Heart failure is a common reason for hospitalization in the elderly and it is associated with significant mortality and morbidity.

The data consist of a retrospective heart failure dataset created by using electronic health data collected from patients who were admitted to a hospital in Sichuan, China between 2016 and 2019.

The dataset includes 168 variables for 2008 patients with heart failure.

GOAL: To predict the readmission at 6 months after hospital discharge by including also information about drug therapy. The drug therapy is an important feature for the outcome?

PHASE 1: DATA PREPROCESSING

As an initial step, we examined our dataset. First, we decided to remove all the features for which more than 50% of the values for the patients were missing, believing that filling so many missing values would inevitably introduce randomness in our model. Thus, we removed 43 columns of our dataset, getting to a dataset with 122 features.

At this point we tried to examine the resulting dataset more in detail, analysing all the columns, one by one, to build ourselves an idea of our sample of patients. We kept track of the very unbalanced features and of the distribution of values of a specific feature (and whether our patients presented in-range values or not). We also identified in this way some patients which had non-physiological values for some features, probably resulting from a measurement error and we changed those incorrect values to NaN.

PHASE 2: DATA CLEANING

At the end of this stage, we proceeded with a cleaning of our dataset.

First of all, we noticed that our target column (‘re.admission.within.6.months’) presented some inconsistencies with respect to other two analogue columns at different times (28 days and 3 months). We proceeded to correct them, including the information contained in the other two columns, and then we removed the other two variables (‘re.admission.within.28.days’ and ‘re.admission.within.3.months’), which were redundant.

At this point, we proceeded to split our dataset in training and test sets (80% vs 20% of the data).

After that, we decided to remove a-priori from our model the features that have (almost) zero variance (the ones with a var<0.05), since a variable which has no dispersion in its values, cannot be used to explain the variability in another variable and so we can affirm that they will not be relevant to determine our output. Thus, our 120 features reduced to 96.

Then, reasoning about the aim of our project, we had to remove from our dataset the 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<=8), many other had the full score (GCS=15), indicating they were fully responsive to every kind of stimulus, but then they died anyway.

We proceeded then to remove all the patients (22) that had >25% of missing values in their features: too little information is known about these patients to be actually of any help in the analysis, and even imputation does not guarantee good results when the missing values are too many.

After that we moved our focus to the analysis of not-continuous variables. First, we distinguished them in: binary, ordinal and categorical, since only the categorical will need one-hot encoding. We wanted to see if all these variables would be useful in our study. In “Data preprocessing” in fact, we noticed a big unbalance in several variables. We thus selected all the features with an unbalanced proportion >=80% and examined if this unbalance was reflected also in the target: we kept those for which this property held and discarded the rest. The reasoning behind this choice was the following: if the unbalance of the feature is reflected in the target (e.g. Consider a binary variable with 99% of 1 and 1% of 0: if all the 0s (which are just a few) are in all in the same class of the target), then this could hold some relevant information; otherwise, the big unbalance in the feature risks to compromise our results. In such a way, we reduced the number of our columns to 85.

After the analysis of these variables, we dedicated ourselves to the observation of the continuous ones. Of course, the correlation matrix is too big to get a clear understanding, so we select the 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, etc. 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. We did similar reasoning for all the other couples (more can be found in the comments of our script). Dropping these features, we get to a dataset with 78 columns.

PHASE 3: JOINING THE DATASET OF THE DRUGS

In parallel, we dedicated ourselves to the dataset of the drugs: we noticed that this dataset had much more rows than the other (12654 vs 2008): so, we wanted to check if there were patients for which there was more than one row in the table (i.e. patients taking more than one drug). There were indeed a lot of them (10650). Thus, we realized we would need to reformulate this dataset in a way such that every patient would correspond to just one row, on which we would have the information about all the drugs he takes.

Before doing so, we did some research about the drugs of our dataset and noticed they can be subdivided into 5 main groups according to their aim and/or their acting principle: 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. We adjusted the dataset adding this categorization, which we believed might come in handy in our future analysis.

After having removed the column with the name of the specific drugs (to only keep into account their aim), we could proceed with the adjustment of our dataset explained above. To do so, we needed to get from each patient ID a row having 1 if the patient has at least a 1 in that column, 0 otherwise. This could be simply achieved by taking the maximum value for each patient on each column.

At this point, we were finally able to merge the drugs dataset with our main dataset over the patient ID (based on the patient ID in the main data frame, because some patients have already been removed for other reasons). We thus obtained our final dataset with 83 columns.

An important thing to notice is that the number of patients in the drug dataset is 2004, while in the original dataset we had 2008. Thus, there should be 4 patients which are not registered because they're not taking any drug.

PHASE 4: MODELS DEFINITION

Before starting our actual analysis, actually we had to take some more steps, and we did so by means of a pipeline: we took the logarithm transformation of some continuous variables with a very skewed distribution (the plots of the distributions can be found in “01\_Data preprocessing”), we standardized the other continuous variables, and we did One-Hot Encoding on our categorical variables. At this point, our dataset counted 108 columns. Outside of the pipeline we also proceeded to the imputation of the remaining missing values, by means of a KNN imputer (we could not do it inside the pipeline because the method ‘transform’ is missing).