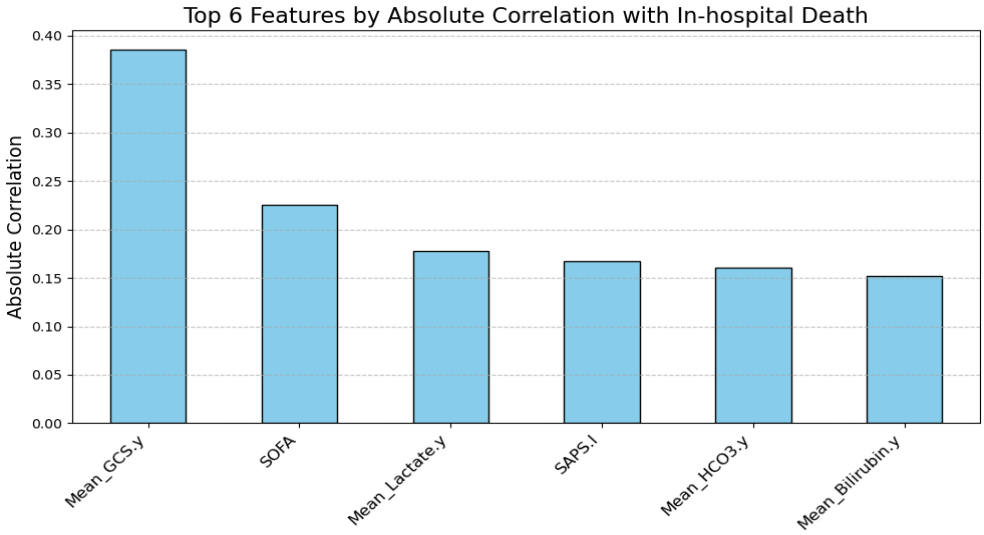

Figure 5: Barplot of Top 6 Absolute correlation of features to the target feature

By eyeballing the barplot, we saw that there is a very slight correlation score difference beginning from SAPS.I which is the top 4th correlation score with the lower correlation score features. Therefore, we decided to use the top 3 absolute correlation which is Mean_GCS.y , SOFA and Mean_Lactate.y in developing our predictive model. The top 3 features with highest correlation value to the target feature can be seen below in Figure 6.

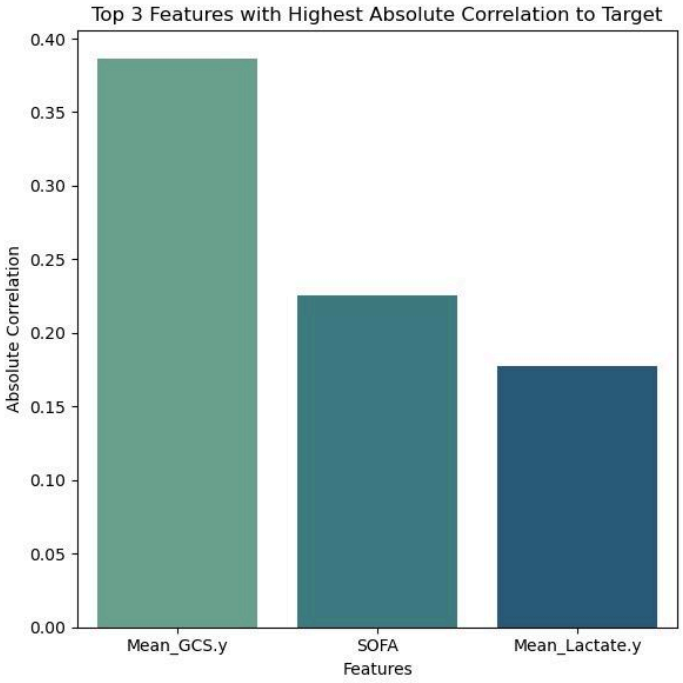

Figure 6: barplot of Top 3 Absolute correlation of features to the target feature

Predictive Model Development and Findings

K- Nearest Neighbors

K-Nearest Neighbors is a supervised learning algorithm where it attempts to class the testing data correctly. This is done by getting the distance of a test data with all the data points of the training data. After this, k number of data points closest to the test data is chosen. It then follows the majority of the k number of points to place the test data into a certain class. (Christopher, 2021).

We decided to use the K-nearest neighbors algorithm as a starting predictive model because it is able to handle classification where in this case is whether a person is in class 1 (has died in ICU) or a person is in class 0 (has survived in ICU). For this algorithm we used the top 3 highest correlations from Figure 6 as the predictors. Firstly, we had to initialize the train-test split for the model. We used an 80-20 split for the training and testing data respectively. Secondly, we had to get a suitable k value for the algorithm and we decided to use the square root method as seen in the code below.

```
# Finding k using the sqrt method
k = np.sqrt(X_train.shape[0])
# k=34.336569426778794
```

Next, we had to create the classifier model with the k = 34 because this is what was obtained from the square root method as seen in the code above. Then we had to train the model and test it. Figure 7 below shows the accuracy of the K-Nearest Neighbor algorithm used for this initial run 1.

	precision	recall	f1-score	support
0	0.73	0.91	0.81	192
1	0.69	0.37	0.48	103
accuracy			0.72	295
macro avg	0.71	0.64	0.65	295
weighted avg	0.72	0.72	0.70	295

Figure 7: Classification report of the k nearest neighbors model in run 1

Based on Figure 7 above we can see that the model has a precision of 0.69 for class 1 where the model correctly predicts that a patient dies 69% of the time. The recall score on the other hand for class 1 is only 0.37 indicating that from all the patients who died, the model is only able to predict 37% of them. The F1 score for class 1 is 0.48 which shows a slightly below average in predicting the death of patients. Furthermore, the accuracy score is 0.72 meaning that it is moderately able to predict the outcome. However, the model could be further improved.

In order to improve the model, we used the elbow method (a method which selects a k-value by looking at the result given by every k) to find the optimal value of k. We chose a k value in the range from 0 to 40. For each value of k in the range, we fitted the K-Nearest Neighbor model on the training set and got the error rate on the testing set. After iterating the process for every k value, we plotted it on a graph (Figure 8).

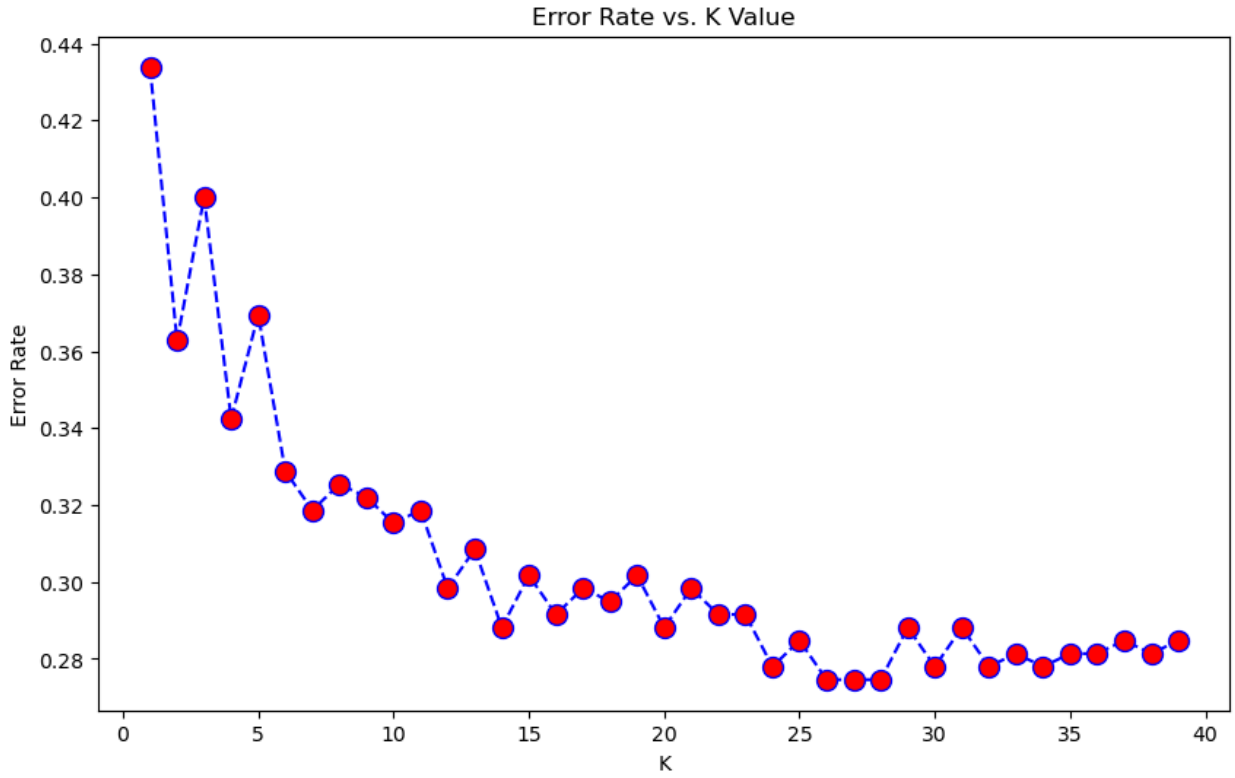

Figure 8: Graph of error rate against K value

As seen in Figure 8, when k = 26, 27 and 28, the error rate is the lowest. Therefore, we can choose one of these 3 values. Here we picked the k value as 26 and then retrained the model with this value. Figure 9 below shows the accuracy of the K-Nearest Neighbor algorithm used for this second run.

	precision	recall	f1-score	support
0	0.73	0.92	0.81	192
1	0.71	0.36	0.48	103
accuracy			0.73	295
macro avg	0.72	0.64	0.65	295
weighted avg	0.72	0.73	0.70	295

Training accuracy: 0.73
Testing accuracy: 0.73

Figure 9: Classification report of k nearest neighbor model in run 2

Therefore as seen in Figure 9 above, the precision score of class 1 has increased in run 2 to 0.71 meaning that the model correctly predicts that a patient dies 71% of the time. In addition, the recall score of class 1 is now 0.36 which means that from all the patients who died, the model is only able to predict 36% of them. The F1 score for class 1 remains the same at 0.48 which still shows a slightly below average in predicting the death of patients. A slight improvement in accuracy score in run 2 than run 1 is seen where it is now 0.73 which means that the prediction of patients for this model of KNN which uses k = 26 is predicting the class correctly 73% of the time. The accuracy of the training and testing is the same at 0.73 which further supports that the model is able to identify patterns within the data which is not overfitting or underfitting.

Logistic Regression Model

Logistic Regression is a supervised machine learning algorithm that is used for binary classification, where the goal is to predict the probability of an outcome (Kanade, 2022). In this project, we aim to predict the mortality of ICU patients, where there are only two possible outcomes (survival or death).

We performed binary logistic regression in a few steps. First, we used the selected features 'Mean_GCS.y', 'SOFA', and 'Mean_Lactate.y' as the predictors. We split them into training and testing sets and then standardized the features to prevent certain features from dominating the model due to their larger values. After that, we trained the logistic regression model and finally, we made predictions and evaluated the model.

	precision	recall	f1-score	support
0	0.75	0.84	0.79	192
1	0.61	0.48	0.54	103
accuracy			0.71	295
macro avg	0.68	0.66	0.66	295
weighted avg	0.70	0.71	0.70	295

Training accuracy: 0.71
Testing accuracy: 0.71

Figure 10: Classification report of logistic regression model

Figure 10 above shows the classification report for the model.

The precision score of 0.61 indicates that out of all the patients that the model predicts will be dead, only 61% actually do. The recall score of 0.48 indicates that out of all the patients who actually die, the model only predicts this result correctly for 48% of those patients. The f1-score of 0.54 tells us that the model has an average performance in predicting whether or not patients will die.

Most importantly, the accuracy score of 0.71 is considerably acceptable as it correctly predicts 71% out of 295 patients whether or not they will die. Besides, both the accuracy of training and testing are the same at 0.71, which tells us that the model is neither overfitting nor underfitting the training data.

We also plotted a confusion matrix of the model for a better understanding of the performance measurement.

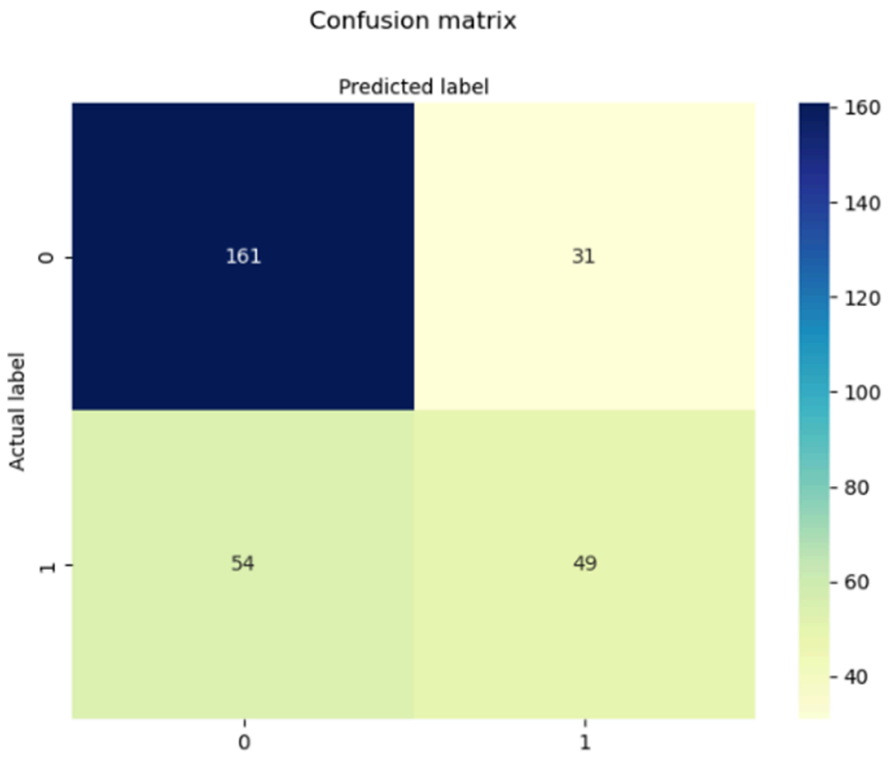

Figure 11: Confusion matrix comparing false and true, positives and negatives as tested by the trained logistic regression model on the rest of the data set

In Figure 11 above, class 0 indicates survival (negative), and class 1 indicates mortality (positive). The model is better at correctly predicting class 0 than class 1. However, there is a notable number of false negatives, indicating the model tends to underpredict class 1.

A ROC (Receiver Operating Characteristic) plot is a graphical representation used to evaluate the performance of a binary classifier, such as logistic regression. It shows how well the model distinguishes between two classes.

In logistic regression, we get probabilities (a score between 0 and 1) for each prediction. These probabilities can be converted into binary decisions by choosing a threshold (e.g., classify as positive if the probability is above 0.5). The ROC curve shows how the TPR (True Positive Rate) and FPR (False Positive Rate) change as we vary this threshold. Thus, we plotted a ROC curve for the model (Figure 12).

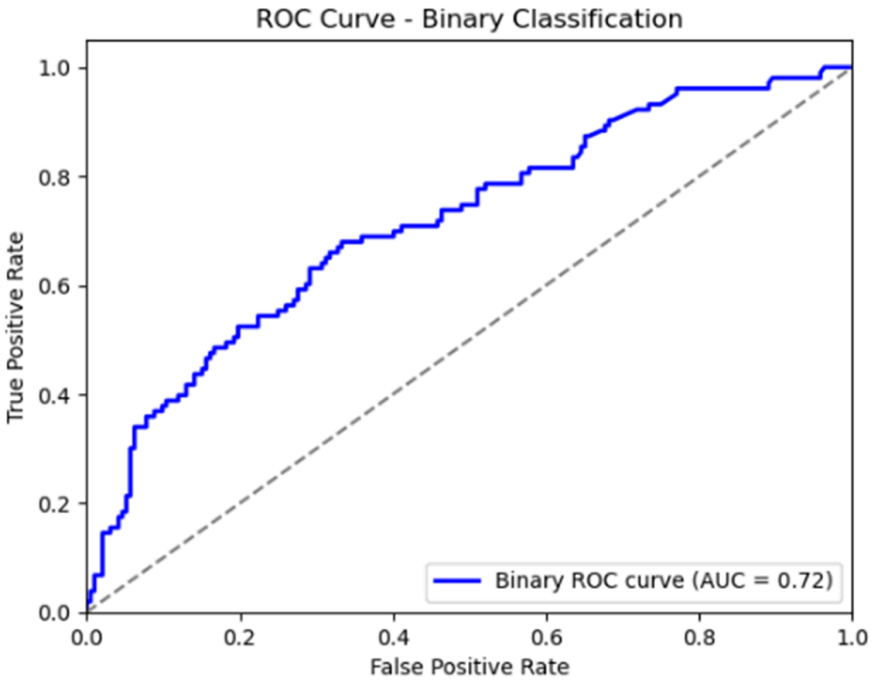

Figure 12: ROC Curve of the logistic regression model

As seen in Figure 12, the shape of the ROC curve is irregular rather than stair-like because the predicted probabilities are continuous rather than discrete. Each threshold produces a different point, creating many distinct FPR and TPR pairs. Besides, the complexity of the model leads to this shape of ROC.

The AUC (Area Under the Curve) value of 0.72 is considerably fair. As it is greater than 0.5, we can say that this model is meaningful as it can distinguish between patients who survived and those who didn't.

In conclusion, the logistic regression model has an average performance in predicting the mortality of patients. The accuracy score of 0.71 and the AUC value of 0.72 are acceptable.

Decision Tree

We chose decision tree as a predictive model because it provides a clear and interpretable structure and it is able to handle large amounts of data. In critical settings like ICU mortality prediction, a decision tree's simple structure would show the most significant feature related to mortality rate. This allows healthcare professionals to make quick decisions on which patients to focus on. As seen in the decision tree (Fig. 13), the most significant feature towards the mortality rate is Mean_GCS.y followed by Mean_Lactate.y and SOFA.

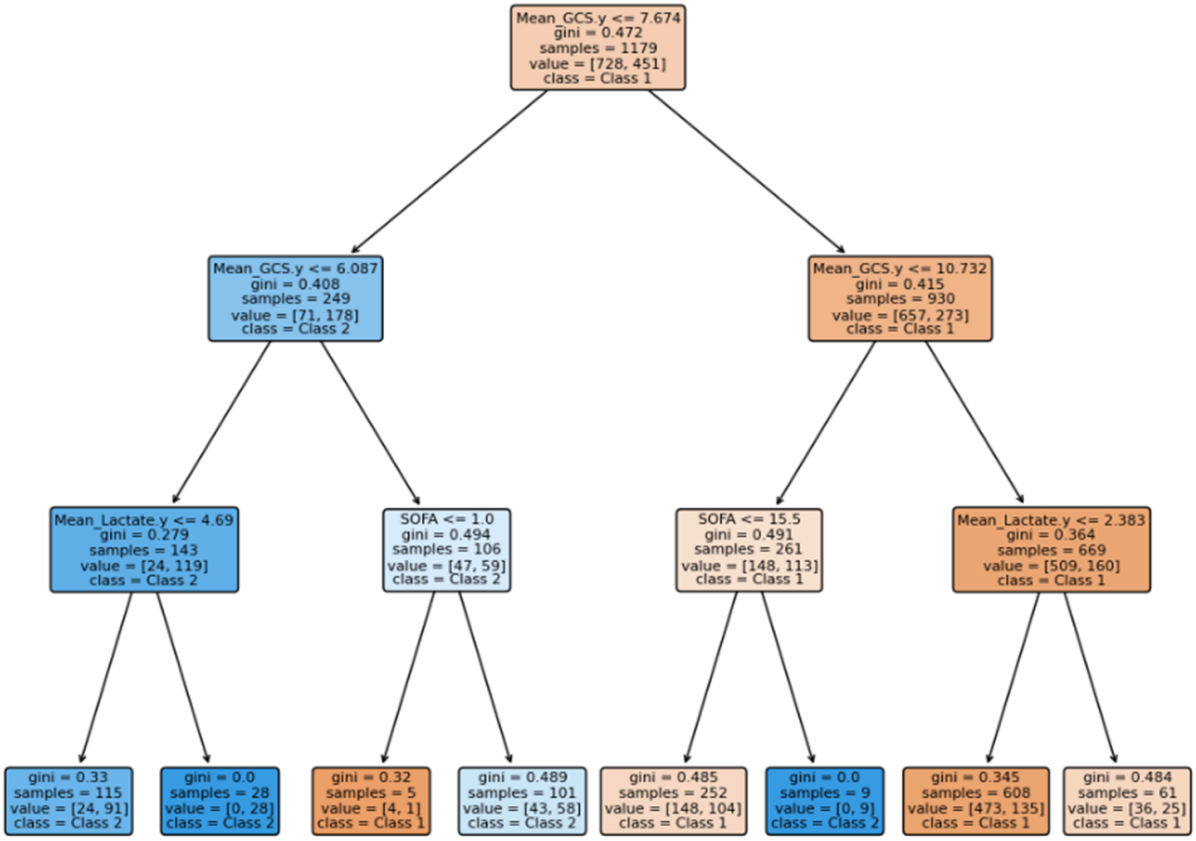

Figure 13: Decision tree

In the confusion matrix (Fig. 14), the patterns are about the same as the previous models. The model predicts survival (class 0) much better than death (class 1). The high number of false negatives indicates that the decision tree model is underpredicting death (class 1).

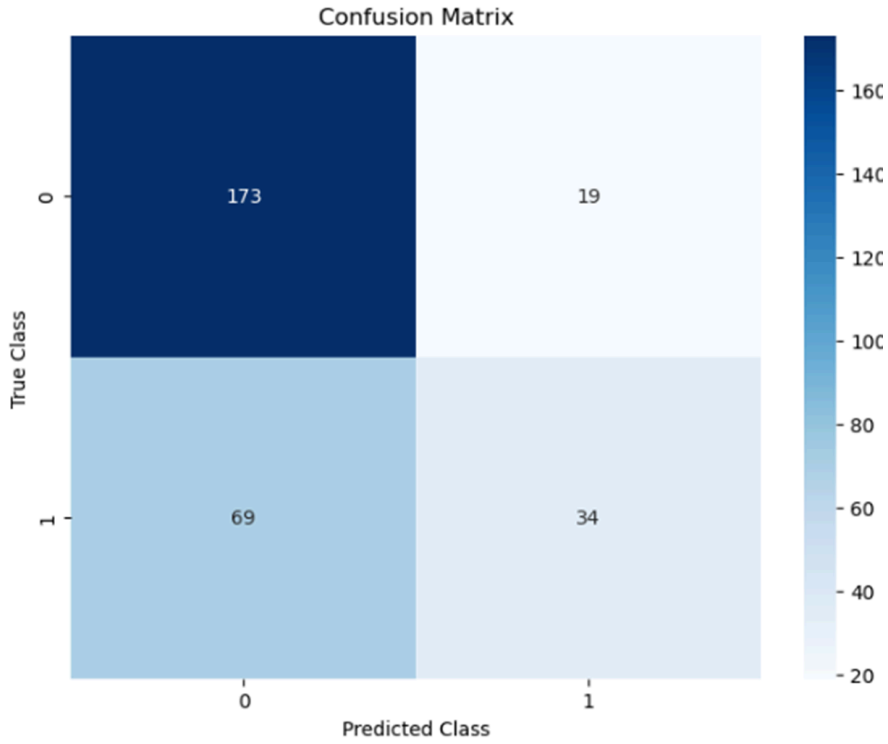

Figure 14: Confusion matrix comparing false and true, positives and negatives as tested by the trained decision tree model on the rest of the data set

As can be seen in the classification report (Fig. 15), the accuracy score of the decision tree is moderately high at 0.70. However, the training accuracy is slightly higher than the testing accuracy. This small gap is fine. To explore a better model, our group decided to experiment with another model related to decision tree modeling.

```
Test set accuracy with selected features: 0.7017

Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.90      0.80       192
           1       0.64      0.33      0.44       103

    accuracy                           0.70       295
   macro avg       0.68      0.62      0.62       295
weighted avg       0.69      0.70      0.67       295

Training accuracy: 0.72
Testing accuracy: 0.70
```

Figure 15: Classification report of the decision tree model

Random Forest Classification

Random forests are an ensemble method, combining multiple decision trees, which would be able to capture more complex and nonlinear relationships in the data. As seen in the confusion matrix (Fig. 16), the model predicts survival (class 0) much better than death (class 1). The high number of false negatives indicates that the model is underpredicting death.

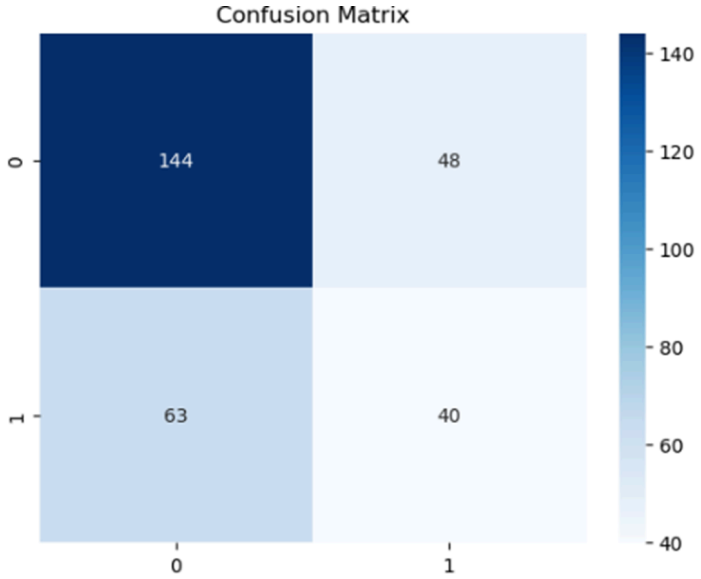

Figure 16: Confusion matrix comparing false and true, positives and negatives as tested by the trained RFC model on the rest of the data set

From the classification report (Fig. 17), the accuracy score using the RFC model is the lowest among all the predictive models used, at 0.62. The training accuracy is way higher than the testing accuracy. The big gap between training accuracy and testing accuracy indicates that the model is highly overfitted. Thus, the performance of the RFC model is far poorer than the decision tree model.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.75      0.72       192
           1       0.45      0.39      0.42       103

    accuracy                           0.62       295
   macro avg       0.58      0.57      0.57       295
weighted avg       0.61      0.62      0.62       295

Training accuracy: 0.96
Testing accuracy: 0.62
```

Figure 17: Classification report of the RFC model

Conclusion

Once we had removed all the (.x), minimum, and maximum columns from the preprocessed dataset, we decided to remove the outliers of the dataset. However, we are left with only 17 rows of data which is not sufficient to conduct data analysis. Hence, we decided to continue with the outliers. After conducting some exploratory data analysis, it was demonstrated that Mean_GCS.y, Mean_Lactate.y, SOFA, SAPS.I and Mean_HCO3.y are significantly correlated to the contribution in predicting ICU mortality rate. Hence all the other variables were excluded.

Overall, our most successful model was the K Nearest Neighbor (KNN) model. This model achieved an accuracy of 72% before adjusting the k-value and an accuracy of 73% after. It has outperformed the Logistic Regression model, which had an accuracy of 71%, the Decision Tree model at 70%, and the Random Forest model at 62%. Notably, the Random Forest model exhibited significant overfitting, with a training accuracy of 96% but a much lower testing accuracy of 62%.

The KNN model's strength lies in its simplicity and its ability to make predictions based on the closest data points in the feature space. By tuning the K value, which determines how many neighbors are considered when making predictions, we could achieve better accuracy. After experimenting with the K value with the lowest error rate, the model's performance improved, demonstrating the KNN model's sensitivity to parameter tuning. The most important variables identified were Mean_GCS.y, Mean_Lactate, and SOFA score. Future improvements for this project would be to include more significant variables in the predictive model to improve the accuracy of the ICU mortality rate prediction.

Reference List

Christopher, A. (2021, February 2). *K-Nearest Neighbor.* Medium. https://medium.com/swlh/k-nearest-neighbor-ca2593d7

Kanade, V. (2022, April 8). *Everything you need to know about logistic regression.* Spiceworks Inc. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

Smith, L. (2022, October 10). *What is an intensive care unit (ICU)?* WebMD. https://www.webmd.com/a-to-z-guides/what-is-an-intensive-care-unit-icu

Taylor, S. (2023, November 21). *Correlation*. Corporate Finance Institute. https://corporatefinanceinstitute.com/resources/data-science/correlation/