Final Project Assignment

1. Objective

   The goal of this project is to predict whether a patient who had a heart failure will survive or not. The selected model is a multilayer perceptron (also known as MLP), and it is expected to assist on the prediction task by identifying which clinical conditions impact the most on the survivability of a heart failure.

2. Description of the data

   The selected dataset is a dataset containing records of patients who had heart failure. It contains the following attributes:

   - Age: age of the patient, in years
   - Anaemia: indicates if patient had decrease of red blood cells or haemoglobin
   - High blood pressure: indicates if patient has hypertension
   - Creatinine phosphokinase: level of CPK enzyme in the blood, in mcg/mL
   - Diabetes: indicates if patient has diabetes
   - Ejection fraction: percentage of blood leaving the heart at each contraction
   - Sex: gender of the patient
   - Platelets: concentration of platelets in the blood, in kiloplatelets/mL
   - Serum creatinine: level of creatinine in the blood, in mg/dL
   - Serum sodium: level of sodium in the blood, in mEq/L
   - Smoking: indicates if the patient smokes
   - Time: follow-up period, in days
   - Death event: indicates if the patient died during the follow-up period

   For this task, the Death event column is going to be the target.

3. Data exploration and processing

   The first thing to be done is checking if the are rows with NaN values. Then, the data distribution in each column. The graphs for each column as shown below.

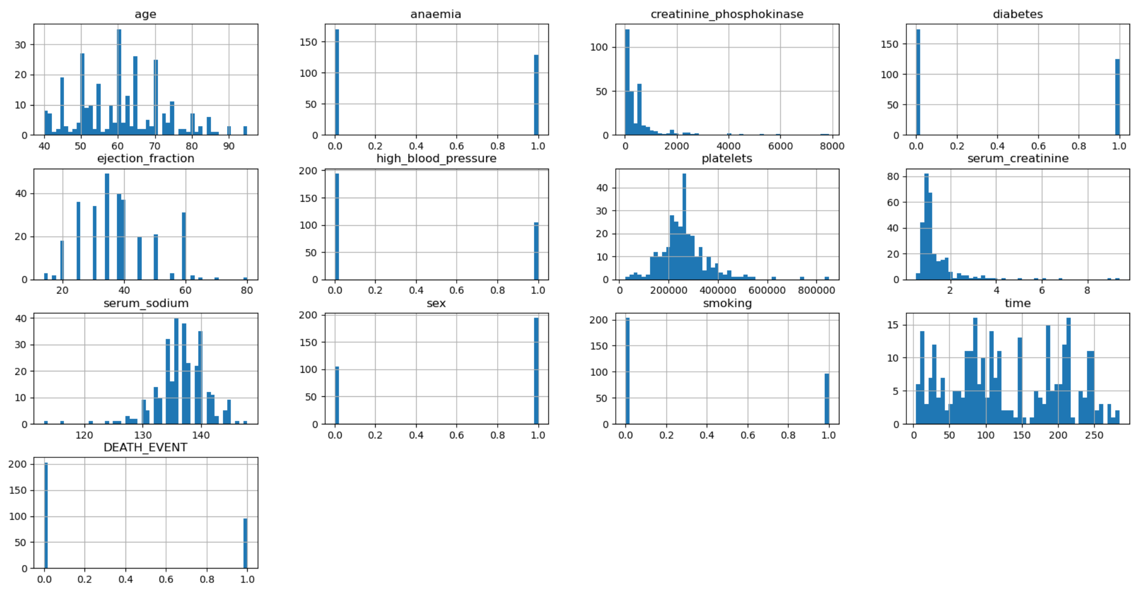


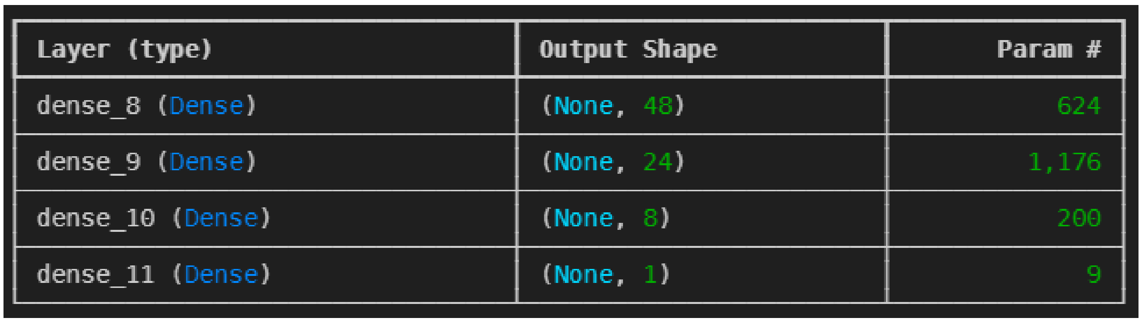
図 1. Data distribution for every column in the dataset

As it can be seen, the data isn't evenly distributed. Each column has a different range of values, and some of them have only two possible values (0 and 1). Also, as seen by the "DEATH_EVENT" graph, which shows the distribution of the target column values, there is an imbalance in the data.

For dealing with the varied data ranges, it is required to scale them. For that, the target column is separated from the others and the rest is scaled. Then, the data is split into training and test data. However, the data imbalance must not be ignored.

4. Training and evaluation

First, it was used a base MLP model which was built in Keras with the following structure:



```
┃ Layer (type)         ┃ Output Shape   ┃    Param # ┃
│ dense_8 (Dense)      │ (None, 48)     │        624 │
│ dense_9 (Dense)      │ (None, 24)     │      1,176 │
│ dense_10 (Dense)     │ (None, 8)      │        200 │
│ dense_11 (Dense)     │ (None, 1)      │          9 │
```

All hidden layers used the ReLU function while the output layer used the sigmoid function. It was trained in 11 epochs (it should've been 25, but the early stopping callback stopped the training), and it returned an accuracy of 83.5% for training data, 72.9% for validation data and 76.67% for test data. The model clearly overfit, as indicated by their accuracy and loss graphs below:

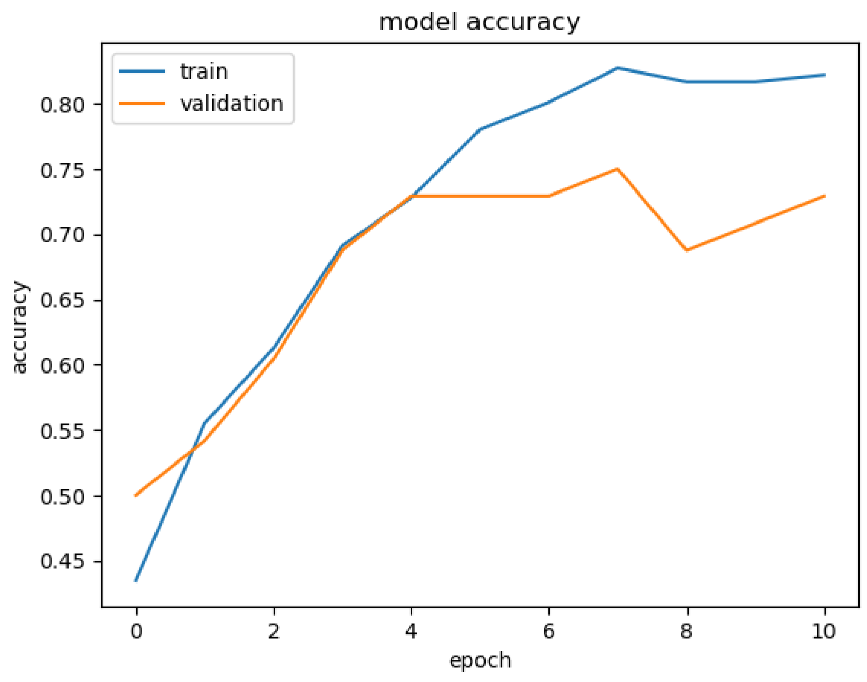


図 2. Model accuracy evolution during training

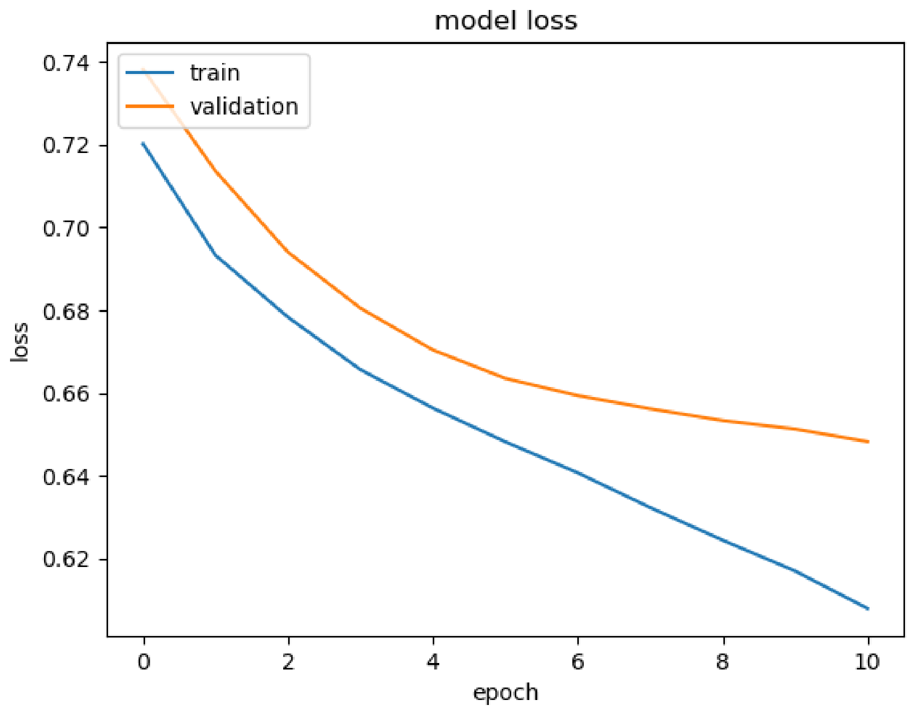


図 3. Model loss evolution during training

And, when we check the confusion matrix, we notice the model could not predict the cases in which the patient died of heart failure.

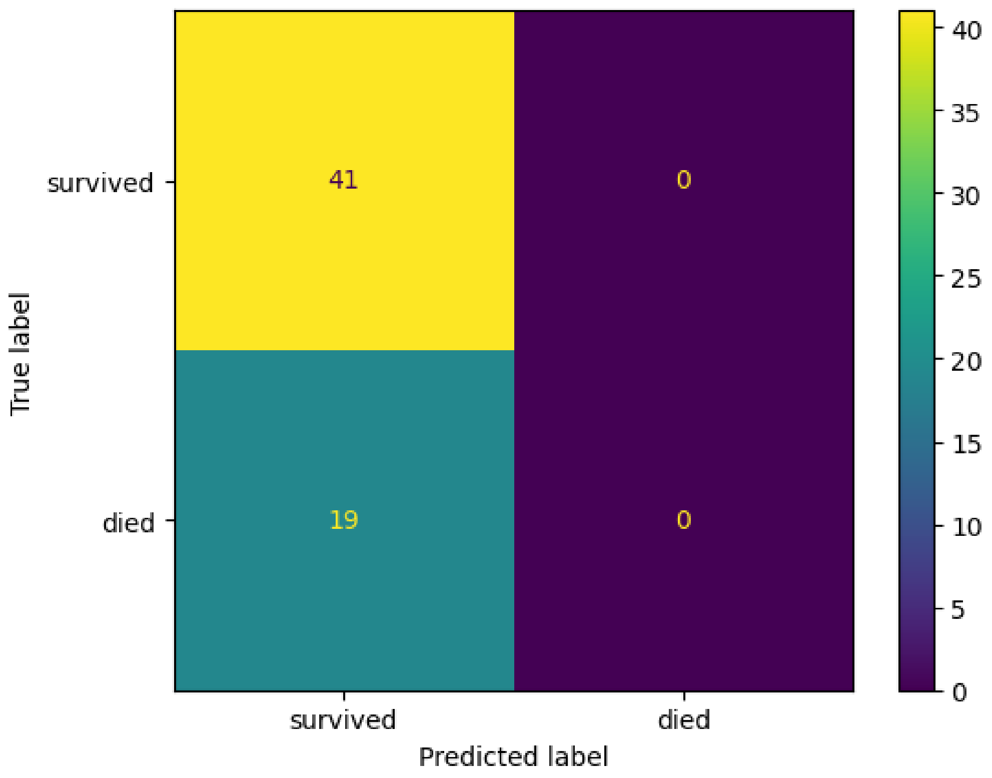
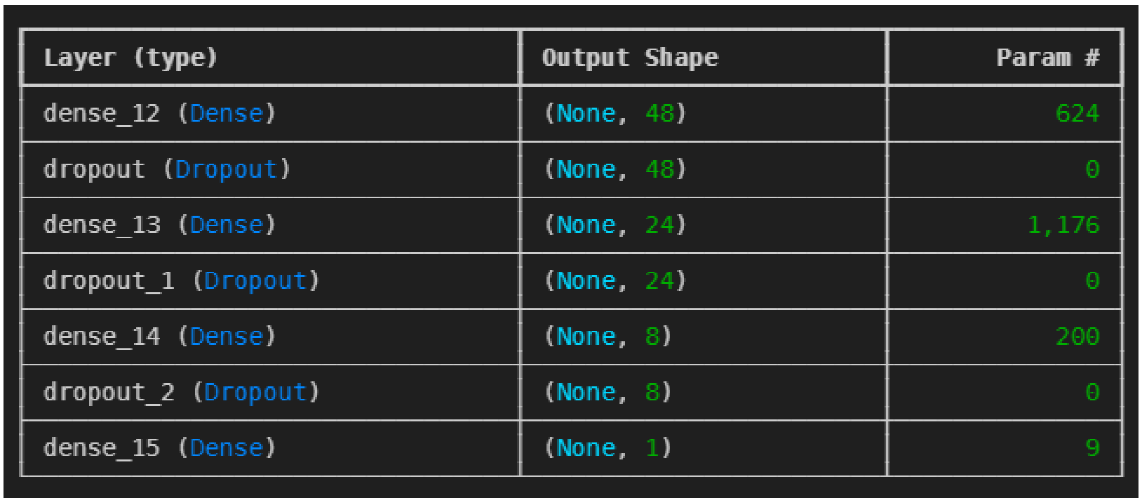


図 4. Confusion matrix comparing predicted and actual labels in test data

After that, we proceed with altering the model architecture by adding a Dropout layer in the hidden layer. The model structure then became



Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 48)	624
dropout (Dropout)	(None, 48)	0
dense_13 (Dense)	(None, 24)	1,176
dropout_1 (Dropout)	(None, 24)	0
dense_14 (Dense)	(None, 8)	200
dropout_2 (Dropout)	(None, 8)	0
dense_15 (Dense)	(None, 1)	9

The activation functions were preserved. It was only trained in 7 epochs because of the early stopping, and it returned an accuracy of 69.4% for training data, 81.25% for validation data and 70% for test data. And, as indicated by their accuracy and loss graphs below, it performed worse:

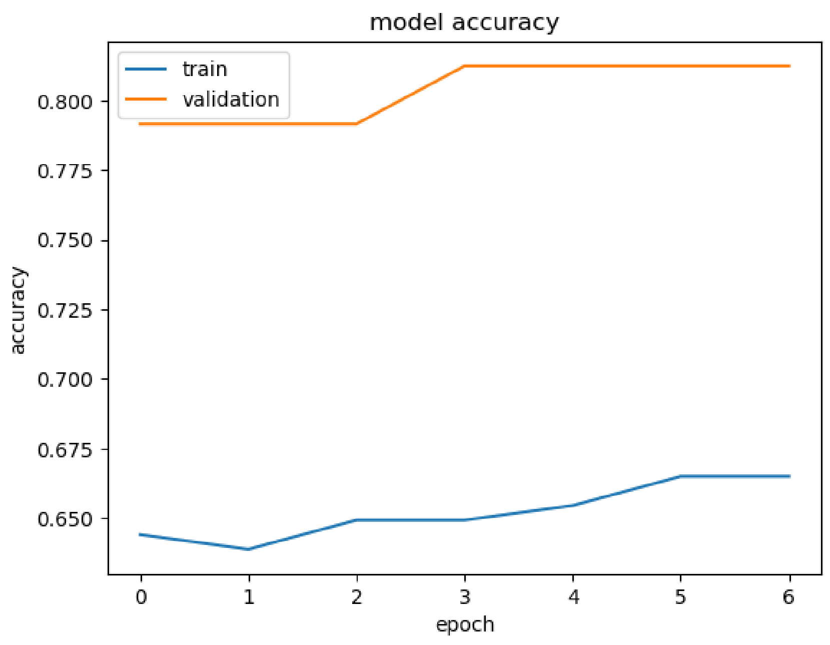


図 5. Model accuracy evolution during training

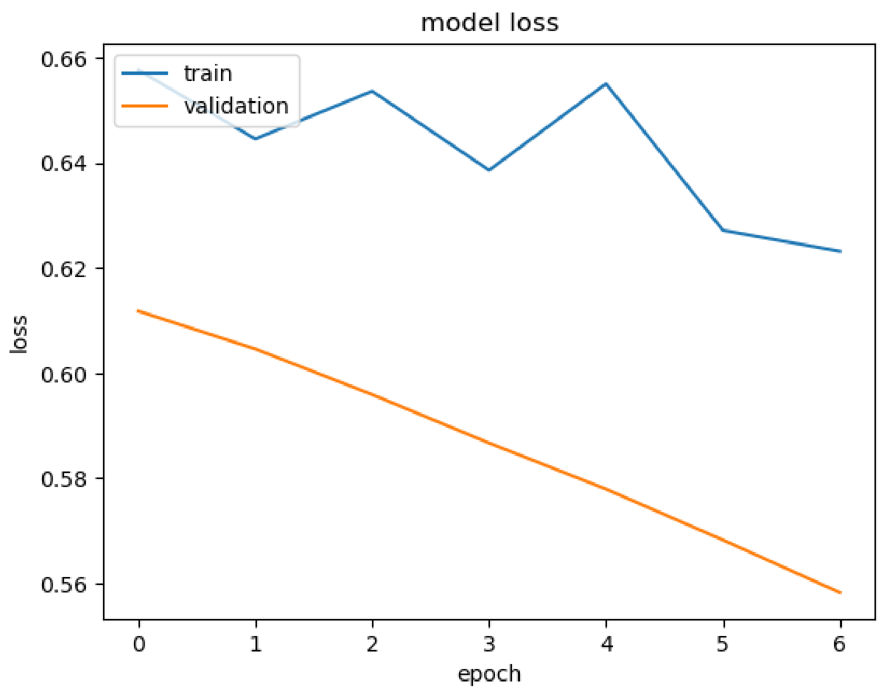


図 6. Model loss evolution during training

And the confusion matrix remained unchanged:

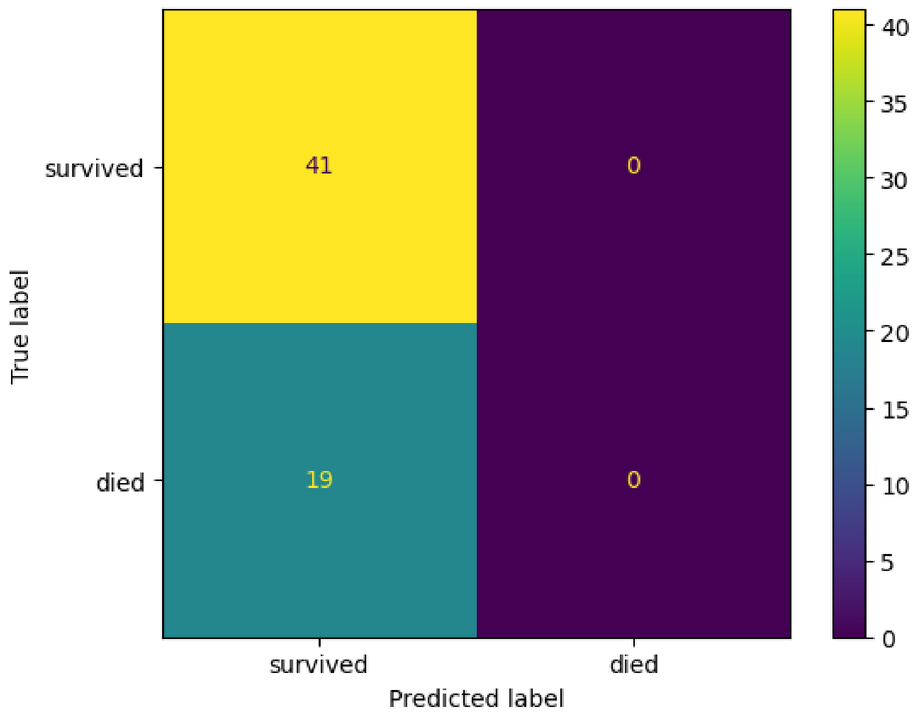
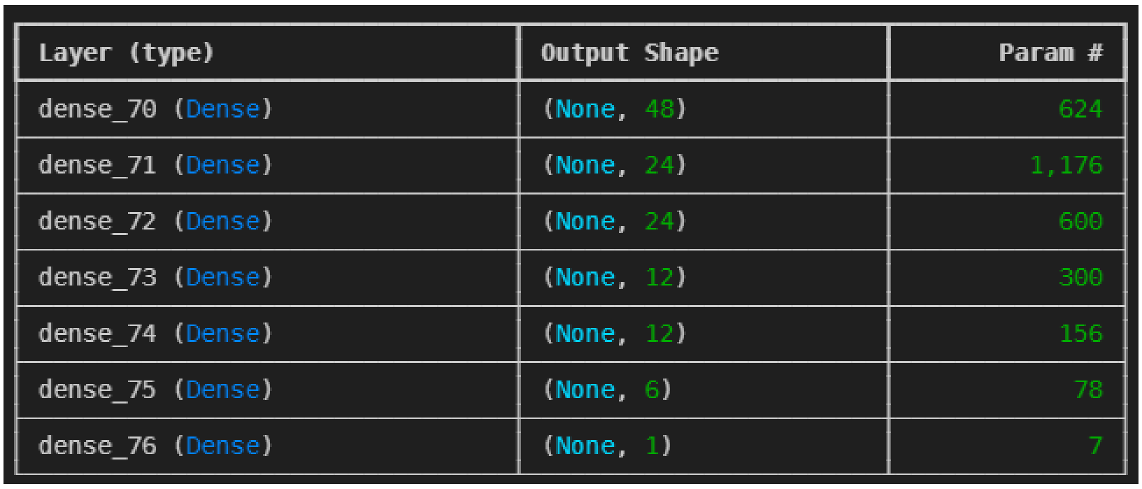


図 7. Confusion matrix comparing predicted and actual labels in test data

To circumvent this issue, the model was once again changed. Its number of layers was increased and the dropout was removed. The structure of the third model became the following:



Layer (type)	Output Shape	Param #
dense_70 (Dense)	(None, 48)	624
dense_71 (Dense)	(None, 24)	1,176
dense_72 (Dense)	(None, 24)	600
dense_73 (Dense)	(None, 12)	300
dense_74 (Dense)	(None, 12)	156
dense_75 (Dense)	(None, 6)	78
dense_76 (Dense)	(None, 1)	7

The activation functions were preserved. It was trained in 13 epochs because of the early stopping (which had an increase in its patience), and it returned an accuracy of 86.1% for training data, 70.83% for validation data and 80% for test data. Despite overfitting, it went better than the previous model, as indicated by the graphs:

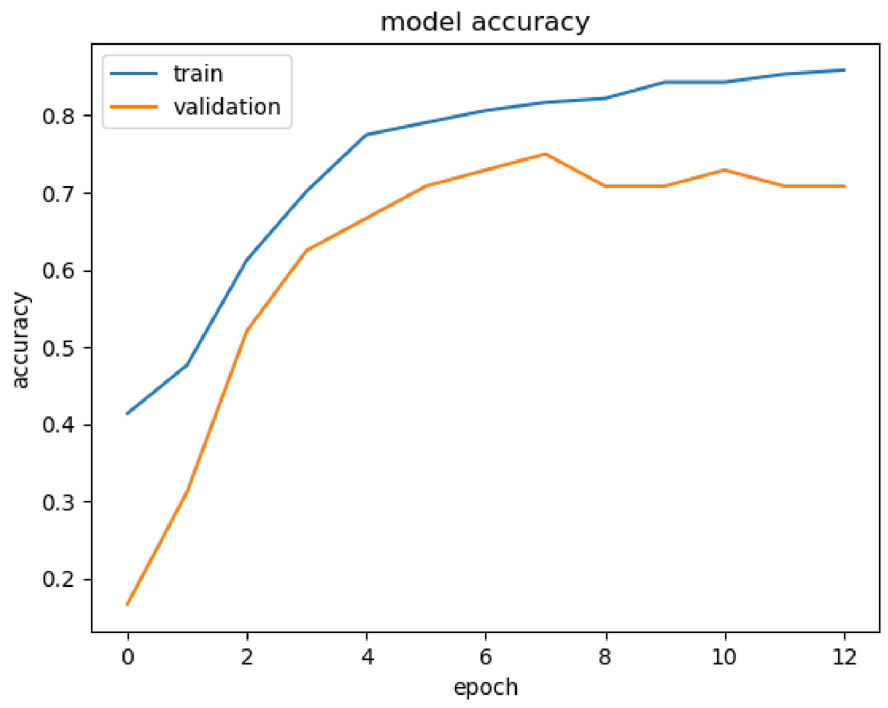


図 8. Model accuracy evolution during training

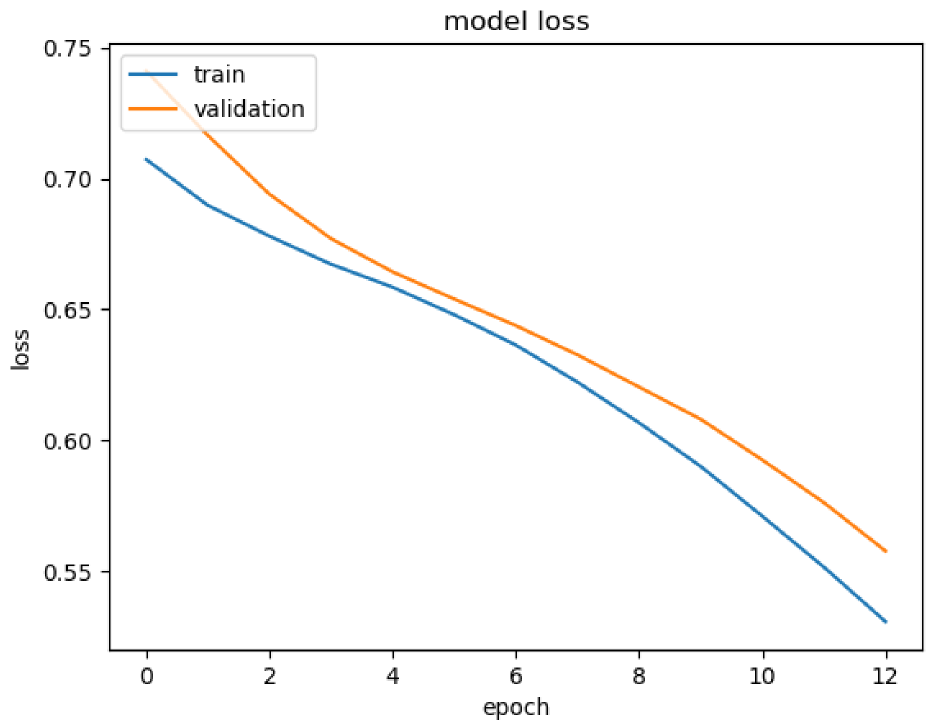


図 9. Model loss evolution during training

However, when checking the confusion matrix, it still could not predict correctly any case of patient dying of heart failure.

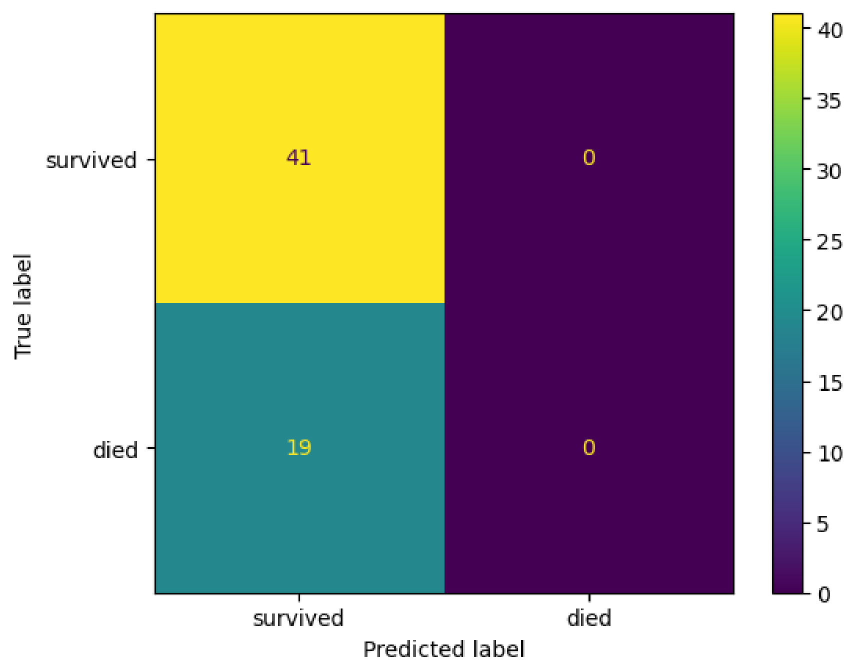


図 10. Confusion matrix comparing predicted and actual labels in test data

5. Recommended model

If we consider solely the accuracy aspect, the third model went the best, despite the first model also being a valid option. However, both models overfit, and, as indicated by the second model, using dropout is not the best approach to solve this issue. Also, in all cases, the model failed to predict the class 1, which indicates it could not learn the factors which contribute to a patient dying after having a heart failure.

6. Insights

The main noticed aspect in the data is its imbalance. Basically, for every 2 samples of patients surviving the heart failure, one sample represents a patient who unfortunately couldn't resist. This imbalance makes harder for the model to predict the class underrepresented (and might have contributed to the failure indicated in the confusion matrix). Also, the differences in the ranges for each feature made data scaling required.

7. Next Steps

The next steps to be taken are focused in two directions: reducing overfitting and dealing with the data imbalance. Strategies like data undersampling or oversampling, using

Dropout more efficiently or finding better alternatives, or even testing with other types of neural networks, are to be considered.

8.