Patient Survival Prediction - A predictive and prescriptive model of ICU patients' mortality

Proag Tanvee, 2422814
University of Mauritius

Abstract

This study is an attempt to create a predictive and prescriptive model of the death rate of patients in the ICU ward using a dataset of characteristics of patients. These include the general and medical attributes of the patient and syndromes they may be suffering from. Logistic Regression and Random Forest Classifier classification methods are used to predict the chance of mortality and the performance is evaluated - the accuracy is above 0.9. Furthermore, to assist in the allocation of resources within a hospital, each patient is evaluated based on their characteristics to determine whether it would be more efficient to assign them to a doctor (i.e., allocate a resource) or place them in palliative care.

1. Introduction

1.1 Problem Statement

The healthcare industry is facing challenges due to the lack of quick and efficient data analysis. Medicine today generates vast amounts of data, including patient records, lab results and clinical observations but much of this information remains underutilized. The slow pace at which this data is processed and analyzed can have serious consequences impacting patient care and healthcare efficiency. One major issue is delayed decision-making. Doctors and medical professionals often rely on their experience and intuition to make critical decisions but without real-time data analysis, there is a risk of overlooking patterns or trends that could indicate a more accurate diagnosis or a better course of treatment.

1.2 Data Analysis in Medicine

Analyzing and predicting deaths using hospital data can improve healthcare outcomes by saving lives. Intensive Care Units (ICUs) treat critically ill patients where timely decisions can significantly impact patient survival. By applying data analysis to ICU patient records, hospitals can identify patterns and factors that influence mortality rates. This helps healthcare professionals make better clinical decisions and improve patient care. Predictive models can assist doctors in assessing the risk of death for each patient based on their medical history, physiological measurements and laboratory results. This facilitates early detection of high-risk patients which leads to faster interventions and personalized treatment plans. Understanding which factors such as age, blood pressure or pre-existing medical conditions contribute most to patient outcomes can also help in developing new treatment protocols.

2. Literature Review

Related past works

1. *Classification based on event in survival machine learning analysis of cardiovascular disease cohort* by Mukhtar & Muhammed (2023) tested how well machine learning models can predict if heart disease patients will survive or not. The dataset had 919 patients which is a large number for a medical study. The researchers agreed that machine learning helps quickly analyze large amounts of data and find hidden patterns that may not be immediately apparent. Different models were used to predict if patients would live or die. The Random Forest model gave the best results with an accuracy score of 0.934 showing it can reliably predict patient outcomes. This shows that classification methods can be used to accurately predict patient outcomes in a clinical setting.

2. *Predicting survival of patients with heart failure by optimizing the hyperparameters* by Akin et al. (2024) uses machine learning to predict survival in 299 heart failure patients based on 11 health features. Seven models were tested, including Decision Trees, SVM and Naive Bayes. SVM and Naive Bayes had the best performance for early diagnosis with fewer false negatives while Decision Trees were easier for doctors to understand but risked missing some diagnoses. The limitation of this study is that models may not apply to all patients as the data is from one country only. Future work could improve model performance with more data and better optimization techniques.

3. In *Pancreatic Cancer Survival Prediction: A Survey of the State-of-the-Art* by Bakasa et al. (2021) reviews machine learning methods used to predict the survival of pancreatic cancer patients, including statistical and deep learning techniques. ML has shown to be effective in predicting survival using data like genetics, clinical information, and medical images. It was concluded that good data collection, design and validation are important for accurate predictions. The authors agree that there is a need for models that can predict survival for each patient and suggest personalized treatments. They believe that deep learning methods are the most promising for combining different types of data and helping doctors make better decisions.

3. Data

This study uses the Patient Survival Prediction dataset from Kaggle by Agarwal (2021), comprising above 91,600 ICU patient records across 149 hospitals. The dataset captures hospital/patient identifiers, demographic characteristics (age, weight, height, BMI etc.), physiological measurements (like respiratory rate, heart rate, systolic and diastolic blood pressure), laboratory results (like potassium levels, glucose levels), medical history (like diabetes mellitus status, prior surgeries) and clinical outcomes. The binary outcome variable *hospitaldeath* indicates patient survival (0 for 83,798 cases) or mortality (1 for 7915 cases) revealing significant class imbalance that reflects real-world ICU patient distributions. To minimize bias, the dataset includes records of patients aged in the wide range of 16 to 89 from 8 different ethnicities. It contains no duplicate entries, combines numeric and categorical features and it has some missing values.

4. Methodology

4.1 Tools

The tools and libraries used in this project are essential for performing data analysis, preprocessing and model training. The PANDAS library is used for loading, manipulating and analyzing structured data efficiently. NUMPY supports numerical operations and array manipulation which are necessary for mathematical computations. MATPLOTLIB and SEABORN are visualization libraries that help in creating graphs and charts to explore the dataset. To handle missing values, the SIMPLEIMPUTER class from SKLEARN.IMPUTE is used. The dataset is split into training and testing sets using the TRAIN_TEST_SPLIT function from SKLEARN.MODEL_SELECTION. Feature scaling is performed using STANDARDSCALER to normalize the input features. For model building, LOGISTICREGRESSION and RANDOMFORESTCLASSIFIER from SKLEARN.ENSEMBLE are employed to predict patient outcomes. The models are evaluated using several metrics such as accuracy, precision, recall and F1-score, which are calculated with functions from SKLEARN.METRICS. Additionally, CONFUSION MATRIX and CLASSIFICATION REPORT provide a detailed performance summary. SCIPY's MINIMIZE function is used to implement Gradient Descent optimization for fine-tuning certain model parameters.

4.2 Data preprocessing

4.2.1 Handle missing values

At first, missing and duplicate values which are common issues in real-world datasets are addressed. A threshold (50% of the dataset's length) is used to drop columns where more than half of the values are missing as such columns are likely to contain insufficient information for analysis. The DROPNA() method with the thresh parameter ensures that only columns with fewer than 50% non-null values are removed. Then, duplicate rows are removed using the DROP_DUPLICATES() method to avoid redundant data that could bias model training. The dataset is then separated into numerical features (columns with data types int64 or float64) and categorical features (columns with object data type) using the SELECT_DTYPES() method. Handling missing values differs based on the feature type. For numerical features, missing values are imputed using the median strategy with the SIMPLEIMPUTER() class. The median is chosen because it is less sensitive to outliers compared to the mean, making it more stable for medical datasets. Categorical features are imputed with the most frequent value (mode) to ensure that missing values are filled with the most common category thus preserving the data distribution. After fitting the imputers to the data, the missing values are replaced using the FIT_TRANSFORM() method and the updated dataset is returned.

4.2.2 Preparing the data

The dataset is preprocessed by preparing it for machine learning algorithms. First, the categorical features are identified using the SELECT_DTYPES() method. Categorical variables are then converted into one-hot encoded features using the PD.GET_DUMMIES() method which creates binary columns representing each category. This transformation is necessary because most machine learning models cannot work directly with categorical data. Next, the target variable *hospital death* is separated from the feature set. The independent features are stored in X while the target variable is stored in y. The drop() method is used to exclude the target column from the feature set. To ensure the machine learning algorithms perform optimally, the numerical features in X are standardized using the StandardScaler() class from Scikit-Learn. Standardization scales the features to have a mean of 0 and a standard deviation of 1 which helps improve model performance by giving equal importance to all features and preventing large magnitude features from dominating smaller ones. The function returns the standardized feature matrix X, target variable y, the list of feature names and the fitted scaler object. This preprocessing ensures that the dataset is clean, consistent and ready for model training.

4.3 Exploratory Data Analysis

The exploratory data analysis (EDA) process begins with an initial inspection of the dataset structure and content. The first five rows are displayed using data.head() to showcase feature values and data formats followed by data.info() to summarize column names, data types and non-null counts. Descriptive statistics for numerical features including central tendencies are generated via data.describe(). Missing values are identified by calculating null counts per column with only columns containing missing values printed to highlight data gaps. The analysis then categorizes features by data type, counting numerical (int64/float64) and categorical (object) columns to understand the composition of the dataset.

Visual exploration starts with histograms for 10 key numerical features (i.e., age, BMI and vital signs) plotted using Seaborn, with kernel density estimation (KDE) curves to assess distributions. The bar chart analyzing the *ethnicity* feature reveals a diverse representation across multiple demographic groups with distinct frequencies observed for each category. This varied distribution of ethnic backgrounds in the dataset helps mitigate potential selection bias ensuring the study accounts for population heterogeneity. Correlation analysis via a heatmap reveals relationships between numerical features. For deceased patients (*hospital death = 1*), a scatter plot compares age and BMI to identify potential mortality patterns, A bar chart is used to quantify comorbidities (e.g., diabetes, cancer) among non-survivors to show the diseases they had prior. Box plots for age, BMI and weight are stratified by survival status to compare distributions and detect outliers.

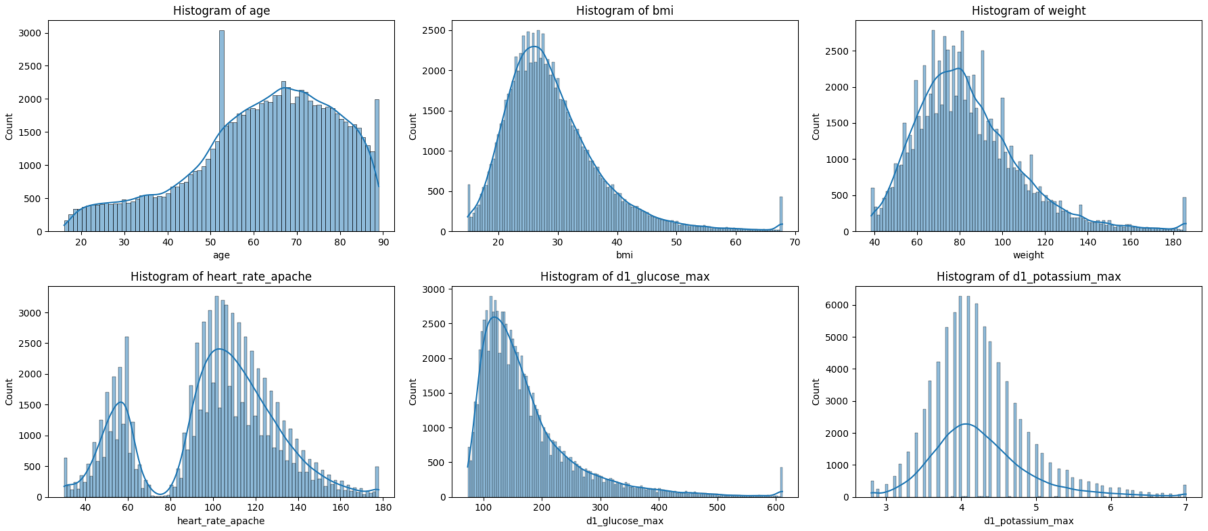

Figure 1: Histograms of patient characteristics

4.4 Modeling Training

4.4.1 Logistic Regression

Logistic Regression is used for binary classification tasks where the goal is to predict one of two possible outcomes, i.e. here, hospital_death = 0 or 1. Logistic regression estimates the probability of an event occurring based on the input features. It uses the logistic (sigmoid) function to map any real-valued input into a range between 0 and 1.

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

- $P(y = 1|x)$ represents the probability that the outcome is 1 (i.e. hospital death) given the input features.
- β_0 is the intercept term.
- $\beta_1, \beta_2, \ldots, \beta_n$ are the model coefficients (weights) for each feature.
- $x_1, x_2, \ldots, x_n$ represent the input features (e.g., age, BMI, glucose levels).
- The exponential function $e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}$ maps the linear combination of input features to a probability between 0 and 1.

It is simple and easy to interpret and provides probability estimates instead of just classifications which can help in medical decision-making. It performs well when there is a linear relationship between the features and the target variable and it can handle both numerical and categorical features. However, logistic regression has some limitations. It assumes that the features have a linear relationship with the log-odds of the outcome which might not always be the case and it is sensitive to outliers.

4.4.2 Random Forest Classification

Random Forest Classification is an ensemble learning method that combines the predictions of multiple decision trees for classification. It builds several individual decision trees during training. Each decision tree makes its own prediction and the final result is obtained through a voting system. Random subsets of the original dataset are created through resampling. Each tree is trained on a random subset of features to make the model more diverse and reducing the chance of overfitting. Decision trees are built independently from each subset by splitting nodes based on the best feature that minimizes the impurity. The final prediction is made by majority voting among all trees.

Random Forest Classification may be better than Logistic Regression because by combining multiple decision trees, it provides more accurate predictions. It can model complex and non-linear relationships between features and the target variable. It reduces the risk of overfitting by averaging multiple trees and automatically ranks features based on their contribution to the prediction. It can handle missing values without needing complex imputation techniques and works well with large datasets. However, training many decision trees can be slow for large datasets and it is more difficult to interpret compared to Logistic Regression.

5. Results

5.1 Predictive Analysis

For the dataset of 91,690 patients, 70% was allocated to the training set and 30% to the test set. Linear Regression and Random Forest Classification models predicted with 16,471 and 16,728 True Positive values respectively. Figure 2 shows the confusion matrices.

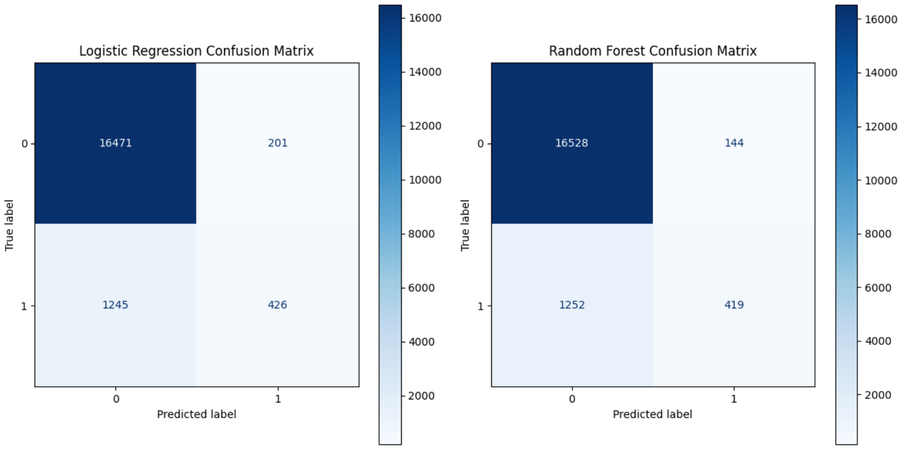

Figure 2: Confusion Matrices

Model	**Precision**	**Recall**	**F1-Score**	**Support**
Logistic Regression				
0.0	0.93	0.99	0.96	16672
1.0	0.68	0.25	0.37	1671
Accuracy	0.92			
Macro avg	0.80	0.62	0.66	18343
Weighted avg	0.91	0.92	0.90	18343
Random Forest				
0.0	0.93	0.99	0.96	16672
1.0	0.74	0.25	0.38	1671
Accuracy	0.92			
Macro avg	0.84	0.62	0.67	18343
Weighted avg	0.91	0.92	0.91	18343

Table 1: Evaluation Results for Logistic Regression and Random Forest

5.1.1 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the proportion of correctly classified instances out of the total instances. Both models have identical accuracy which means that they both classified the same percentage of samples correctly.

5.1.2 Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision is the proportion of correctly predicted positive cases out of all predicted positive cases. For Class 0 (patients who did not die), both models achieved the same precision (93%) meaning both are equally good at minimizing false positives for this class. For Class 1 (patients who died), Random Forest performed better with a precision of 74% compared to Logistic Regression's 68%. This indicates that Random Forest is better at identifying true positive deaths while reducing false positives for this class.

5.1.3 Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall is the proportion of correctly predicted positive cases out of all actual positive cases. Both models have the same recall for Class 0 meaning both are equally good at identifying non-death cases. However, for Class 1 (patients who died), both models have a low recall of 25% which means that both models fail to identify a large portion of the true positives (patients who died).

5.1.4 F1-score

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score is the harmonic mean of precision and recall, balancing both metrics. For Class 0, both models have the same F1-score (96%) indicating a balanced precision and recall. For Class 1, Random Forest has a slightly better F1-score (38%) compared to Logistic Regression's 37%, showing that Random Forest performs slightly better in balancing precision and recall for death prediction.

5.2 Prescriptive Analysis

Prescriptive analysis is used to optimize resource distribution in a hospital by offering clear guidance on how to prioritize patient care based on predicted outcomes. A predicted value of 0 signifies that a patient is likely to survive with treatment while a predicted value of 1 indicates a high likelihood of mortality, suggesting that palliative care would be more appropriate. By classifying patients into these two categories, healthcare providers can make informed decisions on resource allocation more easily, ensuring that medical resources are directed towards those who have the potential to benefit from treatment while those in critical conditions receive appropriate end-of-life care.

In addition to offering recommendations for treatment or palliative care, it considers high-risk patients based on important features identified through feature importance analysis. This allows the system to suggest targeted preventive measures such as encouraging regular check-ups for patients with abnormal values in specific features. This could help mitigate future risks. The system ensures that limited resources are used efficiently by improving patient care while also managing hospital capacity.

6. Discussion

The evaluation results indicate that both Logistic Regression and Random Forest perform well in predicting patient outcomes, achieving an identical accuracy of 92%. However, accuracy alone does not provide a complete picture, especially given the apparent class imbalance in the dataset. Both models show strong performance in identifying non-death cases with a high precision of 93% and a recall of 99% for Class 0, leading to an F1-score of 96%. In contrast, their ability to predict death cases (Class 1) is considerably weaker. Logistic Regression has a precision of 68% and a recall of 25% for Class 1 resulting in an F1-score of 37% while Random Forest slightly outperforms it with a precision of 74%, the same recall of 25% and an F1-score of 38%. The low recall for death cases suggests that both models are struggling to correctly identify patients who did not survive, likely due to the imbalance in the dataset. The macro average metrics show that Random Forest has a slight advantage, particularly in precision (84% vs. 80%) and F1-score (67% vs. 66%) while both models share the same recall of 62%. The weighted averages indicate minimal differences, with Random Forest achieving a better F1-score (91% vs. 90%). Overall, Random Forest demonstrates a slight advantage, especially in handling Class 1 predictions but both models suffer from poor recall in identifying patient deaths. To improve performance, techniques such as resampling, class weighting or the use of more advanced ensemble methods like boosting could be explored to enhance recall for the minority class while maintaining high precision.

7. Conclusion

This study has developped a model that predicts patient outcomes using Logistic Regression and Random Forest with a focus on hospital mortality prediction. First, the structure of the dataset is understood and missing values and data types are identified through exploratory data analysis (EDA) and identify missing values and data types. Through imputation techniques and feature engineering, missing data is handled and categorical variables are transformed into numerical ones. Both models were trained using preprocessed data and their performance was evaluated using classification metrics - accuracy, precision, recall, and F1 score - as well as confusion matrices. The Random Forest model performed well in identifying the most important features influencing hospital mortality which helped for the prescriptive analysis. For prescriptive analysis section, recommendations based on the importance of the features identified by the Random Forest model were given. The analysis suggested preventive measures for high-risk patients such as encouraging regular check-ups and prioritized patients for timely medical interventions or palliative care based on their likelihood of survival. This approach provides valuable predictions about patient mortality and also supports hospital resource optimization by recommending efficient use of resources based on patient priorities. The methodology and results can be directly applied to improve decision-making processes in healthcare settings helping medical staff make better decisions for patient care. Future work could involve refining the models using additional data, experimenting with other machine learning algorithms and incorporating real-time patient data for dynamic decision-making.

8. Abbreviations

- TP = True Positives (Correctly predicted positive cases)

- TN = True Negatives (Correctly predicted negative cases)
- FP = False Positives (Incorrectly predicted positive cases)
- FN = False Negatives (Incorrectly predicted negative cases)

References

Agarwal, M. (2021). Patient survival prediction. https://doi.org/10.34740/KAGGLE/DSV/2972359

Akin, I. M., Jale, S., Sinan, M., Gonce, K. H., & Inci, A. (2024). Predicting survival of patients with heart failure by optimizing the hyperparameters. *International Journal of Knowledge-Based and Intelligent Engineering Systems*. https://doi.org/10.1177/13272314241295925

Bakasa, W., Viriri, S., & Chen, H. (2021). Computational and mathematical methods in medicine. *Computational and Mathematical Methods in Medicine*. https://doi.org/10.1155/2021/1188414

Mukhtar, A. S., & Muhammed, A. N. (2023). Classification based on event in survival machine learning analysis of cardiovascular disease cohort. *BMC Cardiovascular Disorders*, 23(1). https://doi.org/10.1186/s12872-023-03328-2