mathematics

MDPI

Article

Machine Learning at the Service of Survival Analysis: Predictions Using Time-to-Event Decomposition and Classification Applied to a Decrease of Blood Antibodies against COVID-19

Lubomír Štěpánek [1,*], Filip Habarta [1], Ivana Malá [1], Ladislav Štěpánek [2], Marie Nakládalová [2], Alena Boriková [2] and Luboš Marek [1]

[1] Department of Statistics and Probability, Faculty of Informatics and Statistics, Prague University of Economics and Business, W. Churchill's Square 1938/4, 130 67 Prague, Czech Republic
[2] Department of Occupational Medicine, University Hospital Olomouc and Faculty of Medicine and Dentistry, Palacký University Olomouc, I. P. Pavlova 185/6, 779 00 Olomouc, Czech Republic
* Correspondence: lubomir.stepanek@vse.cz

Abstract: The Cox proportional hazard model may predict whether an individual belonging to a given group would likely register an event of interest at a given time. However, the Cox model is limited by relatively strict statistical assumptions. In this study, we propose decomposing the time-to-event variable into "time" and "event" components and using the latter as a target variable for various machine-learning classification algorithms, which are almost assumption-free, unlike the Cox model. While the time component is continuous and is used as one of the covariates, i.e., input variables for various classification algorithms such as logistic regression, naïve Bayes classifiers, decision trees, random forests, and artificial neural networks, the event component is binary and thus may be modeled using these classification algorithms. Moreover, we apply the proposed method to predict a decrease or non-decrease of IgG and IgM blood antibodies against COVID-19 (SARS-CoV-2), respectively, below a laboratory cut-off, for a given individual at a given time point. Using train-test splitting of the COVID-19 dataset ($n = 663$ individuals), models for the mentioned algorithms, including the Cox proportional hazard model, are learned and built on the train subsets while tested on the test ones. To increase robustness of the model performance evaluation, models' predictive accuracies are estimated using 10-fold cross-validation on the split dataset. Even though the time-to-event variable decomposition might ignore the effect of individual data censoring, many algorithms show similar or even higher predictive accuracy compared to the traditional Cox proportional hazard model. In COVID-19 IgG decrease prediction, multivariate logistic regression (of accuracy 0.811), support vector machines (of accuracy 0.845), random forests (of accuracy 0.836), artificial neural networks (of accuracy 0.806) outperform the Cox proportional hazard model (of accuracy 0.796), while in COVID-19 IgM antibody decrease prediction, neither Cox regression nor other algorithms perform well (best accuracy is 0.627 for Cox regression). An accurate prediction of mainly COVID-19 IgG antibody decrease can help the healthcare system manage, with no need for extensive blood testing, to identify individuals, for instance, who could postpone boosting vaccination if new COVID-19 variant incomes or should be flagged as high risk due to low COVID-19 antibodies.

Keywords: time-to-event variable decomposition; time-to-event variable prediction; machine-learning classification algorithms; COVID-19; antibody blood level decrease; multivariate logistic regression; naïve Bayes classifier; support vector machines; decision trees and random forests; artificial neural networks

MSC: 62N02; 62H30

Citation: Štěpánek, L.; Habarta, F.; Malá, I.; Štěpánek, L.; Nakládalová, M.; Boriková, A.; Marek, L. Machine Learning at the Service of Survival Analysis: Predictions Using Time-to-Event Decomposition and Classification Applied to a Decrease of Blood Antibodies against COVID-19 . *Mathematics* **2023**, *11*, 819. https://doi.org/10.3390/math11040819

Academic Editor: Kang Lu

Received: 31 December 2022
Revised: 28 January 2023
Accepted: 30 January 2023
Published: 6 February 2023

1. Introduction

In survival analysis, the time-to-event variable is usually dependent, i.e., a response variable, and describes whether an individual may expect to experience an event of interest and, if so, when they should do it. If an individual does not experience the event of interest for some reason, we call the individual *censored*. The event of interest could be whatever defined and detectable event that is not related to censoring [1].

In statistics and survival analysis, the time-to-event variable is commonly treated as a random variable and modeled using traditional approaches, including estimating, both parametric and non-parametric, of a cumulative distribution function and others, namely survival function and hazard function and its moment characteristics. While the survival function depicts a probability that time to an event of interest is greater than a given time constant, i.e., that an individual would survive a certain time, the hazard function is a rate of probability that the event of interest is registered for in an infinitely short time after a given time period of non-registering the event [2].

The above-mentioned approach enables the application of statistical inference on the time-to-event variable and its derivations for one or more groups of individuals. Besides survival functions, comparison between groups of individuals, considering explanatory covariates, the hazard function might be modeled within a regression framework using a Cox proportional hazard model [3].

The Cox proportional hazard model is routinely used for modeling an association between the covariates as independent variables and the hazard function as a dependent variable. However, the Cox model is relatively strictly limited by statistical assumptions [4]. Supposing that there is a categorical independent variable determining which group an individual belongs to, the limiting fact is that proportions of hazard functions for each pair of groups should be constant across all time points; this is why the Cox model is called "proportional" [5,6]. There are several options for how to overcome the violation of non-proportionality of hazard curves, e.g., some covariates in the Cox model may be stratified for different values or groups of individuals that enable the relaxing of the strict assumption that mutual pair hazard curves' proportions should be equal to one constant [7,8]. Even more, if needed, not only time-invariant covariates but also time-variant explanatory variables could be modified in time-varying models [9], or intervals of time points might be split within time-partitioned models [10]. Various possibilities of the Cox model assumptions' violation treatment within the large family of *Cox non-proportional hazard models* suggest that not one of them is an optimal "remedy" for all new assumptions coming from them. The Cox non-proportional hazard models are highly complex, usually require a sufficient amount of data and advanced erudition and experience in their usage, and could bring new assumptions, sometimes even more complex [11]. Besides the Cox models, parametric survival models use log-normal or Weibull baseline hazard function. Although applied to real-world data, these sometimes work better than the Cox regression [12], some others fail, compared to the Cox model [13]. The parametric survival models might suffer an initial choice of the baseline hazard function. In addition, even these models assume the proportionality of hazard functions among individuals or cohorts across time points [14].

In this study, we address the latter issue and propose an alternative to the Cox proportional hazard model that is almost assumption-free. We introduce a principle of time-to-event variable decomposition into two components; first, a time component that is continuous and is used as an explanatory covariate in a machine-learning classification model, and second, an event component that is binary and could be predicted using various classifiers in the classification models, containing, besides others, the time component as an input covariate. Finally, once a classification model is learned and built, then, for a given combination of values of explanatory covariates, including a time point since the time component is one of the covariates, we may predict whether an event of interest is likely to happen for the given covariates' values combination. The same prediction might be made using the Cox proportional hazard model—the model estimates the posterior distribution

of hazard function, which enables obtaining the point estimate of the probability that an individual with given covariates' values combination would likely experience the event of interest. If the probability is greater than or equal to a defined threshold, we obtain qualitatively the same prediction as using our proposed method. This opens room for comparison of predictive accuracy between our introduced method based on time-to-event variable decomposition and the Cox proportional hazard model. While the Cox proportional hazard model considers the non-experiencing event of interest within a given time scope as the censoring, machine-learning classifiers understand the experiencing and non-experiencing event of interest as two potentially equal states and, thus, might sideline the censoring in this manner. However, if we prefer the prediction paradigm rather than the inference one, censoring as a kind of incomplete information might be sidelined as far as the prediction work with high enough performance.

The first ideas on time-to-event variable decomposition came from the eighties when Blackstone et al. tried to decompose the time-to-event variable into consecutive phases and model a constant hazard function for each phase [15]. The approach was revisited during the last several years by [16,17] since time-varying and time-partitioning models became popular. However, papers dealing with the combination of time-to-event variable decomposition and machine-learning predictions on the components seem to be rare; some initial experiments come from [18], where authors tried to predict survival in patients with stomach cancer.

The machine-learning classification models might differ in their predictive accuracy; however, regardless of the built classifiers, these are usually assumption-free or assumption-almost-free [19]. We consider several classification algorithms in the study—multivariate logistic regression, naïve Bayes classifier, decision trees, random forests, and artificial neural networks.

To obtain an idea of how the proposed methodology works on real data, we apply the time-to-event decomposition and ongoing prediction on a dataset of COVID-19 patients that includes, besides others, a variable depicting whether, and if so, when an individual experienced a decrease of their COVID-19 antibody blood level below a laboratory cut-off. Some of the individuals in the dataset did experience the antibody blood level decrease; the other ones did not. Thus, the variable is appropriate to be treated as a time-to-event one. COVID-19 is an infectious disease caused by the virus SARS-CoV-2 that started with the first clinically manifesting cases at the end of the year 2019 and quickly spread worldwide [20]. Thus, from early 2020 up to the present, there has been more or less a severe long-term pandemic in many regions all around the world [21,22]. Individuals' COVID-19 blood antibodies, particularly IgG antibodies that are more COVID-19-specific, protect them from COVID-19 manifesting disease [23]. Therefore, if we predict their decrease below laboratory cut-off accurately enough, we can, for instance, quickly, and with no need for extensive or expensive blood testing, preselect which individuals should undergo boosting vaccination first if a new COVID-19 variant would income.

Thus, in this article, we address the following *research gap*. First, we investigate how possible it is to predict a decrease of antibodies against COVID-19 in time using various individuals' covariates and a Cox proportional hazard model. Moreover, we introduce a novel method for predicting an event of interest's occurrence in time using time-to-event decomposition, where the event component is classified using machine-learning classification algorithms. In contrast, the time component is used as one of the covariates. Finally, we compare the predictive performance of Cox regression and our proposed method using predictive accuracy and other metrics.

The paper proceeds as follows. Firstly, in Section 2, we introduce and explain ideas of our proposed methodology, mostly the logic and principles of the time-to-event decomposition. Then, we illustrate the principles, assumptions, and limitations of the Cox proportional hazard model. In addition, we describe how the prediction of an event of interest's occurrence could be made using the Cox model. Then, we depict all machine-learning classification algorithms' principles. Next, in Section 3, we show the numerical results we

have obtained so far and, in particular, compare predictions based on the Cox proportional hazard model with other predictions, using classifiers and time-to-event decomposition. Finally, in Section 4, we discuss the results, explain them, and last but not least, we highlight important findings in Section 5.

2. Methodology and Data

A description of the fundamentals of survival variables' characteristics, the Cox proportional hazard model, the proposed methodology, and the dataset we used for algorithms' predictive performance, respectively, follow.

2.1. Fundamentals of Survival Variables' Characteristics

Let us assume that a random variable T is the survival time, i.e., a length of time an individual does not experience an event of interest [24]. Firstly, let us define survival function $S(t)$ as a probability that survival time is greater than some time constant t, so

$$S(t) = P(T > t) = P(T \geq t) - P(T = t) = P(T \geq t) - 0 = P(T \geq t). \tag{1}$$

Let cumulative distribution function $F(t)$ be a probability that survival time T would not be greater than time constant t; then, density function $f(t)$ is a derivative of $F(t)$ with respect to t, so

$$\begin{aligned} f(t) = \frac{\mathrm{d}F(t)}{\mathrm{d}t} = \frac{\mathrm{d}P(T \leq t)}{\mathrm{d}t} = \frac{\mathrm{d}(1 - P(T > t))}{\mathrm{d}t} = \frac{\mathrm{d}1}{\mathrm{d}t} - \frac{\mathrm{d}P(T > t))}{\mathrm{d}t} = \\ = 0 - \frac{\mathrm{d}P(T > t))}{\mathrm{d}t} = -\frac{\mathrm{d}P(T > t)}{\mathrm{d}t} \overset{(1)}{=} -\frac{\mathrm{d}S(t)}{\mathrm{d}t} \end{aligned} \tag{2}$$

and also

$$f(t) = -\frac{\mathrm{d}P(T > t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{P(t + \Delta t > T \geq t)}{\Delta t}. \tag{3}$$

The hazard function, $\lambda(t)$, is an instantaneous probability that an individual experiences the event of interest in an indefinitely short time interval $\langle t, t + \Delta t)$ once they survive through time t up to the beginning of the interval $\langle t, t + \Delta t)$, so

$$\begin{aligned} \lambda(t) = \lim_{\Delta t \to 0} \frac{P(T \in \langle t, t + \Delta t) \mid T \geq t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{P(t + \Delta t > T \geq t \mid T \geq t)}{\Delta t} = \\ = \lim_{\Delta t \to 0} \frac{P(t + \Delta t > T \geq t \ \wedge \ T \geq t)}{\Delta t \cdot P(T \geq t)} = \lim_{\Delta t \to 0} \frac{P(t + \Delta t > T \geq t)}{\Delta t \cdot P(T \geq t)} \overset{(1)}{=} \\ \overset{(1)}{=} \frac{1}{S(t)} \lim_{\Delta t \to 0} \frac{P(t + \Delta t > T \geq t)}{\Delta t} \overset{(3)}{=} \frac{1}{S(t)} f(t) \overset{(2)}{=} -\frac{1}{S(t)} \frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{S'(t)}{S(t)}. \end{aligned} \tag{4}$$

Since hazard function $\lambda(t) = \lim_{\Delta t \to 0} \frac{P(T \in \langle t, t + \Delta t) | T \geq t)}{\Delta t}$ describes an instantaneous probability of an event of interest experiencing just after a specific time t, we could estimate cumulative hazard function $\Lambda(t)$ as an accumulation of the hazard of the event of interest over time until constant t, having estimates $\hat{\lambda}(t_j)$, based e.g., on observed data, for each time point from the beginning to the end of the observed period. That being said,

$$\Lambda(t) \approx \sum_{\forall t_j : t_j \leq t} \hat{\lambda}(t_j),$$

or more precisely as

$$\Lambda(t) = \int_0^t \lambda(\tau) \mathrm{d}\tau, \tag{5}$$

which may help us to derive a relationship between the survival function and hazard function as follows:

$$\Lambda(t) = \int_0^t \lambda(\tau)\mathrm{d}\tau \overset{(4)}{=} \int_0^t -\frac{S'(\tau)}{S(\tau)}\mathrm{d}\tau = [-\log S(\tau)]_0^t,$$

and since $S(0) = P(T > 0) = P(T \geq 0) = 1$, we obtain

$$\begin{aligned}\Lambda(t) &= [-\log S(\tau)]_0^t = -\log S(t) - (-\log S(0)) = \\ &= -\log S(t) + \log 1 = -\log S(t) + 0 = \\ &= -\log S(t). \end{aligned} \tag{6}$$

Finally, by exponentiation of Formula (6), we obtain a direct relationship between the survival function and cumulative hazard function,

$$S(t) = e^{-\Lambda(t)}, \tag{7}$$

which enables us to predict survival probability for a given time point, based on the Cox proportional hazard model.

2.2. Principles, Assumptions and Limitations of Cox Proportional Hazard Model

The Cox proportional hazard model is frequently used for regression modeling of an association between a hazard function for an event of interest as a dependent variable and multiple independent variables, also called covariates.

Sir Cox [3] suggested to model the hazard function $\lambda(t)$ in relation to other $k \in \mathbb{N}$ covariates $\boldsymbol{X}_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,k})^T$ for individual or group i in the following way:

$$\log \lambda(t) = \log \lambda_0(t) + \boldsymbol{\beta}^T \boldsymbol{x}_i, \tag{8}$$

where $\lambda_0(t)$ is the baseline hazard function, i.e., the hazard function when each of the covariates is equal to zero; $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_k)^T$ is a vector of linear coefficients, each matched to an appropriate covariate, and $\boldsymbol{x}_i = (1, x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ is a vector of covariates' values for individual or group i. The linear coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_k)^T$ in the Cox model following Formula (8) can be estimated using the partial maximum likelihood approach [25,26].

By exponentiation Formula (8), we obtain another form of the Cox model,

$$\lambda(t) = e^{\log \lambda_0(t) + \boldsymbol{\beta}^T \boldsymbol{x}_i} = e^{\log \lambda_0(t)} \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_i} = \lambda_0(t) \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}. \tag{9}$$

Considering the Cox model following Formula (9) for two groups with indices r and s,

$$\begin{aligned}\lambda(t \mid \boldsymbol{x}_r) &= \lambda_0(t) \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_r} \\ \lambda(t \mid \boldsymbol{x}_s) &= \lambda_0(t) \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_s},\end{aligned}$$

we could take a proportion of left-hand and right-hand sides of the equations, obtaining

$$\frac{\lambda(t \mid \boldsymbol{x}_r)}{\lambda(t \mid \boldsymbol{x}_s)} = \frac{\lambda_0(t) \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_r}}{\lambda_0(t) \cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_s}} = \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_r}}{e^{\boldsymbol{\beta}^T \boldsymbol{x}_s}}. \tag{10}$$

Assuming the model's coefficient following Formula (8) or Formula (9) are estimated taking into account, for all individuals or groups, $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ for $\forall i \in \{1, 2, \ldots\}$, i.e.,

$$\boldsymbol{\beta}^T \boldsymbol{x}_r = \text{const.} \qquad \text{and} \qquad \boldsymbol{\beta}^T \boldsymbol{x}_s = \text{const.},$$

we may derive—using Formula (10)—that the proportion

$$\frac{\lambda(t \mid \boldsymbol{x}_r)}{\lambda(t \mid \boldsymbol{x}_s)} = \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_r}}{e^{\boldsymbol{\beta}^T \boldsymbol{x}_s}} = \text{const.}$$

of hazard functions for two any groups $r, s \in \{1, 2, \ldots\}$ is supposed to be constant. This is why the Cox model is called the Cox *proportional hazard* model, and the constant proportion of the hazard functions across all time points for any two groups is one of its statistical assumptions. Moreover, if the hazard functions' ratio should be constant, then also survival functions' ratio should be constant, as we may see using Formulas (5) and (7),

$$\frac{\lambda(t \mid \boldsymbol{x}_r)}{\lambda(t \mid \boldsymbol{x}_s)} = \frac{\int \lambda(t \mid \boldsymbol{x}_r) \mathrm{d}t}{\int \lambda(t \mid \boldsymbol{x}_s) \mathrm{d}t} = \frac{e^{-\int \lambda(t|\boldsymbol{x}_r) \mathrm{d}t}}{e^{-\int \lambda(t|\boldsymbol{x}_s) \mathrm{d}t}} = \frac{e^{-\Lambda(t|\boldsymbol{x}_r)}}{e^{-\Lambda(t|\boldsymbol{x}_s)}} = \frac{S(t \mid \boldsymbol{x}_r)}{S(t \mid \boldsymbol{x}_s)} = \text{const.}$$

However, real-world data often violate this assumption in practice, as Figures 1 and 2 illustrate.

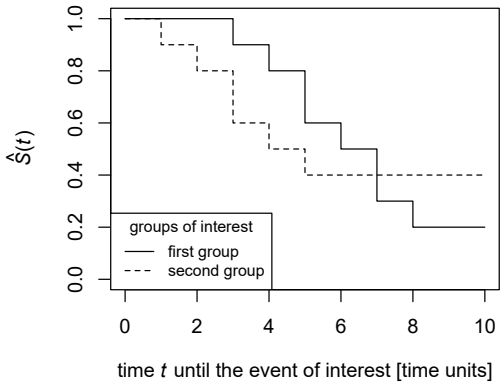

Figure 1. An example of two random survival curves crossing each other.

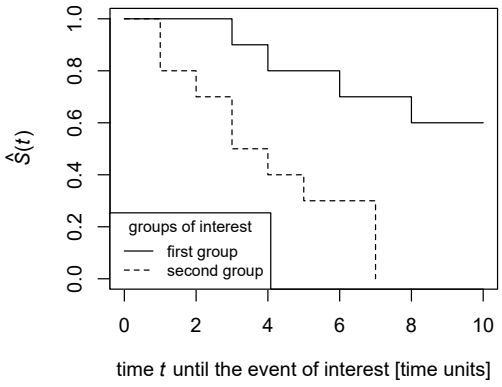

Figure 2. An example of two random survival curves; the first one is leveling off while the second one is dropping to zero.

In Figure 1, the survival functions are estimated in a non-parametric way as polygonal chains cross each other; this means that, while the proportion of the survival function for the solid line to the survival function for the dashed line is greater than 1 till the time point of the crossing, it becomes lower than 1 after the crossing. Thus, the survival functions' proportion could not be constant across all time points.

Similarly, in Figure 2, the proportion of the survival function for the solid line to the survival function for the dashed line is finite until the time point the dashed line drops to zero, then the proportion becomes infinite. Therefore, the survival functions' ratio could not be constant across all time points.

Considering the Cox model from Formula (8) and using Formula (7), we can predict whether an event of interest is likely to be experienced by an individual from a given group at a given time point. Survival function, i.e., a probability that an individual does not experience the event of interest before time t, if ever, is

$$S(t) = P(T > t) = e^{-\Lambda(t)} = e^{-\int_0^t \lambda(t)\mathrm{d}t} = e^{-\int_0^t \lambda_0(t)\cdot e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\mathrm{d}t},$$

and using estimates coming from the Cox model following Formula (8),

$$\hat{S}(t) = \hat{P}(T > t) = e^{-\hat{\Lambda}(t)} = e^{-\int_0^t \hat{\lambda}(t)\mathrm{d}t} = e^{-\int_0^t \hat{\lambda}_0(t)\cdot e^{\hat{\boldsymbol{\beta}}^T \boldsymbol{x}_i}\mathrm{d}t}. \tag{11}$$

2.3. Principles of Proposed Time-to-Event Variable Decomposition and Prediction

The Cox proportional hazard model, following the Formula (8) or Formula (9), estimates the posterior hazard function using covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)^T$, i.e., it models the posterior probability of an event of interest's occurrence in time [27].

Let us mark the time to the event of interest for any individual or group as a random variable T, and the occurrence or non-occurrence of the event of interest for any individual or group as a random variable Y. Obviously, $T_i \geq 0$, and $Y_i \in \{c_1, c_2\}$, or generally $Y_i \in \{c_\ell\}$ considering $\ell \in \{1, 2\}$, for individual or group i, where c_1 stands for the event of interest occurrence before time t and c_2 stands for the event of interest non-occurrence before time t, respectively.

With respect to the task refining above, the Cox proportional hazard model uses covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)^T$ and their values $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})^T$ for estimation, a pair $[T_i, Y_i]$ for individual or group i, i.e., how likely individual or group i would experience $(Y_i = c_1)$, or would not experience $(Y_i = c_2)$ the event of interest until time $T_i = t$.

Inspecting Formula (11), we may see that the time $T = t$ is a parameter of the survival estimate $\hat{S}(t)$. However, the event of interest's occurrence or non-occurrence $Y \in \{c_1, c_2\}$ requires another derivation. Once the Cox model is built, the coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_k)^T$ are estimated using all dataset rows $\boldsymbol{x}_i = (1, x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ for $\forall i \in \{1, 2, \ldots\}$, the survival $\hat{S}(t)$ might be estimated using Formula (11). Since $S(t) = P(T > t)$, we may expect that the event of interest does probably not occur before time t if $\hat{S}(t) = \hat{P}(T > t)$ is high enough. Let us suppose the event of interest is likely not to happen, $Y = c_2$, before time t, if ever does, whenever $\hat{S}(t)$ is greater than or equal to a given threshold $p_{\text{threshold}}$; otherwise, the event of interest does probably occur, $Y = c_1$, before time t. Thus,

$$\hat{Y} = \begin{cases} c_1, \text{ i.e., the event of interest does occur before time } t & \hat{S}(t) < p_{\text{threshold}} \\ c_2, \text{ i.e., the event of interest does not occur before time } t & \hat{S}(t) \geq p_{\text{threshold}} \end{cases} \tag{12}$$

A natural choice for the threshold seems to be $p_{\text{threshold}} = \frac{1}{2}$, but may be grid-searched and adjusted, e.g., to maximize the Cox model's predictive accuracy.

Schematically, the logic of the Cox model is in Figure 3. Covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)^T$ are on input and make the Cox model's estimation possible. Once the Cox model is built, we could predict the event of interest occurrence $Y \in \{c_1, c_2\}$ in time $T = t$ using Formulas (11) and (12). Thus, on the input of the Cox model, there are k covariates $\{X_1, X_2, \ldots, X_k\}$, and on output, there is the time-to-event pair variable $[T, Y]$.

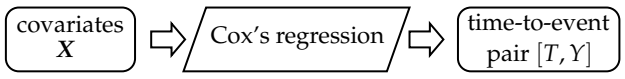

Figure 3. A logic of prediction based on the Cox proportional hazard model. Covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)^T$ on input are used for the Cox model building, which enables us to estimate output, i.e., how likely the event of interest's occurrence Y is in time T (and for specific values of other covariates $\boldsymbol{X}$).

We refine the task of time-to-event prediction the following way. Firstly, we decompose the time-to-event variable into time component T and event component Y. The time component T is continuous and is used as another covariate on input for a machine-learning classification model or, shortly, a classifier. The machine-learning classification model predicts the event component Y on output, i.e., whether the event of interest likely occurs ($Y = c_1$) or not ($Y = c_2$) before time t, using values of covariates $(X_1, X_2, \ldots, X_k)^T = (x_1, x_2, \ldots, x_k)^T$ and a value of time $T = t$ as explanatory variables. Thus, on input of the classifier, there are $k + 1$ covariates $\{X_1, X_2, \ldots, X_k, T\}$, and on output, there is the event component Y, as the scheme in Figure 4 illustrates.

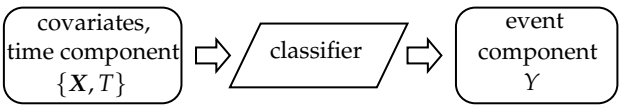

Figure 4. A logic of prediction based on a classification model. Covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k, T)^T$ on input are used for the classifier's building, which enables us to estimate output, i.e., the event of interest's occurrence Y (in time T and for specific values of other covariates $\boldsymbol{X}$). The classifier is an algorithm from the following set: {*multivariate logistic regression*, *naïve Bayes classifiers*, *decision trees*, *random forests*, *artificial neural networks*}.

As a classification algorithm, various approaches, such as multivariate logistic regression, naïve Bayes classifiers, decision trees, random forests, or artificial neural networks, might be chosen, e.g., to maximize the model's predictive accuracy.

2.4. Machine-Learning Classification Algorithms

Following the logic of time-to-event variable decomposition and classification-based prediction of an event of interest's occurrence as demonstrated in the scheme in Figure 4, we describe principles of selected classifying algorithms used for the model building and predicting a COVID-19 antibody blood level decrease below laboratory cut-off.

In general, all classifiers listed below employ covariates $\boldsymbol{X} = (X_1, X_2, \ldots, X_k, T)^T$ on input and classify into two classes, either c_1, or c_2 of a target variable Y, i.e., an event of interest occurrence or non-occurrence before time t.

2.4.1. Multivariate Logistic Regression

Multivariate logistic regression classifies into one of the classes $\{c_1, c_2\}$, i.e., an event of interest occurrence and non-occurrence, of the target variable Y using $k + 1$ covariates $X_1, X_2, \ldots, X_k, T$. A formula of the multivariate logistic regression [28] follows for individual i a form of

$$\log \frac{P(Y_i = c_l \mid \boldsymbol{x}_i, t_i)}{1 - P(Y_i = c_1 \mid \boldsymbol{x}_i, t_i)} = \beta_0 + \sum_{j=1}^{k} \beta_j x_i + \beta_{k+1} t_i + \varepsilon_i,$$

where β_0 is an intercept, β_j are linear coefficients each matched to covariate X_j for $j \in \{1, 2, \ldots, k\}$, β_{k+1} is a linear coefficient matched to time component T, vector $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ contains values of appropriate covariates X_j for individual i, where $j \in \{1, 2, \ldots, k\}$, and ε_i is a residual for individual i.

The linear coefficients are estimated numerically using maximal likelihood [29]. Individual i's value of variable Y is classified using a *maximum-a-posteriori* principle into final class c_ℓ^* so that

$$c_\ell^* = \arg \max_{\ell \in \{1,2\}} \{P(Y = c_\ell \mid \boldsymbol{x}_i, t_i)\}.$$

2.4.2. Naïve Bayes Classifier

Naïve Bayes classifier also predicts the most likely class $c_\ell^* \in \{c_1, c_2\}$ of the target variable Y, where c_1 stands for the event of interest occurrence and c_2 for the event of interest non-occurrence, respectively [30]. Once we apply Bayes theorem, we obtain

$$P(Y_i = c_\ell \mid \boldsymbol{x}_i, t_i) = \frac{P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell)P(Y_i = c_\ell)}{P(\boldsymbol{x}_i, t_i)}. \tag{13}$$

Assuming we have a given dataset, i.e., a matrix containing n vectors such as $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k}, t_i)$ for $\forall i \in \{1, 2, \ldots, n\}$, the probabilities $P(Y_i = c_\ell)$ and $P(\boldsymbol{x}_i, t_i)$ are constant, since

$$\hat{P}(Y_i = c_\ell) = \frac{\text{\# of rows where } Y_i = c_\ell}{n} \tag{14}$$

and

$$\hat{P}(\boldsymbol{x}_i, t_i) = \frac{\text{\# of rows where } \boldsymbol{X}_i = \boldsymbol{x}_i \text{ and } T_i = t_i}{n}. \tag{15}$$

Moreover, if the dataset is well balanced, then for $\forall \ell \in \{1, 2\}$ $\hat{P}(Y_i = c_\ell) = \frac{1}{2}$ and for $\forall p, q \in \{1, 2, \ldots, n\}$ is $\hat{P}(\boldsymbol{x}_p, t_p) = \hat{P}(\boldsymbol{x}_q, t_q)$. Thus, we may write for $\forall \ell \in \{1, 2\}$ and for $\forall i \in \{1, 2, \ldots, n\}$

$$\frac{\hat{P}(Y_i = c_\ell)}{\hat{P}(\boldsymbol{x}_i, t_i)} = \kappa = \text{const.} \tag{16}$$

Thus, using Formulas (14)–(16), we might rewrite Formula (13) as follows:

$$\begin{aligned} P(Y_i = c_\ell \mid \boldsymbol{x}_i, t_i) &= \frac{P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell)P(Y_i = c_\ell)}{P(\boldsymbol{x}_i, t_i)} = \\ &= P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell) \underbrace{\frac{P(Y_i = c_\ell)}{P(\boldsymbol{x}_i, t_i)}}_{=\kappa} = \\ &= P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell) \cdot \kappa \propto \\ &\propto P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell) \end{aligned} \tag{17}$$

Assuming mutual independence of covariates $X_1, X_2, \ldots, X_k, T$, we improve Formula (17) as

$$\begin{aligned} P(Y_i = c_\ell \mid \boldsymbol{x}_i, t_i) &\propto P(\boldsymbol{x}_i, t_i \mid Y_i = c_\ell) \propto \\ &\propto P(X_{i,1} = x_{i,1} \wedge X_{i,2} = x_{i,2} \wedge \cdots \wedge X_{i,k} = x_{i,k} \wedge T_i = t_i \mid Y_i = c_\ell) \propto \\ &\propto \prod_{j=1}^{k} P(X_{i,j} = x_{i,j} \mid Y_i = c_\ell) \cdot P(T_i = t_i \mid Y_i = c_\ell), \end{aligned}$$

and individual i's value of variable Y is classified into final class $c_\ell^* \in \{c_1, c_2\}$, using the maximum-a-posteriori principle, so

$$c_\ell^* = \arg \max_{\ell \in \{1,2\}} \left\{ \prod_{j=1}^{k} P(X_{i,j} = x_{i,j} \mid Y_i = c_\ell) \cdot P(T_i = t_i \mid Y_i = c_\ell) \right\}.$$

The probabilities $P(X_{i,j} = x_{i,j} \mid Y_i = c_\ell)$ and $P(T_i = t_i \mid Y_i = c_\ell)$ for categorical covariates could be estimated as

$$\hat{P}(X_{i,j} = x_{i,j} \mid Y_i = c_\ell) = \frac{\text{\# of rows where } \boldsymbol{X}_{i,j} = \boldsymbol{x}_{i,j} \text{ and } Y_i = c_\ell}{\text{\# of rows where } Y_i = c_\ell}$$

and

$$\hat{P}(T_i = t_i \mid Y_i = c_\ell) = \frac{\text{\# of rows where } T_i = t_i \text{ and } Y_i = c_\ell}{\text{\# of rows where } Y_i = c_\ell},$$

for continuous variables, $\hat{P}(X_{i,j} = x_{i,j} \mid Y_i = c_\ell)$ and $\hat{P}(T_i = t_i \mid Y_i = c_\ell)$ are estimated [31] using conditional normal cumulative distribution function $\hat{P}(X_{i,j} = x_{i,j} \mid Y_i = c_\ell) = \Phi(X_{i,j} = x_{i,j} \pm \epsilon \mid Y_i = c_\ell)$ and $\hat{P}(T_i = t_i \mid Y_i = c_\ell) = \Phi(T_i = t_i \pm \epsilon \mid Y_i = c_\ell)$, both for small $\epsilon > 0$.

2.4.3. Support Vector Machines

Support vector machines natively split the space of all covariates $X_1 \times X_2 \times \cdots \times X_k \times T$ into two subspaces by a hyperplane that maximizes the margins between the hyperplane and the points from both subspaces that are the closest to the hyperplane [32]; see Figure 5.

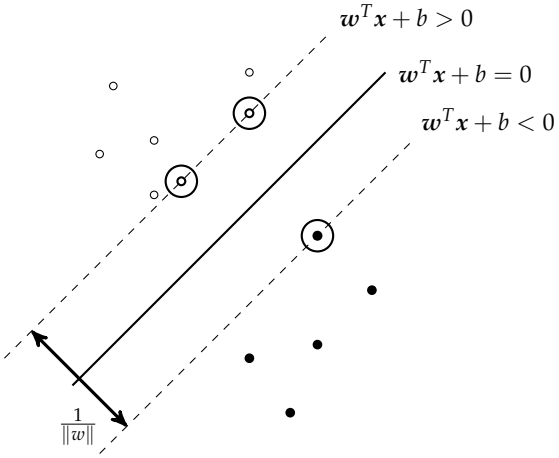

Figure 5. Assuming a two-dimensional case and linear separability of the points belonging to different classes, the margin between the support vector machines' splitting hyperplane (solid line) and both subspaces' closest points defines two boundary hyperplanes (dashed lines), should be maximized. The three points, two "white" and one "black", on the boundary hyperplanes determine the slope of the splitting and boundary hyperplanes and, therefore, are called *support vectors*.

More technically, assuming we have vectors such as $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k}, t_i)^T$, i.e., in other words, these vectors are points in $(k+1)$-dimensional space with coordinates $[x_{i,1}, x_{i,2}, \ldots, x_{i,k}, t_i]$, that belongs either to class $Y = c_1$, or to class $Y = c_2$, any hyperplane in the given universe follows a form of

$$\boldsymbol{w}^T\boldsymbol{x}_i - b,$$

where $\boldsymbol{w}$ is a vector orthonormal, or at least orthogonal to the hyperplane, and b is a tolerated margin's width, i.e., a user's hyperparameter, so that $\frac{b}{\|\boldsymbol{w}\|}$ is the offset of the hyperplane from the universe system of coordinates' origin along the normal vector $\boldsymbol{w}$.

The splitting hyperplane follows an equation $\boldsymbol{w}^T\boldsymbol{x}_i - b = 0$. For all points that belong to class $Y = c_1$ (and are "above" the splitting hyperplane), we suppose to find a boundary hyperplane that is parallel to the splitting hyperplane, so it should have an equation $\boldsymbol{w}^T\boldsymbol{x}_i - b > 0$. Similarly, all remaining points of class $Y = c_2$ should be on or "below" a boundary hyperplane with equation $\boldsymbol{w}^T\boldsymbol{x}_i - b < 0$. If points of class $Y = c_1$ are "above" and points of class $Y = c_2$ are "below" the splitting hyperplane does not matter much and is addressed by $\delta \in \{-1, +1\}$ term in other equations. Assuming linear separability of the points from different classes and formally assigning $Y = c_1 \equiv +1$ and $Y = c_2 \equiv -1$, the distance between these two boundary hyperplanes is equal to $\frac{2}{\|\boldsymbol{w}\|}$ and should be as large as possible, so the searching for max-margin splitting hyperplane means to find

$$\max\left\{\frac{2}{\|\boldsymbol{w}\|}\right\} \qquad \text{subject to} \qquad \delta(\boldsymbol{w}^T\boldsymbol{x}_i - b) \geq 1,$$

where $\delta \in \{-1, +1\}$ term guarantees to work "plus" and "minus" notation if the points of class $Y = c_1 \equiv +1$ or $Y = c_2 \equiv -1$, respectively, would be "below" or "above" the splitting hyperplane, respectively.

If the points that belong to different classes are not linearly separable, we may use the *kernel trick*, i.e., to increase dimensionality of the covariates' universe $X_1 \times X_2 \times \cdots \times X_k \times T$ by one or more dimensions that could enable to find a separating hyperplane [33], as illustrated in Figure 6.

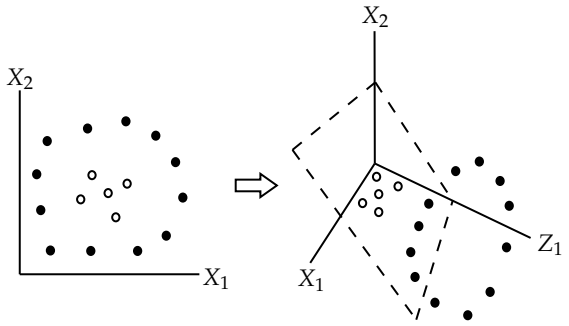

Figure 6. A visualization of the kernel trick's principle. The dimensionality of the original covariates' universe (simplified to) $X_1 \times X_2$ is increased by one (Z_1) or more dimensions, which could enable making the clouds of points from different classes linear separable and could help to find a separating "splitting" hyperplane (with dashed borders).

2.4.4. Decision Trees and Random Forests

Decision trees divide the universe of all covariates $X_1 \times X_2 \times \cdots \times X_k \times T$ into disjunctive orthogonal subspaces related to maximally probably distribution of individual classes of the target variable Y, see Figure 7 for demonstration [34].

Using node rules employing covariates and their grid-searched thresholds that minimize given criterion, e.g., deviance, entropy, or Gini index, a dataset is repeatedly split into new and new branches, while the tree containing node rules, i.e., logic formulas with the covariates and their thresholds, grows up. Once the tree is completely grown, a set of the tree's node rules from a root node to leaves enables the classification of an individual into one of the classes c_1 or c_2.

More technically spoken, let $\pi_{n_\tau,\ell}$ be a proportion of individuals that belong to class c_{ell} based on node n_τ's rule.

If node n_τ is not a leaf one, then the node rule is created so that an *impurity* criterion Q_{n_τ} is minimized. The commonly used impurity criterion is

- misclassification error,
$$Q_{n_\tau} = 1 - \pi_{n_\tau,\ell},$$
- Gini index,
$$Q_{n_\tau} = \sum_{\ell=1}^{2} \pi_{n_\tau,\ell}(1 - \pi_{n_\tau,\ell}),$$
- and deviance (cross-entropy),
$$Q_{n_\tau} = -\sum_{\ell=1}^{2} \pi_{n_\tau,\ell} \cdot \log \pi_{n_\tau,\ell}.$$

If node n_τ is a leaf one, then all observations that are constrained by all node rules from root one up to the leaf one n_t are classified into final class $c^*_\ell \in \{c_1, c_2\}$ of target variable Y, so that

$$c^*_\ell = \arg \max_{\ell \in \{1,2\}} \{\pi_{n_\tau, \ell}\}.$$

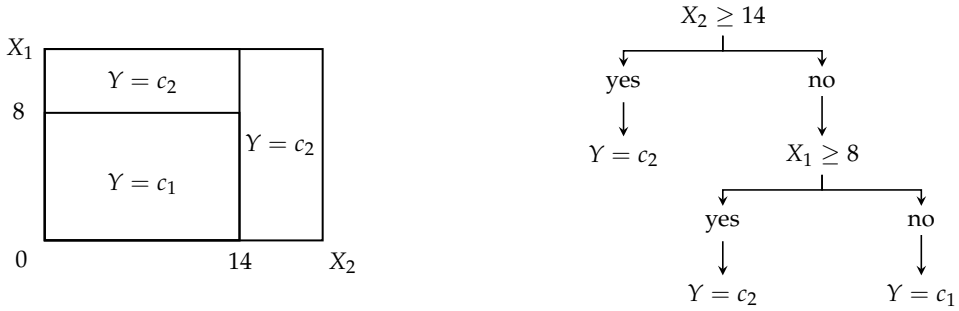

Figure 7. Linear splitting of the variables' (simplified) space $X_1 \times X_2$ according to the likely distribution of target variable Y's classes (on the left) and an appropriate tree representation with two node rules, $X_2 \geq 14$ and $X_1 \geq 8$ (on the right).

Decision trees natively tend to overfit the classification since, whenever there is at least one leaf node classifying into two or more classes, it is repeatedly split into two branches until each leaf node classifies just into one class. To reduce the overfitting, grown trees are *pruned*. A pruned subtree, derived from the grown tree, minimizes the following cost-complexity function,

$$C_\kappa = \sum_{n_t \in \{\boldsymbol{n}_t\}} |\{\boldsymbol{x}_{\text{root} \to n_t}\}| \cdot Q_{n_\tau} + \kappa \cdot |\{\boldsymbol{n}_t\}|,$$

where $\{\boldsymbol{n}_t\}$ is a set of all nodes in the tree, $\{\boldsymbol{x}_{\text{root} \to n_t}\}$ is a set of all nodes in the tree structure from the root to the leaf note n_t, and κ is a tuning parameter governing the trade-off between tree complexity and size (low κ), and tree reproducibility to other similar data (high κ).

Many trees together create a *random forest*. To ensure that trees in a random forest are mutually independent and different enough, each tree is grown using a limited number of covariates that are selected for node rules [35]. Thus, each tree's node might be determined using only $k^* < k + 1$ covariates, which are randomly picked using bootstrap from $\{X_1, X_2, \ldots, X_k, T\}$, which ensures that trees of one random forest are sufficiently different one from another. A voting scheme determines the final class—an individual i is classified into class $c^*_\ell \in \{c_1, c_2\}$ which a maximum of all trees in the random forest classifies into. If there are two classes with maximum trees voting for, one is picked randomly.

2.4.5. Artificial Neural Networks

Artificial neural networks we used in our study are weighted parallel logistic regression models [36], called neurons, as in Figure 8, following an atomic formula:

$$y_{l,j} = \sigma\left(\boldsymbol{w}^T_{l-1} \boldsymbol{z}_{l-1} + b_l\right),$$

where $y_{l,j}$ is a signal, i.e., either 0, or 1, of j-th neuron in l-th layer of neurons aiming at the next $(l+1)$-th layer, $\sigma(\bullet)$ is sigmoid activating function, $\boldsymbol{w}_{l-1}$ is a vector of weights coming from axons of neurons in the previous $(l-1)$-th layer, b_l is l-th layer's activating threshold, and, finally, $\boldsymbol{z}_{l-1}$ is a vector of incoming signals from $(l-1)$-th layer, i.e., it is either a vector of individual i's covariates' values $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k}, t_i)^T$ when the neurons are of the first layer ($l = 1$), so it is a vector of signals $\boldsymbol{y}_{l-1} = (y_{l-1,1}, y_{l-1,2}, \ldots)^T$ coming from $(l-1)$-th layer ($l > 1$).

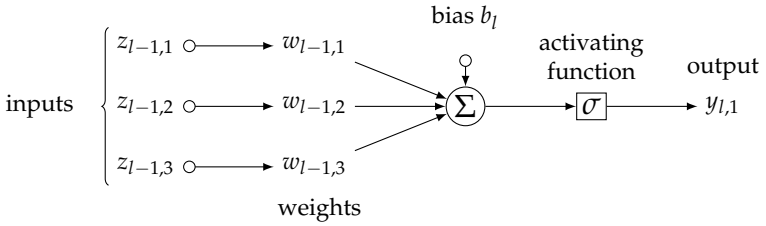

Figure 8. A scheme of one neuron in l-th layer, gaining signals of a form $\Sigma = \boldsymbol{w}_{l-1}^T \boldsymbol{z}_{l-1} = w_{l-1,1}z_{l-1,1} + w_{l-1,2}z_{l-1,2} + w_{l-1,3}z_{l-1,3}$ from $(l-1)$-th layer and transcending signal of a form $y_{l,1} = \sigma(\Sigma + b_l) = \sigma\left(\boldsymbol{w}_{l-1}^T \boldsymbol{z}_{l-1} + b_l\right) = \sigma\left((w_{l-1,1}z_{l-1,1} + w_{l-1,2}z_{l-1,2} + w_{l-1,3}z_{l-1,3}) + b_l\right)$ to next, $(l+1)$-th layer.

A number of layers and neurons in each layer are hyperparameters of the neural network's architecture and should be chosen by an end user. Within a procedure called *backpropagation*, the vectors of weights $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$ are iteratively adjusted by a small gradient per each epoch to minimize loss functions, typically an L_1 or L_2 norm of current and previous neuron's output $y_{l,j}$ and $y'_{l,j}$, i.e., $|y'_{l,j} - y_{l,j}|$ or $|y'_{l,j} - y_{l,j}|^2$, respectively [37]. The learning rate determines the size of the small gradient adjusting weights in each iteration. Although many exist, the commonly applied activating function is a sigmoid one and follows a form of

$$\sigma(\zeta) = \frac{1}{1 + e^{-\zeta}}.$$

In the classification framework, the number of output neurons is equal to the number of classes of target variable Y; each output neuron $y_{\{\#\text{ of layers}\},\,1}$ or $y_{\{\#\text{ of layers}\},\,2}$ represents one of the classes $\{c_1, c_2\}$ and final class $c_\ell^* \in \{c_1, c_2\}$ is

$$c_\ell^* = \arg\max_{\ell \in \{1,2\}} \left\{ y_{\{\#\text{ of layers}\},\,\ell} \right\}.$$

2.5. Evaluation of Classification Algorithms' Performance

Considering individual i's true value of variable Y, i.e., either $Y_i = c_1$, or $Y_i = c_2$, an algorithm may predict the true value correctly, $\hat{Y}_i = Y_i$ or incorrectly, $\hat{Y}_i \neq Y_i$. We may evaluate the performance of a classification algorithm using a proportion of a number of predicted classes equal to true classes to a number of all predicted classes. Assuming a confusion matrix [38] as in Table 1, the number of correctly predicted classes corresponds to $n_{1,1} + n_{2,2}$, while the number of all predictions is $n_{1,1} + n_{1,2} + n_{2,1} + n_{2,2}$.

Table 1. A confusion matrix for two true classes $Y \in \{c_1, c_2\}$ (in rows) and two predicted classes $\hat{Y} \in \{c_1, c_2\}$ (in columns).

		Predicted Class ($\hat{Y}$)	
		c_1	c_2
True class (Y)	c_1	$n_{1,1}$	$n_{1,2}$
	c_2	$n_{2,1}$	$n_{2,2}$

Thus, the predictive accuracy of an algorithm, i.e., what is a point estimate of the probability for which an individual's class would be predicted correctly by the algorithm, is

$$\text{accuracy} = \frac{n_{1,1} + n_{2,2}}{n_{1,1} + n_{1,2} + n_{2,1} + n_{2,2}} = \frac{\sum_{i=1}^{2} n_{i,i}}{\sum_{i=1}^{2} \sum_{j=1}^{2} n_{i,j}}. \tag{18}$$

Even more, assuming that classes $Y_i = c_2$ and $\hat{Y}_i = c_2$ are of "positive" meaning or are of our special interest, then [39], the precision is a point estimate of the probability that an individual classified to $\hat{Y}_i = c_2$ class is truly of class $Y_i = c_2$, so

$$\text{precision} = P\big(Y_i = c_2 | \hat{Y}_i = c_2\big) = \frac{P\big(Y_i = c_2 \wedge \hat{Y}_i = c_2\big)}{P\big(\hat{Y}_i = c_2\big)} = \frac{n_{2,2}}{n_{1,2} + n_{2,2}}, \tag{19}$$

and the recall is a point estimate of the probability that an individual belonging to class $Y_i = c_2$ is classified to class $\hat{Y}_i = c_2$,

$$\text{recall} = P\big(\hat{Y}_i = c_2 | Y_i = c_2\big) = \frac{P\big(\hat{Y}_i = c_2 \wedge Y_i = c_2\big)}{P\big(Y_i = c_2\big)} = \frac{n_{2,2}}{n_{2,1} + n_{2,2}}. \tag{20}$$

Putting the precision and recall together, so-called F1 score is of form,

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{21}$$

and, inspecting Formula (21) above, since F1 score keeps balance between the precision and recall, it is sometimes considered as a metric of an overall model predictive performance [39].

To avoid any bias in the performance measures' estimation, we estimate them multiple times using f-fold cross-validation [40], see Figure 9. Before each iteration of the f-fold cross-validation, the entire dataset is split into two parts following a ratio $(f-1):1$ for training and testing subset, respectively, where $f \in \mathbb{N}$ and $f > 1$. Within each iteration, a portion of $\frac{f-1}{f}$ of all data are used for training of an algorithm, while the remaining $\frac{1}{f}$ portion of all data are used for testing a model based on the trained algorithm.

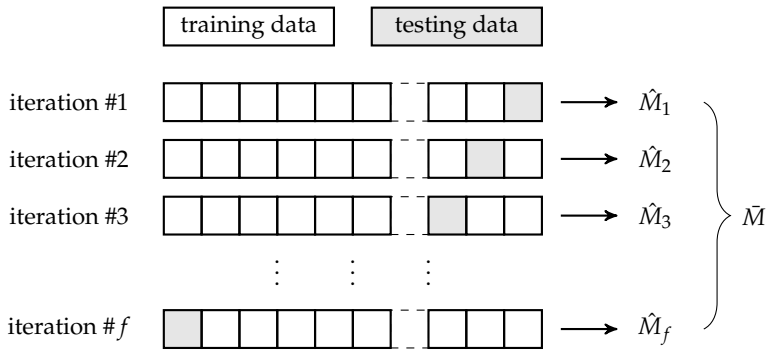

Figure 9. A scheme of f-fold cross-validation, where $f \in \mathbb{N}$ and $f > 1$. Training data are colored in white, while testing data are in grey.

The j-th iteration of the f-fold cross-validation outputs a defined estimate $\hat{M}_j$ of a performance metric, so, in our case, $M \in \{\text{accuracy}, \text{precision}, \text{recall}, \text{F1 score}\}$ following Formulas (18)–(21). The estimates are eventually averaged to obtain a more robust estimate

$$\bar{M} = \frac{1}{f} \sum_{j=1}^{f} \hat{M}_j,$$

thus, in our case,

$$\overline{\text{accuracy}} = \frac{1}{f}\sum_{j=1}^{f}\widehat{\text{accuracy}}_j,$$
$$\overline{\text{precision}} = \frac{1}{f}\sum_{j=1}^{f}\widehat{\text{precision}}_j,$$
$$\overline{\text{recall}} = \frac{1}{f}\sum_{j=1}^{f}\widehat{\text{recall}}_j,$$
$$\overline{\{\text{F1 score}\}} = \frac{1}{f}\sum_{j=1}^{f}\widehat{\{\text{F1 score}\}}_j.$$

Finally, considering all f iterations of the f-fold cross-validation, we also report a confusion matrix containing medians of numbers of matches and mismatches between true classes Y and predicted ones $\hat{Y}$. Using the matrix in Table 1, the *median confusion matrix* is the matrix in Table 2, where $\tilde{n}_{i,j}$ for $\forall i \in \{1,2\}$ and $\forall j \in \{1,2\}$ is median value over all values $n_{i,j}$ from the 1-st, 2-nd, ..., f-th iteration of the f-fold cross-validation.

Table 2. A median confusion matrix for two true classes $Y \in \{c_1, c_2\}$ (in rows) and two predicted classes $\hat{Y} \in \{c_1, c_2\}$ (in columns). Assuming that value ${}_1n_{i,j}$ comes from the 1-st iteration, value ${}_2n_{i,j}$ comes from the 2-nd iteration, ..., value ${}_fn_{i,j}$ comes from the f-th iteration, then $\tilde{n}_{i,j}$ is median over values $\{{}_1n_{i,j}, {}_2n_{i,j}, \ldots, {}_fn_{i,j}\}$.

		Predicted Class ($\hat{Y}$)	
		c_1	c_2
True class (Y)	c_1	$\tilde{n}_{1,1}$	$\tilde{n}_{1,2}$
	c_2	$\tilde{n}_{2,1}$	$\tilde{n}_{2,2}$

Summation of each median confusion matrix should be approximately equal to a portion $\frac{1}{f}$ of all data since each median confusion matrix is produced by the testing, i.e., predicting procedure within an iteration of f-fold cross-validation.

2.6. Asymptotic Time Complexity of Proposed Prediction Based on Time-to-Event Variable Decomposition

Let us briefly analyze the asymptotic time complexity of the proposed method, i.e., the time-to-event variable decomposition and prediction of an event of interest's occurrence in time using time and event components, covariates, and machine-learning classification algorithms.

Assuming we have $n \in \mathbb{N}$ individuals in a dataset in total, time-to-event variable decomposition is a unit time operation made for each of them, so its asymptotic time complexity, using Bachmann–Landau notation [41], is $\Theta(n)$. Picking a machine-learning classifier, let us suppose that it takes $\Theta(\lambda(n))$ time when training on n observations (and all their covariates' values), whereas it takes $\Theta(\pi(n))$ time if testing or predicting an output for n observations (both testing and predicting procedures could be considered the same regarding their asymptotic time complexity since they use a trained model and only output target variable values for input). Firstly, the training, testing, and prediction are linear procedures with respect to a number of observations, i.e., training, testing, or prediction considering $2n$ observations would take approximately double the time than considering only n observations.

$$\begin{aligned}\Theta(\lambda(\ell n)) &\approx \Theta(\ell \cdot \lambda(n)) \approx \ell \cdot \Theta(\lambda(n))\\ \Theta(\pi(\ell n)) &\approx \Theta(\ell \cdot \pi(n)) \approx \ell \cdot \Theta(\pi(n)),\end{aligned} \tag{22}$$

however, using Bachmann–Landau logic, linear multipliers do not change asymptotic time complexity, so it is as well

$$\begin{aligned} \ell \cdot \Theta(\lambda(n)) &\approx \Theta(\lambda(n)) \\ \ell \cdot \Theta(\pi(n)) &\approx \Theta(\pi(n)). \end{aligned} \tag{23}$$

In addition, we may assume that training a model is generally not faster than testing a model or using it for prediction, considering n observations, so

$$\Theta(\lambda(n)) \gtrsim \Theta(\pi(n))$$

and neither training nor testing nor even predicting considering n observations is faster than n-times performed unit time operation, so

$$\Theta(\lambda(n)) \gtrsim \Theta(\pi(n)) \gtrsim \Theta(n). \tag{24}$$

Applying derivations from the previous section, within each iteration of f-fold cross-validation, we build a model on $\frac{f-1}{f}n$ portion of the dataset, test it on $\frac{1}{f}n$ portion of the dataset. This is repeated $f \in \mathbb{N}$ times since there are f iterations in the f-fold cross-validation. Thus, applied to n observations, the time-to-event variable decomposition, f-fold cross-validation containing training and testing all f models and prediction on output, respectively, would asymptotically take $\Theta(\bullet)$, such that

$$\begin{aligned} \Theta(\bullet) &\approx \Theta(n) + f \cdot \left(\Theta\left(\lambda\left(\frac{f-1}{f}n\right)\right) + \Theta\left(\pi\left(\frac{1}{f}n\right)\right)\right) + \Theta(\pi(n)) \overset{(22)}{\approx} \\ &\overset{(22)}{\approx} \Theta(n) + f \cdot \left(\frac{f-1}{f} \cdot \Theta(\lambda(n)) + \frac{1}{f} \cdot \Theta(\pi(n))\right) + \Theta(\pi(n)) \approx \\ &\approx \Theta(n) + (f-1) \cdot \Theta(\lambda(n)) + \Theta(\pi(n)) + \Theta(\pi(n)) \overset{(23)}{\approx} \\ &\overset{(23)}{\approx} \Theta(\lambda(n)) + \Theta(\pi(n)) + \Theta(n), \end{aligned} \tag{25}$$

so, improving Formula (25), it is obviously

$$\begin{aligned} \Theta(\bullet) &\approx \Theta(\lambda(n)) + \Theta(\pi(n)) + \Theta(n) \gtrsim \\ &\gtrsim \Theta(\lambda(n)) \end{aligned} \tag{26}$$

and also

$$\begin{aligned} \Theta(\bullet) &\approx \Theta(\lambda(n)) + \Theta(\pi(n)) + \Theta(n) \overset{(24)}{\lesssim} \\ &\overset{(24)}{\lesssim} \Theta(\lambda(n)) + \Theta(\lambda(n)) + \Theta(\lambda(n)) \approx \\ &\approx 3 \cdot \Theta(\lambda(n)) \overset{(23)}{\approx} \\ &\overset{(23)}{\approx} \Theta(\lambda(n)). \end{aligned} \tag{27}$$

Putting Formulas (26) and (27) together, we obtain

$$\Theta(\lambda(n)) \gtrsim \Theta(\bullet) \gtrsim \Theta(\lambda(n));$$

thus,

$$\Theta(\bullet) \approx \Theta(\lambda(n)),$$

and, finally, asymptotic time complexity $\Theta(\bullet)$ of the time-to-event decomposition, followed by prediction of an event of interest's occurrence in time using machine-learning classification algorithms is approximately equal to asymptotic time complexity of training of the classification algorithm, regardless of its kind. In other words, the proposed method does

not take significantly longer computational time when performed from beginning to end than the classifier used within the technique.

2.7. Description of Used Dataset

The dataset used to confirm the proposed methodology's feasibility comes from the Department of Occupational Medicine, University Hospital Olomouc. The data contain covariates' values in about COVID-19 non-vaccinated 663 patients; there are 34 covariates describing COVID-19 antibody blood level values and their decrease below laboratory cut-off for IgG and IgM antibodies in various time points and multiple biometric and other variables, continuous and categorical. All patients have been informed in advance about using their data for the study, following ideas of Helsinki's declaration [42].

Regarding the covariates, there are

- *continuous* variables such as age (in years), weight (in kilograms), height (in centimeters), body mass index (in kilogram per squared meters), COVID-19 IgG antibody blood level, COVID-19 IgM antibody blood level, a total count of COVID-19 defined symptoms, the time between beginning and end of COVID-19 symptoms (in days), the time between laboratory-based proof of COVID-19 and antibody blood sampling (in days), the time between COVID-19 symptoms' offset and antibody blood sampling (in days);
- *categorical* variables such as sex (male/female), COVID-19 IgG antibody blood level decrease below laboratory cut-off (yes/no), COVID-19 IgM antibody blood level decrease below laboratory cut-off (yes/no), COVID-19 defined symptoms such as headache (yes/no), throat pain (yes/no), chest pain (yes/no), muscle pain (yes/no), dyspnea (yes/no), fever (yes/no), subfebrile (yes/no), cough (yes/no), lack of appetite (yes/no), diarrhea (yes/no), common cold (yes/no), fatigue (yes/no), rash (yes/no), loss of taste (yes/no), loss of smell (yes/no), nausea (yes/no), mental troubles (yes/no), and insomnia (yes/no).

As a time-to-event variable, a COVID-19 IgG antibody blood level decrease below the laboratory cut-off or COVID-19 IgM antibody blood level decrease below the laboratory cut-off, respectively, is considered the event component. In contrast, the time between laboratory-based proof of COVID-19 and antibody blood sampling is taken into account for the time component.

3. Results

We applied the Cox proportional hazard model and proposed methodology handling time-to-event decomposition and classification on the dataset as described in the previous sections. All computations were performed using R statistical and programming language [43].

Firstly, we performed and built the Cox proportional hazard model, considering COVID-19 antibody blood level decrease or non-decrease below laboratory cut-off (for IgG and IgM antibodies, respectively) as an event of interest that does or does not occur when antibody blood sampling is carried out a varying time period after COVID-19 symptoms' onset is laboratory proven.

Assuming *Cox proportional hazard model* for the prediction of COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off, the prediction of IgG antibody decrease below the cut-off was repeated ten times within 10-fold cross-validation and output the predictive accuracy about 0.796, precision about 0.889, recall of 0.951, and F1 score about 0.884; see Table 3. The median confusion matrix for Cox proportional hazard model predicting IgG antibody decrease below cut-off is in Figure 10. An average value of threshold $p_{threshold}$ over all iterations of 10-fold cross-validation for IgG decrease prediction was about $\bar{p}_{threshold} = 0.070$. Cox proportional hazard model predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.627, precision of 0.598, recall of 0.519 and F1 score about 0.507; see Table 4. The median confusion matrix for Cox proportional hazard model predicting IgM anti-

body decrease below cut-off is in Figure 11. An average threshold $p_{threshold}$ over 10-fold cross-validation's iterations when IgM decrease predicted was about $\bar{p}_{threshold} = 0.765$.

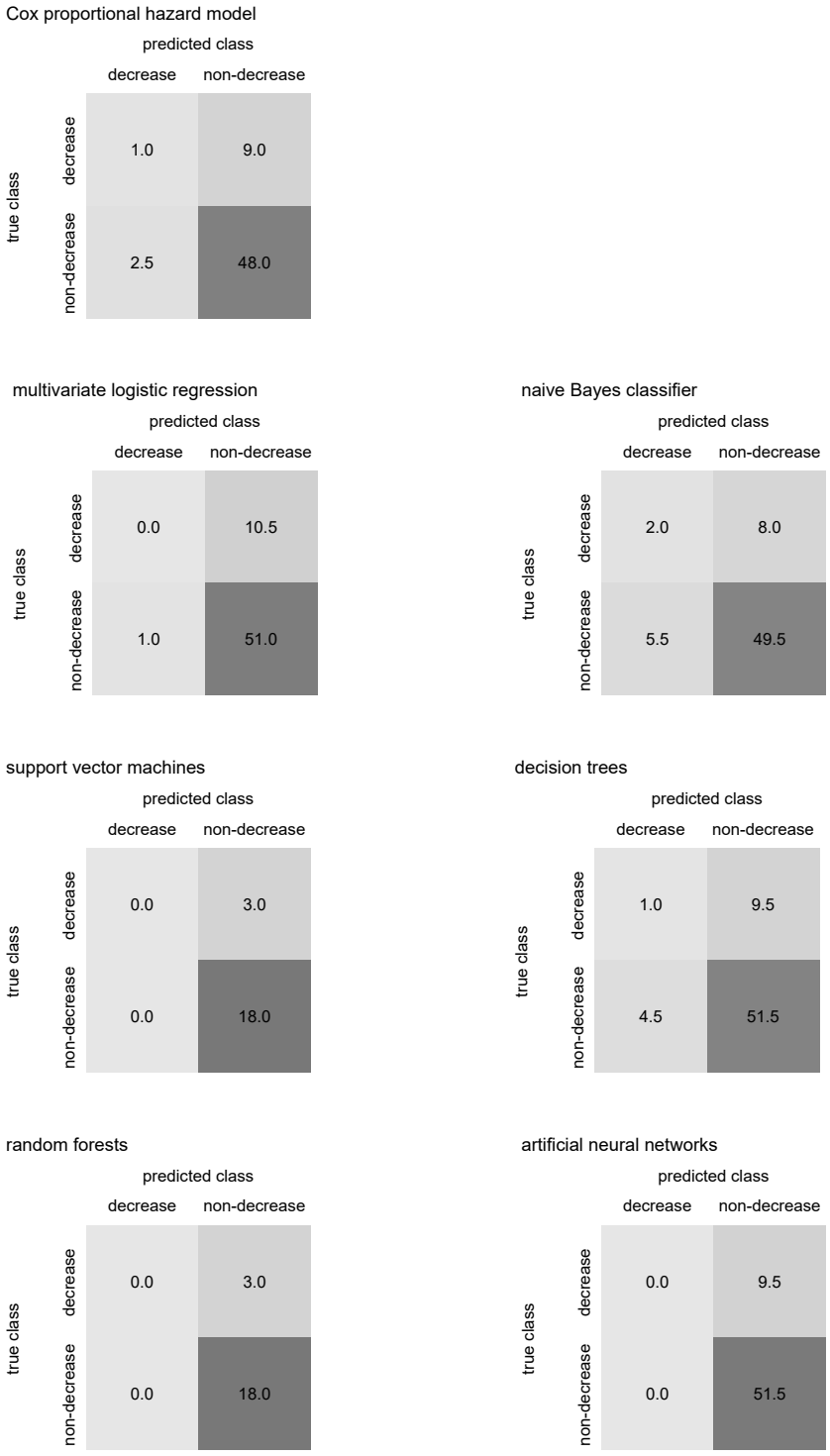

Figure 10. Median confusion matrices for predicting COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off using the listed algorithms.

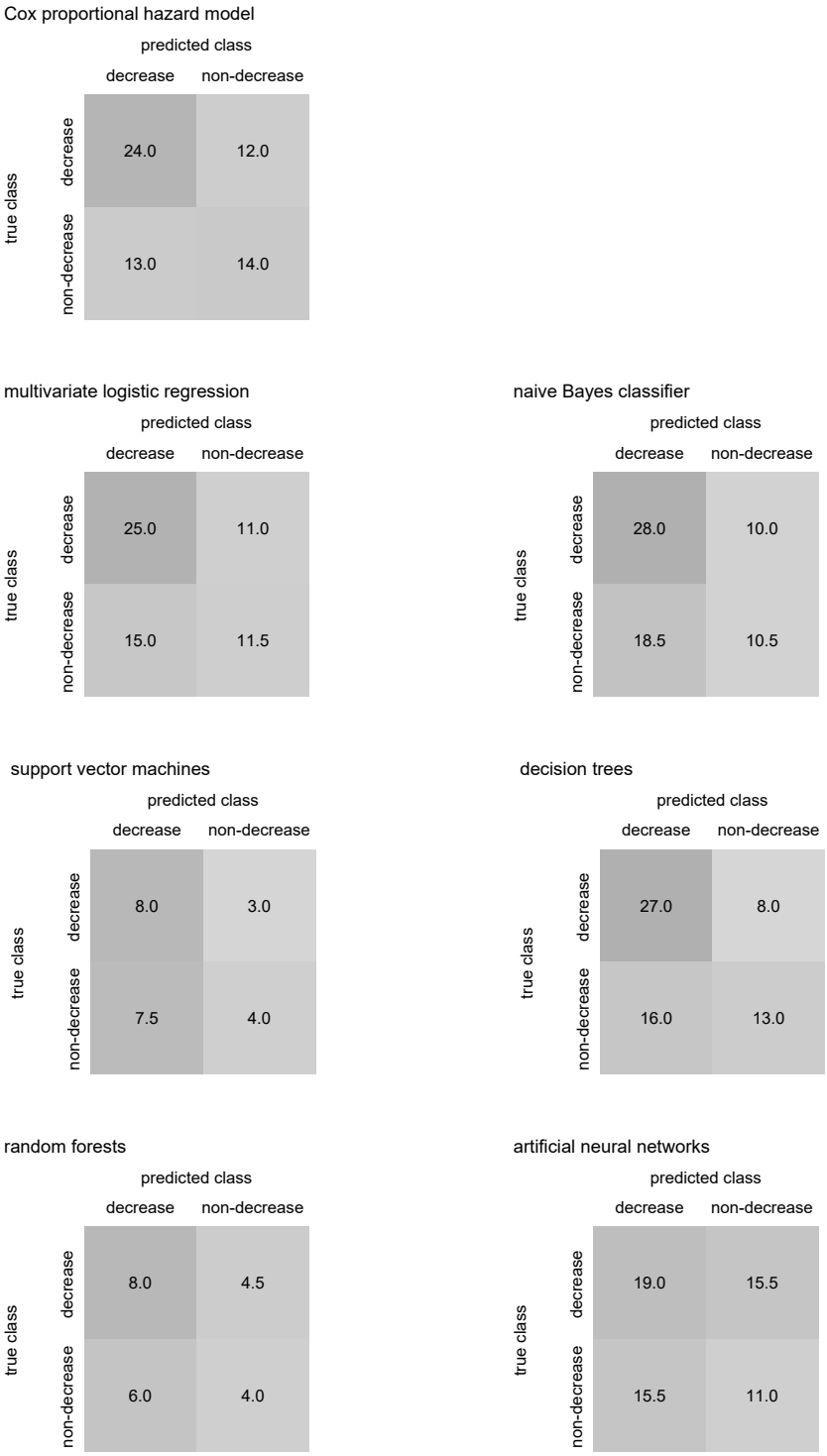

Figure 11. Median confusion matrices for predicting COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off using the listed algorithms.

Let us proceed to the proposed technique. Supposing *multivariate logistic regression* as a tool for the prediction of COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off, the prediction of IgG antibody decrease was repeated ten times within 10-fold cross-validation and ended with the predictive accuracy about 0.811, precision about 0.820, recall of 0.984, and F1 score about 0.894, see Table 3. The median confusion matrix for multivariate logistic regression predicting IgG antibody decrease below cut-off is in Figure 10. Multivariate logistic regression predicted COVID-19 IgM antibody blood level

decrease or non-decrease below laboratory cut-off with predictive accuracy 0.578, precision about 0.530, recall of 0.429, and F1 score about 0.468, see Table 4. The median confusion matrix for multivariate logistic regression predicting IgM antibody decrease below cut-off is in Figure 11.

Table 3. Predictive accuracy, precision, recall and F1 score of COVID-19 IgG antibody blood level decrease below laboratory cut-off's estimation for the listed algorithms, both traditional Cox's model and classifiers using a decomposed time-to-event variable, calculated using 10-fold cross-validation.

Algorithm	Predictive Accuracy	Precision	Recall	F1 Score
Cox proportional hazard model	0.796	0.889	0.951	0.884
multivariate logistic regression	0.811	0.820	0.984	0.894
naïve Bayes classifier	0.771	0.841	0.896	0.865
support vector machines	0.845	0.845	1.000	0.913
decision trees	0.783	0.833	0.920	0.874
random forests	0.836	0.850	0.980	0.908
artificial neural networks	0.806	0.827	0.953	0.881

Table 4. Predictive accuracy, precision, recall and F1 score of COVID-19 IgM antibody blood level decrease below laboratory cut-off's estimation for the listed algorithms, both traditional Cox's model and classifiers using a decomposed time-to-event variable, calculated using 10-fold cross-validation.

Algorithm	Predictive Accuracy	Precision	Recall	F1 Score
Cox proportional hazard model	0.627	0.598	0.519	0.507
multivariate logistic regression	0.578	0.530	0.429	0.468
naïve Bayes classifier	0.574	0.532	0.373	0.428
support vector machines	0.527	0.506	0.347	0.393
decision trees	0.583	0.551	0.412	0.461
random forests	0.555	0.556	0.419	0.467
artificial neural networks	0.516	0.443	0.402	0.518

Naïve Bayes classifier, when predicting COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off, was repeated ten times within 10-fold cross-validation and returned the predictive accuracy about 0.771, precision about 0.841, recall of 0.896, and F1 score about 0.865; see Table 3. The median confusion matrix for the naïve Bayes classifier predicting IgG antibody decrease below cut-off is in Figure 10. Naïve Bayes classifier predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.574, precision about 0.532, recall of 0.373, and F1 score about 0.428, see Table 4. The median confusion matrix for naïve Bayes classifier predicting IgM antibody decrease below cut-off is in Figure 11.

When *support vector machines* using nonlinear kernel trick were considered for prediction of COVID-19, IgG antibody blood level decrease or non-decrease below laboratory cut-off, the algorithm ended with the predictive accuracy about 0.845, precision about 0.845, recall of 1.000, and F1 score about 0.913; see Table 3. The median confusion matrix for support vector machines predicting IgG antibody decrease below cut-off is in Figure 10. Support vector machines predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.527, precision about 0.506, recall of 0.347, and F1 score about 0.393; see Table 4. The median confusion matrix for support vector machines predicting IgM antibody decrease below cut-off is in Figure 11.

Decision trees predicted COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off using 10-fold cross-validation with the predictive accuracy about 0.783, precision of 0.833, recall of 0.920, and F1 score about 0.874, see Table 3. The pruning parameter we used for model training has been adopted from default and recommended setting, i.e., $\kappa = 0.01$. The median confusion matrix for decision trees predicting IgG

antibody decrease below cut-off is in Figure 10. Decision trees predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.583, precision about 0.551, recall of 0.412, and F1 score about 0.461, see Table 4. The median confusion matrix for decision trees predicting IgM antibody decrease below cut-off is in Figure 11.

Random forests were also repeated ten times within 10-fold cross-validation to predict COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off. Each model of the random forests consisted of 1000 decision trees. The random forests' predictive accuracy of IgG antibody decrease below cut-off is about 0.836, precision about 0.850, recall of 0.980, and F1 score about 0.908; see Table 3. The median confusion matrix for random forests predicting IgG antibody decrease below cut-off is in Figure 10. Random forests predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.555, precision about 0.556, recall of 0.419, and F1 score about 0.467; see Table 4. The median confusion matrix for random forests predicting IgM antibody decrease below cut-off is in Figure 11.

Finally, *artificial neural networks* with backpropagation, we performed ten times within 10-fold cross-validation to predict COVID-19 IgG antibody blood level decrease or non-decrease below laboratory cut-off, output the predictive accuracy of IgG antibody decrease below cut-off about 0.806, precision about 0.827, recall of 0.953, and F1 score about 0.881; see Table 3. For each iteration of the 10-fold cross-validation, we used two hidden layers with 10 and 5 neurons, respectively, and set the learning rate to an interval of approximately -0.10 to $+0.30$. The median confusion matrix for artificial neural networks predicting IgG antibody decrease below cut-off is in Figure 10. Artificial neural networks predicted COVID-19 IgM antibody blood level decrease or non-decrease below laboratory cut-off with predictive accuracy 0.516 precision about 0.443, recall of 0.402, and F1 score about 0.518; see Table 4. The median confusion matrix for artificial neural networks predicting IgM antibody decrease below cut-off is in Figure 11.

Median confusion matrices in Figures 10 and 11 may help to identify whether an algorithm predicting COVID-19 antibody blood level decrease below laboratory cut-off tends to over- or underestimate the antibody decrease. Thus, comparing predicted numbers for antibody blood decrease and non-decrease in the median confusion matrices columns could be useful.

4. Discussion

Prediction of the time period till an event of interest is the most challenging and essential task in statistics, particularly when the event has a significant impact on an individual's life. Without a doubt, the blood level of antibodies against COVID-19 and their development in time, especially their decrease below laboratory cut-off, also called *seronegativity*, could determine how an individual would face COVID-19 infection when exposed to it. This is one of the reasons we wanted to compare traditional statistical techniques and machine-learning-based approaches on antibody blood level decrease just for the diagnosis of COVID-19.

Survival analysis provides a toolbox of classical methods that model and predict the time to event. Many methods are commonly used regardless of whether they meet their formal assumptions when applied to real data. Since these methods, including the Cox proportional hazard model for prediction of the time to event, handle the relatively advanced concept of time-to-event modeling, they are limited by relatively strict statistical assumptions. Fortunately, there are various alternatives for how to address the issue of assumptions violation; in the case of the Cox proportional hazard model, we may use rather stratified models that partition an input dataset into multiple parts and build models for them independently, or we might prefer to perform other time-varying and time-partitioning that split the hazard function into several consecutive parts and model the parts individually.

In addition, machine-learning approaches could predict in a time-to-event fashion. A promising fact about the application of machine-learning algorithms not only in survival analysis is that these algorithms usually have very relaxed or not at all prior statistical assumptions. However, the majority of works and papers that employ machine-learning-based techniques for purposes of survival analyses often use mixed methods that combine features from survival analysis and machine-learning, namely scoring and counting models [44], linear-separating survival models such as survival random forests [45], or discrete-time survival models [46]. Nevertheless, if the papers predicted the time to an event of interest and use comparable metrics of performance [47], regardless of data focus, they usually reach predictive accuracy of about 0.7–0.9. These are similar predictive accuracy values we have obtained using the proposed time-to-event decomposition and ongoing classification too. Besides the Cox proportional hazard model, there are parametric survival models such as Weibull or log-normal ones. Those are flexible and might work well on real-world data; however, they require an initial assumption on the parametric choice of the baseline hazard function. If the assumption is wrong, predictions could fail. For example, Valvo in [48] predicts COVID-19-related deaths (instead of antibody decrease) using a log-normal survival model with an average error of about 20%, which is, although a different topic, a similar value to our results.

Going deeper into machine learning applied for prediction in COVID-19 diagnosis, many papers deal with predictions of COVID-19 symptoms' offset, COVID-19 recurrence, long-term COVID-19 disease, post-COVID-19 syndrome [49–51]. However, all the mentioned topics of prediction related to COVID-19 manifestation are usually based on datasets that consist of individual observations of symptoms, biometric data, and laboratory or medical imaging data. Thus, very often, the data size is very large, so machine learning is a legit way for how to analyze these kinds of data; a time-to-event variable is generally missing, so such data are not suitable for survival analysis, though. This may be why papers applying machine-learning prediction on survival data related to COVID-19 are not as frequent as we could initially expect, considering the importance of the topic, or publicly available data are usually aggregated up to a higher level, which is generally not an optimal starting point for survival analysis. Furthermore, some publications are still estimating COVID-19 blood antibody level's development in time [52–54], but using traditional methods, so not applying time-to-event decomposition.

Of papers dealing with COVID-19-related predictions using machine-learning techniques, Willette et al. [55] predicted the risk of COVID-19 severity and risk of hospitalization due to COVID-19, respectively, using discriminant and classification algorithms applied on 7539 observations (!) with variable similar to ours, and received an accuracy of about 0.969 and 0.803, respectively. While our dataset is more than ten times smaller, we obtained similar results. Kurano et al. [56] employed XGBoost (extreme gradient boosting) to classify COVID-19 severity using immunology-related variables. Applied on 134 patients, they received a predictive accuracy between 0.380–0.900 for various lengths of symptoms' onset. Using standard explanatory variables, Singh et al. [57] performed support vector machines to determine infectious COVID-19 status on 1063 recipients with final accuracy greater than 0.700. In addition, Rostami and Oussalah [58] combined feature selectors with explainable trees and others to predict COVID-19 diagnosis. Applied to available 5644 patients' blood test data containing 111 features, they received an accuracy of about 0.877 for XGBoost, about 0.848 for support vector machines, about 0.853 for neural networks, and 0.884 for explainable decision trees, respectively. Cobre at al. [59] introduced a new method for COVID-19 severity, combining various algorithms such as artificial neural networks, decision trees, discriminant analysis, and k-nearest neighbor. Applied on biochemical tests of 5643 patients, they obtained a predictive accuracy of about 0.840. Albeit no COVID-19-related predictions, but classification into acute organ injury or non-acute organ injury, Duan et al. [60] received precision and recall slightly above 0.800, using about 20 features of 339 patients. Thus, the predictive performance we obtained using our proposed method seems comparable with the literature.

However, according to our expectations, when sample sizes, as well as the numbers of features, are enormous and machine-learning algorithms could return outstanding results, as Bhargava et al. [61] showed when they predicted COVID-19 disease using nine large datasets combining laboratory and imaging data and designed the algorithms in deep-learning fashion. As a result, they received an accuracy above 0.900, or, using ensembled algorithms, even close to 0.990.

Our proposed method, based on time-to-event decomposition and machine-learning classification of individuals to an event of interest's occurrence or non-occurrence, using the time to event as an explanatory covariate, might contribute to the application of machine-learning for survival-like prediction in COVID-19 patients. Surely we do not want to claim that the method we are introducing in the paper would work on each dataset of similar size; however, thanks to a robust estimate of the predictive accuracy using 10-fold cross-validation, we may believe that the results we have obtained are not only random but supported by sufficient evidence of perfomance metrics and method's properties.

Considering the IgG antibodies and their blood level decrease below laboratory cut-off in our patients, all performed algorithms show satisfactory performance and output relatively promising results regarding the predictive accuracy, as figured in Table 3. The Cox proportional hazard model, taking into account as a golden standard, reported similar predictive accuracies such as multivariate logistic regression, naïve Bayes classifier, and decision trees. An assumption of the Cox model that survival curves for various combinations of covariates' values should be met—particularly, survival curves would not cross each other or drop to zero, as plotted in Figures 1 and 2. The multivariate logistic regression is a "baseline" model that performs well in prediction but might suffer from multicollinearity between covariates [62]. Naïve Bayes classifier is a relatively simple algorithm and often outputs surprisingly good results; however, it depends on only one assumption, but its performance could be ruined if it is violated—explanatory covariates should be independent, as applied in Formula (17). Decision trees are practically assumption-free [63] but become more powerful when creating an entire random forest model. Support vector machines and random forests seem to noticeably outperform the Cox model in predictive accuracy, precision, recall, and F1 score—both algorithms reached all metrics higher than Cox's regression using our dataset. The support vector machines are sophisticated algorithms. Using the kernel trick, these can find a separating hyperplane even for linear non-separable points of different classes. The random forest is the only algorithm among others that is natively ensembled, i.e., consists of a large number of other classification algorithms—decision trees. This is why the random forest typically shows good predictive performance. Among all algorithms applied to the data and predicting IgG antibody decrease, artificial neural networks reported the predictive performance slightly better than the Cox's model—neural networks are universal classifiers, and their performance could be even improved when larger subsets are used for training, different activating function applied or various architecture of hidden layers investigated. Performance metrics other than predictive accuracy, as listed in Table 3, are more than satisfying too—and each of them is greater than 0.800.

Prediction of IgM antibody decrease might be tricky since IgM antibodies are less related to a cause it induced their growth, i.e., COVID-19 exposure [64]. This might be why all algorithms performed mutually similar (and relatively poor) outputs of predictive accuracy and other metrics, as reported in Table 4.

In the case of IgG decrease prediction, support vector machines, random forest, and neural networks do not predict the decrease class; however, the number of individuals with antibody decrease is significantly lower than those with no antibody decrease, see Figure 10, so the predictive accuracy is not affected. Still, the possible lack of data in the "IgG decrease" class in training sets could cause the mentioned algorithms to fail to classify a few individuals from the class correctly. Mainly, neural networks are sensitive to this. If the median confusion matrix summation in Figures 10 and 11 is lower than approximately one-tenth of the entire dataset size, then the antibody decrease is likely

not predicted for some individuals. This could happen due to their missing covariates' values, which occurred for the three previously mentioned algorithms. Inspecting median confusion matrices in Figure 11, there seems to be no algorithm that would fail in the prediction of one of the classes in IgM antibody decrease. However, in general, the accuracy of IgM decrease is relatively low, likely caused by the advanced complexity of IgM (!) decrease prediction (compared to IgG decrease prediction).

As for the *limitation* of the work, the fact that the proposed methodology works with good performance, particularly for IgG antibody decrease prediction, does not guarantee that it would work on another dataset with similar or even better performance. In addition, classification into classes that stand for an event of interest's occurrence and non-occurrence consider both classes equal, which sidelines the censoring; however, it seems not to affect prediction performance. In addition, when the classifiers are mutually compared within the proposed method, one should take into account the fact that some of the algorithms did not classify all observations, as we see in Figures 10 and 11 (i.e., when the summation of a median confusion matrix is less than one-tenth of the dataset), and consider such a comparison rather only as approximative. Machine-learning classification algorithms within the proposed method may bias any inference potentially carried out using the event of interest's occurrence estimates [65]—in this article, we are primarily interested in prediction paradigm and predictive performance. Finally, albeit not a limitation, but rather a note, varying tuning parameters set for classifiers in the introduced method and different sample sizes used for classifiers' training, testing, and predicting may lead to other predictive performances.

5. Conclusions

Whether an individual would likely experience an event of interest or not, and if so, when exactly, is an essential predictive task in survival analysis. Unfortunately, statistical assumptions often limit methods commonly used to perform this forecasting.

In this work, we address the issue of assumption violation and employ machine learning, usually assumption-free or assumption-relaxed, in the predictive task. We decompose the time-to-event variable into two components—a time and an event one. While the event component is classified using various machine-learning classification algorithms, the time component becomes one of the covariates on the input of the classification models. Classifying into an event of interest's occurrence and non-occurrence enables us to compare our proposed method with the traditional Cox proportional hazard model.

We apply the introduced methodology to COVID-19 antibody data where we predict IgG and IgM antibody blood level decrease below laboratory cut-off, also considering other explanatory covariates besides the time component.

The asymptotic time complexity of the proposed method equals the computational time of a classifier employed in the method. Compared to the Cox model, the classification does not take into account any censoring and considers both classes, i.e., an event of interest occurrence and non-occurrence, as equal. However, predictive performance measured using predictive accuracy and other metrics is for some models, particularly when IgG antibody decrease is estimated, higher than for the Cox model. Namely, multivariate logistic regression (with an accuracy of 0.811), support vector machines (with an accuracy of 0.845), random forests (with an accuracy of 0.836), and artificial neural networks (with an accuracy of 0.806) seem to outperform the Cox's regression (with an accuracy of 0.796), applied as classifiers in the proposed method on the COVID-19 data and predicting IgG antibody decrease below the cut-off. The precision, recall, and F1 score of the four named classifiers are constantly high, generally above 0.800. Regarding IgM antibody decrease below cut-off, Cox's regression and the proposed method employing various classifiers perform relatively poorly, likely due to a weak association between COVID-19 exposure and IgM antibody production. In comparison, Cox's model reached an accuracy of about 0.627, and all classifiers within the introduced method predicted with an accuracy of about 0.520–0.583.

The proposed method seems to be a promising tool, currently applied to COVID-19 IgG antibody decrease prediction. Since IgG antibodies are closely related to protection against COVID-19 disease, an effective and accurate forecast of their (non-)decrease below laboratory cut-off could help—with no explicit need for time-consuming and expensive blood testing—to early identify COVID-19 non-vaccinated individuals with sufficient antibody level that could not undergo boosting vaccination when new COVID-19 outbreak incomes, or the non-vaccinated with IgG antibody decrease could be detected early as in risk of severe COVID-19, using only the algorithm and variables from their case history.

As for the future outlook, the proposed method could benefit from ensembled classifiers, increasing their predictive accuracy. In addition, various tuning parameters' settings might improve predictive performance. Probably, the proposed method broadly applies to similar problems based on two-states survival prediction tasks.

Author Contributions: Conceptualization, L.Š. (Lubomír Štěpánek); methodology, L.Š. (Lubomír Štěpánek); software, L.Š. (Lubomír Štěpánek); validation, L.Š. (Lubomír Štěpánek), I.M., L.M.; formal analysis, L.Š. (Lubomír Štěpánek); investigation, L.Š. (Lubomír Štěpánek); resources, L.Š. (Lubomír Štěpánek); data acquisition: L.Š. (Ladislav Štěpánek), M.N., A.B.; data curation, L.Š. (Ladislav Štěpánek), M.N., L.Š. (Lubomír Štěpánek); writing—original draft preparation, L.Š. (Lubomír Štěpánek); writing—review and editing, L.Š. (Lubomír Štěpánek), L.Š. (Ladislav Štěpánek), M.N., F.H.; visualization, L.Š. (Lubomír Štěpánek), F.H.; supervision, I.M., L.M., M.N.; project administration, L.Š. (Lubomír Štěpánek); funding acquisition, L.Š. (Lubomír Štěpánek). All authors have read and agreed to the submitted version of the manuscript.

Funding: This paper is supported by the grant OP VVV IGA/A, CZ.02.2.69/0.0/0.0/19_073/0016936 with no. 18/2021, which has been provided by the Internal Grant Agency of the Prague University of Economics and Business, and by the grant IGA_LF_2022_005 provided by Palacký University Fund.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The clinical part of the study was approved by the Ethics Committee of the University Hospital Olomouc and Faculty of Medicine and Dentistry, Palacký University Olomouc (reference No. 18/21). Details are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Leung, K.M.; Elashoff, R.M.; Afifi, A.A. Censoring issues in survival analysis. *Annu. Rev. Public Health* **1997**, *18*, 83–104. [CrossRef]
2. Collett, D. *Modelling Survival Data in Medical Research*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015. [CrossRef]
3. Cox, D.R. Regression Models and Life-Tables. In *Springer Series in Statistics*; Springer: New York, NY, USA, 1992; pp. 527–541. [CrossRef]
4. Harrell, F.E. Cox Proportional Hazards Regression Model. In *Regression Modeling Strategies*; Springer: New York, NY, USA, 2001; pp. 465–507. [CrossRef]
5. Bradburn, M.J.; Clark, T.G.; Love, S.B.; Altman, D.G. Survival Analysis Part II: Multivariate data analysis—An introduction to concepts and methods. *Br. J. Cancer* **2003**, *89*, 431–436. [CrossRef] [PubMed]
6. Štěpánek, L.; Habarta, F.; Malá, I.; Marek, L. A random forest-based approach for survival curves comparing: Principles, computational aspects and asymptotic time complexity analysis. In Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Virtual, 2–5 September 2021; IEEE: Piscataway, NJ, USA, 2021. [CrossRef]
7. In, J.; Lee, D.K. Survival analysis: Part II—Applied clinical data analysis. *Korean J. Anesthesiol.* **2019**, *72*, 441–457. [CrossRef] [PubMed]
8. Mehrotra, D.V.; Su, S.C.; Li, X. An efficient alternative to the stratified Cox model analysis. *Stat. Med.* **2012**, *31*, 1849–1856. [CrossRef] [PubMed]
9. Zhang, Z.; Reinikainen, J.; Adeleke, K.A.; Pieterse, M.E.; Groothuis-Oudshoorn, C.G.M. Time-varying covariates and coefficients in Cox regression models. *Ann. Transl. Med.* **2018**, *6*, 121. [CrossRef] [PubMed]
10. Woods, B.S.; Sideris, E.; Palmer, S.; Latimer, N.; Soares, M. Partitioned Survival and State Transition Models for Healthcare Decision Making in Oncology: Where Are We Now? *Value Health* **2020**, *23*, 1613–1621. [CrossRef] [PubMed]
11. Bellera, C.A.; MacGrogan, G.; Debled, M.; de Lara, C.T.; Brouste, V.; Mathoulin-Pélissier, S. Variables with time-varying effects and the Cox model: Some statistical concepts illustrated with a prognostic factor study in breast cancer. *BMC Med. Res. Methodol.* **2010**, *10*, 20. [CrossRef]

12. Hosseini Teshnizi, S.; Ayatollahi, S.M.T. Comparison of Cox Regression and Parametric Models: Application for Assessment of Survival of Pediatric Cases of Acute Leukemia in Southern Iran. *Asian Pac. J. Cancer Prev.* **2017**, *18*, 981–985. [CrossRef] [PubMed]
13. Hoseini, M.; Bahrampour, A.; Mirzaee, M. Comparison of Weibull and Lognormal Cure Models with Cox in the Survival Analysis Of Breast Cancer Patients in Rafsanjan. *J. Res. Health Sci.* **2017**, *17*, e00369.
14. Dumonceaux, R.; Antle, C.E. Discrimination Between the Log-Normal and the Weibull Distributions. *Technometrics* **1973**, *15*, 923–926. [CrossRef]
15. Blackstone, E.H.; Naftel, D.C.; Turner, M.E. The Decomposition of Time-Varying Hazard into Phases, Each Incorporating a Separate Stream of Concomitant Information. *J. Am. Stat. Assoc.* **1986**, *81*, 615–624. [CrossRef]
16. Betensky, R.A.; Mandel, M. Recognizing the problem of delayed entry in time-to-event studies: Better late than never for clinical neuroscientists. *Ann. Neurol.* **2015**, *78*, 839–844. [CrossRef] [PubMed]
17. Walsh, D.P.; Dreitz, V.J.; Heisey, D.M. Integrated survival analysis using an event-time approach in a Bayesian framework. *Ecol. Evol.* **2015**, *5*, 769–780. [CrossRef] [PubMed]
18. Štěpánek, L.; Habarta, F.; Malá, I.; Marek, L.; Pazdírek, F. A Machine-learning Approach to Survival Time-event Predicting: Initial Analyses using Stomach Cancer Data. In Proceedings of the 2020 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 29–30 October 2020; IEEE: Piscataway, NJ, USA, 2020. [CrossRef]
19. Lorena, A.C.; Jacintho, L.F.; Siqueira, M.F.; Giovanni, R.D.; Lohmann, L.G.; de Carvalho, A.C.; Yamamoto, M. Comparing machine learning classifiers in potential distribution modelling. *Expert Syst. Appl.* **2011**, *38*, 5268–5275. [CrossRef]
20. Hu, B.; Guo, H.; Zhou, P.; Shi, Z.L. Characteristics of SARS-CoV-2 and COVID-19. *Nat. Rev. Microbiol.* **2020**, *19*, 141–154. [CrossRef] [PubMed]
21. Wu, Y.C.; Chen, C.S.; Chan, Y.J. The outbreak of COVID-19: An overview. *J. Chin. Med. Assoc.* **2020**, *83*, 217–220. [CrossRef]
22. Adil, M.T.; Rahman, R.; Whitelaw, D.; Jain, V.; Al-Taan, O.; Rashid, F.; Munasinghe, A.; Jambulingam, P. SARS-CoV-2 and the pandemic of COVID-19. *Postgrad. Med. J.* **2020**, *97*, 110–116. [CrossRef]
23. Wei, J.; Pouwels, K.B.; Stoesser, N.; Matthews, P.C.; Diamond, I.; Studley, R.; Rourke, E.; Cook, D.; Bell, J.I.; Newton, J.N.; et al. Antibody responses and correlates of protection in the general population after two doses of the ChAdOx1 or BNT162b2 vaccines. *Nat. Med.* **2022**, *28*, 1072–1082. [CrossRef]
24. Kleinbaum, D.G.; Klein, M. *Survival Analysis*, 3rd ed.; Statistics for Biology and Health; Springer: New York, NY, USA, 2011.
25. Cox, D.R. Partial likelihood. *Biometrika* **1975**, *62*, 269–276. [CrossRef]
26. Chen, M.H.; Ibrahim, J.G.; Shao, Q.M. Maximum likelihood inference for the Cox regression model with applications to missing covariates. *J. Multivar. Anal.* **2009**, *100*, 2018–2030. [CrossRef]
27. Chen, M.H.; Ibrahim, J.G.; Shao, Q.M. Posterior propriety and computation for the Cox regression model with applications to missing covariates. *Biometrika* **2006**, *93*, 791–807. [CrossRef]
28. Chambers, J. *Statistical Models in S*; Chapman & Hall/CRC: Boca Raton, FL, USA, 1992.
29. Albert, A.; Anderson, J.A. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **1984**, *71*, 1–10. [CrossRef]
30. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian Network Classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [CrossRef]
31. Izenman, A.J.; Silverman, B.W. Density Estimation for Statistics and Data Analysis. *J. Am. Stat. Assoc.* **1988**, *83*, 269. [CrossRef]
32. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
33. Wang, J.; Lee, J.; Zhang, C. Kernel Trick Embedded Gaussian Mixture Model. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 159–174. [CrossRef]
34. Breiman, L. *Classification and Regression Trees*; Chapman & Hall: New York, NY, USA, 1993.
35. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Hecht-Nielsen, R. Theory of the backpropagation neural network. In Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, USA, 18–21 June 1989; IEEE: Piscataway, NJ, USA, 1989. [CrossRef]
37. Rojas, R. The Backpropagation Algorithm. In *Neural Networks*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 149–182. [CrossRef]
38. Provost, F.J.; Fawcett, T.; Kohavi, R. The Case against Accuracy Estimation for Comparing Induction Algorithms. In Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, Madison, WI, USA, 24–27 July 1998; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1998; pp. 445–453.
39. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
40. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence—Volume 2, IJCAI'95, Montreal, QC, Canada, 20–25 August 1995; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995; pp. 1137–1143.
41. Knuth, D.E. Big Omicron and big Omega and big Theta. *ACM SIGACT News* **1976**, *8*, 18–24. [CrossRef]
42. Association, W.M. World Medical Association Declaration of Helsinki: Ethical principles for medical research involving human subjects. *JAMA* **2013**, *310*, 2191. [CrossRef]
43. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2021.

44. Medina-Olivares, V.; Calabrese, R.; Crook, J.; Lindgren, F. Joint models for longitudinal and discrete survival data in credit scoring. *Eur. J. Oper. Res.* **2022**, *in press*. [CrossRef]
45. Ishwaran, H.; Kogalur, U.B.; Blackstone, E.H.; Lauer, M.S. Random survival forests. *Ann. Appl. Stat.* **2008**, *2*, 841–860. [CrossRef]
46. Suresh, K.; Severn, C.; Ghosh, D. Survival prediction models: An introduction to discrete-time modeling. *BMC Med. Res. Methodol.* **2022**, *22*, 207. [CrossRef] [PubMed]
47. Spooner, A.; Chen, E.; Sowmya, A.; Sachdev, P.; Kochan, N.A.; Trollor, J.; Brodaty, H. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Sci. Rep.* **2020**, *10*, 20410. [CrossRef]
48. Valvo, P.S. A Bimodal Lognormal Distribution Model for the Prediction of COVID-19 Deaths. *Appl. Sci.* **2020**, *10*, 8500. [CrossRef]
49. Nemati, M.; Ansary, J.; Nemati, N. Machine-Learning Approaches in COVID-19 Survival Analysis and Discharge-Time Likelihood Prediction Using Clinical Data. *Patterns* **2020**, *1*, 100074. [CrossRef]
50. Altini, N.; Brunetti, A.; Mazzoleni, S.; Moncelli, F.; Zagaria, I.; Prencipe, B.; Lorusso, E.; Buonamico, E.; Carpagnano, G.E.; Bavaro, D.F.; et al. Predictive Machine Learning Models and Survival Analysis for COVID-19 Prognosis Based on Hematochemical Parameters. *Sensors* **2021**, *21*, 8503. [CrossRef]
51. Kim, G.; Yoo, C.D.; Yang, S.J. Survival Analysis of COVID-19 Patients With Symptoms Information by Machine Learning Algorithms. *IEEE Access* **2022**, *10*, 62282–62291. [CrossRef]
52. Röltgen, K.; Powell, A.E.; Wirz, O.F.; Stevens, B.A.; Hogan, C.A.; Najeeb, J.; Hunter, M.; Wang, H.; Sahoo, M.K.; Huang, C.; et al. Defining the features and duration of antibody responses to SARS-CoV-2 infection associated with disease severity and outcome. *Sci. Immunol.* **2020**, *5*, eabe0240. [CrossRef]
53. Shirin, T.; Bhuiyan, T.R.; Charles, R.C.; Amin, S.; Bhuiyan, I.; Kawser, Z.; Rahat, A.; Alam, A.N.; Sultana, S.; Aleem, M.A.; et al. Antibody responses after COVID-19 infection in patients who are mildly symptomatic or asymptomatic in Bangladesh. *Int. J. Infect. Dis.* **2020**, *101*, 220–225. [CrossRef]
54. Štěpánek, L.; Magdaléna, J.; Štěpánek, L.; Nakládalová, M.; Boriková, A. The kinetics and predictors of anti-SARS-CoV-2 antibodies up to 8 months after symptomatic COVID-19: A Czech cross-sectional study. *J. Med. Virol.* **2022**, *94*, 3731–3738. [CrossRef]
55. Willette, A.A.; Willette, S.A.; Wang, Q.; Pappas, C.; Klinedinst, B.S.; Le, S.; Larsen, B.; Pollpeter, A.; Li, T.; Brenner, N.; et al. Using machine learning to predict COVID-19 infection and severity risk among 4510 aged adults: A UK Biobank cohort study. *medRxiv* **2020**. [CrossRef] [PubMed]
56. Kurano, M.; Ohmiya, H.; Kishi, Y.; Okada, J.; Nakano, Y.; Yokoyama, R.; Qian, C.; Xia, F.; He, F.; Zheng, L.; et al. Measurement of SARS-CoV-2 Antibody Titers Improves the Prediction Accuracy of COVID-19 Maximum Severity by Machine Learning in Non-Vaccinated Patients. *Front. Immunol.* **2022**, *13*, 811952. [CrossRef]
57. Singh, P.; Ujjainiya, R.; Prakash, S.; Naushin, S.; Sardana, V.; Bhatheja, N.; Singh, A.P.; Barman, J.; Kumar, K.; Gayali, S.; et al. A machine learning-based approach to determine infection status in recipients of BBV152 (Covaxin) whole-virion inactivated SARS-CoV-2 vaccine for serological surveys. *Comput. Biol. Med.* **2022**, *146*, 105419. [CrossRef] [PubMed]
58. Rostami, M.; Oussalah, M. A novel explainable COVID-19 diagnosis method by integration of feature selection with random forest. *Inform. Med. Unlocked* **2022**, *30*, 100941. [CrossRef]
59. De Fátima Cobre, A.; Stremel, D.P.; Noleto, G.R.; Fachi, M.M.; Surek, M.; Wiens, A.; Tonin, F.S.; Pontarolo, R. Diagnosis and prediction of COVID-19 severity: Can biochemical tests and machine learning be used as prognostic indicators? *Comput. Biol. Med.* **2021**, *134*, 104531. [CrossRef] [PubMed]
60. Duan, Z.; Song, P.; Yang, C.; Deng, L.; Jiang, Y.; Deng, F.; Jiang, X.; Chen, Y.; Yang, G.; Ma, Y.; et al. The impact of hyperglycaemic crisis episodes on long-term outcomes for inpatients presenting with acute organ injury: A prospective, multicentre follow-up study. *Front. Endocrinol.* **2022**, *13*, 1057089. [CrossRef]
61. Bhargava, A.; Bansal, A.; Goyal, V. Machine learning-based automatic detection of novel coronavirus (COVID-19) disease. *Multimed. Tools Appl.* **2022**, *81*, 13731–13750. [CrossRef]
62. Kim, J.H. Multicollinearity and misleading statistical results. *Korean J. Anesthesiol.* **2019**, *72*, 558–569. [CrossRef]
63. Song, Y.Y.; Lu, Y. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130–135.
64. Jia, X.; Zhang, P.; Tian, Y.; Wang, J.; Zeng, H.; Wang, J.; Liu, J.; Chen, Z.; Zhang, L.; He, H.; et al. Clinical Significance of an IgM and IgG Test for Diagnosis of Highly Suspected COVID-19. *Front. Med.* **2021**, *8*, 569266. [CrossRef]
65. Gianfrancesco, M.A.; Tamang, S.; Yazdany, J.; Schmajuk, G. Potential Biases in Machine Learning Algorithms Using Electronic Health Record Data. *JAMA Intern. Med.* **2018**, *178*, 1544. [CrossRef]