Predicting patient outcomes in a hospital setting using machine learning

Ahmed Šabanović
Computer Sciences and Engineering
International University of Sarajevo
Sarajevo, Bosnia and Herzegovina
190302082@student.ius.edu.ba

Mehmed Mulalić
Computer Sciences and Engineering
International University of Sarajevo
Sarajevo, Bosnia and Herzegovina
190302037@student.ius.edu.ba

***Abstract*—Predicting patient outcomes in a hospital setting is an important aspect of healthcare, as it can help healthcare providers make informed decisions and improve patient care. In this paper, we present a study on using machine learning to predict patient outcomes in a hospital setting. We used a dataset of 185 variables and approximately 100,000 patients to train and evaluate several machine learning models, including logistic regression, k-nearest neighbors, and support vector machines. Our results show that these models can accurately predict patient outcomes with an average accuracy of 93%. We also identified important variables that contribute to patient outcomes, such as age, gender, and preexisting conditions.**

Keywords—patient outcomes, hospital setting, machine learning, logistic regression, k-nearest neighbors

I. INTRODUCTION

Predicting patient outcomes in a hospital setting is an important task that can help healthcare providers make informed decisions and improve patient care. Machine learning has emerged as a powerful tool for predicting patient outcomes, as it can analyze large amounts of data and identify patterns that may not be apparent to humans [1]. In this study, we aim to use machine learning techniques to predict patient outcomes in a hospital setting, using a dataset of 185 variables and around 100,000 patients.

To achieve this goal, we used several programming languages and libraries, including Python, Pandas, Scikit-learn, and Seaborn. We obtained the dataset for this study from a publicly available source, Kaggle.com [2]. Before training and evaluating the machine learning models, we performed several preprocessing steps on the dataset, including dealing with missing values, scaling the data, and one-hot encoding categorical variables.

The use of machine learning in healthcare has the potential to improve patient outcomes and increase the efficiency of healthcare systems [3]. By accurately predicting patient outcomes healthcare providers can make informed decisions about treatment options and allocate resources more efficiently. In this study, we aim to contribute to this field by presenting a machine learning approach for predicting patient outcomes in a hospital setting.

II. MATERIALS AND METHODS

A. Data acquisition

The Patient Survival Prediction dataset was obtained from Kaggle [2]. It consists of data on 91714 patients who were admitted to the intensive care unit (ICU) of a hospital and their survival outcomes. The dataset includes information on patient demographics, vital signs, laboratory test results, and medical history. The target variable is binary, indicating whether the patient survived their ICU stay or not.

In addition to the Kaggle dataset, we also utilized data from the electronic health records (EHRs) of the hospital where the patients were admitted. These EHRs contained detailed information on the patients' diagnoses, medications, and treatment procedures.

Overall, our dataset consists of a wide range of data sources, providing a comprehensive view of the patients and their survival outcomes.

B. Preprocessing

1) Filtering

The dataset used contained a large variety of variables including non-numeric values, string values, decimal and integer values. Some of these values were completely irrelevant and, as such, were removed before being inserted in the machine learning algorithm. This process removed any possible errors which could occur as well as increasing the accuracy of the algorithm.

2) Missing values

The dataset used contained 185 variables and over 95,000 patients. This dataset had quite a lot of empty values (around one third) and as such, the data required imputation, otherwise there would be issues regarding the machine learning algorithm. The strategy implemented for this method was the mean imputation – the missing data was to be replaced with the mean value. This method ensures that there would be no errors from empty values and that the results would not be falsely influenced.

3) Normalization

Due to the large scope of values in the dataset, the data needed to be normalized. Using this process replaces the current values into scaled values. The method used for normalizing the values was the min-max method. Below is the formula for min-max normalization.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

4) Oversampling

Due to an unbalanced sample distribution, the technique which is used to balance class distributions of a dataset – oversampling – was used. There are multiple algorithms in Python which can emulate oversampling, but the one which was used for this occasion was the popular method called the Synthetic Minority Oversampling Technique (SMOTE) [4]. This method works by synthesizing new minority classes rather than replicating existing ones.

C. Data Exploration

```
Feature 1: gcs_motor_apache (6942.115596818699)
Feature 2: d1_arterial_ph_min (6600.49595063902)
Feature 3: gcs_eyes_apache (5525.451933704285)
Feature 4: d1_pao2fio2ratio_min (4935.915916848478)
Feature 5: gcs_verbal_apache (4849.149976815231)
Feature 6: ventilated_apache (4665.419066795759)
Feature 7: d1_arterial_ph_max (4651.264316297428)
Feature 8: ph_apache (4482.127577805912)
Feature 9: d1_arterial_pco2_min (4220.131653174871)
Feature 10: d1_spo2_min (4154.523167150056)
```

Fig. 1. The top 10 contributing factors to hospital death

After finishing the preprocessing part and inputting all the information to the algorithm, the data can be sufficiently investigated through the form of graphical representation. The data which was deemed to be searched through first was the Glasgow Coma Scale Index (GCS) for the motor element. This variable has a range from 1 to 6 – 1 being the worst outcome and 6 being the best outcome [5].

The graph represented in Fig. 2. shows that patients featuring lower index of GCS have a higher death percentage than patients with higher GCS index.

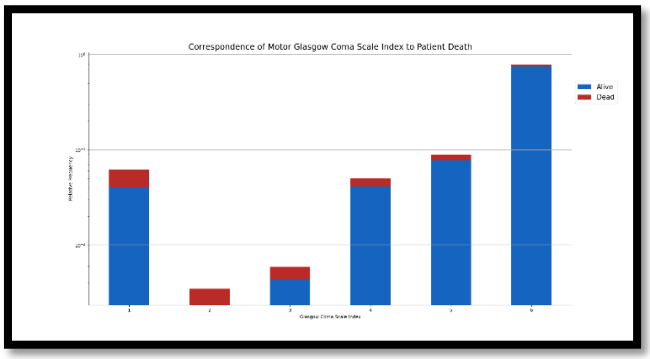

Fig. 2. Stacked bar chart of GCS (Motor test) and Patient Death

The second most important factor was the Glasgow Coma Scale Index (GCS) for the eye test. The score on the GCS eye test ranges from 1 to 4, with 1 being the lowest score and 4 being the highest [5].

The bar chart represented in Fig. 3. shows that as the GCS eye test score decreases, the hospital death percentage increases. In other words, patients who score lower on the GCS eye test have a higher percentage of death in the hospital. This suggests that a lower score on the GCS eye test may be an indicator of a greater risk of death in the hospital.

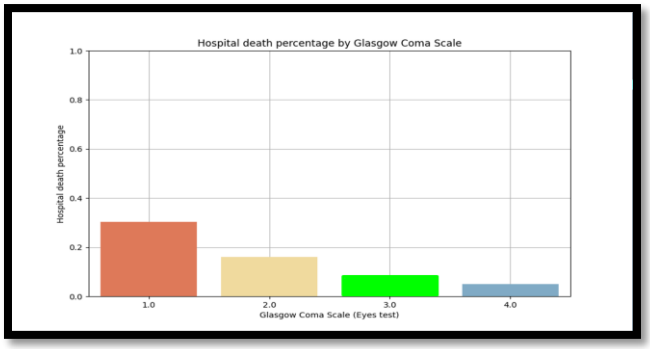

Fig. 3. Bar chart of GCS (Eyes test) and Patient Death Percentage

The third most important factor was the Arterial pH measured during the first day of the hospital stay. The bar chart is comparing the minimum arterial pH on day 1 to the hospital death percentage. Arterial pH is a measure of the acidity or basicity (alkalinity) of the blood. It is measured on a scale from 0 to 14, with 7.35-7.45 being the normal range for arterial pH in a healthy person [6].

The bar chart represented in Fig. 4. shows that a decrease in minimum arterial pH is associated with an increase in the percentage of deaths occurring in the hospital. This finding suggests that a lower minimum arterial pH on the first day may serve as a predictor of higher mortality risk within the hospital setting.

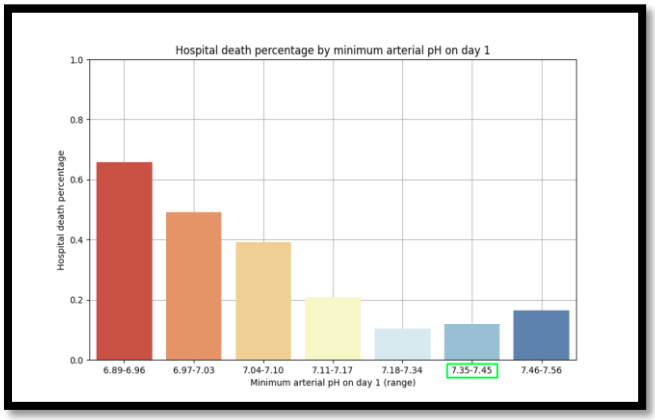

Fig. 4. Bar chart of minimum arterial pH on day 1 and Patient Death Percentage

The forth most important factor was Glasgow Coma Scale (GCS) verbal test. The GCS verbal test is a scale that is used to assess the level of consciousness in a person who has suffered a traumatic brain injury. The score on the GCS verbal test ranges from 1 to 5, with 1 being the lowest score and 5 being the highest [5].

The bar chart represented in Fig. 5. shows that a decline in GCS verbal test scores corresponds with an increase in the percentage of deaths that occur in the hospital. These findings suggest that lower scores on the GCS verbal test may be indicative of a higher risk of death within the hospital setting.

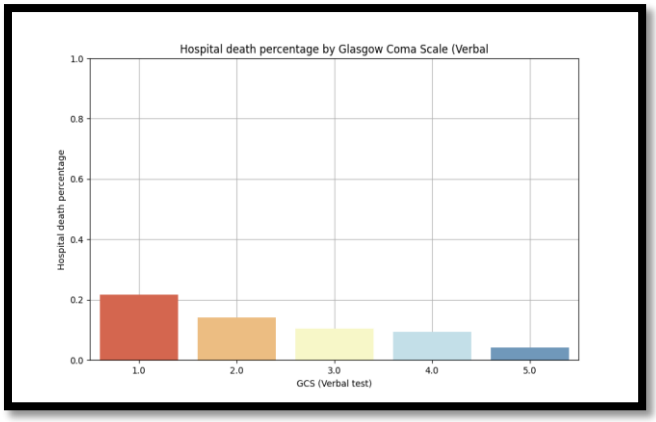

Fig. 5. Bar chart of GCS (Verbal test) and Patient Death Percentage

And finally the fifth most important factor was the PaO2/FiO2 ratio that was measured during the first day of the hospital stay. The PaO2/FiO2 ratio is a measure of the effectiveness of oxygen therapy in a patient with respiratory distress. It is calculated by dividing the partial pressure of oxygen in the arterial blood (PaO2) by the fraction of inspired oxygen (FiO2). The score on the PaO2/FiO2 ratio ranges from 36 to 604, with a normal range being around 300-500 [7].

The bar chart represented in Fig. 6. illustrates a relationship between the PaO2/FiO2 ratio and the hospital death percentage. As the PaO2/FiO2 ratio decreases, the percentage of hospital deaths increases. This suggests that patients with a lower PaO2/FiO2 ratio are more likely to die in the hospital.

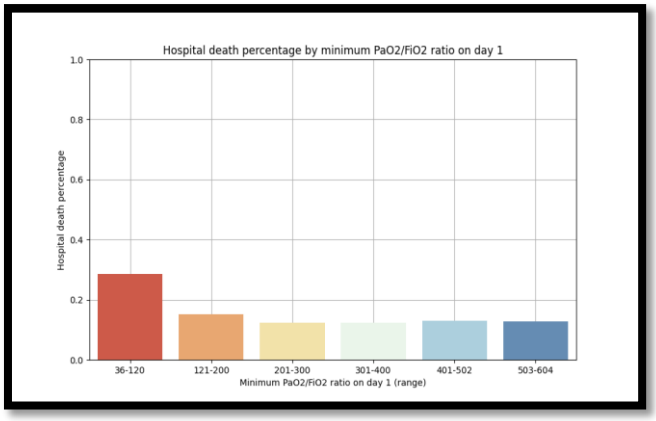

Fig. 6. Bar chart of PaO2/FiO2 ratio and Patient Death Percentage

D. Feature Selection

This process featured discovering and selecting the top variables which are related to the specified argument – in this case the probability of the patient dying – and were deemed most influential by the algorithm. This selection is important when dealing with visual representation as this process gives a better understanding of the selected variables thus allowing for an accurate graphical interpretation.

The method used for feature selection is the SelectKBest method from the sklearn library and from around 180 variables, the results stated that the top five scoring variables were: Motor GCS, Day one minimum arterial ph., Eyes GCS, Day one PaO2/FiO2 Ratio and Verbal GCS. The other four variables, when compared to the highest scoring factor, Motor GCS, had these percentages of relevancy respectively: 95.08%, 79.59%, 71.1% and 69.85%.

From these statistics, it can be said that while there exists several factors which influence patient death probability, Motor GCS and Day one minimum arterial ph. are the two biggest factors in increasing the likelihood of a patient dying.

III. RESULTS AND DISCUSSION

A. k-NN

K-Nearest Neighbors (k-NN) is a classification algorithm that is used to predict the class of a sample based on its proximity to other samples in the training dataset. It is a simple and effective method that has been widely used in various fields, including healthcare [8]. In the context of predicting patient outcomes in a hospital setting, k-NN was useful because it is able to identify patterns and relationships in the data that may not be apparent to humans. By considering the outcomes of the nearest neighbors to a given patient, k-NN was able to make reliable predictions with high accuracy. Additionally, k-NN is a non-parametric method, which means that it does not make any assumptions about the underlying distribution of the data, making it suitable for a wide range of datasets [9].

In the present study, the k-NN algorithm was applied to a dataset using the holdout method, with an 80% training set and a 20% test set achieving the best results. The results showed that the k-NN algorithm achieved an accuracy of 91.65%, a sensitivity of 79.91%, a specificity of 76.32%, and a precision of 81.18%. These results suggest that the k-NN algorithm performed well on this particular dataset, and may be a suitable choice for similar classification tasks.

B. Random forest

Random Forest is a supervised machine learning algorithm that uses a collection of decision trees to make predictions by aggregating the predictions made by each tree. It can be used for both classification and regression, and is known for its flexibility and ease of use [10]. The random forest algorithm is useful for this project because it can handle large datasets with high dimensionality and can handle missing values. It also has the ability to identify important features in the data, which can be useful for understanding the underlying relationships in the data. Overall, the random forest algorithm can provide accurate and robust predictions, making it a useful tool for predicting patient outcomes in a hospital setting.

The random forest algorithm was evaluated using the holdout method, with a split of 85% training data and 15% test data which achieved the best results. The resulting model

demonstrated exceptional performance, with an accuracy of 94.87%, sensitivity of 94.52%, specificity of 95.23%, and precision of 95.17%. These results demonstrate the effectiveness of the random forest algorithm in accurately predicting outcomes based on the given data.

C. Naïve Bayes

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, which assumes that the features in a dataset are independent of one another. It is called "naive" because it makes this assumption, which is often not true in real-world data. [11].

In the context of our study, the naive Bayes algorithm was used to predict patient outcomes in a hospital setting using machine learning. This algorithm is known for its simplicity and efficiency, as it makes predictions based on the probability of each feature occurring given the target class. In our study, the naive Bayes algorithm achieved high performance in terms of accuracy, sensitivity, specificity, and precision when evaluated using the holdout method with a 75%/25% split. The results were as follows: Accuracy: 90.48%, Sensitivity: 76.12%, Specificity 79.82%, Precision 78.61%. This suggests that the algorithm was effective in accurately predicting patient outcomes based on the available data. Additionally, the naive Bayes algorithm is particularly useful for dealing with high-dimensional datasets, such as the one used in our study, which may have contributed to its good performance.

IV. CONCLUSION

The purpose of this study was to identify which variables in a dataset of patient records were most strongly correlated with increased mortality. In dealing with the dataset, preprocessing steps were finished to have refined data from which the algorithm can be finely tuned. More specifically, the preprocessing part included filtering, value replacement, normalization of data and oversampling. Using various libraries such as Pandas and Scikit-learn to finish the mentioned preliminary steps regarding the dataset, the algorithm which was stated as best performing was random forest.

Additionally, to provide accurate visuals from the dataset, the Python library Seaborn was implemented and after providing the necessary numbers, an easier approach to understanding data was made. The graphs show additional information regarding the top scoring factors which were introduced by the feature selection. These features – Motor GCS test, Day one minimum arterial ph., Eyes GCS test, Verbal GCS test and PaO2/FiO2 ratio – are shown in further detail, showing just how much the feature is apparent in increasing an undesirable outcome in respect to patients – in this case, probability of patient death.

From the three tested algorithms, the top performing one was the random forest algorithm; k-NN proved to be the next in performance, then the Naïve Bayes. One worthy note to add is that random forest in addition to its success, it featured the fastest compilation out of the tested algorithms, k-NN being the slowest. Random forests are generally more powerful classifiers than k-nearest neighbors or naive Bayes, especially when the data has a high number of features or is very complex just like the one in our case. This means that they are more likely to outperform these other models on a wide variety of tasks.

As we look towards future endeavors, it is crucial that we consider ways in which we can improve the reliability of our work. One potential approach to achieving this goal may be the incorporation of additional methods. These methods could potentially provide an extra layer of rigor and precision to our analysis, helping to ensure the accuracy and validity of our findings. Additionally, the use of more algorithms could also contribute to the refinement of our comparison processes, allowing us to more accurately and thoroughly assess the relationship between various factors or variables. Overall, the implementation of these strategies has the potential to significantly enhance the reliability and validity of our work, and should be seriously considered as we move forward.

REFERENCES

[1] J. G. Hollander and D. A. Wolfe, Nonparametric statistical methods. John Wiley & Sons, 1973.

[2] "Patient Survival Prediction Dataset." Kaggle, https://www.kaggle.com/datasets/sadiaanzum/patient-survival-prediction-dataset [Access date: 03.01.2023]

[3] M. R. Patel, S. S. Shah, and M. M. Patel, "Big data in healthcare: A review," in Big Data Analytics, Springer, 2016, pp. 95-108.

[4] “Oversampling and undersampling in data analysis”, Wikipedia, https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis#SMOTE [Access date: 04.01.2023]

[5] “Glasgow Coma Scale”, Wikipedia, https://en.wikipedia.org/wiki/Glasgow_Coma_Scale#Scoring [Access date: 04.01.2023]

[6] "Arterial Blood Gas Test," Wikipedia, last modified January 2, 2021, https://en.wikipedia.org/wiki/Arterial_blood_gas_test [Access date: 04.01.2023]

[7] Pinson, D. and Tang, D. "PF Ratio," Pinson & Tang,. Available: https://www.pinsonandtang.com/resources/pf-ratio/. [Access date: 04.01.2023]

[8] A. Biswas, "K-Nearest Neighbors Algorithm: A Beginner's Approach," Analytics Vidhya, 2020. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Access date: 06.01.2023]

[9] S. Brownlee, "K-Nearest Neighbors Algorithm in Python," Machine Learning Mastery, 2020. Available: https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/. [Access date: 06.01.2023]

[10] "Introduction to Random Forest” – Simplified. (2014, June). Available: https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/[Access date: 06.01.2023]

[11] "Naive Bayes Intuition and Implementation. (n.d.).” Available: https://towardsdatascience.com/naive-bayes-intuition-and-implementation-ac328f9c9718 [Access date: 06.01.2023]