POLITECNICO
MILANO 1863

Statistical Learning for Healthcare Data: Heart Failure project

Authors: Nicole Fontana, Annachiara Rossi, Eleonora Spizzi

GitHub repository for the project code here

Data: July 8, 2022

---

1. Introduction

Heart Failure (HF) is a common reason for hospitalization in the elderly and it is associated with significant mortality and morbidity.
The goal of this report is to illustrate results on the prediction of patients' risk of readmission at six months after hospital discharge following HF and to identify the relevant patients' characteristics for the task. In particular, our main concern is to understand whether drug therapy information is a significant addition to all the other features in the distinction among the two classes.

2. Materials and Methods

The data consists of a retrospective HF dataset (available at PhysioNet [3]), created by using electronic health data collected from patients who were admitted to a hospital in Sichuan, China (2016-2019).
The dataset includes 165 variables for 2008 patients with HF. Among them, we have demographic data, clinical data, comorbidities, laboratory findings, drugs and the outcome (i.e. readmission within six months).

2.1. Data exploration

A step of data exploration was conducted as first analysis. Initially, we decided to remove all the features for which more than 50% of the values were missing. Indeed, filling so many missing values would inevitably introduce randomness in our study. This led to 122 features.
At this point we examined the resulting dataset more in detail, analyzing the features one by one, to have a better understanding of their distribution and of the sample of patients at our disposal.
We kept track of the very unbalanced and of very skewed features. We looked at the distributions to highlight the presence of outliers, observations presenting out-of-range values, along with non-physiological values. The latter were probably due to measurement errors, as a cross check with other values of the patients proved, and as such we decided to flag them as NaN, to eventually be filled.

2.2. Data cleaning

First of all, a redundant feature was noticed, indicating if a patient was readmitted to the emergency department. Clearly, it gives us information which should not be known at the time we want to predict the possibility of readmission of a patient. Hence, it was excluded from our dataset.
Moreover, looking closely at all columns, we noticed that some were including information about readmission at shorter periods in time with respect to our target. They presented some inconsistencies, as patients registered to be readmitted within 28 days or 3 months were not at 6 months. We updated this information in our target and dropped the other columns.
Another important note is about patients who died before leaving the hospital: they will be of no use in trying to predict the readmission after 6 months. Before discarding these patients, we tried to analyse them qualitatively and we observed their degree of consciousness expressed by *'GCS'* (and the 3 features *'eye.opening'*, *'movement'* and *'verbal.response'*). We noticed something curious: while 5 of these 16 patients reported the minimum score in all 3 categories, indicating a state of severe coma (we have coma for GCS $\leq$ 8), many other had the full score (GCS=15), indicating they were fully responsive to every kind of stimulus, but then they died anyway.

After these modifications which are not influ-

enced by the use we will make of our data, we splitted our samples in training and test set (80% and 20%). We stratified over the outcome, since it is sligthly unbalanced (36% of samples in the positive class), to have the same proportions in the two sets. From now on, all the inspections will be referred to the train set and the transformations will be inferred on the test set using only training information. This allows to keep a portion of data totally unseen and, in the end, to have a reliable estimate of what would be the predictive capability of the model built.

We started removing (almost) zero-variance features, choosing a very low threshold of 0.05, since a variable which has no dispersion in its values cannot be used to explain the variability in another variable. Hence, it would not be relevant in the explicability of the outcome.
Patients for which much information was not registered (more than 25% of missing features) were removed from the dataset. Indeed, imputing their missing values from the other features might bias too much our predictions and we deemed better not to include them in our study.
Then, we focused on the analysis of non-continuous variables. First, we distinguished them in binary, ordinal and categorical, since only the latter will need one-hot encoding. In our first inspection, in the file "1_Data_preprocessing", we noticed a strong unbalance in several variables. We selected all the features with an unbalanced proportion $\geq 80\%$ and examined if this unbalance was also reflected in the target. We only kept the features satisfying this property (i.e. those that induced a significant difference among the classes of the target) and discarded the rest.
Regarding continuous variables, a correlation analysis was our first step. As the complete matrix is too big to get a clear understanding, we looked closer at couples of features with a correlation >0.9 and examined them one by one, deciding which of the two features to keep based on literature, meaning of the variables and their possible inclusion in one another.
Just to make an example: we had a big correlation (0.92299) between *'map'* (i.e. Mean Arterial Pressure) and *'diastolic.blood.pressure'*. Knowing that in our dataset we also had the systolic blood pressure and knowing that MAP is the synthesis of the 2 quantities, we decided to discard MAP, to be sure not to have redundant information, which could lead to a loss in interpretability for some models (collinearity problem). Similar reasonings were conducted on other features.

2.3. Joining the drugs dataset

Information about drug therapy has been added to our main dataset at this point. Since several drugs are taken into consideration, we deemed necessary to group them according to their aim and/or their acting principle, to avoid the addition of 18 dummy variables to an already high-dimensional dataset. After some reaserch, five groups were identified: diuretics, drugs to cure hypertension, drugs to treat heart failure, drugs to treat Angina pectoris and other cardiac pathologies and, finally, drugs to lower high cholesterol.
Since every patient takes an arbitrary quantity of drugs (also none), we merged the two datasets by patient ID over the main dataset. In this way, patients not taking drugs would get all 0's and patients already removed in the original dataset would not appear again. Obviously, each patient will have a 1 in the column concerning the group it takes drugs from. After this pre-processing phase, we got to a dataset with 82 columns.

2.4. Feature selection techniques

In view of the high-dimensionality of the dataset, various feature selection techniques were implemented, as a prior step for some classification models. In particular:

a) SelectKBest: it looks at the differences among classes in the outcome induced by each feature using Anova testing, and removes all but the k features having smallest p-value;

b) SelectFromModel: it finds the best features based on the coefficients obtained fitting the given estimator (in our case, the Elastic-net Logistic Regression which puts the weight of some features automatically to 0);

c) Recursive Feature Elimination: it uses the given estimator (Perceptron [*c*1] and DecisionTree [*c*2] in our case) to compute coefficients of the features, and recursively eliminates the one having smaller weight. The optimal number of features is selected through a CV loop.

2.5. Models' definition

Prior to any modelling, some data transformations should be applied. As cited in subsection 2.1, some continuous variables present a very skewed distribution: a logarithm transformation might be of help for models (the plots of the distributions can be found in "01_Data_preprocessing").
All the pre-processing of our data and its modelling has been set up through a Pipeline. This allowed us to properly and easily cross-validate the models to compare them with one another and avoid optimistic estimates of the generalization error. Indeed, extracting the training set after the pre-processing Pipeline would apply the transformations to the whole training set. Instead, with our procedure, every time a K-fold cross-validation is applied, K-1 folds are used as training and the transformations on the last fold (the validation fold) will be inferred from that (`fit_transform` vs `transform`). All this avoids data-leakages, which give away information about the validation set in the training and one should be very cautious on that.
The Pipeline includes the standardization of continuous variables, and the encoding of categorical variables into dummies using the One-Hot technique. Missing values filling has been carried out with a KNNImputer, which infers a missing value using the 5 nearest neighbors over the other features.
We tried several models and ways of treating the imbalance of our target classes. We deemed appropriate the use of penalized learning algorithms, that increase the cost of classification mistakes on the minority class. At the end, a trial was also done using SMOTE: this method over-samples the minority class by adding some synthetic points with similar characteristics to a minority class point. We didn't expect this to be the best approach, since it generates new, fictitious, observations.
In the following, we give some insights on our choices and motivations.

Logistic regression: Since collinearity among features might still be present, we considered an Elastic net penalty, which takes into account both the L2 norm and the L1 norm, which induces sparsity. We used the `saga` solver, since it is the only one supporting the ElasticNet regularization. Moreover, we increased the number of iterations to guarantee convergence. The hyperparameters `C`, the inverse of regularization strength, and `l1_ratio` have been tuned to get a strong enough regularization and promote feature selection.

SVM: We tried this method to deal with the high-dimensionality of the data, thanks to the kernel trick. For this reason, no prior feature selection is implemented in this case. We used a Radial Basis Function kernel, which is the most seen in literature, for which we tuned the parameter γ. Also here, we tuned the inverse of the regularization `C`.

Decision Tree: The classification tree is able to fit training data until every observation is in a leaf. For this reason, if no constraints are added, it heavily overfits the training set. To avoid this, we did some trials: it emerged that using the balanced `class_weight` the performance on validation set improves and setting the `max_depth` parameter to be small reduces the overfitting a lot. So, we tuned the `max_depth`, the `criterion` to be *Entropy* or *Gini* and the `min_samples_split`, which is the minimum number of samples in a node to proceed in splitting again.

Random Forest: This algorithm averages the estimates from many trees to reduce the variance, while keeping the bias low. As such, it will be able, tuning the right parameters of the base tree, to generalize better. We did a feature selection with different techniques as explained in subsection 2.4 and fine-tuned `min_impurity_decrease, min_samples_leaf, min_samples_split, max_leaf_nodes, n_estimators, max_depth`. For more details, please see the related code "4_Models".

Extreme Gradient Boosting: The last idea was to use this state-of-the-art ensemble method, which is a more regularized form of Gradient Boosting. Hyper-parameters were tuned following the project in [8].

Logistic Regression with SMOTE: Since the over-sampling should only be done during the training phase, it should be added to a Pipeline able to fit the transformation over the training set, but not act on the test set in the transform

step. For this, we introduced a new Pipeline from the `imblearn` library. Then, we repeated the same process of logistic regression.

3. Results

We evaluated the methods presented in subsection 2.5 by cross-validating the performance on the training set. The same CV scheme is used for all models, and consequently results are comparable. For the algorithms which also needed tuning, a nested cross-validation strategy was implemented: it allows to find parameters trough a Grid-Search for every cross validation fold and at the same time to get an estimate of the chosen score. This ensures the significance of the estimates and avoids optimistic measures of performance, which would then be overturned by testing the final model on the hold-out set.
The scoring used is the *F1*. We choose to favor this metric as it is an average between Recall and Precision, taking into account both mistakes on the positive and on the negative class. Indeed, from the hospital perspective having a false positive means that a patient that would be likely readmitted actually won't. Thus they would prepare resources that then won't be used. On the contrary, having a false negative would mean to have a patient we would not expect. Hence, we judged both mistakes as crucial.
We reported the value of the *F1-score* for training and validation sets in Table 1.

	train	validation
LR-ElasticNet	0.613	0.548
SVM	0.686	0.527
DT	0.553	0.524
RF	0.561	0.505
RF & *a*	0.563	0.501
RF & *b*	0.566	0.510
RF & *c*(1)	0.567	0.508
RF & *c*(2)	0.555	0.504
XGBoost	0.728	0.511
LR-ElasticNet, Smote	0.609	0.545

Table 1: F1 scoring on training and validation data

Looking at these results, the model having the best performance in terms of F1 is the Logistic Regression with Elastic-Net penalty. We can see that the variant with SMOTE has similar performance, but lower. SVM and XGBoost have still some overfitting problems, while the other methods work well in this sense. Comparing the Random Forests using the four feature selection methods, we notice that the best one comes from the covariates of the Elastic-Net model. In the end, the chosen model is the first one: the performance on validation is acceptable and interpretability is ensured thanks to the fact that the model is linear. Thus, in this case a simpler model leads to the best results.

We fitted the final model on the whole training set and tested it on our hold-out set to get an estimate of its predictive capabilities. In Table 2 are reported the F1 and the AUC on train and test set.

Best Model: LR-ElasticNet

	AUC	F1
Train	0.661	0.602
Test	0.607	0.539

Table 2: AUC & F1 on train and test set

To have an idea of the drivers in the prediction of the readmission within six months, we show in Figure 1 the 15 highest coefficients in the Logistic Regression model.

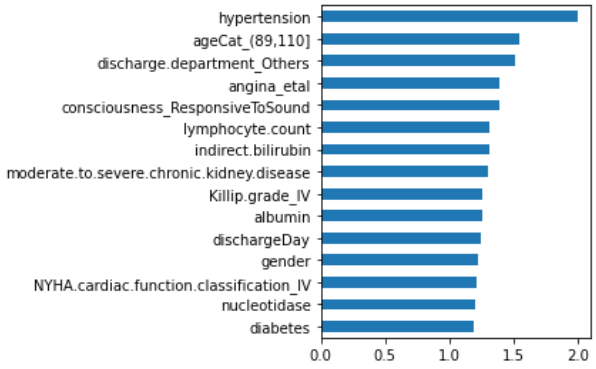

Figure 1: Features' importance from LR-ElasticNet model

4. Discussion and Conclusions

Our main research question is whether the information about drug therapy is a relevant addition to the other clinical features. From Figure 1 we notice that some categories of drugs result to be important predictors for the readmission after 6 months. In particular, *'hypertension'*, but also *'angina_etal'* and *'diabetes'*. We think that this significance might be a consequence of the presence of a pathology, which is trying to be cured (diabetes, hypertension and cardiac diseases like Angina pectoris), rather than the side effects induced by a certain type of drug. As a matter of fact, being subject to another disease, increases the risk of readmission to the hospital. The presence of the Kidney disease can be similarly interpreted.
Regarding the other important features, the Killip grade 4 expresses a high risk of death within the first 30 days after myocardial infarction. Among the socio-demographic variables, we find *'ageCat_(89,110]'* and *'gender'*: the elderly have a greater risk of readmission and from literature we can state that, in general, men have a lower life expectancy and are more likely to suffer of some kind of disease. All these findings are supported by previous studies in the field, like [1], which includes age, gender and comorbidity conditions as relevant factors.

We can state that all significant features lead to a reasonable explanation. However, predictive capabilities of the model are contained (see Table 2). Nevertheless, our results are in line with those of literature. In [1] a systematic review of several statistical models for readmission after HF supports our unsatisfactory results. The best results are also in these cases given by multivariate logistic regression models. We even managed to improve them a little bit: they show a C-statistic (equivalent to our ROC-AUC) of 0.6, very close to our result of 0.607. We might be tricked into believing that a better result is achieved by Felker et al. in [1], with a C-statistic equal to 0.69, but this model was never validated, so we cannot be sure of its performances on newly seen data.

Looking at our results and to the systematic review in [1], we can affirm patient characteristics provide only modest information about the readmission risk. A possible explanation is that other non-patient related factors may be more important in assessing readmission risk, like the medical insurance, the hospital in which the patient was admitted the first time, the post-discharge care. In other words, the dependence might be more on the healthcare system's rather than on the patient's characteristics: to verify this hypothesis, a different kind of data would be needed, focusing on administrative data instead of clinical ones. This kind of study has indeed been taken into consideration by M. S. Khan [2].
To conclude, information about drug therapy seems beneficial for the prediction of readmission after 6 months from the HF of a patient, even if from a clinical perspective it is difficult to stratify patient risk and more advanced solutions should be explored.

References

[1] J. S. Ross, G. K. Mulvey et al., *Statistical Models and Patient Predictors of Readmission for Heart Failure - A Systematic Review*, American Medical Association, 2008.

[2] M. S. Khan, J. Sreenivasan et al., *Trends in 30- and 90-Day Readmission Rates for Heart Failure*, Circulation: Heart Failure, 2021.

[3] Z. Zhang, L. Cao et al., *Electronic healthcare records and external outcome data for hospitalized patients with heart failure*, Springer Nature, 2021.

[4] *All You Need To Know About Different Types Of Missing Data Values And How To Handle It*
`https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/`

[5] *Data Leakage in Machine Learning*
`https://machinelearningmastery.com/data-leakage-machine-learning/`

[6] *How to Handle Missing Values in Cross Validation* `https://towardsdatascience.com`

[7] *How to select best features to improve the accuracy of your machine learning model* `https://medium.com`

[8] *XGBoost Parameters*
`https://xgboost.readthedocs.io/en/stable/parameter.html`

[9] *Nested versus non-nested cross-validation*
https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html

[10] *Recursive Feature Elimination for Feature Selection in Python*
https://machinelearningmastery.com/rfe-feature-selection-in-python/

[11] SMOTE for Imbalanced Dataset
https://iq.opengenus.org/smote-for-imbalanced-dataset/

[12] Information about drugs and clinical variables:
https://en.wikipedia.org

[13] Information about drugs and clinical variables:
https://www.humanitas.it/enciclopedia/principi-attivi

[14] Information about drugs and clinical variables:
https://my.clevelandclinic.org