TÉCNICO LISBOA

Intelligent Systems

MEMEC

Project [EN]

Authors:

Afonso Almeida (100114) afonsofdealmeida@tecnico.ulisboa.pt
João Moura (103693) joao.n.moura@tecnico.ulisboa.pt

Group 3

2025/2026 – 1st Semester, P1

Contents

GitHub Repository

1 Introduction

Septic cardiomyopathy (SCM) is a transient yet potentially life-threatning cardiac dysfunction that develops in patients with sepsis. It is characterised by impaired cardiovascular performance, manifesting as systolic or diastolic dysfunction, and occurs when the host response to an infection becomes sufficiently dysregulated to cause organ dysfunction [1]. However, its diagnosis remains a challenge, making early detection difficult and contributing to higher mortality rates in septic patients. Cardiac dysfunction is estimated to affect $40 - 70\%$ of adult patients with sepsis or septic shock, and is primarily driven by the release of cytokines, mitochondrial dysfunction and tissue hypoxia leading to myocardial injury. Some studies have associated SCM with increased mortality, while others have not found such evidence, demonstrating how its impact on outcomes remains unclear. The lack of reliable diagnostic methods complicates the identification of this dysfunction, reinforcing the need for data-driven approaches to improve diagnostic accuracy [2].

The goal of this project is to develop a method for accurately diagnosing SCM based on patients' personal and medical information, while managing and preprocessing the given data. This project also aims to identify relevant patterns that could support early and reliable diagnosis of SCM. To achieve these goals, a dataset containing 207 attributes (features) collected from 100 patients (samples) was used. These attributes include, among others, clinical and laboratory data, which together provide an overview of the condition of each patient.

2 Methodology

Given the dimension of the dataset, some preprocessing was required, not only to reduce the dimensionality of the feature space but also to fix the problem of missing values, which were present in many features, both continuous and binary.

First, the target variable was defined and certain irrelevant features were excluded. Then, the categorical features were converted into numeric values and the missing values were filled. Once the dataset was prepared, feature selection methods were applied to further reduce its dimensionality and, finally, a model was developed to predict the behaviour of the target variable.

2.1 Features of interest

The initial step in data processing involved defining the target variable. Considering the goal of the project, the chosen target was "OBITO", a binary feature that indicates whether the patient died or not. The correlations between the remaining features and the target were then computed, revealing three features with substantially higher correlation values:

- "Recov": stratifies how each patient recovered, being 0 if no recovery was observed, 1 if the patient fully recovered, 2 when they partially recovered or 3 if the disease was aggravated;

- "Dias_ate_morte": how many days the patient lived from hospital admission until death;
- "RecovTime": number of days it took each patient to recover.

These features were initially discarded to avoid data leakage, as they contain information that would only be available after the outcome of the target was known. Including them in the training process would allow the model to learn information that would indirectly reveal the value of the target variable.

Having defined the target, there were still too many features, so a number of them were logically discarded, without applying any formal method. These included the patient's identification number, the corresponding process number and all date features, such as birth dates and hospital admission dates were excluded, as the patients' ages and the number of days spent in each phase of hospitalization were already captured by other variables. It is worth noting that these dates could have held relevance for the classification task if they spanned a wider time range such as different decades potentially reflecting variations in treatment practices. However, after verification, all records were found to fall within a two-year period, so this possibility was ruled out.

2.2 Categorical features

There were two categorical features in the dataset: "SEXO", corresponding to the sex of the patient, and "Agent", which indicates the bacteria or fungus detected on the patient, which caused the disease.

The first variable was binarized, with "M" (male) coded as 1 and "F" (female) coded as 0. For the second one, one-hot-encoding was considered, but after verifying there was a high number of different variables, this possibility was discarded as it would increase the dimensionality in the dataset. Therefore, embedding vectors were used. Each category of "Agent" was converted into a four-dimensional numerical vector, allowing the model to use the information contained in the feature.

2.3 Missing values

Approximately 4% of the dataset consisted of missing values, with "Vrenais_padrão$\geq 0,8$", "Vrenais_padrao$< 0,7$", "Indice_resistencia", "Arenais_Vmin" and "ARenais_Vmax" being the features with the highest percentage of missing values at around 43% each, followed by "CD4_CD8" at 37% and "PVC" at 22%. Therefore, before selecting the features to input into the model, the missing values were dealt with. For all features except "Dias_ate_morte" and "RecovTime", different options were considered, from simpler methods to more complex ones, which are described below.

- Mean: fill the missing values with the mean of the existing values for the same feature;
- Median: fill the missing values with the median of the existing values for the same feature;
- Regression: train a regression model for each feature with missing values, using the other features as predictors and the available values of the target feature as the response. Then, use the trained model to predict the missing values for that feature, using the

corresponding values of the other features for each patient. Repeat the process for each feature with missing values independently;

- Multiple Imputation by Chained Equations (MICE): fill missing values for each feature iteratively, using models based on the remaining features and a certain initial estimate. In each iteration, update the current imputations and repeat the process multiple times until convergence or until a specified maximum number of iterations is reached:
  - MICE with regressions: use regression models to update and fill the missing values;
  - MICE with decision trees: use decision tree models to update and fill the missing values.

It is important to note that continuous and binary features must be treated separately, since applying some of these methods to binary features could result in values other than 0 or 1.

After careful consideration, MICE with decision trees was chosen to handle the missing values. Missing values for continuous features were imputed using a Decision Tree Regressor, while missing values for binary features were estimated using a Decision Tree Classifier. Since this is a real-world dataset containing medical information, it is acknowledged that data imputation may not always represent the most accurate or ethically ideal approach, as it involves estimating potentially sensitive patient information. However, this method was necessary to enable the completion of the subsequent stages of the analysis, which could not proceed with incomplete data. MICE with decision trees was selected as it is considered a principled and robust approach that leverages feature relationships to produce the most plausible imputations possible.

Although "Dias_ate_morte" and "RecovTime" had already been removed, their missing values were still filled for data treatment purposes. These values were imputed through a different approach than the remaining features. In fact, some patients did not die or did not recover, so the missing values should not be handled based on the values of other patients. Therefore, for these features, the missing values were set to approximately twice the highest existing value, namely 800 for "Dias_ate_morte" and 400 for "RecovTime". However, imputing values that are much higher than the existing ones would cause the model to lose sensitivity to smaller variations. For instance, regarding "Dias_ate_morte", a difference of 5 days out of 800 may appear insignificant, whereas in reality it represents a meaningful change, especially when compared to the range of around 400 days observed in the original data. To overcome this limitation, logarithmic transformations of both features were used instead of their original values. To asses which base was the most appropriate, four different logarithmic bases were tested, and the results for each feature are presented in Figures 1 and 2.

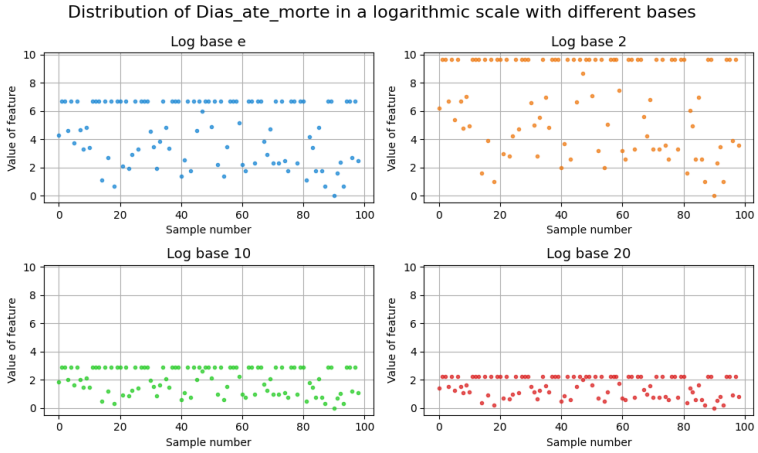


Figure 1: Distribution of Dias_ate_morte in a logarithmic scale with different bases

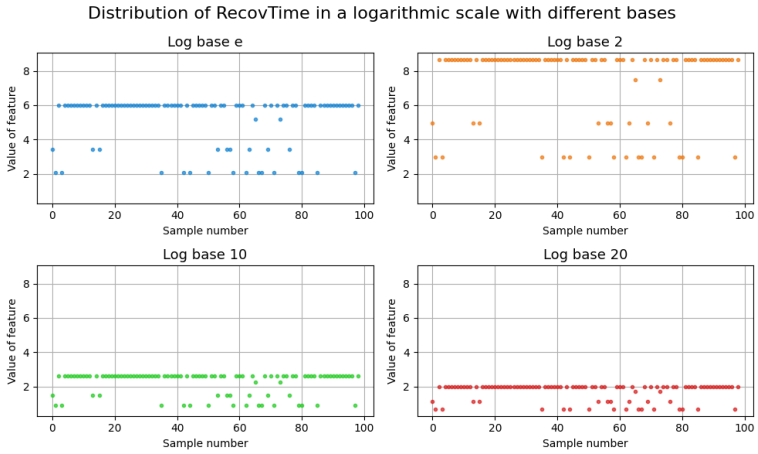


Figure 2: Distribution of RecovTime in a logarithmic scale with different bases

Based on the observed distributions, base 10 was selected, to actively capture the difference between the higher values and the other ones, while being able to keep sensitivity.

2.4 Feature selection

After dataset preprocessing, feature selection was conducted to reduce the initial set of 197 variables. Given the high dimensionality of the data, it was likely that several variables exhibited linear dependencies, allowing the underlying information to be represented using a smaller subset of features. Reducing this redundancy not only simplifies the model but also mitigates potential overfitting and improves computational efficiency. To achieve this, a two-stage feature selection approach was implemented. In the first stage, correlation-based feature selection was applied to identify and remove highly correlated variables. Subsequently, Principal Component Analysis (PCA) was employed to extract the most informative components, thereby retaining the features that captured the highest variance in the dataset. The correlation matrix was computed to assess how the features correlate to each other. Then, by defining a correlation

threshold, when two features had a correlation higher than the threshold, one of them was discarded to prevent redundancy. This threshold was defined as follows, considering the absolute values of the correlations for the calculations, where k is a constant empirically defined, in this case, $k = 2$:

$$\text{threshold} = \text{mean of correlations} + k \times \text{standard deviation} \tag{1}$$

When a correlation higher than the threshold was found, a tie-breaking criterion was applied in order to choose the variable to be dropped. For each pair of features with a correlation exceeding the threshold, the absolute values of the correlations with all other features were summed and the feature with the higher sum was discarded. Using this method, 129 features of the 197 total features were removed, leaving 68 remaining features.

After reducing the number of features using the correlation matrix, PCA was applied to the standardized data. Given the relatively small sample size ($n = 99$), the analysis was restricted to twenty principal components to ensure stability and interpretability of the results. The selection of the number of components was further supported by examining the cumulative variance explained (CEV), which quantifies how much of the dataset's total variance is captured as additional components are included. As illustrated in the subsequent graph of cumulative variance explained versus the number of principal components, the components account for the majority of the variance, indicating that they adequately represent the underlying structure of the data, as is displayed in Figure 3.

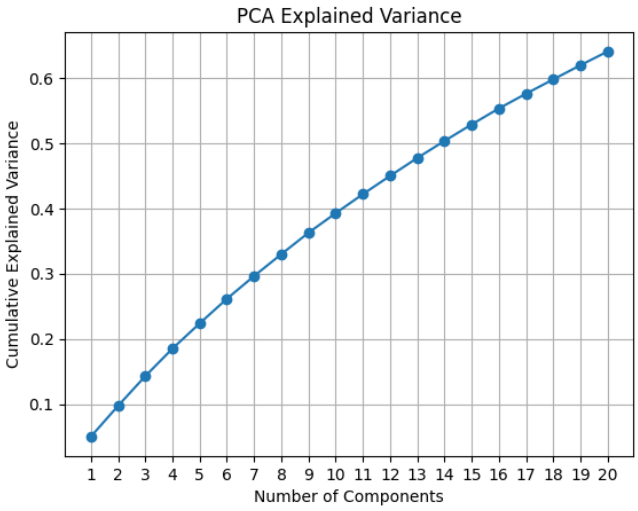


Figure 3: Evolution of CEV over number of components selected

This choice also reflects the inherent trade-off between model simplicity and variance retention: increasing the number of components captures more variance but adds complexity and potential noise, while limiting the model to a smaller set of components promotes interpretability and reduces overfitting, even if some minor variance is lost.

2.5 Final Model

The first step before building the model itself was to split the data into training and testing sets. Consequently, 20% of the samples were used for testing, while the rest of the data was

used for training and validation using K-Fold Cross Validation. The use of a separate validation set for hyperparameter tuning was discarded given that the number of samples in the dataset was too small, therefore using a separate validation set would both take meaningful data from the training and testing sets.

Subsequently, Fuzzy C-means clustering was applied, and 2 clusters were created, so that a Gaussian membership function could be obtained for each of them. With this information, a Takagi-Sugeno-Kang (TSK) model was built, in which the parameters for the antecedents and consequents were determined using the hybrid ANFIS training method. The antecedent parameters were adjusted through gradient descent, whereas the consequent parameters were adjusted by a least-squares approach. Then, K-Fold Cross-Validation was applied to tune the number of clusters and the fuzziness parameter m. Finally, the model was trained once again on the training set with the optimal hyperparameters, and evaluated on the testing set.

3 Results and discussion

After completing hyperparameter tuning and training, the model was evaluated on the test set. Given the limited size of the dataset, there was a significant risk of overfitting, whereby the model could learn patterns specific to the training data rather than generalizable relationships. To mitigate this, the final training phase was deliberately restricted to a small number of epochs (five). Model performance was then assessed using several complementary metrics, namely: accuracy, precision, recall and F1-score, which are presented in Table 1.

Table 1: Final performance metrics obtained

Accuracy	**F1 Score**	**Recall**	**Precision**
0.6500	0.6316	0.6667	0.6000

To obtain a clearer understanding of the model's classification performance, the confusion matrix for the test set was plotted and is shown in Figure 4. This analysis is particularly relevant in medical contexts, where minimizing false negatives is crucial, as these represent cases in which a patient is incorrectly classified as healthy when they are not.

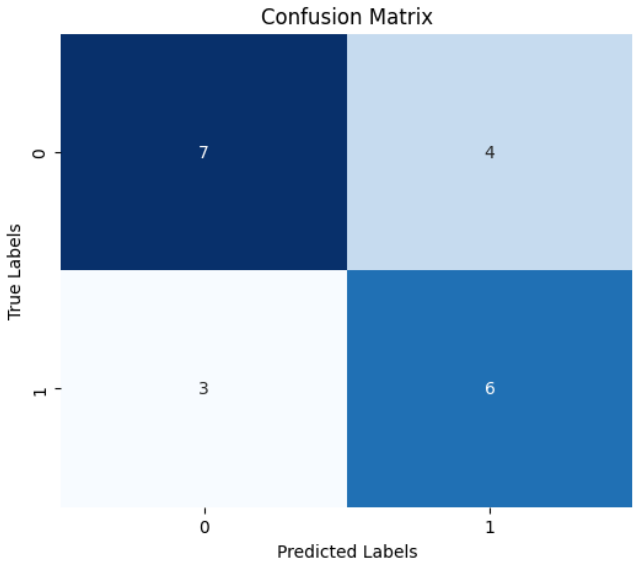


Figure 4: Confusion Matrix

As observed, the results obtained were suboptimal, with the model achieving an accuracy of only 65% for the binary classification task. Nevertheless, the confusion matrix indicates that false negatives were the least frequent error type, which is particularly relevant in medical applications, as previously discussed. To better understand the reasons behind the model's limited predictive performance, the outputs of the TSK model on the testing set were further analyzed. These outputs, ranging from 0 to 1, are converted into binary predictions, where values above 0.5 correspond to class 1 and those below or equal to 0.5 to class 0. To visualize this behavior, the predicted confidence values were plotted, with correctly classified samples shown in green and misclassified samples in red, as illustrated in Figure 5. This analysis provided insight into the model's confidence and its relationship to prediction accuracy.

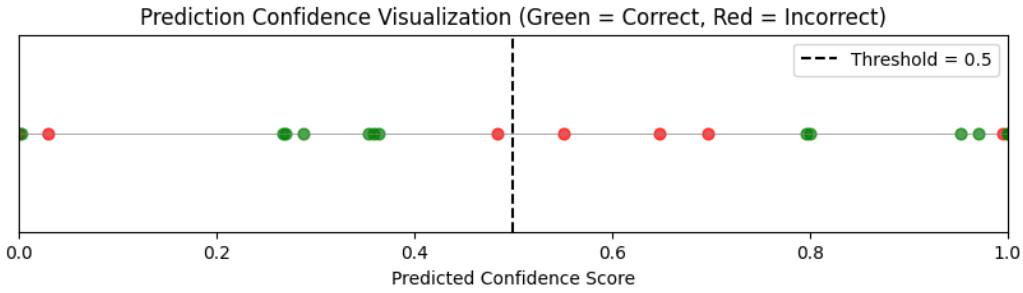
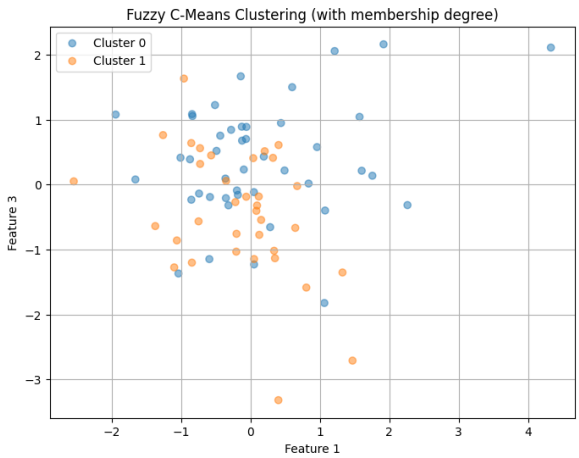


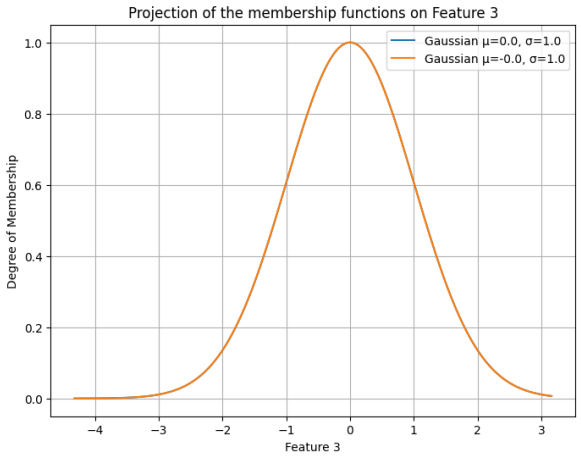
Figure 5: Prediction confidence scores

Ideally, the model's misclassifications would correspond to predictions with confidence values close to 0.5, indicating that although incorrect, the model was uncertain and nearly correct in its estimation. However, this was not consistently observed. While some misclassifications did occur near the decision boundary, several false predictions were made with very high confidence values, close to 0 or 1. This suggests that the model failed to adequately capture the underlying patterns in the data, leading to overconfident yet incorrect predictions.

There are several plausible reasons for this behavior. First, the dataset used in this study is a real-world medical dataset characterized by complex, highly interdependent variables and relatively low variance among samples. Such homogeneity makes it inherently difficult for the model to identify clear discriminative boundaries between classes. This challenge is reflected in the low clustering performance obtained during training (clustering score of 0.3333), which indicates that the input feature space lacked well-defined separability, which is worth noting since these clusters form the basis of the model's fuzzy rule structure. This can also be infered from the clustering results presented in Figure 6.



(a) Data is not very separable (although it's only plotted on two features)



(b) Overlapping membership functions for one of the features

Figure 6: Clustering results visualization.

Additionally, the small sample size represents a significant limitation. With only 79 training samples, the model had very limited data from which to learn meaningful relationships. This constraint also necessitated restricting the number of training epochs to prevent overfitting, further reducing the model's ability to generalize. Consequently, this probably prevented the TSK model from achieving better results.

4 Conclusion

This study applied a TSK model to a small, real-world medical dataset for binary classification, achieving an accuracy of 65%. Analysis of the model outputs showed that some incorrect predictions were made with high confidence, indicating that the model struggled to capture the underlying patterns in the data. The small sample size and low variance in the dataset further limited the model's ability to generalize.

For future work, alternative models such as LSTMs could be explored, alongside a more detailed analysis of the dataset to reduce the need for imputation. Techniques such as transfer learning may also help improve training and evaluation when working with very small datasets, such as this.

References

[1] Sarah J. Beesley, Jeff Sorensen, Allan J. Walkey, Joseph E. Tonna, Michael J. Lanspa, Ellie Hirshberg, Colin K. Grissom, Benjamin D. Horne, Rebecca Burk, Theodore P. Abraham, Robert Paine, and Samuel M. Brown. Long-term implications of abnormal left ventricular strain during sepsis. *Journal not specified*, 2022. Available in PMC 2022 April 01.

[2] Mariana de Braga Lima Carvalho Canesso, Isabela Nascimento Borges, Thiago Adriano de Deus Queiroz Santos, Tijmen Hermen Ris, Marcio Vinicius Lins de Barros, Vandack Nobre, and Maria Carmo Pereira Nunes. Value of speckle-tracking echocardiography changes in monitoring myocardial dysfunction during treatment of sepsis: potential prognostic implications. *Journal not specified*, March 2019. Received: 16 August 2018; Accepted: 31 December 2018; Published online: 7 March 2019; © Springer Nature B.V. 2019.