POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Project report

Myocardial infarction complications prediction

Statistical learning for healthcare data

Authors: Sofia Di Filippo, Chiara Mocetti, Gabriele Moro and Filippo Pagella

Academic year: 2022-2023

1. Introduction

Myocardial infarction (MI) is a term for an event of heart attack. MI occurs when blood stops flowing properly to a part of the heart and the heart muscle gets injured because of lack of oxygen supply, leading to possible necrosis if reperfusion of the organ is not performed in short times. Despite the remarkable advances in its treatment, MI remains a very dangerous disease and the most common cause of heart failure. It is widely spread in all countries, especially in the developed ones where people are more exposed to chronic stress and unhealthy lifestyles. The course of MI in patients is very diverse, thus it is very difficult to foresee the development of complications which can even lead to death.

In the present study, we analyse a database collected in the Krasnoyarsk Interdistrict Clinical Hospital in Russia between 1992 and 1995 with the goal to present a model able to classify a patient in two categories: dead or alive. We also want to show how the classification changes if we consider only the information obtained when they are admitted and those collected in the three days of hospitalisation.

2. Data and methods description

The dataset contains information about 1700 patients through 124 features: variables from 2 to 112 describe the clinical picture of the patient, from 113 to 124 represent possible complications of the MI. In particular, the last column is called 'LET_IS' and it represents the lethal outcomes: it is a categorical variable where value '0' means that the patient is alive, whereas values from '1' to '7' are the different causes of death.

Since we are only interested in classifying the patients in dead or alive, we can immediately drop columns from 113 to 123 as they represent other possible outcomes, and we make variable 'LET_IS' binary where value '0' means once again that the patient is alive and value '1' that they died. The obtained data matrix has 1700 rows and 113 columns.

2.1. Data cleaning, missing values imputation and preparation

The vast majority of the features are ordinal or binary, while only nine are continuous. There are no duplicate data in the original dataset.
Real-life clinical data typically contain missing values, so the first problem we had to face was their imputation (initially the 8.32% of data were missing). In this section we explain the procedures we followed to clean the dataset and impute the missing values.

2.1.1 Dropping irrelevant features

The first thing we did was to get rid of all features containing more than 60% of missing values and all rows with more than 20% of missing values. By doing so, we lost the features 'IBS_NASL' (heredity on Chronic Heart

Disease (CHD)), ‘S_AD_KBRIG’, ‘D_AD_KBRIG’ (systolic and diastolic blood pressure according to the Emergency Cardiology Team (ECT)) and ‘KFK_BLOOD’ (serum CPK content). Given the high percentage of missing values, the loss of these features is not an issue, furthermore for the blood pressure we still have the measurements according to the Intensive Care Unit (ICU) in features ‘S_AD_ORIT’, ‘D_AD_ORIT’. By choosing to get rid of data with more than 20% of missing values, we still remain with 92% of data.
At the end of this initial and brutal cleaning, we have a dataframe of dimension 1566x109 and the percentage of missing values has dropped to the 3.77%.

2.1.2 Outliers detection

We performed outliers detection on ‘AGE’, ‘S_AD_ORIT’, ‘D_AD_ORIT’, ‘K_BLOOD’ (potassium concentration), ‘NA_BLOOD’ (sodium concentration), ‘ALT_BLOOD’, ‘AST_BLOOD’ (transaminases concentration), ‘L_BLOOD’ (white blood cells count) and ‘ROE’ (erythrocyte sedimentation rate) which are continuous features. We noticed some values were null which is physically impossible, so we decided to treat null values of the blood pressure as missing values. For the other features, we saw there were a lot of outliers, but we could not find enough evidence in the literature to get rid of them since, in case of heart attack, these values can differ remarkably from the nominal ones. However, we could observe a pattern in the outlier distribution: case by case, we got rid of values which were too spread, but only if the corresponding patient had some missing values. We obtained a dataframe with dimension 1557x109 (3.76% of missing values).

2.1.3 A "manual" missing values imputation

Now we discuss a first approach on the missing values imputation. We started by considering the variables with less than 20% of missing data. In the ordinal and binary features, we imputed those values using the Scikit-Learn’s SimpleImputer which replaces missing values using a descriptive statistic (in our case: “most frequent”, the mode) along each column. For the continuous variables, we imputed the missing data using the mean of the columns or, when the difference was significant, the mean differentiating between male and female patients, dead or alive.
For the remaining features with more than 20% of missing values, we noticed that some of them indicate the use of drugs which can be given to the patient both by the ECT (‘NA_KB’, ‘LID_KB’ and ‘NOT_NA_KB’) or in the ICU (‘NA_R_1_n’, ‘NA_R_2_n’ and ‘NA_R_3_n’; ‘LID_S_n’; ‘NOT_NA_1_n’, ‘NOT_NA_2_n’ and ‘NOT_NA_3_n’). Since we have already imputed the missing data in the features related to the ICU, we decided to eliminate the corresponding ECT features. Like we did with the blood pressure, if we have the same data according to the ECT and the ICU, we consider the most accurate one, which are the ones with less missing values.
The last four features we imputed were ‘GIPO_K’ (hypokalemia (<4 mmol/l), binary) and ‘K_BLOOD’ (serum potassium content, continuous), ‘GIPER_NA’ (increase of sodium in serum (>150 mmol/l), binary) and ‘NA_BLOOD’ (serum sodium content, continuous). The binary features represent a content in the serum higher than a specific value. Since the imputation of continuous features is easier, we decided to drop ‘GIPO_K’ and ‘GIPER_NA’ and to impute the missing values in ‘K_BLOOD’ and ‘NA_BLOOD’ using the Deterministic Regression Imputation, which replaces the missing data with the values predicted by a regression model and repeat this process for each variable.

2.1.4 Multiple Imputation by Chained Equations

We also considered an automated procedure to solve the problem of missing values: the Multiple Imputation by Chained Equations (MICE), implemented by the Scikit-Learn’s IterativeImputer. The algorithm initializes the imputation of the missing values through a simple strategy (i.e. mean or most frequent value) and then it imputes them through an iterative series of predictive models. In each iteration, each specified variable in the dataset is imputed using the other variables until all specified variables have been imputed. The procedure lasts for 10 iterations.
We divided the dataset into continuous and categorical variables, applying the algorithm with two different initializations for the missing values: mean and most frequent value, respectively. We obtained two imputed dataset that, after applying the imputation algorithm, we merged to obtain our final imputed dataset. The parameters of the IterativeImputer method have been set to the default ones.
We obtained a completely imputed dataset with 0 missing values.

2.1.5 Data preparation

The last problem we needed to tackle was organizing our data. There are nine features which refer to different times: after 24 hours from the admission, after 48 and after 72. We decided to merge the following features: 'R_AB_1_n', 'R_AB_2_n' and 'R_AB_3_n' in 'R_AB' (relapse of the pain); 'NA_R_1_n', 'NA_R_2_n' and 'NA_R_3_n' in 'NA_R' (Use of opioid drugs in the ICU); 'NOT_NA_1_n', 'NOT_NA_1_n' and 'NOT_NA_1_n' in 'NOT_NA' (use of NSAIDs in the ICU). This seemed reasonable since, in order to evaluate any change in the prediction of our model, we were interested in knowing if at least once in the three days following the admission in the hospital, the patient had a relapse of the pain or needed the drugs.

2.2. Methods

We tackled the problem by making use of three different approaches, taking into account their theoretical aspects and comparing their performances in order to choose the final model. Considering the medical framework the problem lives in, we decided to construct our own performance index, which could help us choose the optimal parameters to improve our models. We called this measure "MoP" (Measure of Performance) and we define it as follows:

$$\text{MoP} = 0.5 * \text{Sensitivity} + 0.3 * \text{Precision} + 0.2 * \text{Specificity}$$

Sensitivity has half of the weight because our goal is to build a classifier that misclassifies True Deads very few times. Precision and Specificity have the other half of the weight because we still want to have a good trade-off in terms of accuracy and reliability of the results.

2.2.1 Data pre-processing

We split our dataset in train and test set assigning them, respectively, the 70% and the 30% of the initial data. We noticed the ratio of elements of class '1' in the train set was around 15%. Working with imbalanced classes is dangerous since the model could perform badly. To avoid that, we decided to oversample the minority class. New examples can be synthesized from the existing ones. This type of data augmentation is referred to as the Synthetic Minority Oversampling Technique, or SMOTE for short.

Before training the models, data have been normalized using Scikit-Learn's MinMaxScaler.

2.2.2 First Method

The first model we tried was a Random Forest which consists of a large number of individual, uncorrelated decision trees that operate as an ensemble. The advantages of using this model are many: Random Forests work very well with categorical data and, since the output is based on majority voting or averaging, they solve the problem of overfitting. On the other hand, a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In order to optimize our Random Forest Classifier we performed a tuning of the hyper-parameters of the model, such as number of threes, depth of the forest, number of leaves, threshold, using Scikit-Learn's RandomizedSearchCV method: we defined a grid of hyper-parameter ranges, and randomly sample from the grid, performing 5-folds Cross-Validation with each combination of values.

The main limitation and the reason why, even given the very good performance of this model, we decided to go in another direction is that Random Forests are a predictive and not a descriptive tool. If we are interested in a description of the relationships in our data, other approaches are to be preferred.

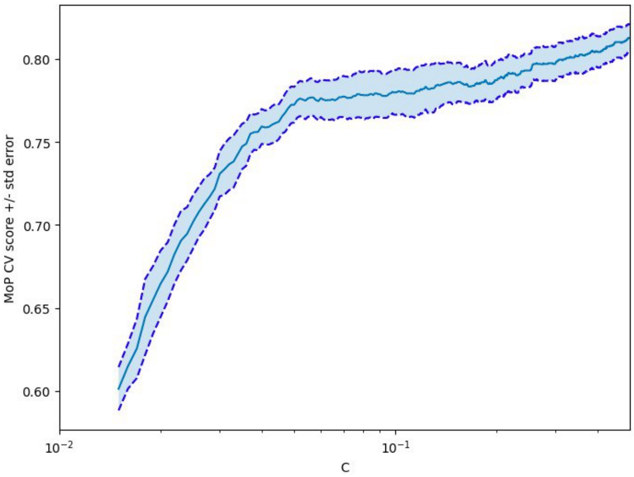
2.2.3 Second Method

With the aim of exploiting the Random Forest power without losing interpretability, we decided to order the features from the most to the least important one according to the previous Random Forest model. From this, using the forward stepwise selection, we built a logistic regression classifier. L2 penalty was used and the performance measures of the model were computed with 5-folds Cross-Validation. After confronting the performance measures with respect to the number of features, we decided to keep the first 20 variables since that represented the "elbow" value.

To increase the sensitivity of the model, we decided to tune the threshold of the logistic regression, which until now was set to the default value of 0.5, via Cross-Validation, maximizing MoP. This led us to choose the value 0.45 for the threshold.

2.2.4 Third Method

The third method consists in fitting a pure logistic regression classifier to our data. To make our results as much interpretable as possible, we considered necessary to perform some feature selection in order to reduce the dimensionality of our problem. Therefore, we added an L1 penalty that allowed us to reach a model with good performances, making use of 28 out of the original variables. The choice of the optimal penalty parameter C has been made through the maximisation of the MoP, searching on a grid of values which allowed us to significantly shrink the set of covariates.



Observing the plot, for the regularization parameter C we chose the value 0.1, in correspondence of the elbow. Finally, we considered the possibility of tuning the value of the threshold in order to increase the sensitivity, but the results we obtained were too expensive in terms of all the other measures decreasing, therefore we decided to stick with the default parameter of 0.5.

3. Results

In this section, we report the principal results obtained testing the three models presented before on the two datasets where missing values have been imputed "manually" and automatically with the MICE algorithm.

	"Manual" missing values imputation			MICE		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
Accuracy	0.8137	0.6724	0.7559	0.8158	0.7388	0.7944
Precision	0.4417	0.3020	0.3525	0.4286	0.3434	0.4071
Sensitivity	0.7260	0.8356	0.6712	0.6857	0.8143	0.8143
Specificity	0.8299	0.6421	0.7716	0.8388	0.7254	0.7909
MoP	0.6615	0.6368	0.5957	0.6392	0.6552	0.6874
AIC		40.5	41.07		43.02	59.02
AUC		0.7389	0.7214		0.7699	0.8026

As we have already mentioned, we decided to award a model that not only could perform well, but also give interpretable results. Thus, we decided to go in the direction of the logistic regression. In particular, the best model, in terms of performances and explainability, is model 3 trained and tested using the MICE dataset. We report below, as a comparison, the confusion matrices of model 1 and model 3 using the MICE dataset.

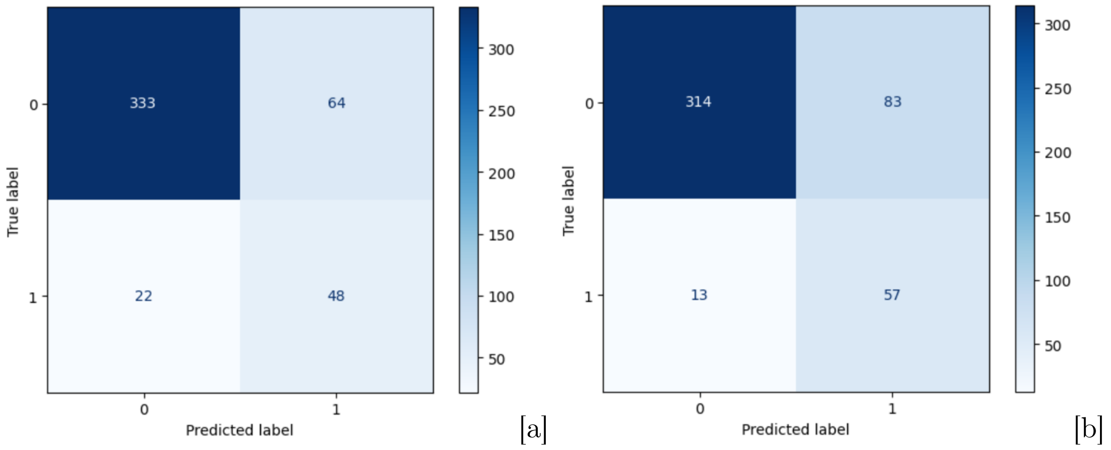


Figure 1: [a] Confusion matrix of model 1, [b] Confusion matrix of model 3.

Even if the Random Forest performs better, with the Logistic Regression we can better control the false negatives.

Finally, we tested model 3 using the 28 selected features plus the 3 additional ones we mentioned in section 2.1.5. By doing so, we could investigate how the performance changed if we had information during the three days of hospitalization of the patient.

	At the time of admission	In the 3 days period
Accuracy	0.7944	0.7880
Precision	0.4071	0.4000
Sensitivity	0.8143	0.8286
Specificity	0.7909	0.7809
MoP	0.6874	0.6905
AIC	59.02	77.28
AUC	0.8026	0.8047

We can see that the model maintains a good level of performance and we are pleased to notice a slightly improvement in the sensitivity.

Once again, we report the confusion matrix, this time of model 3 trained with the data obtained at the time of admission and those of the three days period.

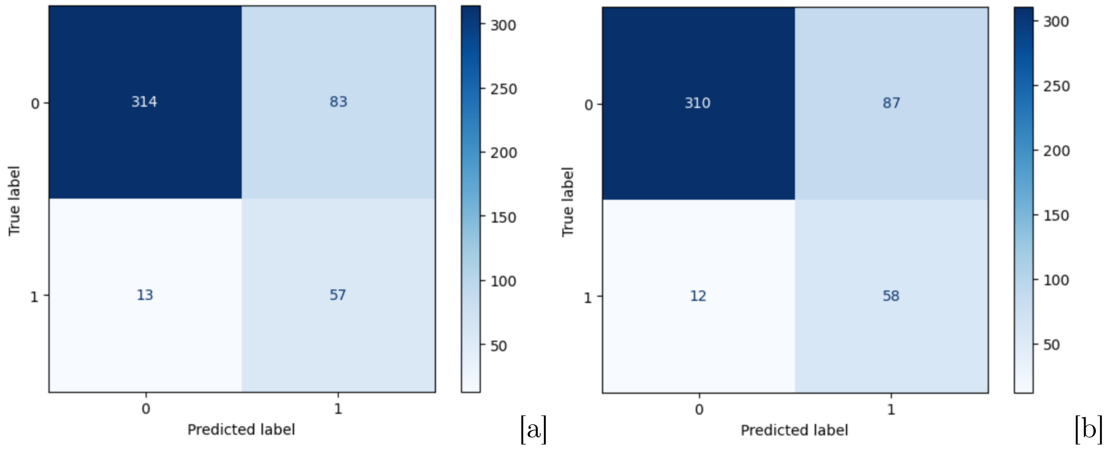


Figure 2: [a] Confusion matrix of model 3 at the time of admission, [b] Confusion matrix of model 3 after the 3 days period.

4. Conclusions

A deep understanding of the risk factors that can lead to mortality after MI is important to guide clinicians to recognize patients at higher risk. Therefore, we consulted literature and made a comparison between the current models used to estimate a patient's risk of mortality after MI and ours. Those models take a complete picture of the patient's condition, considering demographic and clinical information prior to hospitalisation and the presentation characteristics at the admission to the hospital, when the myocardial infarction occurs [2]. Our final model is able to do so, selecting only 28 of the original features. In particular, it considers:

- Sociodemographic characteristics (Age, Sex)
- Risk factors (Diabetes, Obesity, Hypertension)
- Presence of comorbidities (Chronic bronchitis and Chronic obstructive bronchitis in the anamnesis)
- Presentation Characteristics (Arrhythmias such as atrial fibrillation, Cardiogenic shock)
- Initial Diagnostic Studies (Electrocardiogram)
- Pharmacological treatments, used to lower blood pressure, to reduce heart's oxygen demand and to limit the risk of blood clots (Drugs administered in Intensive Care Unit: Beta-blockers, Nitrates, Calcium Channel Blockers, Aspirin)

Comparing the final model with the others, we can say that it performs better in terms of features selection. For instance, it considers diabetes which is a well-known risk factor: several studies clearly show that diabetic patients have a higher mortality rate after MI with respect to non-diabetic ones [3, 5]. Other important factors taken into account are presence of comorbidities, such as chronic obstructive pulmonary disease (COPD), and arrhythmias. From literature we know that COPD is common in patients with MI who have substantially greater mortality rate than patients without. Collectively, these findings suggest that COPD is not only common but also a significant marker for adverse clinical outcomes after a MI [6]. Arrhythmias (atrial fibrillation and right bundle branch block) discovered at the time of admission through the electrocardiogram (ECG) are evaluated as important risk factors by the major recognized prediction model scores [2].

If we look at the model performance using the information collected during the three days period, we see a very slight improvement. The three additional variables appear not to be determinant to assess positive or negative outcome after MI. Even in the literature, the discussion about the effects of opiods and other drugs on cardiovascular patients is still open [1, 4].

In conclusion, we are satisfied by the final results. Our final model considers a complete set of risk factors that match those of affirmed models already in use, and are supported by clinical studies. This indicates that it could be applied to real clinical scenarios.

References

[1] Mickael Bonin, Nathan Mewton, Francois Roubille, Olivier Morel, Guillaume Cayla, Denis Angoulvant, Meyer Elbaz, Marc J Claeys, David Garcia-Dorado, Céline Giraud, et al. Effect and safety of morphine use in acute anterior st-segment elevation myocardial infarction. *Journal of the American Heart Association*, 7(4):e006833, 2018.

[2] Yulanka Castro-Dominguez, Kumar Dharmarajan, and Robert L McNamara. Predicting death after acute myocardial infarction. *Trends in Cardiovascular Medicine*, 28(2):102–109, 2018.

[3] Sean M Donahoe, Garrick C Stewart, Carolyn H McCabe, Satishkumar Mohanavelu, Sabina A Murphy, Christopher P Cannon, and Elliott M Antman. Diabetes and mortality following acute coronary syndromes. *Jama*, 298(7):765–775, 2007.

[4] Ji Quan Samuel Koh, Himawan Fernando, Karlheinz Peter, and Dion Stub. Opioids and st elevation myocardial infarction: a systematic review. *Heart, Lung and Circulation*, 28(5):697–706, 2019.

[5] Lene Rytter, Svend Troelsen, and Henning Beck-Nielsen. Prevalence and mortality of acute myocardial infarction in patients with diabetes. *Diabetes care*, 8(3):230–234, 1985.

[6] Adam C Salisbury, Kimberly J Reid, and John A Spertus. Impact of chronic obstructive pulmonary disease on post-myocardial infarction outcomes. *The American journal of cardiology*, 99(5):636–641, 2007.