Myocardial infarctions complications dataset - Preprocessing, classification and evaluation

Evgeny Manturov

March 14, 2024

1 About the dataset

Link to the dataset:

```
https://archive.ics.uci.edu/dataset/579/myocardial+infarction+complications
```

The dataset has been created and analysed by S. E. Golovenkin et.al.. The dataset represents the records of patients, whose data was collected at the Krasnoyarsk state medical university. It contains information about 1700 patients and 110 features characterizing the clinical phenotypes and 12 features representing possible complications of the myocardial infarction disease [1].

2 Data Exploration

The dataset contains 15974 missing values out of $1700 \times 111 = 188700$ data entries. There are no missing values in column representing age and in label columns. The columns contain different datatypes: continuous (float), integer, categorical (encoded state) and boolean. Specifically, there are 11 boolean and 1 categorical variable labels. This means that they will have to be processed and predicted separately by the model. This affects performance, since the model has to be re-trained for each variable.

3 Data Preprocessing

Firstly, as suggested and done by Golovenkin et.al., the dataset is cleansed of entries and parameters that are too impure (contain too many NaN values). The thresholds are set to 20% of values for rows and 30% of values for columns. After their removal, the amount of NaN values is reduced to 4473, with 544 entries now containing no NaN values.

In order to attempt further preprocessing, it is needed to fill the remaining NaN values with sensible and as accurate as possible data. In contrast to performing SVD on the 544 full rows and then projecting the remaining data onto the domain of reduced dimensionality, performed by Golovenkin et.al.[1], the experimental multivariate iterative imputation of variables was used. This algorithm uses surrounding data and similar entries to predict the missing values and fill them in. This method may produce different inference in future versions of sklearn, thus it should be used with care, and can be replaced with a simpler imputation algorithm.

Then:

- Correlation matrix was built, with several rows having more than 0.8 correlation coefficient:

  ```
  STENOK_AN FK_STENOK 0.8450335438356769
  S_AD_ORIT D_AD_ORIT 0.8331132905991112
  MP_TP_POST ritm_ecg_p_02 0.845850627969873
  #If LET_IS is one-hot-encoded:
  RAZRIV LET_IS_3 1.0
  ```

  Here, "STENOK_AN" and "FK_STENOK" are Exertional angina pectoris in the anamnesis and Functional class of angina pectoris, and "S_AD_ORIT" and "D_AD_ORIT" are Systolic and Diastolic blood pressures. These are not entirely the same and it was decided to keep them.

  Paroxysms of atrial fibrillation, represented by "MP_TP_POST" and "ritm_ecg_p_02", are essentially the same variable and either of the 2 can be removed to reduce set dimensionality without much effect on the final metrics.

  In general, a myocardial rupture is almost always lethal. For this dataset, "RAZRIV" is a label for rupture, and "LET_IS_3" is a death with a cause being rupture. These labels represent the same, so "RAZRIV" can be removed.

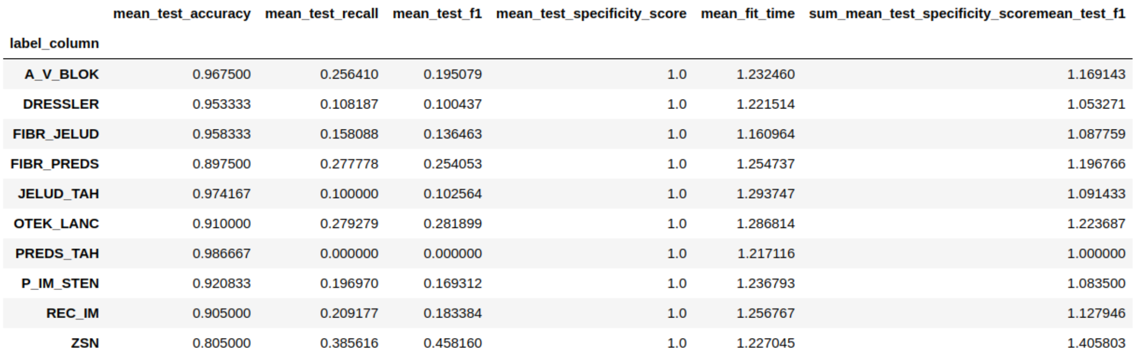
The overall step is there to discover possible patterns, which can be discussed/confirmed with a real physician (e.g. if a myocardial rupture is 100% lethal).

- Categorical data cannot be passed over to a classification model - it has to be one-hot-encoded via **pandas.get_dummies**.
- In an attempt to reduce the set's dimensionality, Principal Component Analysis was performed. This SVD algorithm attempts to keep the dataset's properties for the model, but reduce the dimensionality, sacrificing part of the data, which cannot be explained by variance. The dataset was reduced to 52 columns, retaining 90% of data. This, however, is not included in the final model, since the marginal improvement in fit/predict time leads to reduction in F1 score and specificity.

4 Classification methods and performance evaluation

The classification algorithm is to be selected by performance, since it is a dataset with vast class imbalance, few entries and too many dimensions. SVD is not applicable here by design, so the 4 models tested were Extreme Gradient Boosting (xGB), Random Forests, Decision Trees and Multinomial Naive Bayes.

- All 4 models were iteratively given to a 6-fold (and later 3-fold) Grid search cross-validation algorithm. The metrics selected were F1 score (the score representing both TN and TP rates well) and specificity score, best for medical usage (it's better to falsely predict a condition that falsely predict its absence). The sorting algorithm cannot perform unbiased sort of a combination of metrics, so their sum was used as a representative of both metrics being highest possible.
- Categorical label ((non)lethality and reason of death) was grid-searched first. It was determined that **Decision trees with GINI criterion** were the best of all algorithms.
  Decision trees is a simple criteria-based selection algorithm, assigning criteria with strong class separation (high GINI) to the top of the tree and weaker criteria to the bottom of the tree. It is equivalent to multiconditional assignment (if... and... and..., then the features have label X).
- For binary labels, the same grid-search CV proved that the set imbalance is, in fact, so high, that some classes were never seen in the evaluation part of the set, which meant that the values could not be properly defined:



	mean_test_accuracy	mean_test_recall	mean_test_f1	mean_test_specificity_score	mean_fit_time	sum_mean_test_specificity_scoremean_test_f1
label_column						
A_V_BLOK	0.967500	0.256410	0.195079	1.0	1.232460	1.169143
DRESSLER	0.953333	0.108187	0.100437	1.0	1.221514	1.053271
FIBR_JELUD	0.958333	0.158088	0.136463	1.0	1.160964	1.087759
FIBR_PREDS	0.897500	0.277778	0.254053	1.0	1.254737	1.196766
JELUD_TAH	0.974167	0.100000	0.102564	1.0	1.293747	1.091433
OTEK_LANC	0.910000	0.279279	0.281899	1.0	1.286814	1.223687
PREDS_TAH	0.986667	0.000000	0.000000	1.0	1.217116	1.000000
P_IM_STEN	0.920833	0.196970	0.169312	1.0	1.236793	1.083500
REC_IM	0.905000	0.209177	0.183384	1.0	1.256767	1.127946
ZSN	0.805000	0.385616	0.458160	1.0	1.227045	1.405803

Figure 1: Binary labels classification metrics

Specificity can be considered ill-defined if there are unrepresented classes. The weighed multiclass specificity for lethality on train data is around 0.5, but is below 0.2 on test data, which means that the classic algorithms cannot accurately predict myocardial complications.

5 Conclusions

Although the classic classification models have proven use in data science for medicine, they work best with well-balanced and low-dimensional data or data with a large and representative cardinality. The dataset presented requires a unique principal tree algorithm, which allows tracking and construction of figure-based paths, which has been built by the researchers' team as shown in Figure 2 [1]. Projecting on real-life, rare cases of death and rare complications are to be evaluated individually, with models usable only if enough data is collected throughout time in the whole world.

Other possible improvement may be the use of imbalanced-learn library for data oversampling, which may still be questionable, because if SMOTE or ADASYN are used, the generated samples train off imputed values, and these values are fed to the classifier. No regenerative algorithm will fully replace real data, and both scores for classifiers and diversity for generative models fall if they are trained on model-generated content.

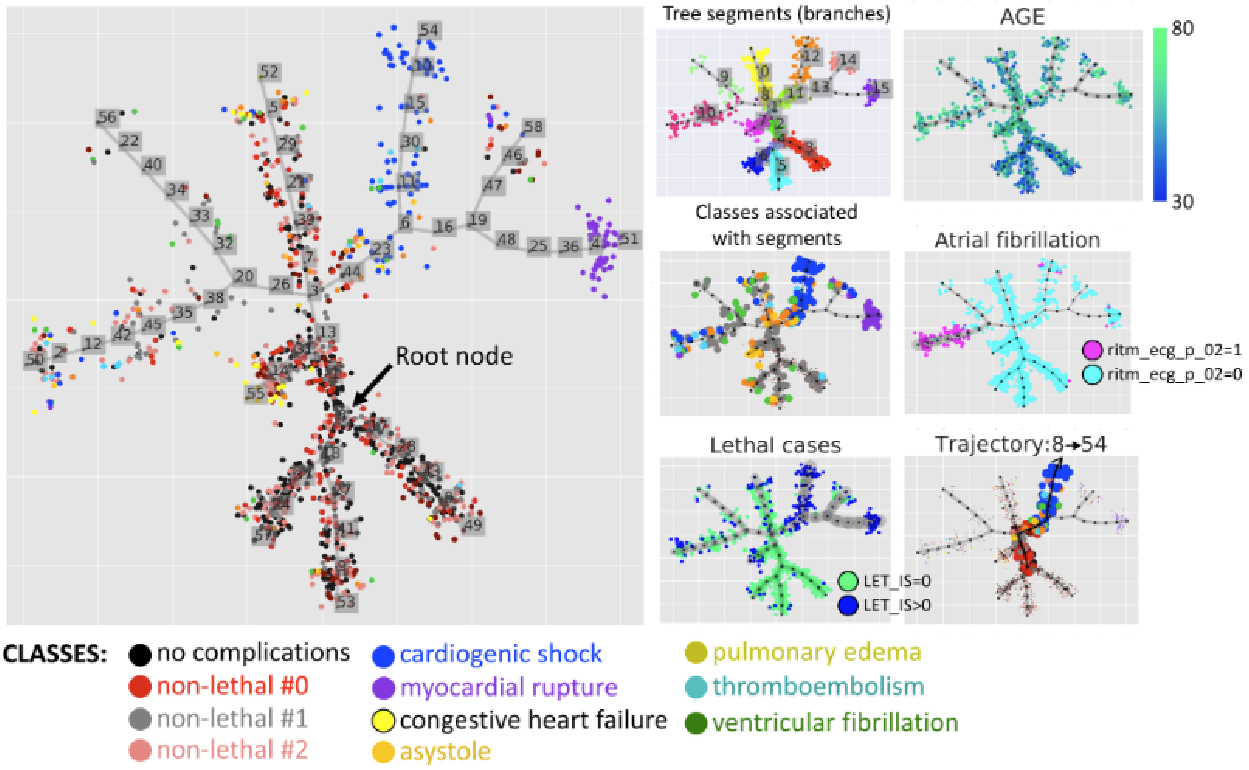


Figure 2: Principal tree for the miocardial dataset [1]

The overall effectiveness prevents this inference from being useful when communicating with a physician. The ML models used are interpretable, but are not so easy to understand (specifically the most effective of them by F1 score - xGB and Random forests - are ensembles, which may use inference of many different algorithms, including black box algorithms such as SVM).

The attempt to construct a proper DNN algorithm was also performed. Attempted Tensorflow-Keras implementation can be observed in the **main-keras.py** file. This idea was scrapped due to overall better performance of conventional algorithms.[3]

References

[1] Golovenkin, S.E., Bac, J., Chervov, A.V., Mirkes, E.M., Orlova, Y.V., Barillot, E., Gorban, A.N., & Zinovyev, A.Y. (2020). Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data. GigaScience, 9.

[2] Gorban AN, Rossiev DA, Butakova EV, Gilev SE, Golovenkin SE, Dogadin SA, et al. Medical and Physiological Applications of MultiNeuron Neural Simulator. In: International Neural Network Society Annual Meeting; Lawrence Erl-baum Associates, vol. 1; 1995. p. 170–175.

[3] Ravid Shwartz-Ziv, Amitai Armon, Tabular data: Deep learning is not all you need, Information Fusion, Volume 81, 2022, Pages 84-90.