Artif Intell Rev
https://doi.org/10.1007/s10462-018-9625-3

# A comparative study of pattern recognition algorithms for predicting the inpatient mortality risk using routine laboratory measurements

Narayan Schütz[1] · Alexander B. Leichtle[1] · Kaspar Riesen[2]

**Abstract** Laboratory tests are a common and relatively cheap way to assess the general health status of patients. Various publications showed the potential of laboratory measurements for predicting inpatient mortality using statistical methodologies. However, these efforts are basically limited to the use of logistic regression models. In the present paper we use anonymized data from about 40,000 inpatient admissions to the Inselspital in Bern (Switzerland) to evaluate the potential of powerful pattern recognition algorithms employed for this particular risk prediction. In addition to the age and sex of the inpatients, a set of 33 laboratory measurements, frequently available at the Inselspital, are used as basic variables. In a large empirical evaluation we demonstrate that recent pattern recognition algorithms (such as random forests, gradient boosted trees or neural networks) outperform the more traditional approaches based on logistic regression. Moreover, we show how the predictions of the pattern recognition algorithms, which cannot be directly interpreted in general, can be calibrated to output a meaningful probabilistic risk score.

**Keywords** Comparative study on pattern recognition algorithms · Laboratory measurements · Predicting inpatient mortality risk

## 1 Introduction and related work

In the last decades healthcare costs have been constantly increased in many countries. In particular, in developed countries various chronic diseases and the aging population can

✉ Kaspar Riesen
kaspar.riesen@fhnw.com

Alexander B. Leichtle
Alexander.Leichtle@insel.ch

[1] University Institute of Clinical Chemistry, Bern University Hospital (Inselspital), University of Bern, Bern, Switzerland

[2] Institute for Information Systems, University of Applied Sciences FHNW, Olten, Switzerland

Published online: 10 March 2018

be seen as the origin of exploding costs in the health sector. In Switzerland, for instance, the healthcare costs are known to contribute almost 12% of the *gross domestic product* (Köthenbürger and Sandqvist 2016).

While some expenses are actually necessary, with regard to an advanced healthcare system, there is great potential for savings in different ways. The Minnesota Department of Health, for instance, has shown that during the year 2012 about 1.2 million patient visits and 1.3 billion in dollars were caused by preventable emergency department visits (Minnesota Department of Health 2015).

With the rise of digitalization in the field of healthcare, huge amounts of data have become readily available. This data stems from numerous sources, such as *electronic health records* (EHR), *personalized genome scans*, or even live-data obtained by *wearables*. This development could help to reduce costs by leveraging modern methods from big data analytics to healthcare. Furthermore, it could also increase the accessibility to healthcare in less developed countries and ultimately improve the overall quality of care (Raghupathi and Raghupathi 2014).

A key component to leverage the accessibility to large amounts of complex data are pattern recognition algorithms (Duda et al. 2001; Bishop 2008). Pattern recognition algorithms basically allow machines to learn models from large amounts of data. In the scenario of *supervised learning* the underlying training data is labeled with a specific class value (often assessed by human experts). Eventually, the learned model can be used to predict the label of unseen data (known as *classification*).

The main contribution of the present paper is the thorough evaluation of powerful methods from pattern recognition in order to predict the risk of mortality of inpatients. The ultimate goal of this prediction is to automatically identify patients at risk and in turn take proactive measures (for instance, an intensified monitoring). Actually, pattern recognition methods gained much attention in the field of medical predictions. In Churpek et al. (2016), for instance, demographic variables, laboratory values, and vital signs are employed in a discrete-time survival analysis framework to predict the combined outcome of cardiac arrest, intensive care unit transfer, or death. In Goldstein et al. (2017) the use of elaborated methods for risk prediction models in the context of acute myocardial infarction diagnosis is reviewed.

In our scenario the inpatients are formally described by various laboratory measurements as well as their age and sex. Laboratory tests are usually samples of blood (or urine), which a physician sends to a laboratory to assess their values. Laboratory tests are often ordered as part of routine health-checks. Yet, they may also be used to check for certain conditions when a patient exhibits specific symptoms (for instance, factors indicating heart failure). Compared to other medical tests, like *magnetic resonance imaging* (MRI) or *computerized tomography* (CT), laboratory tests are relatively cheap and fast and thus more commonly available.

There are several publications available that confirm the benefit of using statistical models applied on laboratory tests. Most of these approaches are based on *logistic regression*. Pine et al., for instance, developed an early model based on logistic regression which is restricted to analyze inpatients with certain diagnoses (Pine et al. 1997). In Froom and Shimoni (2005) a similar model is built that takes into account the age of the inpatient as well as twelve chemistry and complete blood count variables. Recently, in Tabak et al. (2014) a scoring model to quickly assess a patients risk is proposed [termed *Acute Laboratory Risk of Mortality Score* (*ALaRMS*)]. ALaRMS employs a set of available laboratory tests to assess the overall health status of a patient with a risk score. Although the proposed model is a scoring system, its parameters originate from a logistic regression model.

In the present paper we aim at evaluating the performance of very recent pattern recognition algorithms in conjunction with standard laboratory tests, age and sex as input variables. To this end, we train and evaluate four pattern recognition algorithms on a recent real world data set. In particular, we make use of a *Support Vector Machine* (Chang and Lin 2001), a *Multilayer Perceptron* (Al-Rfou et al. 2016), as well as two classifier ensembles based on *Decision Tree* algorithms (Breiman 2001; Chen and Guestrin 2016) and compare these four methods with a standard logistic regression model applied on the same data.

The remainder of this paper is organized as follows. Next, in Sect. 2 the data set is thoroughly introduced. Moreover, as preliminary work has shown the high variance in the availability of laboratory tests between patients (Nakas et al. 2016), we also review a strategy for handling missing values in this section. Next, in Sect. 3 the different classification models actually employed in this study are briefly reviewed. The experimental setup as well as the test results are presented and discussed in Sect. 4. Finally, Sect. 5 concludes the paper and provides suggestions for possible future research activities.

## 2 Data set

### 2.1 Variables

The *Inselspital* is the university hospital in the city of Bern (Switzerland).[1] The raw data employed in the present study includes all hospital admissions to the Inselspital during the year 2015.[2] The initial data set consists of the age and sex as well as several laboratory measurements of the patients.

First, we select patients where the personal and administrative as well as the laboratory data are available resulting in 39,887 inpatients in total. For patients with multiple hospital entries during the considered time span, only the first entry is considered. Multiple laboratory measurements are processed in a similar way. That is, only the laboratory test closest to the initial entry is considered. Additionally, only the most frequent laboratory measurements with less than 80% missing cases are included in our final data set (reducing the available laboratory data to 33 measurements[3]). Finally, non-numerical laboratory measurements as for instance "$< x$", "$> x$", "positive", or "negative" are transformed to the next lower or greater discrete feature value, to the maximum value or to zero, respectively.

In summary, our final data set consists of $N = 39{,}887$ patients (45.37% female and 54.63% male) described with $n = 35$ features (age, sex and 33 laboratory measurements). Additionally, the discharge status (*dead* or *alive*), which is actually the variable to be predicted in the present study, is also known for every inpatient in the data set. The overall mortality rate in our selection amounts to 2.26%. That is, we observe 905 death inpatients and 38,982 living inpatients (termed *negative* and *positive*) from now on.

Table 1 summarizes the 33 laboratory measurements employed in the present study including the mean value ($\mu$), the standard deviation ($\sigma$) as well as the minimum and maximum value for each feature (the statistics of the variable *Age* are also given). Details on the medical descriptions of the laboratory measurements can be obtained from AACC (2001), Smith et al. (2005).

[1] The Inselspital Bern serves as University Hospital with 50,000 inpatients with quaternary care per year, providing about 900 beds and 78 departments.

[2] Due to ethical considerations and regulatory requirements all data has been completely anonymized beforehand.

[3] The threshold of 80% is somehow arbitrary and other thresholds could have been used, of course.

**Table 1** Laboratory measurements employed in the present study including the mean value ($\mu$), the standard deviation ($\sigma$), the minimum and maximum value as well as the relative amount of patients where this particular measurements is missing (Miss)

| Variable | $\mu$ | $\sigma$ | Min | Max | Miss (%) |
|---|---|---|---|---|---|
| Age (years) | 52.3 | 25.4 | 0 | 101 | 0.0 |
| Lipaemic | 16.3 | 24.6 | 0 | 2300 | 12.3 |
| Potassium (plasma, μmol/L) | 4.1 | 0.6 | 1.6 | 41.6 | 12.5 |
| MCV (fL) | 90.3 | 6.3 | 45 | 140 | 15.5 |
| MCHC (g/L) | 335.1 | 19.9 | 252 | 3365 | 15.5 |
| Erythrocytes (T/L) | 4.2 | 0.7 | 0.7 | 8.5 | 15.5 |
| RDW (%) | 14.6 | 2.1 | 11.1 | 57.7 | 15.5 |
| Thrombocytes (G/L) | 239.5 | 101.9 | 1 | 3420 | 15.5 |
| Leukocytes (G/L) | 9.3 | 7.0 | 0.0 | 597.4 | 15.5 |
| MPV (fL) | 8.575 | 1.3 | 5.4 | 30.7 | 15.5 |
| Hemoglobin (g/L) | 127.1 | 49.9 | 22 | 8400 | 15.5 |
| Sodium (μmol/L) | 138.2 | 3.9 | 76 | 194 | 17.9 |
| Creatinine (plasma, μmol/L) | 92.4 | 80.4 | 5 | 1737 | 22.4 |
| Glucose (plasma, μmol/L) | 6.7 | 2.9 | 0.4 | 63.56 | 23.2 |
| C-reactive protein (plasma, mg/L) | 47.4 | 68.8 | 3 | 637 | 51.4 |
| Urea (plasma, μmol/L) | 7.9 | 6.2 | 0.6 | 113.1 | 54.0 |
| Calcium total (plasma, μmol/L) | 2.2 | 0.2 | 0.6 | 6.4 | 62.3 |
| Quick venous (plasma, %) | 75.6 | 23.6 | 8 | 100 | 63.1 |
| pH (urine) | 5.9 | 0.8 | 5 | 9 | 65.1 |
| pCO2 (blood, mmHg) | 40.7 | 9.9 | 8 | 176 | 65.6 |
| pO2 (blood, mmHg) | 85.6 | 78.1 | 0 | 498 | 66.0 |
| Monocytes (G/L) | 0.6 | 1.3 | 0 | 97.68 | 66.6 |
| Lymphocytes (G/L) | 1.5 | 6.2 | 0 | 573.5 | 66.6 |
| Neutrophils (G/L) | 7.5 | 5.3 | 0 | 143.27 | 66.6 |
| Lactate (blood, μmol/L) | 2.0 | 2.1 | 0.3 | 29 | 70.1 |
| Bicarbonate (blood, μmol/L) | 23.0 | 4.2 | 1.1 | 57.1 | 70.3 |
| Alkaline phosphatase (plasma, U/L) | 112.0 | 142.8 | 12 | 5342 | 70.4 |
| Phosphate (plasma, μmol/L) | 1.1 | 0.4 | 0.1 | 5.67 | 72.0 |
| Calcium ionized (whole blood, μmol/L) | 1.2 | 0.1 | 0.23 | 1.99 | 78.1 |
| Magnesium (plasma, μmol/L) | 0.8 | 0.3 | 0.21 | 18.1 | 78.1 |
| aPTT (plasma, s) | 37.2 | 18.5 | 17.4 | 291.8 | 79.0 |
| Base excess (whole blood, μmol/L) | −1.8 | 4.5 | −29.8 | 28.9 | 79.4 |

Due to financial constraints and/or a physicians prior knowledge, not every possible test is carried out for every inpatient. Hence, our basic data set consists of a high amount of inpatients that are not described by means of the full feature set. In the last column of Table 1 we indicate the relative amount of inpatients where this particular measurements is missing (Miss).

We observe that laboratory features are missing in between 12.3 and 79.4% of all inpatients. Regarding the availability of the laboratory data we note quite a large gap between the first

13 features (*Lipaemic* to *Glucose*), which are missing in between 12.3 and 23.2% of all inpatients, and the remaining laboratory data (missing in more than 51.4% up to almost 80% of all inpatients). This observation is due to the available panels of tests that can be ordered as sets by physicians.

## 2.2 Handling missing data

There are some classification algorithms (e.g. decision tree algorithms Breiman et al. 1984) that can be applied to data sets with missing feature values. The way these methods deal with missing values is usually based on probabilistic estimates such as maximum likelihood and/or adding additional categories for missing variables (Schmitt et al. 2015).

Yet, several other statistical classifiers, such as for instance support vector machines (Shawe-Taylor and Cristianini 2004), generally need data sets with complete feature sets. Hence, for these algorithms the objects with missing values have to be processed before the classification can be carried out. Generally one distinguishes three types of missing data, viz. data that is *missing completely at random*, data that is *missing at random*, and data that is *missing not at random* (Donders et al. 2006).[4]

For our data set it is clear that specific tests are dependent on their actual value and thus fall in the third category of missing data. That is, the available distribution might be heavily biased towards a small fraction of patients associated with an elevated probability of having non-standard values and in turn elevated mortality risk.

This can actually be observed in Fig. 1. It is clearly visible that certain variables are associated with significantly higher mortality risks than others.[5] For instance, the *Creatinine* test is not associated with any increased risk. Yet, we observe that for inpatients where an infrequent test (e.g. *Base Excess*) is carried out, the mortality risk is about three times higher than the average. These tests are highly specific and are thus normally not carried out during a routine health-check but rather when a physician has a certain suspicion.

In the most simple scenario one would discard all observations with missing values. Obviously, this is not suitable in our particular use case (with large amounts of missing values). Another idea is to substitute missing values of a feature $f$ by the global mean $\mu_f$ of this particular feature (or by any other statistical measurement). However, this has also major drawbacks such as a significant decrease in the variance of the variable as well as a substantial loss of information. Moreover, in preliminary experimental evaluations we observed that the substitution of missing values by the global mean leads in general to worse results than the more elaborated imputation algorithm actually used.[6]

In the present paper we make use of a classifier to predict a missing variable by means of the available features. Assuming the missing value is not completely independent from all other measurements, this allows a more elaborated guess of the true laboratory value. Formally, we use the *MissForest imputation algorithm* proposed by Stekhoven and Bühlmann (2012).[7]

[4] Missing Completely at Random (MCAR) means that data is missing completely independent of the observed and unobserved values, while Missing at Random (MAR) means that the observed data might explain the missing data. Thus given the observed data, unobserved data are missing independently of their actual value.

[5] The overall mortality risk in our data set is $905/39{,}887 \approx 2.27\%$. The values shown in Fig. 1 indicate the relative increase of the probability of negative discharge status given the specific test has been carried out. In the worst case a test is carried out on 20.6% of the patients. Hence, the relative increase of mortality risks are computed on at least 8200 patients.

[6] Using mean imputation leads to a reduction of the area under ROC-curves of a about 2–3% points when compared to the imputation method employed.

[7] We employ the available *R* implementation provided by the same authors.

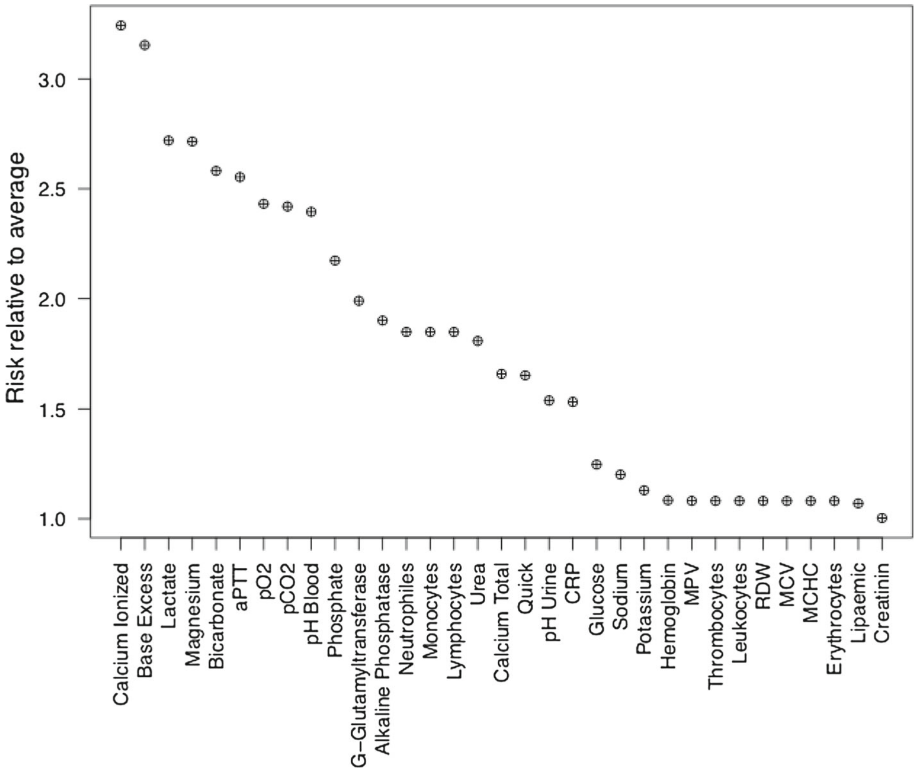

**Fig. 1** The mortality risk is dependent on the laboratory tests carried out

This algorithm uses *random forests* to predict the value of missing variables. One of the main advantages of this particular algorithm is that no parameters have to be defined by the user. Moreover, in Waljee et al. (2013) it is shown that this imputation technique achieves state-of-the-art accuracy.

## 2.3 Normalization

Several classification algorithms rely on distances computed on the feature vectors, and thus, a normalization of the dynamic range of numerical features is necessary. We use a $z$-score based on the mean $\mu_f$ and the standard deviation $\sigma_f$ of feature $f$ computed over all available observations in the data set. Formally, we compute

$$\hat{f} = \frac{f - \mu_f}{\sigma_f} \tag{1}$$

for all features $f$. Hence, we scale all variables to have zero mean and unit variance. Without this particular normalization variables with bigger magnitude would automatically have a larger impact, leading to potentially undesirable effects.

## 3 Pattern recognition algorithms actually applied

Regarding our basic data set, we note that the objects under consideration (i.e. the inpatients) can be interpreted as points in a $n$-dimensional real space, i.e. $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$. Thus, any algorithm from the rich repository of algorithmic tools for statistical pattern recognition can be employed for our prediction task (Duda et al. 2000; Shawe-Taylor and Cristianini 2004; Bishop 2008). In the present paper we make use of both well-established and very recent pattern classification algorithms. In the following subsections the five algorithms actually used are briefly reviewed.

### 3.1 Logistic regression

*Logistic Regression* (LR) is still widely used for binary classification tasks. In particular, in medical research it is often the method of choice due to its simplicity and easy interpretability. With a regularization term (weighted by a regularization parameter $\lambda$) one can penalize too complex parameters which in turn helps with collinearity and prevents overfitting (Minka 2003; Peng et al. 2002). The main limitation of LR is its linear nature. Thus, it often fails to capture non-linear relationships or needs heavy feature-engineering to make the underlying problem linear.

In the present work we make use of *Pythons Scikit-learn* module (Pedregosa et al. 2011) for LR classification.

### 3.2 Support vector machine

Introduced in 1995 by Cortes and Vapnik (1995) the *Support Vector Machine* (SVM) has become one of the most frequently used classifiers in pattern recognition (see Byun and Lee 2003 for an early survey on pattern recognition applications using SVMs).

In contrast with LR, which tries to calculate the *posterior* probability of the class, SVMs aim at finding a hyperplane in the feature-space, which optimally divides the given classes. Besides its solid theoretical foundation and the convex optimization setting (leading to a guaranteed optimum) the major benefit of SVMs is the extension to non-linear decision boundaries by means of the *kernel trick*. The kernel trick allows to map the inner-product of two vectors into a higher dimensional (or even infinite) Hilbert space, where the data is more likely to be linearly separable (Schölkopf and Smola 2002).

One of the most widely used kernels is the *Radial Basis Function* (RBF), which only depends on one parameter $\sigma$ and often leads to very decent results with non-linearly separable data (Shawe-Taylor and Cristianini 2004). The second important parameter for SVMs is the weighting parameter $C$ that controls the trade off between the requirements of a large margin of the hyperplane and few misclassifications.

For SVM classification we use the *Scikit-learn* module (Pedregosa et al. 2011), which in turn uses the *libsvm* implementation (Chang and Lin 2001).

### 3.3 Multilayer perceptron

The basic idea behind neural networks, also termed *Multilayer Perceptrons* (MLP), is actually quite old (established in the 1940s McCulloch and Pitts 1943). With the introduction of the *Backpropagation algorithm*, efficient training of MLPs has become possible in the mid of the eighties of the last century (Werbos 1990; Bishop 1996). Currently, neural networks undergo a true renaissance (in particular due to the massive increase of computational power and the resulting concept of *deep learning* LeCun et al. 2015).

The basic intuition behind MLPs is to use nodes that represent neurons with activation functions and edges with associated weights (representing the strength of connections between the neurons). Using training data, one then adjusts the weights and usually a bias term to approximate the expected output for given input data (the strength of the weight adaptation is commonly controlled by a learning rate $\eta$). Multiple layers allow the network to learn also abstract feature representations, thus compressing information with regard to the target variable and capturing non-linear effects. Training is often done via stochastic gradient descent using a certain number of random training instances per training step.

Our MLP models are built using the *Theano* and *Lasagne* frameworks (Al-Rfou et al. 2016; Dieleman 2016).

## 3.4 Decision tree

The intuition behind *decision trees* (Breiman et al. 1984) is to build a tree where internal nodes represent decisions based on features, partitioning samples with respect to their individual values. Leaf nodes are then associated with classes (or values). The formal basis for building decision trees can be found in information theory, as one wants to maximize information gain with every decision (this can be achieved by means of the *Shannon Entropy*, for instance).

Finding the best split at a specific node may be computed via a greedy search for the best variable and respective cut-off point across that variable's values (or categories). From a theoretic point of view, one can repeat this until all leaf nodes are pure in a sense that they only contain instances of a single class. However, this usually leads to a massive overfitting to the training samples. In order to prevent this, one can define, for instance, the minimum information gain required to carry out a split, set the maximal depth of the tree or the maximal number of features employed, or apply a pruning of the tree (Mohamed et al. 2012).

### *3.4.1 Random forest*

*Random forests* (RF) are ensembles of several decision trees. The basic idea is to create trees with some randomness such that they do not highly correlate with each other. This may be achieved, for instance, by *bootstrapping*, i.e. using a random subsample of the training set for each tree. Sometimes randomization is extended to subsample a certain number of variables per tree (Breiman 1996, 2001).

For RF based classification we use the *Scikit-learn* module (Pedregosa et al. 2011).

### *3.4.2 Tree boosting*

*Tree Boosting* is part of a broader family of ensemble techniques called *Gradient Boosting* (Friedman 2001; Schapire 2003). The basic idea behind tree boosting is similar to the RF approach. However, instead of training each decision tree randomly, one tries to iteratively add a tree which optimally improves prediction in combination with the already existing ensemble.

*XGBoost* (XGB) is a recent implementation of a tree boosting algorithm which is able to handle missing values out of the box. XGB needs a great variety of parameters. For instance, the learning rate $\eta$ broadly describes how big the step from one tree to the next is in terms of minimizing the objective function. Moreover, we also have more classical tree parameters such as the percentage of random variables the tree should use or the maximum allowed depth of a tree in the ensemble. Another more specific regularization parameter is the minimal sum of instance weight, which stops tree growth if a partition would end up with a sum of instance weights below a specified threshold.

The original XGB implementation is used for our experiments (Chen and Guestrin 2016).

# 4 Experimental evaluation

## 4.1 Experimental setup

Rather than directly predicting binary classes (*Positive* or *Negative*), we use the different classification algorithms to predict the risk of mortality $p_0 \in [0, 1]$. The LR, MLP, and XGB algorithms output the score of a *sigmoid* function. In case of the SVM classifier we make use of the probabilistic SVM described in Wu et al. (2004) to output a probability value for each inpatient. For RF we take the mean of all predicted probabilities from the tree ensemble.

Given the risk value $p_0$ given by a specific classifier, we use a global threshold $\theta \in [0, 1]$ for assignments of inpatients to one of the two classes. Formally, if the score $p_0$ output by a certain classifier is greater than $\theta$, a negative classification is returned, otherwise a positive classification is returned. Obviously, threshold $\theta$ has a crucial impact on our model. The higher the threshold is defined, the lower is the relative amount of false negatives. Yet, with higher threshold values we have to also accept higher amounts of false positives.

For each threshold value $\theta$ we thus compute the false positive rate FPR as well as the the true positive rate TPR.[8]

$$FPR = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad \text{and} \quad TPR = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

As basic evaluation tool we then use *ROC-curves* that plot the TPR as a function of the FPR. In order to condense the ROC-curves into one single metric, we compute the *Area Under the Curve*. The AUC is not only independent from the actual decision threshold, but also invariant to prior class probabilities (which is actually desired as we have highly skewed class probabilities with only 2.26% negatives).

## 4.2 Validation of meta parameters

The full data set is randomly split into 30,000 patients for training as well as a 9,887 patients for testing. In order to learn and optimize the meta parameters for the different classification models we apply stratified five-fold cross-validation with random shuffling, where stratifying refers to even patient-outcome distributions. Since the *MissForest* imputation is computationally intensive, we use mean imputation during the optimization process (i.e. every missing value of feature $f$ is replaced by the global mean $\mu_f$). *MissForest* is then applied for the subsequent test. For XGB the internal imputation technique is applied in both cases.

For LR the regularization parameter $\lambda$ is the sole value to be optimized, while for the SVM the weighting parameter $C$ as well as the RBF parameter $\sigma$ are optimized by means of a grid search.

The actual network architecture of our MLP consists of an input layer with 35 nodes, 3 hidden layers with 35 nodes each and one output unit. The Dropout is set to 30% for each layer, including the input layer. For the nodes in the hidden layers, *rectified linear units* (*ReLu*)

[8] TP, TN, FP, and FN refer to the number of true positives and true negatives as well as the number of false positives and false negatives, respectively.

**Table 2** Validated parameters and optimal values for each algorithm

| Classifier | Parameter | Validated values | Optimal value |
|---|---|---|---|
| LR | $\lambda$ | $\{2^{-10}, 2^{-9} \dots, 2^{19}\}$ | $\lambda = 2^{-7}$ |
| SVM | $C$ | $\{2^{-9}, 2^{-8}, \dots, 2^{10}\}$ | $C = 2^{0}$ |
|  | $\sigma$ | $\{2^{-9}, 2^{-8}, \dots, 2^{10}\}$ | $\sigma = 2^{-5}$ |
| MLP | $\eta$ | $\{2^{-19}, 2^{-18}, \dots, 2^{0}\}$ | $\eta = 2^{-13}$ |
| RF | $d_{\max}$ | $\{3, 4, \dots, 20\} \cup \infty$ | $d_{\max} = \infty$ |
|  | $f_{\max}$ | $\{3, 4, \dots, 20\}$ | $f_{\max} = 3$ |
| XGB | $d_{\max}$ | $\{3, 4, \dots, 10\}$ | $d_{\max} = 9$ |
|  | $c$ | $\{0.5, 0.6, \dots, 0.9\}$ | $c = 0.5$ |
|  | $\eta$ | $\{0.01, 0.02, \dots, , 0.1\}$ | $\eta = 0.08$ |
|  | $w_{\min}$ | $\{0.5, 1.0, 1.5, 2.0\}$ | $w_{\min} = 1.5$ |

were used as activation functions. Weights are initialized using the method proposed in Glorot and Bengio (2010) and the updates are calculated using *Nesterov Momentum* (Sutskever et al. 2013) with learning rate $\eta$. With regard to the output layer, we use a Logistic function to map the hidden layer activation to a probability between 0 and 1. Finally, to quantify the offset between target-labels and predictions we used binary cross-entropy (Bengio 2012).

For the RF classifier the parameters that control the maximum depth of the tree $d_{\max}$ and the maximum number of variables employed $f_{\max}$ are optimized. With respect to computational resources available, the number of trees is fixed to 700 during optimization (for the final prediction task we increase this value to 2000).

For XGB the parameters for the maximal depth $d_{\max}$ and the subsample ratio of columns $c$ are tuned. Additionally the learning rate $\eta$ and minimal sum of weights per instance $w_{\min}$ are subject to the grid search optimization. Early stopping is set to 20 rounds. During the parameter optimization the maximum number of trees is set to 100 while in the final prediction task the hard-limit is set to 2000.

In Table 2 the validated parameters as well as the optimal values per parameter are shown for every algorithm.

### 4.3 Test results and discussion

The different classification algorithms are applied with optimized parameters on the independent test set that consists of about 10,000 patients. In Fig. 2 the ROC-curves of three algorithms are shown (for the sake of readability we omit two curves here), while Fig. 3 displays the AUC scores of all algorithms.

The boosted tree ensemble XGB achieves the highest AUC score of 94.19%. The Random Forest classifier and the Multilayer Perceptron show similar performances resulting in AUC of 92.14 and 92.35%, respectively. The worst performing classifiers are the Logistic Regression and Support Vector Machine with AUC scores of 90.51 and 90.59%, respectively.

Regarding the relatively high correlation between the laboratory tests carried out and the mortality risk (as observed in Fig. 1), the question arises whether the pure presence of a certain laboratory measurement is sufficient for a reliable prediction. In order to answer this question, we evaluate the best performing classifier, viz. XGB, on binarized laboratory measurements. Formally, if a laboratory measurement is available for a certain inpatient, we

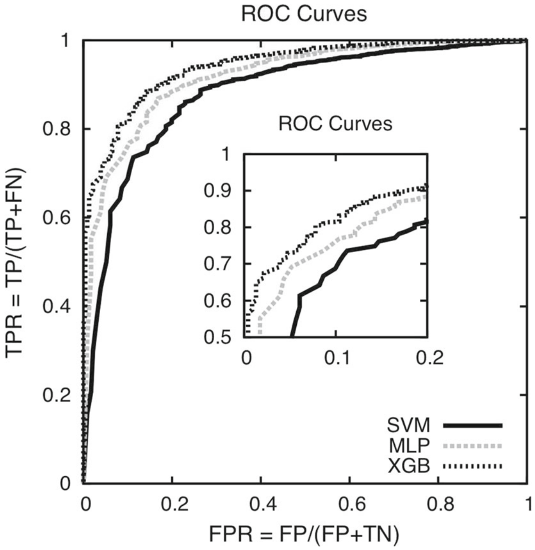

**Fig. 2** The ROC-curves of the different classification algorithms

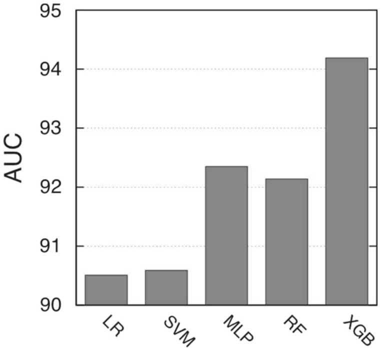

**Fig. 3** The area under ROC-curves (AUC) of the different classification algorithms

replace the value by 1. Likewise, absent measurements are encoded by 0. On this particular data set, XGB achieves a relatively low AUC of 89.10%. Hence, we conclude that the actual values of the laboratory measurement are crucial to make an accurate prediction (or vice versa, the mere presence or absence of laboratory tests is not sufficient, in general).

We additionally applied a feature selection algorithm in conjunction with the best individual classifier XGB. We employ a recursive feature elimination that starts with all features and iteratively removes the least important one. In case of XGB the feature importance is measured by means of the frequency of splits a variable has been used in. This actually resulted in a subset of only six laboratory variables[9] plus age and sex with a resulting AUC of 90.07%.

The AUC metric indicates the performance of the classification algorithms regardless the threshold actually used. Yet, in order to further compare the different classification algorithms, we compare the systems with a fixed threshold value. For each algorithm we choose the lowest threshold that achieves a false positive rate of 10%. That is, such a system classifies

[9] These six features are *Lactate*, *Quick venous*, *C-reactive Protein*, *Creatinin*, *Leukocytes*, and *Glucose*. Note that three out of six selected features have low risk factors. Thus, a feature with high risk factor does not necessarily mean that its predictive power for mortality is also high.

**Table 3** The classification accuracy (Acc), the $F_1$-score, the precision (P), and the recall (R) of the five algorithms when the threshold value is fixed such that the false positive rate is 10%

| Classifier | Acc | $F_1$-score | P | R |
|---|---|---|---|---|
| Logistic regression | 75.50 | 85.70 | 99.68 | 75.16 |
| Support vector machine | 61.83 | 81.39 | 99.76 | 61.06 |
| Mulitlayer perceptron | 75.48 | 88.40 | 99.71 | 75.11 |
| Random forest | 77.28 | 86.70 | 99.69 | 76.96 |
| Tree boosting | 81.67 | 89.67 | 99.71 | 81.47 |

90% of the negative inpatients correctly as negative, while 10% of the negative samples are wrongly classified as positive.[10] For these particular thresholds we compare the algorithms with respect to their classification accuracy (ACC), the $F_1$ score, the precision (P), as well as the recall (R).

In Table 3 the corresponding results are shown. All algorithms achieve similar (and very high) precision scores of about 99.7%. That is, only 3 out of 1000 positive predictions are actually wrong. These high precision scores are due to the highly skewed data reinforced with the low number of false positives evoked by the fixed threshold. Regarding the other three metrics, however, we observe significant differences among the algorithms. In particular, we observe that XGB outperforms all other methods with respect to all three metrics. For instance, regarding the recall XGB outperforms the other classifiers by five or more percentage points.

### *4.3.1 Calibration of the probabilities*

The risk outputs by the classification algorithms are actually not corresponding to the real probability of the class membership. Therefore, it is necessary to adequately calibrate the probability outputs. A possible approach is to use a logistic regression layer behind the classifier, which calculates the *posterior* probability of the class (*dead* or *alive* in our case) given the data.

This method is formally introduced in Platt (1999) in the context of SVMs. Yet, it can be used with any other classifier as well. We actually use this method in conjunction with the tree boosting method XGB (see Fig. 4). Obviously, the probability estimates of XGB are initially too high. Yet, we observe that the actual mortality rate fits quite well with the calibrated model.

## 5 Conclusions and future work

In an effort to predict inpatient mortality risk from laboratory tests plus age and sex, we attempt to compare different pattern recognition algorithms with each other. We employ a fully anonymized electronic health records data of more than 39,000 patient admissions. While similar approaches in literature are mostly limited to logistic regression based models, we evaluate the predictive performance of Support Vector Machines, Random Forests, Gradient Boosted Trees, and Multilayer Perceptrons. We use the area under ROC-curves as basic performance evaluation metric. The best performance is achieved using a gradient boosting tree algorithm in the form of XGBoost, which resulted in an AUC score of about 94%. Last but

[10] Note that these thresholds are actually determined on the test data and thus the results have to be seen as upper bounds.

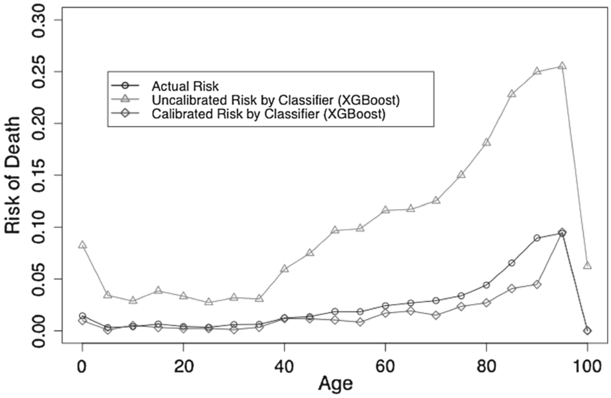

**Fig. 4** Calibrating the output risk to the actual risk of mortality

not least, we calibrate the probability outputs to the real probability of the class membership and observe that the actual mortality rate fits quite well with the calibrated model.

In summary, we show that despite the high amount of missing values in laboratory tests, the use of appropriate imputation and pattern recognition algorithms allows for well calibrated models with competitive predictive performance and the potential to be applied in a real-time decision support system.

We see several rewarding avenues to be pursued in future work. First, we aim at evaluating more feature selection algorithms in conjunction with the classification models in order to further improve the classification performance (or reduce the number of necessary laboratory tests). Additionally, our evaluation is based on laboratory measurements with less than 80% missing cases and we will evaluate other threshold scenarios in our future work. Moreover, one could try to predict other outcome variables such as the length of stay, the requirement of intensive care, the destination after hospital stay, and others. Last but not least, one could use multiple entries for one patient and investigate the changes among those entries and extract features that make use of all the entries instead of using the initial one.

**Acknowledgements** We would like to thank Christos T. Nakas and Lars Clarfeld for their valuable help on the experimental evaluation.

## References

AACC (2001). https://labtestsonline.org/site/

Al-Rfou R, Alain G, Almahairi A, Angermüller C, Bahdanau D, Ballas N, Bastien F, Bayer J, Belikov A, Belopolsky A, Bengio Y, Bergeron A, Bergstra J, Bisson V, Snyder JB, Bouchard N, Boulanger-Lewandowski N, Bouthillier X, de Brébisson A, Breuleux O, Luc Carrier P, Cho K, Chorowski J, Christiano PF, Cooijmans T, Côté M-A, Côté M, Courville AC, Dauphin YN, Delalleau O, Demouth J, Desjardins G, Dieleman S, Dinh L, Ducoffe M, Dumoulin V, Kahou SE, Erhan D, Fan Z, Firat O, Germain M, Glorot X, Goodfellow IJ, Graham M, Gülçehre Ç, Hamel P, Harlouchet I, Heng J-P, Hidasi B, Honari S, Jain A, Jean S, Jia K, Korobov M, Kulkarni V, Lamb A, Lamblin P, Larsen E, Laurent C, Lee S, Lefrançois S, Lemieux S, Leonard N, Lin Z, Livezey JA, Lorenz C, Lowin J, Ma Q, Manzagol P-A, Mastropietro O, McGibbon R, Memisevic R, van Merriënboer B, Michalski V, Mirza M, Orlandi A, Joseph Pal C, Pascanu R, Pezeshki M, Raffel C, Renshaw D, Rocklin M, Romero A, Roth M, Sadowski P, Salvatier J, Savard F, Schlüter J, Schulman J, Schwartz G, Vlad Serban I, Serdyuk D, Shabanian S, Simon É, Spieckermann S, Ramana Subramanyam S, Sygnowski J, Tanguay J, van Tulder G, Turian JP, Urban S, Vincent P, Visin F, de Vries H, Warde-Farley D, Webb DJ, Willson M, Xu K, Xue L, Yao L, Zhang S, Zhang Y (2016) Theano: a python framework for fast computation of mathematical expressions. CoRR. http://arxiv.org/abs/1605.02688

Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. CoRR. arXiv:1206.5533

Bishop C (1996) Neural networks for pattern recognition. Oxford University Press, Oxford

Bishop C (2008) Pattern recognition and machine learning. Springer, Berlin

Breiman L (1996) Bagging predictors. Mach Learn 24:123–140

Breiman L (2001) Random forests. Mach Learn 45:5–32

Breiman L, Friedman J, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Medina County

Byun H, Lee S (2003) A survey on pattern recognition applications of support vector machines. Int J Patt Recognit Artif Intell 17(3):459–486

Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines. Software available at www.csie.ntu.edu.tw/cjlin/libsvm

Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. CoRR. http://arxiv.org/abs/1603.02754

Churpek MM, Yuen TC, Winslow C, Meltzer DO, Kattan MW, Edelson DP (2016) Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. Crit Care Med 44(2):368–374

Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20:273–297

Dieleman S (2016). http://lasagne.readthedocs.io/en/latest/index.html

Donders AR, van der Heijden GJ, Moons KG, Stijnen T (2006) Review: a gentle introduction to imputation of missing values. J Clin Epidemiol 59(10):1087–1091

Duda R, Hart P, Stork D (2000) Pattern classification, 2nd edn. Wiley, New York

Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York

Friedman JH (2001) Greedy function approximation: a gradient booosting machine. Ann Stat 5:1189–1232

Froom P, Shimoni Z (2005) Prediction of hospital mortality rates by admission laboratory tests. Clin Chem 52(2):325–328

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the international conference on artificial intelligence and statistics

Goldstein BA, Navar AM, Carter RE (2017) Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. Eur Heart J 38(23):1805–1814. https://doi.org/10.1093/eurheartj/ehw302

Köthenbürger M, Sandqvist P (2016) KOF health expenditure forecast, spring: moderate trend followed by growing dynamics. https://doi.org/10.3929/ethz-a-010803967

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:463–444

McCulloch W, Pitts W (1943) A logical calculus of ideas immanen in nervous activity. Bull Math Biophys 5(4):115–133

Minka T (2003) A comparison of numerical optimizers for logistic regression. https://www.microsoft.com/en-us/research/publication/comparison-numerical-optimizers-logistic-regression/

Minnesota Department of Health (2015). http://www.health.state.mn.us/news/pressrel/2015/hcevents.html

Mohamed W, Haizan WN, Salleh M, Najib M, Halim OA (2012) A comparative study of reduced error pruning method in decision tree algorithms. In: Proceedings of the IEEE international conference on control system, computing and engineering

Nakas C, Schütz N, Werners M, Leichtle A (2016) Accuracy and calibration of computational approaches for inpatient mortality predictive modeling. PloS ONE 11(17):1–11

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Machine learning in python. J Mach Learn Res 12:2825–2830

Peng J, Lee K, Ingersoll G (2002) An introduction to logistic regression analysis and reporting. J Educ Res 96(1):3–14

Pine M, Norusis M, Jones B, Rosenthal GE (1997) Predictions of hospital mortality rates: a comparison of data sources. Ann Intern Med 126(5):347–354

Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Proc Adv Large Margin Classif 10:61–74

Raghupathi W, Raghupathi V (2014) Big data analytics in healthcare: promise and potential. Health information science and systems. Health Inf Sci Syst 2(1):3

Schapire RE (2003) The boosting approach to machine learning: an overview. In: Denison DD, Hansen MH, Holmes C, Mallick B, Yu B (eds) Nonlinear estimation and classification (Lecture Notes in Statistics), vol 171. Springer

Schmitt P, Mandel J, Guedj M (2015) A comparison of six methods for missing data imputation. J Biom Biostat 6(224):1

Schölkopf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

Smith et al. (2005). http://www.webmd.com/

Stekhoven D, Bühlmann P (2012) Missforest—non-parametric missing value imputation for mixed-type data. Bioinformatics 28(1):112–118

Sutskever I, Martens J, Dahl GE, Hinton GE (2013) On the importance of initialization and momentum in deep learning. ICML 3(28):1139–1147

Tabak Y, Sun X, Nunez C, Johannes R (2014) Using electronic health record data to develop inpatient mortality predictive model: acute laboratory risk of mortality score (alarms). J Am Med Inf Assoc 21:455–463

Waljee AK, Mukherjee AG, Singal A, Zhang Y, Warren J, Balis U, Marrero J, Zhu J, Higgins P (2013) Comparison of imputation methods for missing laboratory data in medicine. BMJ Open 3(8):e002847

Werbos PJ (1990) Backpropagation through time: what it does and how to do it. Proc IEEE 78(10):1550–1560

Wu CFJ, Lin CJ, Weng RC (2004) Probability estimates for multi-class classification by pairwise coupling. J Mach Learn Res 5:975–1005