The Effect of Missing Data when Predicting Readmission in Heart Failure Patients

Filip Plesinger[1], Zuzana Koscova[1], Eniko Vargova[1], Jan Pavlus[1], Radovan Smisek[1], Ivo Viscor[1], Veronika Bulkova[2]

[1]Institute of Scientific Instruments of the CAS, Brno, Czech Republic
[2]Medical Data Transfer, s. r. o., Brno, Czech Republic

Abstract

Background: *The discharge of patients from hospital care is regulated by guidelines. Still, readmission of heart failure (HF) patients is a common issue, and several calculators have been published to predict it.*

Aims: *We elaborate on how the prediction performance decreases when features become missing. We also elaborate on which features should a user include every time to reach acceptable prediction performance.*

Method: *We prepared a balanced dataset from HF patients in the MIMIC-III database (N=2,204) with 16 features. Using training data (80%) in a four-fold cross-validation manner, we evaluated all feature combinations ($N=2^{16}-1$) and found the optimal feature set for the logistic regression model. We also evaluated feature presence in top-performing models (N=655) and identified mandatory features. Finally, we trained the resultant model using all training data and evaluated the effect of missing features ($N=2^8$ combinations) using separate test data (20%).*

Results: *We identified three mandatory features (age, blood urea nitrogen, and systolic blood pressure) and eight optional. This led to a resultant model with eleven features. The hazard ratio (HR) using test data showed a value of 2.08 (95%CI 1.66-2.61) when all eleven features were present. It also showed an HR of 1.73 (95%CI 1.39-2.17) when only three mandatory features were present, and others were missing (i.e., replaced by zeros).*

1. Introduction

Predicting readmission in heart failure patients can be solved using machine learning methods implemented as a web service. In this paper, we elaborated on the optimal implementation strategy if the user is allowed to leave some feature input fields unused: should we use a single model and the missing features strategy or specifically trained reduced models? How these strategies affect prediction results? And could we identify mandatory features, which shall be entered every time by the user?

2. Data

We extracted 2,204 records from MIMIC-III database[1] accessed via PhysioNet[2]. Each record was described by manually selected 16 features. The feature set consists of demographic features (gender, age), laboratory features (creatinine, sodium, blood urea nitrogen - BUN, potassium, glucose, hematocrit, magnesium, phosphate, chloride), and features describing rhythm (systolic blood pressure - SBP, diastolic blood pressure - DBP, heart rate - HR, and respiratory rate - RR). All features were acquired as a mean from the last 24 hours before discharge (excepting laboratory values). Each record was accompanied by follow-up (up to 3 years) and event type (readmission or all-cause death). We used patient state in one month after discharge from the intensive care unit as the output (balanced) for training models.

3. Method

Dataset (Fig.1A) is split into training and testing part (Fig.1B). Training part is used for feature exploration and selection (Fig.1C). To minimize a risk of overfitting, feature exploration and selection is done in four-fold cross-validation (CV). For each CV iteration, we standardized data and built logistic regression model for all features. Next, we iterated over all feature combinations ($N = 2^{16}-1 = 65,535$). For each combination step, specific features were replaced by zero and tested using data from the resting fold.

Therefore, we obtained four F1 scores for each of 65,535 feature sets. Based on mean F1 and its standard deviation, we manually selected optimal feature set (Fig.1E). Based on feature presence in 1% of top performing models, we also identified features which should be treated as "substantial" (Fig.1F; also Fig.2); rest of features was treated as "optional".

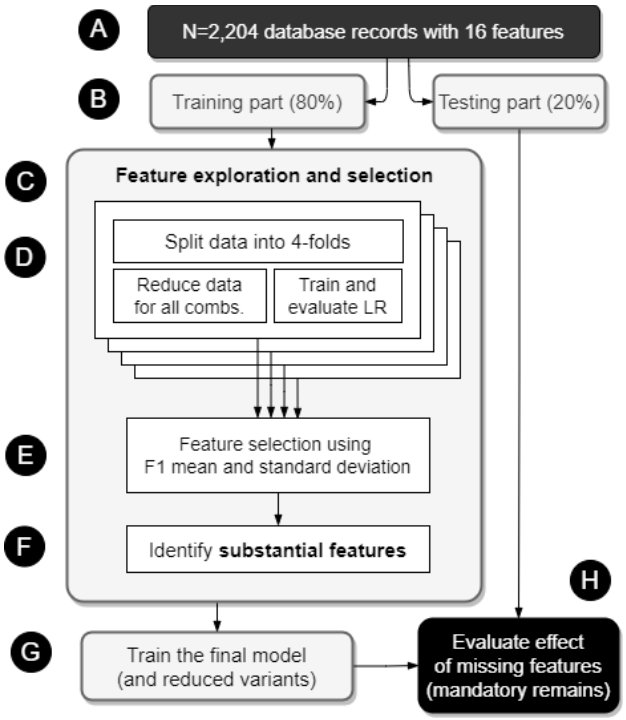

Figure 1. Method flowchart. Data(A) are split into training and testing part (B). Testing part is used for feature effect exploration (C), consisting of cross-validation evaluation (D) of all feature combinations. Final feature set is identified (E) as well as substantial features(F). Final model is trained (G) and the effect of missing features is evaluated (H).

Next, we trained the final model using all training data (standardized) and evaluated its performance on all combinations of missing optional features (mandatory features were present every time). We also trained reduced models from respective feature sets for later comparison.

The method was implemented in the Python 3.10[3] supported with Pandas[4], scikit-learn[5], and matplotlib[6] packages. We used Wilcoxon non-parametric pair test to evaluate differences between approach with reduced models and approach using missing values strategy from scikit-learn Python package[5]. Hazard ratio including confidence intervals (CI) was evaluated using GraphPad Prism 9.5.1 software (GraphPad Software, LLC, MA, USA).

3. Results and Discussion

3.1. Feature exploration and selection

Figure 2 shows 1% (655) of the best performing models; reported F1 score was computed as mean from 4-fold cross validation. It shows that age, BUN, and SBP should be considered mandatory since they are present in all of these best models (Fig2., blue-highlighted features). We also manually removed five features due to linear links or expected inaccessibility in our future project application (Fig2, gray-highlighted features).

A different point of view on the same results is shown in Figure 3, telling how often each feature shares a model with the other one; it also reflects linear relationship.

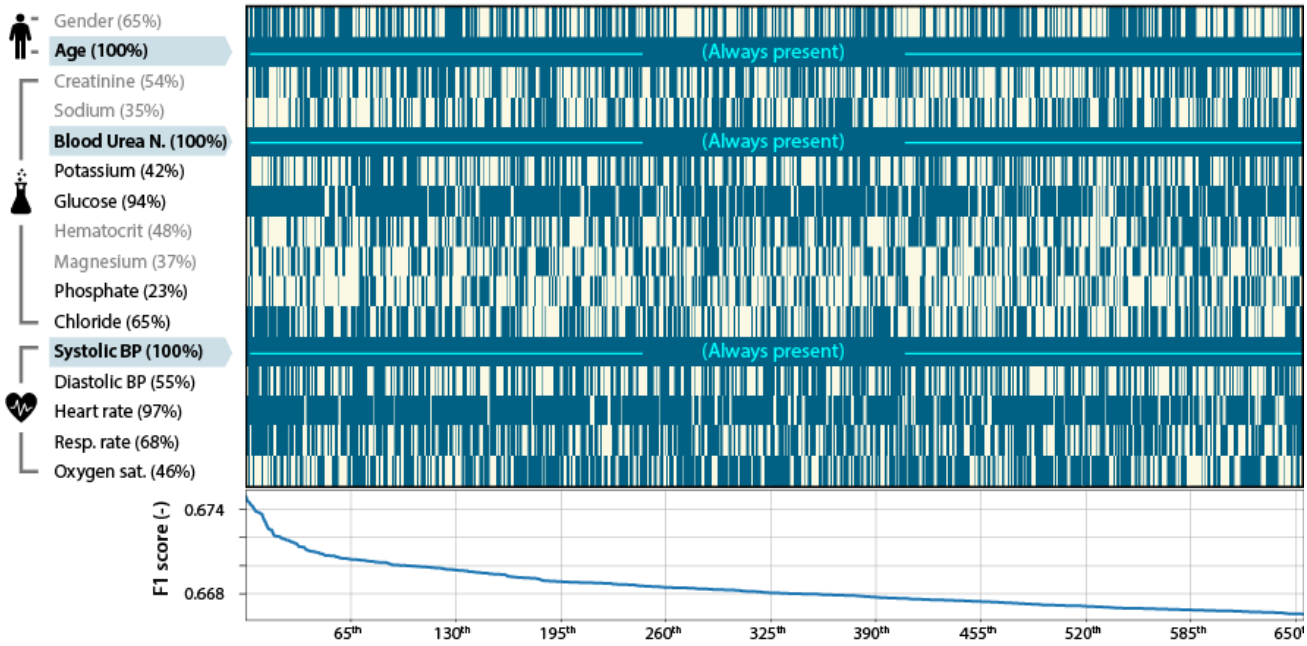

Figure 2. Features in 1% top performing models (N=655). Models are ordered by F1 performance (X-axis); feature presence is highlighted in dark blue; otherwise, the feature was replaced by zero (light yellow). Feature names in gray signalize those not selected for the final model. Feature labels in black refer to "substantial" features, light blue highlights "mandatory" features, and feature labels in grey point to features removed from the final ("full") feature set.

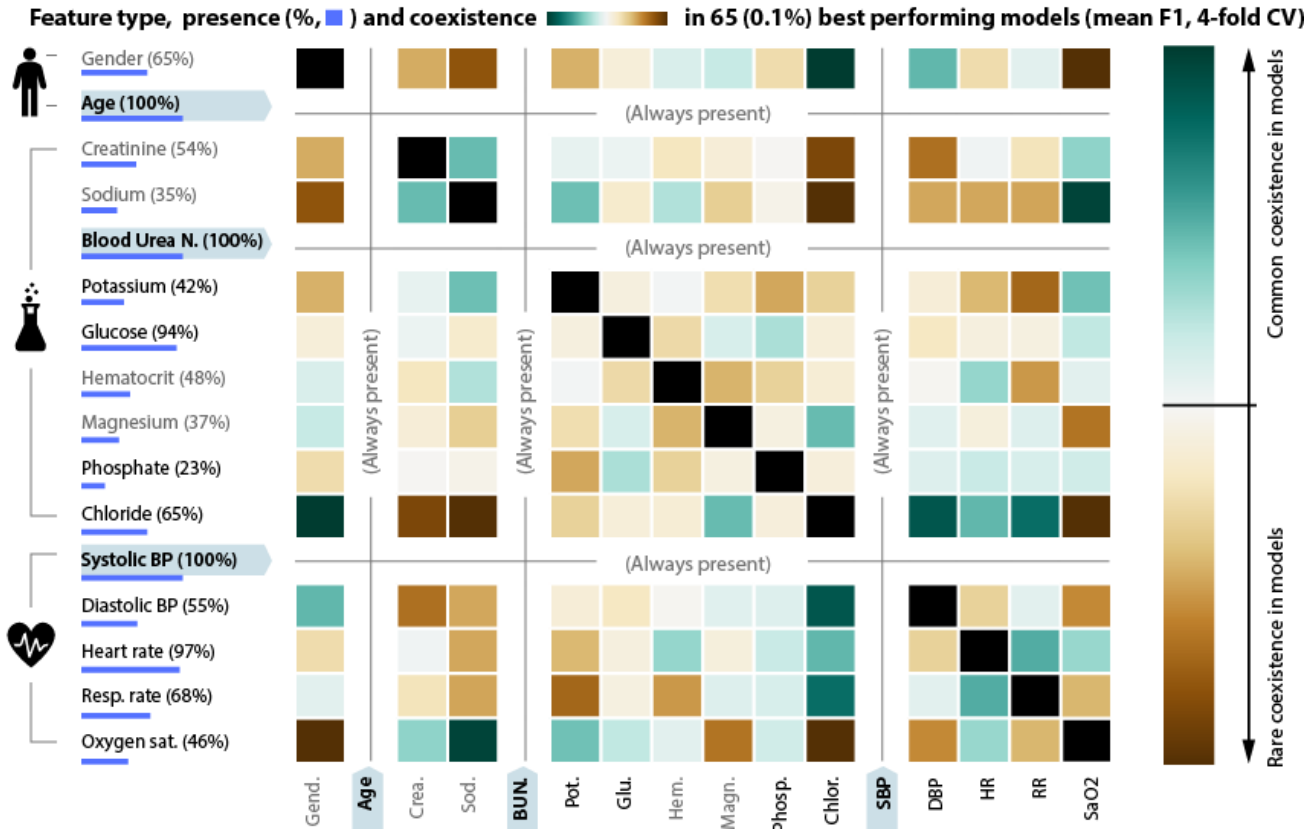

Figure 3. Features presence and coexistence in 0.1% (N=65) of the best performing models. Blue tone signalizes that a specific feature pair form models often while brown tone signalizes the opposite (likely in correlated features). Feature labels in black refer to "substantial" features, light blue highlights "mandatory" features, and feature labels in grey point to features removed from the final ("full") feature set.

3.2. Final model evaluation

Model variants consisting of three mandatory and eight optional features were evaluated using test data (N=401). We received F1 score of 0.681, and 0.629 when all features are present (N_F=11), and all optional features were missing (N_F=3), respectively. Tab.1 shows final model features.

Table 1. Features of the final logistic regression model. The Mandatory features are highlighted.

Feature	Coefficient
Age	0.353
Blood urea nitrogen	0.421
Potassium	-0.115
Glucose	0.167
Phosphate	0.044
Chloride	0.228
Systolic blood pressure	-0.418
Diastolic blood pressure	-0.014
Heart rate	0.257
Respiratory rate	0.065
Oxygen saturation	-0.404

3.3. Importance of optional features

We evaluated performance by presence of optional features in models (Fig.4). Chloride or glucose level, for example, is present in better performing combinations.

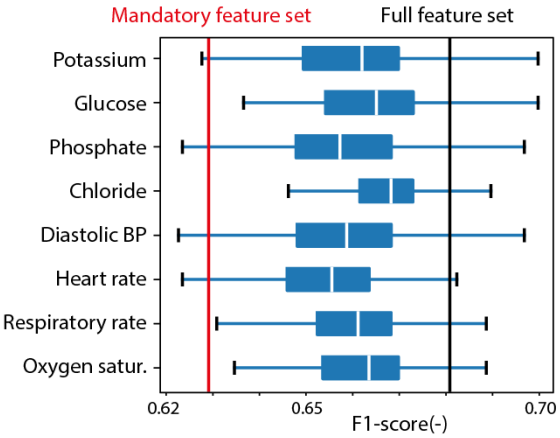

Figure 4. Performance (test set) of models if specific features are present (N_F>=4). The perf. of mandatory feature set (N_F=3) is red; the full set (N_F=11) is black.

3.4. Comparison of using missing values and reduced models

Contrary to our expectations, models trained for specific feature combinations did not show generally better performance than the strategy with missing values. Furthermore, Fig. 5 shows that scores for reduced models are significantly lower ($p<0.0001$) than scores for the missing value strategy.

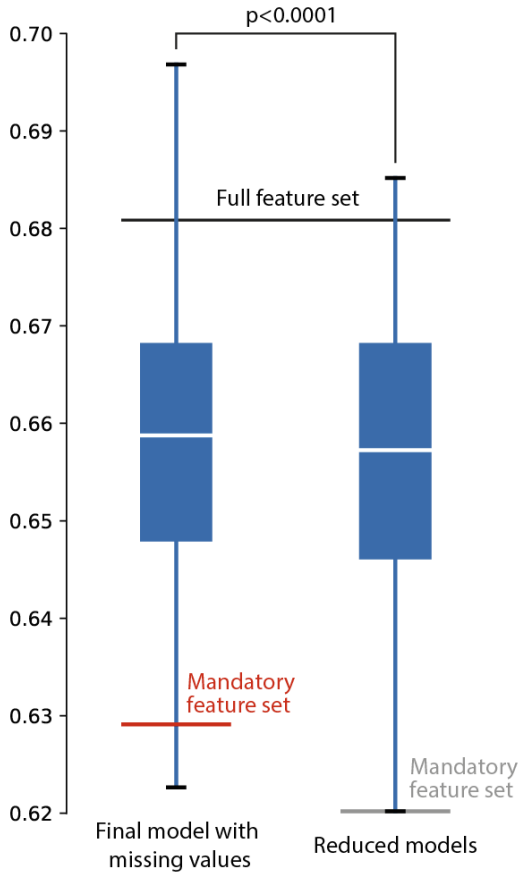

Figure 5. Performance comparison (F1-score) of using one model and missing values (left) and specifically trained reduced models (right). It also shows performance for the full feature set scenario (black, N_F=11). Performance for the mandatory feature set (N_F=3) in model with missing values is red; performance for mandatory feature set for reduced model is gray.

3.5. Performance in follow-up

Since data in MIMIC-III are accompanied with follow-up information, we evaluated dichotomization ability (Fig.6) of the proposed models in the full and minimal configuration. These two feature sets lead to dichotomizations with hazard ratios of 2.08 (95%CI 1.66-2.61) and 1.73 (95%CI 1.39-2.17), respectively.

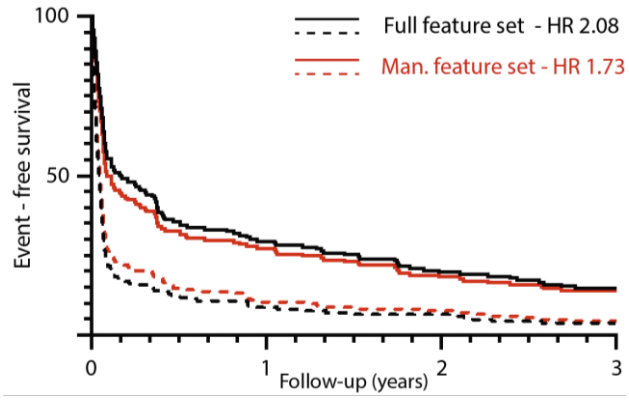

Figure 6. Data dichotomization by the final model when all features are present (black, N_F=11) and when only mandatory features are present (red, N_F=3).

4 Conclusion

We elaborated on optimal solution for web service estimating readmission in case the user does not include complete feature set in a user interface. We found that in case of logistic regression model and a given feature set, more efficient solution is to use a single logistic regression model and treat other features as missing instead of training specifically reduced models. For our specific case, we identified three features as mandatory and evaluated how missing feature affect hazard ratio in a three-years follow up.

Acknowledgments

The research was supported by the Czech Technological Agency grant number FW06010766 and the project RVO:68081731 by the Czech Academy of Sciences.

References

[1] A. E. W. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Sci. Data 2016 31*, vol. 3, no. 1, pp. 1–9, May 2016.

[2] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals.," *Circulation*, vol. 101, no. 23, pp. E215–E220, 2000.

[3] G. Van Rossum *et al.*, "Python 3 Reference Manual," *Nature*, vol. 585, no. 7825, pp. 357–362, 2009.

[4] W. McKinney, "Data Structures for Statistical Computing in Python," *Proc. 9th Python Sci. Conf.*, pp. 56–61, 2010.

[5] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, 2011.

[6] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

Address: F. Plesinger, ISI of the CAS, Kralovopolska 147, Brno 612 00, Czech Republic. E-mail: fplesinger@isibrno.cz