Predicting the outcome of a PCR test for COVID-19 using a routine blood exam: A replication

An example of how AI can help identify possible infections in a primary care/triage system

By

Lukas Schießer

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Cognitive Science
to the
Institute of Cognitive Science at the Osnabrück University
February 22, 2021

Thesis Supervisor:
M. Sc. Johannes Schrumpf, Institute of Cognitive Science, Osnabrück University
Thesis Supervisor:
Dr. Tobias Thelen, Institute of Cognitive Science, Osnabrück University

Abstract

This thesis replicates a study training different classifiers to predict the outcome of a PCR test for COVID-19 based on a routine blood exam [6]. It reproduces the Random Forest and Logistic Regression classifiers. They were trained using 5-fold nested cross validation and compared to the classifiers' performance in the original paper. The replicated classifiers underperformed when contrasted with the original classifiers. However, their performance was still statistically significant and showed congruency with regards to findings from medical research. By focusing testing efforts on patients with a high probability of having contracted the virus, models like these can assist medical staff during situations where testing capacities are restricted due to shortages in material or personnel.

Acknowledgements

I want to thank...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In December 2019, a novel coronavirus named Sars-CoV-2 emerged in Wuhan, China. It spread rapidly worldwide, leading the World Health Organization (WHO) to declare a global pandemic in March 2020. To date, more than one year post-outbreak, the coronavirus disease COVID-19 caused by the Sars-CoV-2 coronavirus has infected more than 100 million people and has claimed more than 2 million dead [15]. Due to its unprecedented nature and its high ability to spread in the general population, it is crucial to identify infectious people early and isolate them as soon as possible to prevent the spread of the virus. Especially the high proportion of asymptomatic but infectious patients is a major challenge for the containment of this pandemic [32]. The primary method to identify the virus is the reverse polymerase chain reaction (PCR) or reverse transcriptase-PCR (RT-PCR) technique. It identifies the virus by detecting its genetic material in a sample taken by nasopharyngeal swab, which collects nasal secretion from the back of the nose and throat. Unfortunately, the test's execution is very time-consuming and takes at least four to five hours under optimal conditions. Furthermore, the execution requires the use of special equipment and reagents, the involvement of skilled and qualified staff to collect and analyze samples, and relies on the proper genetic conservation of the RNA sequences [45, 12].

Medical staff in emergency departments have to assess the probability of a Sars-CoV-2 infection for each patient during triage and decide who should be tested. In the event of a shortage of testing capabilities, this can be particularly difficult to do. Algorithmic solutions could assist and reduce the uncertainty regarding who should be tested for the disease. There are different approaches to this algorithmic solution. One strategy is based on computer tomography scans or chest X-rays [49]. Models based on chest X-rays are often associated with high rates of false negative

predictions [49, 47]. Whereas models based on CT scans are not advantageous in comparison with a PCR test as they share common requirements like specialized equipment, long duration, and high costs. Another approach is the use of blood tests, which are generally available, deliver results much faster, taking only a minimum of 15 minutes in an emergency setting, and are much more cost-effective. Therefore, this thesis replicates a feasibility study implementing different classifiers to predict the outcome of a PCR test for COVID-19 based on a routine blood exam [6].

Chapter 2

Materials and Methods

2.1 Data

The data used to train the classifiers was provided by Brinati et al. [6]. It was collected between the end of February 2020 and mid of March 2020 from patients admitted to the *IRCSS Ospedale San Raffaele* in Milan, Italy. It consists data from 279 individuals who were selected randomly. For each individual, the data set provides their age, gender, the results of a routine blood screening and a PCR test for Sars-CoV-2. A complete overview of the recorded variables is provided in Table A.1. The target variable *Swab* is binary and indicates the result of a PCR-test for Sars-CoV-2 taken by a nasopharyngeal swab. 0 indicates a negative test, while 1 indicates a positive test. The data set is slightly imbalanced towards positive cases, with 102 (37%) negative cases and 177 (63%) positive cases.

The data was cleaned and wrangled in order to analyze it. Since the variable *Gender* was provided as a string, it was transformed into two binary numerical variables called *female* and *male* by one-hot encoding. Further, two values of the variable *Age* were replaced with a missing value indicator, specifically the age 0 and 1. This was sensible, seeing that there was no other data recorded from minors. Thus these two values can be presumed to be input errors during the collection process. Further, the variable *Lymphocytes* had to be recast to a numeric format because there was an input error for one of its values rendering the column's contents a string instead of a numerical value. This process created one missing value in place of the erroneous value.

Table A.4 provides standard statistics for the data set's numerical features, specifically their measurement unit, mean, standard deviation, and median. These standard statistics, combined with the kernel density plots provided in Figure B.1, show

that all features are skewed relative to a (standard) normal distribution. Blood values are slanted positively, exhibiting a right-tailed distribution because, for most blood values, a lower or centered value is a sign of better health. Age is skewed negatively, exhibiting a left-tailed distribution since COVID-19 affected older individuals more severely than younger individuals, especially in Italy [35].
In order to confirm the observation of non-normality, a Shapiro-Wilk test for normality was performed. The test probes the null-hypothesis that the samples of a variable come from a normal distribution. If the test's p-value is smaller than 0.05, a commonly chosen α-level, the null-hypothesis is rejected, meaning the data is not normally distributed [43]. The test results are provided in Table A.5 and show that all of the numerical variables are non-normally distributed. This observation will be important during the assessment of the imputation method.
Table A.3 shows that most features have missing values. 196 samples have at least one feature missing, which amounts to 70% of the data. Due to the data set's small size, it is not feasible to exclude these individuals from the analysis process. It is rather more constructive to use an imputation method that models the missing values based on the data set's observed values. Therefore, Brinati et al. chose to use *Multivariate Imputation by Chained Equations*.

2.2 Multivariate Imputation by Chained Equations

Multivariate Imputation by Chained Equations, or *MICE* for short, is an imputation method proposed by Buuren and Groothuis-Oudshoorn [8]; it is also known as *fully conditional specification* (FCS). MICE is a method that imputes missing data by estimating a set of possible values from distributions of observed data. Each variable with missing data x_k is regressed separately on the set of all other variables $x_1, ..., x_n$. This set is restricted to occurrences where the incomplete variable x_k has observed values. MICE assumes that the data are missing at random (MAR). That is, the probability of missing data does not depend on the unobserved data but is only dependent (conditional) on the observed data. Due to its individualistic approach, MICE can handle variables of different types using different modeling choices for different variable types. Such as linear regression for normally distributed continuous variables, logistic regression for binary variables, multinomial logistic regression for categorical variables with more than two classes, and predictive mean matching for skewed variables or variables that do not fit any of the types mentioned before [48, 4].

The imputation process is based on the following four main steps [48, 4]: Firstly, all missing values are imputed using a simple imputation method (e.g., mean imputation). These imputations can be thought of as "place holders" used during the first modeling phase. During Step 2, the "place holder" imputations for one variable x_k are set back to missing. In Step 3, all observed values from variable x_k are regressed on the imputation model's other variables. Since this is the model building phase, this step only uses samples where x_k has observed values. Therefore, x_k is the dependent variable, and all other variables are independent variables used in the regression model. In Step 4, the missing values in x_k are replaced with imputations (predictions) from the regression model built in Step 3. All values of x_k, the observed and the imputed values, are then used in subsequent regression models of other variables.
Steps 2-4 are repeated for every variable with missing data. After the algorithm cycled through all variables, one iteration is completed. Steps 2-4 are repeated for a user-specified number of cycles (iterations). Generally, ten to twenty cycles should suffice to stabilize the imputation results, indicating that the parameters controlling the imputations should have converged by then. This imputation process is usually repeated m times, creating m slightly differently imputed data sets, which are then used in the subsequent analysis. According to [7, 4, 48], already a small number of imputed data sets, usually three to ten, is sufficient to provide sensible results during analysis. It can be advantageous to use a higher number of imputed data sets to get a broader range of estimates. However, setting m higher requires more computations and storage and may not be worth preoccupying these extra resources [7].

2.3 Classifiers

The original paper implements eight classifiers, namely Random Forest, Logistic Regression, Decision Tree, K-nearest Neighbors, Naive Bayes, Support Vector Machine, Extremely Randomized Trees, and three-way Random Forest [6]. Here, I will reproduce the Random Forest, Logistic Regression, and Decision Tree classifiers. The following section introduces them.

2.3.1 Decision Tree

Since Decision Trees are the basic building block of Random Forests, it is necessary to take a closer look at them [5]. Consequently, the use of Random Forests will be motivated. Moreover, a Decision Tree will be used during the validation phase to make the Random Forest model more intuitive. A Decision Tree has a flowchart-like structure where each internal (decision) node tests an attribute. Each branch corresponds to one attribute value, and each leaf node represents a classification. It is built up using the ID3 algorithm [40].
ID3 tries to determine the most informative attribute of a given data set given the distribution of its values. This attribute is then used as a root of the Decision Tree, and a branch is created for every value this attribute can take, which also creates a subset of the data set that only has the attribute value of the branch. This process is repeated for every branch with the remaining subset of the training set until a leaf node is reached [38].
ID3 can use many different measures to decide which is the most informative attribute. Two of the most popular ones are *Information Gain* and the *Gini Impurity*. Information Gain uses the *entropy* measure to compute the impurity, also known as an attribute's heterogeneity. Entropy originates from information theory and describes the average information content of an attribute's possible outcome. It is calculated, as shown in Equation 2.1, where S is a set of samples and C are the possible outcome labels of the target variable. A high entropy represents a high average information content in an attribute [21]. Information Gain measures the expected reduction in entropy of a set S caused by learning the state of a random attribute A. It is calculated, as shown in Equation 2.2, and can be described as the difference between the entropy of a set S and the weighted average of the child entropies $E(S_v)$ where S_v is a subset of S where the attribute A has the value v. This is computed for all attributes, the attribute that maximizes Information Gain is then selected as a new node [38].

$$E(S) = \sum_{i=1}^{C} -p(i)\ log_2(p(i)) \tag{2.1}$$

$$Gain(S, A) = E(S) - \sum_{v}^{V} \frac{|S_v|}{S} E(S_v) \tag{2.2}$$

The Gini Impurity or Gini index measures the impurity in a set S that is the probability of incorrect classification of a random sample. It is calculated as shown in Equation 2.3. The measure, which the algorithm uses to determine which attribute is chosen as a node, is called the Gini gain. It is calculated as shown in Equation 2.4

and can be described as the average Gini Impurity. The attribute with the lowest value is selected for the Gini gain since we want to minimize the attribute's incorrect classification [40].

$$G = 1 - \sum_{i=1}^{C} p(i)^2 \tag{2.3}$$

$$Gini(S, A) = \sum_{i=1}^{C} \frac{|S_i|}{|S|} G(S_i) \tag{2.4}$$

ID3 is not able to process continuous features since it needs categorical attributes. Here, an extension called *C4.5* is consulted. C4.5 obtains a categorical attribute from the continuous valued attribute by applying thresholds to it. The thresholds are chosen such that the information gain of the attribute is maximized. Initially, all attribute values are sorted according to their values, either in descending or ascending order. The algorithm then looks for "boundaries" within the values where the classification (target variable changes); the found "boundaries" are then used as thresholds. The information gain is calculated for each possible threshold. The threshold with the highest information gain is then used to split the samples in order to obtain a categorical attribute. The resulting categorical attribute is then used to build the Decision Tree further [36, 37].

2.3.2 Random Forest

The Random Forest Classifier is an ensemble classifier that uses multiple Decision Tree instances to classify the given data. It uses two methods to diversify the different classifications of the Decision Tree instances called *tree bagging* and *feature bagging* [5].

During tree bagging, the Decision Tree is not trained on the whole training set but a subset of the training set. The subsets are generated by sampling the original data set with replacement. A certain percentage of samples is selected from the original data set, and the remaining percentage are duplicates of the already selected samples. This process generates new bagged training sets S_i with the same size as the original training set S. If the bagged set size is smaller than the size of the original set, this process is called sub-bagging [5, 42]. The generated subsets are then used to build the Decision Trees.

Feature bagging or feature subset selection limits the number of features the individual Decision Trees can use at each new split in consideration. For each decision node consideration, several features are randomly selected and presented to the Decision Tree algorithm. A hyperparameter controls the number of features available

to the algorithm. For classification problems, this is usually set to $\sqrt{p}$ where p is the number of features present in the complete data set [5, 3].
Each Decision Tree in the forest generates a classification. A majority vote over all Decision Trees determines the final classification of the Random Forest. Using tree and feature bagging, Random Forests average over all individual Decision Tree models, reducing the classification variance and avoiding overfitting [3].

2.3.3 Logistic Regression

Logistic Regression is a modification of linear regression that models the probability $p(X)$ that a sample X belongs to a specific category, e.g., zero for a negative case and one for a positive case. To do so, it uses a special form of the sigmoid function called the logistic function where $\mathbf{X}$ is a vector or matrix of samples and $\vec{\beta}$ is a vector of the coefficients [42, 3]:

$$p(\mathbf{X}) = \phi_{sig}(\mathbf{X}) = \frac{e^{\mathbf{X}\vec{\beta}}}{1 + e^{\mathbf{X}\vec{\beta}}} = \frac{1}{1 + e^{-\mathbf{X}\vec{\beta}}} \tag{2.5}$$

After taking the inverse of the logistic function and exponentiating both sides, we can see

$$\frac{p(\mathbf{X})}{1 - p(\mathbf{X})} = e^{\mathbf{X}\vec{\beta}} \tag{2.6}$$

The left-hand side of Equation 2.6 is called odds. It can take a value between 0 and ∞ indicating very low or very high probabilities of having a particular category. After applying the logarithm to both sides of Equation 2.6, the log-odds or logit is obtained (Equation 2.7), which is linear in X [30, 3].

$$ln\Big(\frac{p(\mathbf{X})}{1 - p(\mathbf{X})}\Big) = \mathbf{X}\vec{\beta} \tag{2.7}$$

Since β_0 and all β_i of the vector $\vec{\beta}$ are unknown, they must be estimated on the available training data. The method that is used to achieve this is called maximum likelihood estimation (MLE). MLE tries to find estimates for the coefficients β such that the estimated probability $\hat{p}(x_i)$ of having a particular category for each individual observation x_i is as close as possible to the individual's observed category. In other words, MLE tries to find estimates for the coefficients β of the model $P(\mathbf{X})$ described in Equation 2.5 such that the resulting probability is close to one if the individual's observed category is a positive case and close to zero if the individual's label is a negative case. This can be formalized as a likelihood function, as shown in Equation 2.8. The estimates for β are then chosen to maximize the likelihood

function [3].

$$L(\vec{\beta}|\vec{y}, \mathbf{X}) = \prod_i p(x_i)^{y_i}(1 - p(x_i))^{(1-y_i)} \tag{2.8}$$

2.4 Model Training

The Random Forest and Logistic Regression are trained using 5-fold nested cross validation. In each outer fold of the nested cross validation, the data is imputed using MICE and then split into training and test data according to a 4:1 ratio. Since MICE usually generates several imputed data sets to account for the uncertainty in the imputation process, there are five different models for each classifier per fold. The cross validation's inner fold is used to determine the best hyperparameter combination for the classifiers using *Grid Search*, also called *hyperparameter tuning* [34]. Figure B.2 explains this process in further detail. The parameters for the Random Forest are:

- Number of trees in the forest: [10, 100, 500]
- Number of features to consider looking for the best split: [4, 8, 12, 16]
- Quality of split measure: Gini Impurity or Information Gain

For Logistic Regression, Grid Search seeks for an optimum among the following hyperparameters:

- Inverse of regularization strength C: [0.001, 0.01, 0.1, 1, 10, 100, 1000]
- Penalty: L1 or L2 regularization
- Algorithm to use during optimization problem: liblinear or saga
- Maximal iteration: [200, 300, 400, 500]

The remaining hyperparameters use the default values specified in the documentation of `sklearn` for Random Forests and Logistic Regression [34]. Grid Search selects the best hyperparameters according to the highest accuracy value.
After the training, the models are evaluated using accuracy, balanced accuracy, sensitivity, specificity, positive predictive value, and area under the ROC curve, which are explained in the next section. The hyperparameters of the best performing models for Random Forest and Logistic Regression are then used to obtain the final

classifiers. For the validation of the final classifiers, the whole data set is split into training and validation data according to an 80:20 ratio. The data set is imputed before the split due to how the `mice()` package runs in R. The classifiers are trained and validated using the split validation data. In addition to the evaluation metrics above, the final classifiers are also evaluated using the ROC curve. Additionally, the precision-recall curve is used to assess the classifiers' performance regarding the slight imbalance in the data set. The Decision Tree is only trained and validated during the validation phase since it is mainly used to provide an interpretable overview of the Random Forest's classification reasoning. It is trained using the standard hyperparameters according to the sklearn implementation [34].
To implement this procedure, the Python libraries `scikit-learn`, `pandas`, and `numpy` are used. To implement the imputation method, the R library `mice()` is used and connected to the Python implementation using a bridge package called `rpy2`. The R implementation of the imputation method is used instead of a Python implementation since it provides more functions to inspect the MICE procedure and since it can handle non-normally distributed data. The libraries `seaborn` and `matplotlib` are used to visualize the results.

2.5 Evaluation Metrics

This section introduces and defines the metrics used to evaluate and validate the classifiers. The following terms will be used in the equations to describe them: True Positive, False Positive, False Negative, and True Negative. They are further characterized in Table 2.1.

		True condition	
		Condition positive	Condition negative
Predicted condition	Condition predicted positive	True Positive (TP)	False Positive (FP)
	Condition predicted negative	False Negative (FN)	True Negative (TN)

Table 2.1: Confusion matrix for binary classification

Accuracy describes the closeness of a predicted label to its true label. It is calculated as shown in Equation 2.9. Accuracy alone is not a sensible measure to determine the goodness of fit for a model since the model can be affected by imbalanced data or poor parameter initialization. Accordingly, other measures should be considered

during model selection [21].

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.9}$$

Balanced accuracy is used in addition to the average accuracy. Balanced accuracy is often chosen when a data set is imbalanced to investigate the effect of the imbalance on the classifier. If the balanced accuracy differs significantly from the average accuracy, the imbalance affects the classifier regarding class prevalence. Consequently, it is biased against one of the outcomes of the target variable. The balanced accuracy is defined as the average of sensitivity and specificity [21, 6].

$$balanced\ accuracy = \frac{\frac{TP}{TP+FN}\frac{TN}{TN+FP}}{2} \tag{2.10}$$

In the context of this data set, sensitivity is interpreted as the proportion of people infected with a disease who will test positive for this disease. It is calculated as shown in Equation 2.11. Since it is only calculated using the part of the population with the disease, sensitivity can only give evidence about the true positive rate and not the test's false positive rate [1].

$$sensitivity = \frac{TP}{TP + FN} \tag{2.11}$$

Specificity can be defined as the proportion of people without the disease that are identified correctly, i.e., who tested negative for the disease. It is calculated as shown in Equation 2.12. Specificity only gives information about the proportion of people without the disease and a negative test, i.e. the false positive rate, and cannot disclose anything about the false negative rate [1].

$$specificity = \frac{TN}{TN + FP} \tag{2.12}$$

In a medical setting, the positive predictive value (PPV) measures the post-test probability of the disease given a positive test. Therefore, it refers to the probability that a subject with a positive test has indeed the disease. It is calculated as shown in Equation 2.13 [1].

$$PPV = \frac{TP}{TP + FP} \tag{2.13}$$

The receiver operating characteristic curve (ROC curve) is a commonly used metric to determine a model's performance describing the trade-off between specificity and sensitivity. It is calculated using the prediction scores of a model that are either discriminant values or posterior probabilities [41]. In fact, most models do not produce a classification label as their output but rather a prediction score that is then

thresholded to correspond to one of the target variable levels. The metrics mentioned above are also called *single-threshold* metrics since they are defined only for the classifier's thresholded output [41, 21]. Since ROC uses the prediction scores instead of a mdoel's thresholded output, it can compute the sensitivity and specificity for every sensible threshold. The resulting values are then plotted with sensitivity or true positive rate as the vertical axis and the false positive rate or 1 - specificity as the horizontal axis. The points are interpolated linearly to create the ROC curve [41]. Usually, a reference line is added to the plot illustrating a model's performance that only makes random predictions. The curve for the trained model is expected to be above the reference line at all times. A model with perfect performance will appear in the top left-hand corner of the ROC space where the sensitivity is one, and the false positive rate is zero [21, 16].
The area under the ROC curve can also be measured and is called either AUC for area under the ROC curve, or ROC index. It can give a numeric summary of the ROC curve. It is calculated using the ROC curve's integral, which can be easily done using the trapezoidal method since the ROC curve is discrete and stepped. A large AUC value indicates a better model performance. As a heuristic, an AUC over 0.7 indicates a strong predictive model, and an AUC below 0.6 indicates a weak model. AUC and ROC both are quite robust against imbalanced data, but other threshold-free metrics can be used in this case as well [21, 16].

The Precision-Recall curve (PRC) is another threshold-free metric that is especially useful when dealing with an imbalanced data set. It calculates the precision (or positive predicitive value) and recall (or sensitivity) values for each feasible threshold. The resulting values are then plotted with precision as the y-axis and recall as the x-axis. The points are interpolated non-linearly to create a curve. In contrast to the ROC curve, which uses a fixed reference line, the PRC's baseline is determined by the ratio of positive (P) and negative (N) samples in the data set and represented as in Equation 2.14. Each point in the PRC plot corresponds to exactly one point in the ROC plot since both plots chart values for sensitivity. The area under the curve can also be calculated for the precision-recall curve similar to the AUC of the ROC curve and is then denoted by AUC (PRC) or average precision. Similarly, a large value for the AUC (PRC) indicates a strong model, and a lower value indicates a weak model [13, 41].

$$y = \frac{P}{P + N} \tag{2.14}$$

Chapter 3

Results

This chapter presents the results of the reproduction of the classifiers according to the procedure described in Chapter 2. In addition, the performance of the reproduced classifiers and the classifiers are from the original work [6] are compared.

3.1 Results of the Reproduction

Table A.6 and A.7 show the mean and standard deviation for models' the accuracy and balanced accuracy during nested cross-validation. The authors of the original paper provided a 95% confidence interval for these two measures. Since there is no information on how they computed this interval, it is assumed that they used a two-sided Student's t-distribution, which is used to estimate the mean of normally distributed populations where the sample size is small. It is calculated using Equation 3.1 where $t_{\alpha,v}$ is the t-value which can be taken from a table with α as the degrees of freedom $n-1$ and v as the confidence level, s is the sample variance, $\bar{x}$ is the sample mean, and n is the number of samples [19]. For the Random Forest, the 95% confidence interval of the average accuracy is [0.76, 0.80], andof the average balanced accuracy is [0.73, 0.77]. For Logistic Regression, the confidence interval of the average accuracy is [0.72, 0.77], and of the average balanced accuracy is [0.68, 0,72]. Since the confidence intervals of the accuracy and balanced accuracy for both classifiers overlap or are near each other, it can be assumed that the slight imbalance of the data set had no significant effect on the classifiers and that the classifiers, therefore, are not biased against the outcomes of the target variable. All

evaluation metrics for each fold and imputed data set are provided in Table A.9.

$$\bar{x} \pm \frac{t_{\alpha,v} s}{\sqrt{n}} \tag{3.1}$$

The best models were selected by searching for the model with the highest evaluation measures. The best Random Forest Classifier used maximally four features, 500 estimators, and Gini impurity. It achieved an accuracy of 87.5% and a sensitivity of 90%. The best Logistic Regression Classifier used a C of 10, a maximal iteration of 200, the 'l2' penalty, and the 'liblinear' solver. It achieved an accuracy of 83.9% and a sensitivity of 85%. These settings were then used to retrain the classifiers and compute the validation metrics.
The Random Forest Classifier is also the best performing classifier after retraining it with the best hyperparameter with an accuracy of 76.8% and a balanced accuracy of 73.7%. The Logistic Regression achieves an accuracy of 73.2% and a balanced accuracy of 69.4%. Accordingly, there is no significant difference between the accuracy and balanced accuracy for all three classifiers, meaning no classifier is biased. The metrics mentioned above are lower than the evaluation metrics since the data set used to train during the validation phase is different from the one used during the evaluation phase due to the way MICE imputes missing data.
All single-threshold metrics are provided in Table A.8. The two threshold-free metrics, ROC and PRC curve, are shown in Figure B.3. Both plots show that the classifiers perform better than the reference line. The values for the area under the ROC and under the PRC curve suggest that Logistic Regression (AUC: 80.8%, AUC (PRC): 85.5%) performs better than the Random Forest (AUC: 80.5%, AUC (PRC): 82.9%) although all other metrics indicate better performance of the latter. It should be noted that the difference between the AUC scores for both curves is minimal and is not meaningful enough to declare the Logistic Regression as the better classifier. The more apparent differences in the AUC (PRC) for both classifiers, on the other hand, could indicate that the Logistic Regression is less affected by the imbalance in the data set than the Random Forest is.
The Decision Tree Classifier exhibits a lower discriminative performance, as indicated by the AUC and AUC (PRC) with 68.5% and 69.8% indicating a relatively weak classifier. Additionally, it only achieves an accuracy of 71.4% and a balanced accuracy of 68.5%. Despite these low metrics, the Decision Tree can still be used to help clinicians make quick decisions and interpret the result of the Random Forest.

Figure B.4 shows the feature importance plots for the three classifiers. The feature importance for the Decision Tree and Random Forest was obtained using the sklearn implementation. The implementation calculates the feature importance by

quantifying the mean decrease in impurity. It should be noted that impurity-based feature importance can suffer from favoring features with a high number of unique values and from making predictions based on statistics derived from the training set. Therefore, this method is not necessarily informative about whether a feature makes a good prediction or not [23]. For Logistic Regression, the feature importance is inferred by looking at the each feature's coefficients in log space. A positive coefficient can be interpreted as contributing to a positive (1) classification, and a negative coefficient can be interpreted as contributing to a negative (0) classification [30]. It is difficult to compare non tree-based feature importance with tree-based importance because they have different scales and calculations. Nontheless, the absolute values of the coefficients in log space can be compared with the impurity-based importance features of the tree based-methods.
Both tree-based methods show a similar feature importance plot, with AST being the most important feature. The remaining features are differently distributed but mostly exhibit similar values. Consequently, the Decision Tree and its visualization (Figure B.5), combined with the feature importance plots for both tree-based classifiers, can be used to give some rough insights into the Random Forest classification procedure. Section 4.1.2 will take a closer look at the feature importance of the three classifiers and compare them to findings from medical research.

3.2 Comparison with the Original Paper

The 95% confidence intervals for the accuracy and balanced accuracy of this replication overlap with the ones from the original paper. The average accuracy for the Random Forest in the original is [0.74, 0.80] (replication: [0.76, 0.80]) and [0.70, 0.81] (replication: [0.72, 0.77]) for Logistic Regression. The average balanced accuracy in the original work is [0.70, 0.82] (replication: [0.73, 0.77]) for Random Forest and [0.65, 0.74] (replication: [0.68, 0.72]) for Logistic Regression. Since there is no information about individual values during the nested cross validation and it is also not reported how the authors computed the 95% confidence intervals, it can be difficult to compare them. Nevertheless, the overlap of the intervals suggests that the classifiers can be compared to a certain degree. Nevertheless, the intervals' overlap suggests that the classifiers have similar discriminatory abilities and are therefore comparable. Unfortunately, there is no more information about other evaluation metrics.
The validation metrics of the final classifiers show lower performance than the classifiers in the original paper. In this replication, the Random Forest has an accuracy of

0.77, and the Logistic Regression an accuracy of 0.73. In the original work, the Random Forest achieves an accuracy of 0.82, and the Logistic Regression an accuracy of 0.78. Though the replicated classifiers perform worse than the original ones, their accuracies exhibit a similar difference of 0.04. Further, all single-threshold metrics except for the AUC are higher for the Random Forest in the replication. The sensitivity is equal for both classifiers in the reproduction. In contrast, in the original work, only accuracy and specificity are higher for the Random Forest, AUC and sensitivity are higher for the Logistic Regression. When setting aside the sensitivity, the classifiers show a similar pattern where accuracy and specificity are better for the Random Forest and AUC is better for the Logistic Regression. Therefore, the replicated classifiers show a similar but less accurate performance than classifiers in the original work. The ROC plot for both classifiers (Figure B.3a) exhibits a similar course as in the original paper since all classifiers operate above the reference line (chance). Still, the higher AUC of the original classifiers is also reflected in the plot because they reach further to the upper left corner. Seeing that the precision-recall curve only plots the Random Forest's performance, we can only compare these results. The replication (Figure B.3b) and the original differ considerably in this plot. The replicated classifier has a more abrupt course due to fewer computed thresholds. Nonetheless, both curves exhibit similar behavior as both curves drop rapidly in the interval recall = [0.0, 0.2], suggesting a reduced classification reate that improves afterward. Unfortunately, there was no information regarding the validation metrics of the Decision Tree classifier. Nevertheless, the reproduced classifier's validation metrics can be compared with the cross-validation's 95% confidence intervals of the original. The reproduced Decision Tree achieves an accuracy of 0.71, which lies in the confidence interval [0.70, 0.78], and a balanced accuracy of 0.69, which lies in the confidence interval [0.64, 0.71].

Chapter 4

Discussion

4.1 Discussion of Results

4.1.1 Statistical Validity

Although the classifiers perform worse than those in the original paper, most recorded metrics still certify them a good discriminative performance. With a sensitivity of 90% for both models and accuracy values between 73% and 76%, the classifiers can still serve as additional decision tools to help physicians make decisions about allocating testing resources based on their predictions and other indicators. A high sensitivity is preferable in this scenario since it makes a negative test result more meaningful than a positive result. A test with high sensitivity rarely misdiagnoses patients who have the disease and thereby is a useful tool to rule out the disease. This is preferable because we want to sort out the patients who are not infected and only include those where we are sure or not certain if they contracted the disease [1]. Further, the AUC and AUC (PRC) values, which are over 80% for both classifiers, indicate that they are indeed effective classifiers [21]. Due to the slight imbalance of the data set, it is favorable to examine the classifiers regarding this potential source of bias. If there were a significant difference between the accuracy and the balanced accuracy for a classifier, it would entail a bias against one of the target outcomes. Moreover, the high AUC (PRC) values suggest that the classifiers are not significantly affected by the data's imbalance. The high sensitivity signals that 90% of patients with the disease will get a positive test result and will be correctly identified. While the specificity is comparatively low, it does not necessarily pose a problem. A low specificity in combination with a high sensitivity indicates a higher

false positive rate than a high false negative rate [1]. Therefore, if the classifier's result is negative, it will be safe to assume that a PCR test for COVID-19 will also be negative, and thus the patient will not have the disease. This is more desirable because it is better to be over cautious and recommend more tests.

4.1.2 Theoretical Validity

Since the question regarding the classifiers' relevance has been clarified, this paragraph deals with the classifiers' internal classification reasoning and their accordance with findings from medical studies. When inspecting the feature importance plots for the tree-based classifiers, both plots (Figure B.4a and B.4b) show that the AST, WBC, CRP, and lymphocyte counts are among the five most important features. The 5 most important feature also includes the LDH count for the Random Forest and the age variable for the Decision Tree. For the Logistic Regression (Figure B.4c), the eosinophil, WBC, basophil, lymphocyte counts, and gender are the most important features. All feature importance measures have the WBC and lymphocyte count in common. Multiple studies also show that these two blood values are significant indicators for a COVID-19 infection [17, 44, 24]. The WBC or white blood cell count refers to the actual number of white blood cells per volume of blood, while the lymphocytes are a subgroup of white blood cells involved in eliciting an immune response to foreign agents [28, 39]. During an infection with Sars-CoV-2, the WBC and lymphocyte counts will decrease since this is a sign of a viral infection and the body's response to it [17, 24, 44]. AST or aspartate aminotransferase, the most important feature in the tree-based methods, is an enzyme mainly found in the heart and liver; its level increases when the muscles of these organs are injured [26, 39]. This is also the case during a COVID-19 infection since the virus attacks not only the upper respiratory tracts but the whole body, including the liver and heart [46]. CRP or C-reactive protein is another substance found during a COVID-19 infection. It is produced in the liver and discharged after tissue damage, the start of an infection, or other inflammatory causes. Increased volumes of this protein are often the first indication of an infection or inflammation in the body [27, 39]. Elevated levels of this protein can also be observed during a COVID-19 infection since it is an infection that primarily targets lung tissue [14, 17]. LDH or lactate dehydrogenase is an enzyme involved in metabolic cycles for energy production. It is present in almost all cells in the body, especially in the heart, liver, lungs, kidneys, muscles, and blood cells. An increase in LDH can indicate acute kidney or liver disease, hypoxia, or heart and lung infarction [29, 39]. According to [17, 14], a blood test reveals elevated levels of LDH because Sars-CoV-2 primarily attacks the

upper respiratory tracts, which leads to lung damage and the discharge of LDH into the bloodstream. Age as the fourth important feature in the Decision Tree can also be a predictor for a positive result. Poletti et al. [35] determine that individuals older than 70 years are more susceptible to a severe course of the disease or even death. Further, the study revealed a higher risk for male individuals, one of the most important features in Logistic Regression. For eosinophils and basophils, [17, 44] reveal a significant decrease for patients with a positive test result. However, [17] notes that these differences might not have any clinical implication during diagnosis since the count in healthy individuals is also rather low and exhibits a large variability. In conclusion, the comparison of every classifier's most important features with the findings from medical research reveals that the models use features that medical research deemed as significant to identify patients with COVID-19. Thereby, they can be considered valid regarding their reasoning and classification in a theoretical or laboratory setting.

4.1.3 Real-World validation

The two paragraphs above certify the models' statistical and theoretical validity, but they lack evidence that these models will also benefit medical personnel in the real world. Relational and ecological validity inspect how a model impacts the clinical workflow and the overall social environment. Specifically, relational validity refers to which extent different user groups, e.g., medical workers or patients, can relate to the model, i.e., how much they trust the model's predictions [11]. Users can either under-rely on the model, ignoring all the predictions presented to them, or over-rely on the model accepting all predictions without questioning them. The former is disadvantageous because, in this case, the model would not have any benefit in a clinical setting. The latter could be seen as detrimental as this can lead to biased decision making or automation bias, affecting patients negatively [11, 9]. Another factor contributing to relational validity is the explainability of the models and their classification reasoning, which will be discussed in Section 4.2.2.
Ecological validity investigates the impact of the technology, further scrutinizing the intersection of the clinical and social settings in which a model is deployed. A simple way to assess this type of validity is to compare performance, outcome and practice oriented measures of different medical teams utilizing the model and relying on traditional technology in the same setting [9]. Moreover, relational and ecological validity also assesses the sustainability of a model by continuously examining the model's effect over time. As Parikh, Obermeyer, and Navathe [33] note, "unlike a drug or device, algorithms are not static products [as] their inputs [...] can change

with context". Accordingly, AI models should be assessed periodically, examining if they continue to provide net benefits comparable to earlier times [9]. Therefore, these kinds of validities need to be investigated in future research.

4.2 Discussion of Methods

The discussion of the methods is mainly focused on the imputation method MICE since it is a fairly new technique and is not commonly used to impute missing values.

4.2.1 MICE

Assessment of the Imputation

To assess the imputation procedure, this section will look at the method the imputation model used to impute the missing values and how the imputations compare to the original (non-imputed) data. The `mice()` package in R provides functions to inspect the imputation model used after the imputation is done. Since MICE can handle variables of different types and distributions and due to its individualistic approach, every variable can utilize a different model type to impute the missing values. In our case, the imputation method used the same model type for all variables as a consequence of the non-normality of the data called *Predictive Mean Matching* (PMM). PMM imputes missing values by using a small subset of complete observations called the *donor pool*. The donor pool contains three to ten donor candidates that exhibit the same or similar observed values compared to the missing entry's predicted value. One donor is randomly selected from the candidates, and the observed value of the donor is then used to replace the missing value [7, 2, 31]. It is usually used to model data where the assumptions of normality and homoscedasticity are in question, as is the case for the data used in this thesis [31].
The model's imputations can be assessed by generating summary statistics for the observed and the imputed values separately and jointly for each variable. Especially comparing the differences in mean and standard deviation between the observed and imputed values of each variable can help identify variables of concern [4]. Thus, summary tables presenting the mean and standard deviation for the observed and imputed data were created. Table A.10 includes all samples, the observed and imputed, of the imputed data sets, while Table A.11 only includes samples with imputed values. Additionally, kernel density plots comparing the observed and im-

puted data sets were created. Figure B.6 and Figure B.7 plot the observed data versus all samples of the imputed sets or only the imputed samples of each imputed set, respectively. Most variables exhibit similar values for the summary statistics when comparing the mean of the observed data with the mean of all samples of the imputed sets. Only ALP, GGT, LDH, and Neutrophils exhibit apparent deviations from the observed data. These deviations are also noticeable in Figure B.6. It should be pointed out that most deviations are not significant. There are apparent deviations in every variable when comparing the observed data's summary statistics and only the imputed samples for each data set. However, it should be noted that with 70% of samples missing at least one feature, it is hard to compare the statistics of the observed values with the imputed ones. Especially since 117 of the 196 missing samples have a negative test result, which amounts to almost 60% of the missing samples and 42% of all samples, and only 79 of the missing samples have a positive test result. This imbalance is also reflected in the samples where all features are present. Out of 83 complete samples, only 23 samples have a negative test result, and 60 have a positive result. Although the target variable was not available to the imputation model at the time of imputation, the target's characteristics might have transpired in the data. Since MICE operates on the data's characteristics, this imbalance could be reflected in the imputation model. As reported in the section above, the values for lymphocytes, eosinophils, and basophils decrease in patients infected with COVID-19 [17, 22]. When inspecting the summary statistics of the imputed sets with this in mind, the lower mean and standard deviation are more sensible. It should also be noted that ALP and GGT have the highest missing value proportion. Since MICE has fewer data to base its imputations on, it can be expected that the imputations exhibit apparent deviations from the observed values. MICE also accounts for the uncertainty during the imputation process, meaning a higher number of missing values represents a higher uncertainty. Therefore the imputation model cannot make as educated predictions as for other variables with more observed data [4, 48].

In conclusion, it is tough to make statements about the goodness of the imputation model. Most imputed sets do not exhibit drastic deviations from the observed set of observations. The variables that show apparent deviations can be ascribed to an imbalance when looking at the samples with missing values. Further, there is no standard procedure to assess MICE's imputations and can therefore be a very subjective procedure. Intuitively, the imputations still seem like a rasonably good approximation of the data and are therefore appropriate to use during the models' training. For further research, I would nonetheless recommend using another method to impute missing values that is more commonly used and has a theoretical basis. For example, Brinati et al. use K-nearest neighbor imputation in a subsequent

study [10], which uses a more extensive data set than the one in [6].

Discussion of the Implementation

The implementation of MICE used to impute the missing values is provided in the R package `mice()` since the `sklearn` implementation could not impute non-normal data correctly at the time. The R implementation, though much better suited, has its own limitation. Specifically, it cannot apply the imputation model to other data but only the data it is fitted. Usually, when using cross-validation the data is split into training and testing data before the imputation is done. The imputation method is then trained on the training data, and the trained imputation model is applied to the training and the testing data. Using this process, the risk of *data leakage* is lowered. Data leakage is a term in machine learning that describes a model's contamination with information from outside the training data where the model could learn something that it otherwise could not [20]. In the context of this thesis, data leakage could take place during the imputation process. Due to the limitation of the R implementation, the whole data set is used to fit the imputation model because it is not possible to apply the model to different data sets. Therefore, test data's characteristics are used to build the imputation for the training data and vice versa. However, the data is split into training and test data after the imputation, consequently lowering the risk of leakage. Moreover, it could be argued that it is not necessarily data leakage as no test data is used to train the classifiers themselves but only to perform the imputation. Even if it had been possible to fit the imputation model only on the training data and apply it to both the training and test data, the appropriateness would have been questionable. As MICE is built on the data's facets, applying the imputation model based on the training data to test data could lead to false or unreliable imputations since the training data does not necessarily carry the characteristics of the test data. Hence, training the imputation model on the whole data set could be advantageous as it would encompass the whole data set's characteristics. Additionally, training MICE on an inadequate sample size cannot account for the complex relationships between the features due to the reduced variability and increased sparseness of the data set. Regardless of the implementation, MICE is also not a commonly used imputation method. Thus, there are currently no guidelines or procedures to identify good imputations. Further, MICE lacks a theoretical basis since fitting a series of conditional probabilities may not be consistent with a proper joint distribution of the features [4]. The method relies on empirical studies rather than a theoretical basis to justify its procedure. However, this may not be a large issue in practice,

but further research is needed here. Despite these shortcomings, MICE provides a great advantage over other imputation techniques regarding flexibility regarding variable type and the number of variables to impute [4, 48]. After all, MICE should be used in cases where traditional imputation methods face their limits due to the small sample size but where the characteristics of the variables and their relations are nonetheless recognizable.

4.2.2 Discussion of the Classifiers

In addition to the assessment of MICE, this section examines the classifiers regarding their explainability. Random Forest is known as a black-box model. It cannot provide interpretable insights into how its predictions are made since these predictions are produced by averaging the forest's Decision Trees [5]. In contrast, the Decision Tree is completely explainable since its decision structure can be visualized in a tree-like structure [40]. Accordingly, a Decision Tree can also help shed some light on the Random Forest's predictions by approximating its decision-making steps [18]. Nonetheless, the tree can only give limited insights and should be used in combination with feature importance plots. The Logistic Regression is also only interpretable up to a point since the magnitude of its coefficients only provides information about the importance of each feature. It cannot be used as a comparative measure of how much more important feature A is than another feature B, only that feature A is more important than feature B. Further, the coefficient's interpretation can be unintuitive because it is multiplicative rather than additive. Moreover, the coefficients have to be interpreted in the model's context since they depend on all other features [30].

4.2.3 Discussion of the Original Work

Another explanation for the differing results of this re-implementation is how the authors provide information on their implementation of the model training and selection process. They neither provide detailed information about which hyperparameters are tuned during cross-validation nor their ranges. Further, the structure of the cross-validation process was not clear at all times.
Moreover, Brinati et al. do not report all evaluation metrics used to determine the best classifiers. They only report 95% confidence intervals for the average accuracy and balanced accuracy and point references for the classifiers they deemed as best performing. There was no further information on other evaluation metrics like sen-

sitivity or AUC. Additionally, the authors only provide the configuration for one of the best performing models, namely the Random Forest. Unfortunately, this does not inform about the ranges for each hyperparameter used during Grid Search. Nevertheless, the authors provide their data and a description of each variable in the data set. It should also be noted that [6] is only a feasibility study to explore if machine learning classifiers can help identify patients with COVID-19 solely based on a routine blood exam where the focus is not on replicability but feasibility. However, such research that has a potential effect on an individual's health should strive to be as transparent and comprehensible as possible.

4.3 Conclusion

In conclusion, this thesis replicates two of the eight classifiers in [6], namely the Random Forest and Logistic Regression. A Decision Tree was constructed to provide support for the interpretation of the predictions from other models. The classifiers predict the outcome of a PCR test for COVID-19 using a routine blood exam. Although the classifiers perform worse than in the original paper, they still can be regarded as substantial due to a 90% sensitivity and an accuracy above 70% for both classifiers performing above chance level. Further, a Decision Tree model is built to provide an approximation of the Random Forest's decision-making steps. A comparison with medical research findings reveals that features identified as most important by the classifiers are also deemed pivotal in diagnosing COVID-19. Thereby, the patterns are confirmed and their theoretical validity is certified. This implementation uses Multiple Imputation using Chained Equations to impute the missing values. This method is used due to the small sample size of 279 individuals and the proportionally high samples with at least one missing value. Though MICE lacks a theoretical basis and justifies its procedure with experimental studies, it provides great advantages, especially where the sample size is too small for more conventional and theoretically justified imputation methods and where the variables differ in terms of their type and distribution. Seeing that the replicated paper is a feasibility study investigating if it is possible to predict a PCR test result from a range of blood values, further research should be done on this topic. For instance, Brinati et al. did a follow-up study using a data set comprising more samples, more blood values, and symptomatic observations at the time of triage or admittance to a hospital [10]. Importantly, it should always be brought to attention that these models should never replace the actual medical test. They should be seen as a valuable though non-conclusive tool to detect patients with a particular disease earlier and better,

resulting in improved management of these patients while awaiting confirmation by the medical test.

Code & Data Availability

The code and data used to train the classifiers is made available at the GitHub repository https://github.com/lschiesser/ba. The data set is also available at the original source https://zenodo.org/record/3886927#.YBPnRXkxlEY.
A data drive with the code and data was submitted along a hard copy version of this thesis to the examination office.

Declaration

I declare that...

Appendix A

Supplementary Tables

Feature	Data Type
Gender	Categorical
Age	Numerical (discrete)
WBC (White blood cell count)	Numerical (continuous)
Platelets	Numerical (continuous)
Neutrophils	Numerical (continuous)
Lymphocytes	Numerical (continuous)
Monocytes	Numerical (continuous)
Eosinophils	Numerical (continuous)
Basophils	Numerical (continuous)
CRP (C-reactive protein)	Numerical (continuous)
AST (aspartate aminotransferase)	Numerical (continuous)
ALT (alanine aminotransferase)	Numerical (continuous)
ALP (alkaline phosphatase)	Numerical (continuous)
GGT (gamma glutamyl transferase)	Numerical (continuous)
LDH (lactate dehydrogenase)	Numerical (continuous)
SWAB	Categorical

Table A.1: Overview of all features of the data set

Feature	Unit	Mean	Std	Median
Age	Years	61.33	18.05	64
White Blood Cell Count (WBC)	10^9/L	8.49	4.89	7.10
Platelets	10^9/L	224.91	102.61	204.00
Neutrophils	10^9/L	4.64	4.50	3.90
Lymphocytes	10^9/L	0.88	0.87	0.80
Monocytes	10^9/L	0.45	0.44	0.40
Eosinophils	10^9/L	0.04	0.12	0.00
Basophils	10^9/L	0.01	0.03	0.00
C-reactive protein (CRP)	mg/L	88.93	94.32	53.10
Aspartate Aminotransferase (AST)	U/L	53.81	57.59	36.00
Alanine Aminotransferase (ALT)	U/L	42.82	45.43	30.00
Alkaline Phosphatase (ALP)	U/L	42.21	75.71	68.00
Gamma Glutamyl Transferase (GGT)	U/L	40.20	101.29	0.00
Lactate dehydrogenase (LDH)	U/L	264.54	238.53	254.00

Table A.2: Descriptive statistics for numerical features in the data set (including missing values as in [6])

Feature	Number of NaN (in %)
Gender	0 (0 %)
Age	2 (0.72 %)
WBC (White blood cell count)	2 (0.72 %)
Platelets	2 (0.72 %)
Neutrophils	70 (25.09 %)
Lymphocytes	71 (25.45 %)
Monocytes	70 (25.09 %)
Eosinophils	70 (25.09 %)
Basophils	71 (25.45 %)
CRP (C-reactive protein)	6 (2.15 %)
AST (aspartate aminotransferase)	2 (0.72 %)
ALT (alanine aminotransferase)	13 (4.66 %)
ALP (alkaline phosphatase)	148 (53.05 %)
GGT (gamma glutamyl transferase)	143 (51.25 %)
LDH (lactate dehydrogenase)	85 (30.47 %)
SWAB	0 (0 %)

Table A.3: Number of missing values and their proportion of the total number of data points

Feature	Unit	Mean	Std	Median
Age	Years	61.78	17.81	64
White Blood Cell Count (WBC)	10^9/L	8.55	4.86	7.10
Platelets	10^9/L	226.5	101.2	205.00
Neutrophils	10^9/L	6.20	4.17	5.10
Lymphocytes	10^9/L	1.19	0.80	1.00
Monocytes	10^9/L	0.61	0.41	0.50
Eosinophils	10^9/L	0.06	0.13	0.00
Basophils	10^9/L	0.01	0.04	0.00
C-reactive protein (CRP)	mg/L	90.89	94.42	54.20
Aspartate Aminotransferase (AST)	U/L	54.20	57.61	36.00
Alanine Aminotransferase (ALT)	U/L	44.92	45.50	31.00
Alkaline Phosphatase (ALP)	U/L	89.89	89.09	71.00
Gamma Glutamyl Transferase (GGT)	U/L	82.48	132.70	41.00
Lactate dehydrogenase (LDH)	U/L	380.45	193.98	328.00

Table A.4: Descriptive statistics for numerical features in data set (excluding missing values)

Feature	Test statistic	p-value
Age	0.976	0.000
WBC (White blood cell count)	0.873	0.000
Platelets	0.930	0.000
Neutrophils	0.838	0.000
Lymphocytes	0.785	0.000
Monocytes	0.811	0.000
Eosinophils	0.457	0.000
Basophils	0.395	0.000
CRP (C-reactive protein)	0.836	0.000
AST (aspartate aminotransferase)	0.556	0.000
ALT (alanine aminotransferase)	0.629	0.000
ALP (alkaline phosphatase)	0.420	0.000
GGT (gamma glutamyl transferase)	0.500	0.000
LDH (lactate dehydrogenase)	0.877	0.000

Table A.5: Results of Shapiro-Wilk test for normality for numerical features of the data set; a p-value under 0.05 refutes the null hypohtesis that the variable is normally distributed

	Fold1		Fold2		Fold3		Fold4		Fold5		Combined	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
RF	0.72	0.01	0.84	0.03	0.81	0.03	0.75	0.03	0.77	0.03	0.78	0.05
LR	0.71	0.03	0.81	0.02	0.74	0.03	0.70	0.06	0.73	0.04	0.74	0.05

Table A.6: Mean accuracies and their standard deviations (std) for each training fold and all training folds combined. RF = Random Forest, LR = Logisitic Regression

	Fold1		Fold2		Fold3		Fold4		Fold5		Combined	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
RF	0.69	0.01	0.81	0.04	0.79	0.03	0.73	0.03	0.71	0.04	0.75	0.05
LR	0.67	0.02	0.78	0.03	0.72	0.03	0.68	0.06	0.67	0.04	0.70	0.06

Table A.7: Mean balanced accuracies and their standard deviations (std) for each training fold and all training folds combined. RF = Random Forest, LR = Logisitic Regression

Model	Accuracy	Balanced Accuracy	Sensitivity	Specificity	PPV	AUC	AUC (PRC)
RF	0.768	0.737	0.909	0.565	0.750	0.802	0.829
LR	0.732	0.694	0.909	0.478	0.714	0.808	0.855
DT	0.714	0.685	0.848	0.522	0.718	0.685	0.698

Table A.8: Validation metrics for all classifiers; RF = Random Forest, DT = Decision Tree, LR = Logisitic Regression

Fold	Imputed Set	Model	Model Count	Accuracy	Balanced Accuracy	Sensitivity	Specificity	PPV	AUC
0	1	LR	0	0.750	0.709	0.939	0.478	0.721	0.816
0	2	LR	1	0.714	0.679	0.879	0.478	0.707	0.837
0	3	LR	2	0.696	0.663	0.848	0.478	0.700	0.806
0	4	LR	3	0.696	0.663	0.848	0.478	0.700	0.797
0	5	LR	4	0.679	0.642	0.848	0.435	0.683	0.826
1	1	LR	5	0.804	0.787	0.825	0.750	0.892	0.845
1	2	LR	6	0.839	0.831	0.850	0.812	0.919	0.872
1	3	LR	7	0.804	0.787	0.825	0.750	0.892	0.856
1	4	LR	8	0.786	0.738	0.850	0.625	0.850	0.822
1	5	LR	9	0.804	0.769	0.850	0.688	0.872	0.833
2	1	LR	10	0.732	0.713	0.818	0.609	0.750	0.742
2	2	LR	11	0.768	0.750	0.848	0.652	0.778	0.764
2	3	LR	12	0.768	0.757	0.818	0.696	0.794	0.823
2	4	LR	13	0.732	0.720	0.788	0.652	0.765	0.773
2	5	LR	14	0.696	0.683	0.758	0.609	0.735	0.758
3	1	LR	15	0.714	0.695	0.771	0.619	0.771	0.774
3	2	LR	16	0.750	0.733	0.800	0.667	0.800	0.788
3	3	LR	17	0.643	0.619	0.714	0.524	0.714	0.686
3	4	LR	18	0.750	0.733	0.800	0.667	0.800	0.777
3	5	LR	19	0.625	0.605	0.686	0.524	0.706	0.680
4	1	LR	20	0.800	0.723	0.972	0.474	0.778	0.801
4	2	LR	21	0.727	0.680	0.833	0.526	0.769	0.781
4	3	LR	22	0.709	0.629	0.889	0.368	0.727	0.773
4	4	LR	23	0.691	0.627	0.833	0.421	0.732	0.754
4	5	LR	24	0.727	0.680	0.833	0.526	0.769	0.785
0	1	RF	0	0.732	0.694	0.909	0.478	0.714	0.792
0	2	RF	1	0.714	0.685	0.848	0.522	0.718	0.801
0	3	RF	2	0.732	0.694	0.909	0.478	0.714	0.782
0	4	RF	3	0.732	0.700	0.879	0.522	0.725	0.739
0	5	RF	4	0.714	0.692	0.818	0.565	0.730	0.760
1	1	RF	5	0.875	0.856	0.900	0.812	0.923	0.848
1	2	RF	6	0.821	0.800	0.850	0.750	0.895	0.842
1	3	RF	7	0.839	0.812	0.875	0.750	0.897	0.850
1	4	RF	8	0.839	0.812	0.875	0.750	0.897	0.809
1	5	RF	9	0.804	0.750	0.875	0.625	0.854	0.806
2	1	RF	10	0.804	0.787	0.879	0.696	0.806	0.825
2	2	RF	11	0.857	0.839	0.939	0.739	0.838	0.839
2	3	RF	12	0.786	0.772	0.848	0.696	0.800	0.816
2	4	RF	13	0.804	0.781	0.909	0.652	0.789	0.847
2	5	RF	14	0.804	0.787	0.879	0.696	0.806	0.831
3	1	RF	15	0.750	0.714	0.857	0.571	0.769	0.826
3	2	RF	16	0.696	0.700	0.686	0.714	0.800	0.783
3	3	RF	17	0.768	0.738	0.857	0.619	0.789	0.848
3	4	RF	18	0.786	0.762	0.857	0.667	0.811	0.812
3	5	RF	19	0.768	0.729	0.886	0.571	0.775	0.793
4	1	RF	20	0.782	0.734	0.889	0.579	0.800	0.836
4	2	RF	21	0.764	0.695	0.917	0.474	0.767	0.798
4	3	RF	22	0.727	0.655	0.889	0.421	0.744	0.848
4	4	RF	23	0.764	0.708	0.889	0.526	0.780	0.798
4	5	RF	24	0.800	0.760	0.889	0.632	0.821	0.777

Table A.9: Evaluation metrics for Random Forest (RF) and Logisitc Regression (LR) during the nested cross validation

	AGE		WBC		Platelets		Neutrophils		Lymphocytes		Monocytes		Eosinophils		Basophils		CRP		AST		ALT		ALP		GGT		LDH	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
original	61.776	17.816	8.553	4.855	226.532	101.174	6.200	4.173	1.187	0.806	0.606	0.410	0.055	0.132	0.014	0.039	90.889	94.421	54.202	57.613	44.917	45.503	89.893	89.090	82.478	132.703	380.448	193.984
1.0	61.839	17.773	8.574	4.863	226.654	100.820	6.597	4.482	1.260	0.899	0.649	0.455	0.066	0.147	0.017	0.041	92.581	96.875	54.208	57.458	47.459	49.868	82.362	66.091	79.681	119.072	379.516	199.693
2.0	61.832	17.768	8.557	4.840	225.951	101.067	6.597	4.301	1.236	0.807	0.655	0.458	0.052	0.140	0.015	0.039	91.247	94.973	54.269	57.422	47.405	50.648	81.803	65.194	73.470	103.199	365.706	178.115
3.0	61.871	17.792	8.600	4.870	226.288	100.896	6.650	4.439	1.220	0.801	0.659	0.473	0.054	0.123	0.014	0.038	91.278	94.059	54.161	57.417	46.774	48.087	83.484	67.061	75.283	106.761	367.760	191.073
4.0	61.728	17.768	8.557	4.839	226.571	100.911	6.625	4.353	1.229	0.807	0.643	0.429	0.054	0.124	0.016	0.043	91.447	94.246	54.029	57.452	46.864	48.285	83.065	66.928	75.384	113.305	374.771	195.309
5.0	61.832	17.806	8.534	4.843	226.446	101.323	6.592	4.412	1.209	0.799	0.642	0.446	0.055	0.126	0.015	0.039	91.255	94.812	54.004	57.453	45.943	45.804	83.964	67.706	76.444	117.722	374.405	196.479
6.0	61.839	17.773	8.574	4.863	226.654	100.820	6.597	4.482	1.260	0.899	0.649	0.455	0.066	0.147	0.017	0.041	92.581	96.875	54.208	57.458	47.459	49.868	82.362	66.091	79.681	119.072	379.516	199.693
7.0	61.832	17.768	8.557	4.840	225.951	101.067	6.597	4.301	1.236	0.807	0.655	0.458	0.052	0.140	0.015	0.039	91.247	94.973	54.269	57.422	47.405	50.648	81.803	65.194	73.470	103.199	365.706	178.115
8.0	61.871	17.792	8.600	4.870	226.288	100.896	6.650	4.439	1.220	0.801	0.659	0.473	0.054	0.123	0.014	0.038	91.278	94.059	54.161	57.417	46.774	48.087	83.484	67.061	75.283	106.761	367.760	191.073
9.0	61.728	17.768	8.557	4.839	226.571	100.911	6.625	4.353	1.229	0.807	0.643	0.429	0.054	0.124	0.016	0.043	91.447	94.246	54.029	57.452	46.864	48.285	83.065	66.928	75.384	113.305	374.771	195.309
10.0	61.832	17.806	8.534	4.843	226.446	101.323	6.592	4.412	1.209	0.799	0.642	0.446	0.055	0.126	0.015	0.039	91.255	94.812	54.004	57.453	45.943	45.804	83.964	67.706	76.444	117.722	374.405	196.479
11.0	61.839	17.773	8.574	4.863	226.654	100.820	6.597	4.482	1.260	0.899	0.649	0.455	0.066	0.147	0.017	0.041	92.581	96.875	54.208	57.458	47.459	49.868	82.362	66.091	79.681	119.072	379.516	199.693
12.0	61.832	17.768	8.557	4.840	225.951	101.067	6.597	4.301	1.236	0.807	0.655	0.458	0.052	0.140	0.015	0.039	91.247	94.973	54.269	57.422	47.405	50.648	81.803	65.194	73.470	103.199	365.706	178.115
13.0	61.871	17.792	8.600	4.870	226.288	100.896	6.650	4.439	1.220	0.801	0.659	0.473	0.054	0.123	0.014	0.038	91.278	94.059	54.161	57.417	46.774	48.087	83.484	67.061	75.283	106.761	367.760	191.073
14.0	61.728	17.768	8.557	4.839	226.571	100.911	6.625	4.353	1.229	0.807	0.643	0.429	0.054	0.124	0.016	0.043	91.447	94.246	54.029	57.452	46.864	48.285	83.065	66.928	75.384	113.305	374.771	195.309
15.0	61.832	17.806	8.534	4.843	226.446	101.323	6.592	4.412	1.209	0.799	0.642	0.446	0.055	0.126	0.015	0.039	91.255	94.812	54.004	57.453	45.943	45.804	83.964	67.706	76.444	117.722	374.405	196.479
16.0	61.839	17.773	8.574	4.863	226.654	100.820	6.597	4.482	1.260	0.899	0.649	0.455	0.066	0.147	0.017	0.041	92.581	96.875	54.208	57.458	47.459	49.868	82.362	66.091	79.681	119.072	379.516	199.693
17.0	61.832	17.768	8.557	4.840	225.951	101.067	6.597	4.301	1.236	0.807	0.655	0.458	0.052	0.140	0.015	0.039	91.247	94.973	54.269	57.422	47.405	50.648	81.803	65.194	73.470	103.199	365.706	178.115
18.0	61.871	17.792	8.600	4.870	226.288	100.896	6.650	4.439	1.220	0.801	0.659	0.473	0.054	0.123	0.014	0.038	91.278	94.059	54.161	57.417	46.774	48.087	83.484	67.061	75.283	106.761	367.760	191.073
19.0	61.728	17.768	8.557	4.839	226.571	100.911	6.625	4.353	1.229	0.807	0.643	0.429	0.054	0.124	0.016	0.043	91.447	94.246	54.029	57.452	46.864	48.285	83.065	66.928	75.384	113.305	374.771	195.309
20.0	61.832	17.806	8.534	4.843	226.446	101.323	6.592	4.412	1.209	0.799	0.642	0.446	0.055	0.126	0.015	0.039	91.255	94.812	54.004	57.453	45.943	45.804	83.964	67.706	76.444	117.722	374.405	196.479
21.0	61.839	17.773	8.574	4.863	226.654	100.820	6.597	4.482	1.260	0.899	0.649	0.455	0.066	0.147	0.017	0.041	92.581	96.875	54.208	57.458	47.459	49.868	82.362	66.091	79.681	119.072	379.516	199.693
22.0	61.832	17.768	8.557	4.840	225.951	101.067	6.597	4.301	1.236	0.807	0.655	0.458	0.052	0.140	0.015	0.039	91.247	94.973	54.269	57.422	47.405	50.648	81.803	65.194	73.470	103.199	365.706	178.115
23.0	61.871	17.792	8.600	4.870	226.288	100.896	6.650	4.439	1.220	0.801	0.659	0.473	0.054	0.123	0.014	0.038	91.278	94.059	54.161	57.417	46.774	48.087	83.484	67.061	75.283	106.761	367.760	191.073
24.0	61.728	17.768	8.557	4.839	226.571	100.911	6.625	4.353	1.229	0.807	0.643	0.429	0.054	0.124	0.016	0.043	91.447	94.246	54.029	57.452	46.864	48.285	83.065	66.928	75.384	113.305	374.771	195.309
25.0	61.832	17.806	8.534	4.843	226.446	101.323	6.592	4.412	1.209	0.799	0.642	0.446	0.055	0.126	0.015	0.039	91.255	94.812	54.004	57.453	45.943	45.804	83.964	67.706	76.444	117.722	374.405	196.479

Table A.10: Numerical summaries for all variables of the original not-imputed data set (denoted by original) and the imputed data sets (denoted by a number)

	AGE		WBC		Platelets		Neutrophils		Lymphocytes		Monocytes		Eosinophils		Basophils		CRP		AST		ALT		ALP		GGT		LDH	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
original	61.776	17.816	8.553	4.855	226.532	101.174	6.200	4.173	1.187	0.806	0.606	0.410	0.055	0.132	0.014	0.039	90.889	94.421	54.202	57.613	44.917	45.503	89.893	89.090	82.478	132.703	380.448	193.984
1.0	70.500	7.778	11.500	7.071	243.500	2.121	7.783	5.153	1.475	1.105	0.780	0.553	0.099	0.182	0.025	0.044	169.583	171.826	55.000	41.012	99.462	93.888	75.696	33.854	77.021	104.884	377.388	213.331
2.0	69.500	6.364	9.100	2.404	145.500	37.477	7.783	4.488	1.382	0.795	0.803	0.557	0.043	0.161	0.017	0.038	107.533	127.200	63.500	19.092	98.308	103.728	74.642	30.147	64.902	62.903	332.059	129.954
3.0	75.000	7.071	15.150	0.778	192.500	50.205	7.993	4.947	1.320	0.784	0.817	0.600	0.051	0.088	0.014	0.035	109.000	80.700	48.500	17.678	84.769	78.776	77.811	37.641	68.441	73.996	338.800	182.065
4.0	55.000	8.485	9.050	1.485	232.000	74.953	7.891	4.659	1.352	0.800	0.754	0.468	0.050	0.097	0.023	0.051	116.850	89.815	30.000	18.385	86.692	80.529	77.020	37.065	68.636	91.091	361.812	198.849
5.0	69.500	20.506	5.900	0.566	214.500	168.999	7.760	4.910	1.273	0.779	0.750	0.527	0.056	0.104	0.017	0.038	107.900	120.274	26.500	3.536	66.923	48.786	78.716	39.890	70.706	101.595	360.612	202.547
6.0	70.500	7.778	11.500	7.071	243.500	2.121	7.783	5.153	1.475	1.105	0.780	0.553	0.099	0.182	0.025	0.044	169.583	171.826	55.000	41.012	99.462	93.888	75.696	33.854	77.021	104.884	377.388	213.331
7.0	69.500	6.364	9.100	2.404	145.500	37.477	7.783	4.488	1.382	0.795	0.803	0.557	0.043	0.161	0.017	0.038	107.533	127.200	63.500	19.092	98.308	103.728	74.642	30.147	64.902	62.903	332.059	129.954
8.0	75.000	7.071	15.150	0.778	192.500	50.205	7.993	4.947	1.320	0.784	0.817	0.600	0.051	0.088	0.014	0.035	109.000	80.700	48.500	17.678	84.769	78.776	77.811	37.641	68.441	73.996	338.800	182.065
9.0	55.000	8.485	9.050	1.485	232.000	74.953	7.891	4.659	1.352	0.800	0.754	0.468	0.050	0.097	0.023	0.051	116.850	89.815	30.000	18.385	86.692	80.529	77.020	37.065	68.636	91.091	361.812	198.849
10.0	69.500	20.506	5.900	0.566	214.500	168.999	7.760	4.910	1.273	0.779	0.750	0.527	0.056	0.104	0.017	0.038	107.900	120.274	26.500	3.536	66.923	48.786	78.716	39.890	70.706	101.595	360.612	202.547
11.0	70.500	7.778	11.500	7.071	243.500	2.121	7.783	5.153	1.475	1.105	0.780	0.553	0.099	0.182	0.025	0.044	169.583	171.826	55.000	41.012	99.462	93.888	75.696	33.854	77.021	104.884	377.388	213.331
12.0	69.500	6.364	9.100	2.404	145.500	37.477	7.783	4.488	1.382	0.795	0.803	0.557	0.043	0.161	0.017	0.038	107.533	127.200	63.500	19.092	98.308	103.728	74.642	30.147	64.902	62.903	332.059	129.954
13.0	75.000	7.071	15.150	0.778	192.500	50.205	7.993	4.947	1.320	0.784	0.817	0.600	0.051	0.088	0.014	0.035	109.000	80.700	48.500	17.678	84.769	78.776	77.811	37.641	68.441	73.996	338.800	182.065
14.0	55.000	8.485	9.050	1.485	232.000	74.953	7.891	4.659	1.352	0.800	0.754	0.468	0.050	0.097	0.023	0.051	116.850	89.815	30.000	18.385	86.692	80.529	77.020	37.065	68.636	91.091	361.812	198.849
15.0	69.500	20.506	5.900	0.566	214.500	168.999	7.760	4.910	1.273	0.779	0.750	0.527	0.056	0.104	0.017	0.038	107.900	120.274	26.500	3.536	66.923	48.786	78.716	39.890	70.706	101.595	360.612	202.547
16.0	70.500	7.778	11.500	7.071	243.500	2.121	7.783	5.153	1.475	1.105	0.780	0.553	0.099	0.182	0.025	0.044	169.583	171.826	55.000	41.012	99.462	93.888	75.696	33.854	77.021	104.884	377.388	213.331
17.0	69.500	6.364	9.100	2.404	145.500	37.477	7.783	4.488	1.382	0.795	0.803	0.557	0.043	0.161	0.017	0.038	107.533	127.200	63.500	19.092	98.308	103.728	74.642	30.147	64.902	62.903	332.059	129.954
18.0	75.000	7.071	15.150	0.778	192.500	50.205	7.993	4.947	1.320	0.784	0.817	0.600	0.051	0.088	0.014	0.035	109.000	80.700	48.500	17.678	84.769	78.776	77.811	37.641	68.441	73.996	338.800	182.065
19.0	55.000	8.485	9.050	1.485	232.000	74.953	7.891	4.659	1.352	0.800	0.754	0.468	0.050	0.097	0.023	0.051	116.850	89.815	30.000	18.385	86.692	80.529	77.020	37.065	68.636	91.091	361.812	198.849
20.0	69.500	20.506	5.900	0.566	214.500	168.999	7.760	4.910	1.273	0.779	0.750	0.527	0.056	0.104	0.017	0.038	107.900	120.274	26.500	3.536	66.923	48.786	78.716	39.890	70.706	101.595	360.612	202.547
21.0	70.500	7.778	11.500	7.071	243.500	2.121	7.783	5.153	1.475	1.105	0.780	0.553	0.099	0.182	0.025	0.044	169.583	171.826	55.000	41.012	99.462	93.888	75.696	33.854	77.021	104.884	377.388	213.331
22.0	69.500	6.364	9.100	2.404	145.500	37.477	7.783	4.488	1.382	0.795	0.803	0.557	0.043	0.161	0.017	0.038	107.533	127.200	63.500	19.092	98.308	103.728	74.642	30.147	64.902	62.903	332.059	129.954
23.0	75.000	7.071	15.150	0.778	192.500	50.205	7.993	4.947	1.320	0.784	0.817	0.600	0.051	0.088	0.014	0.035	109.000	80.700	48.500	17.678	84.769	78.776	77.811	37.641	68.441	73.996	338.800	182.065
24.0	55.000	8.485	9.050	1.485	232.000	74.953	7.891	4.659	1.352	0.800	0.754	0.468	0.050	0.097	0.023	0.051	116.850	89.815	30.000	18.385	86.692	80.529	77.020	37.065	68.636	91.091	361.812	198.849
25.0	69.500	20.506	5.900	0.566	214.500	168.999	7.760	4.910	1.273	0.779	0.750	0.527	0.056	0.104	0.017	0.038	107.900	120.274	26.500	3.536	66.923	48.786	78.716	39.890	70.706	101.595	360.612	202.547

Table A.11: Numerical summaries for all variables of the original not-imputed data set (denoted by original) and the imputed data sets (denoted by a number) only for the imputed values

Appendix B

Supplementary Figures

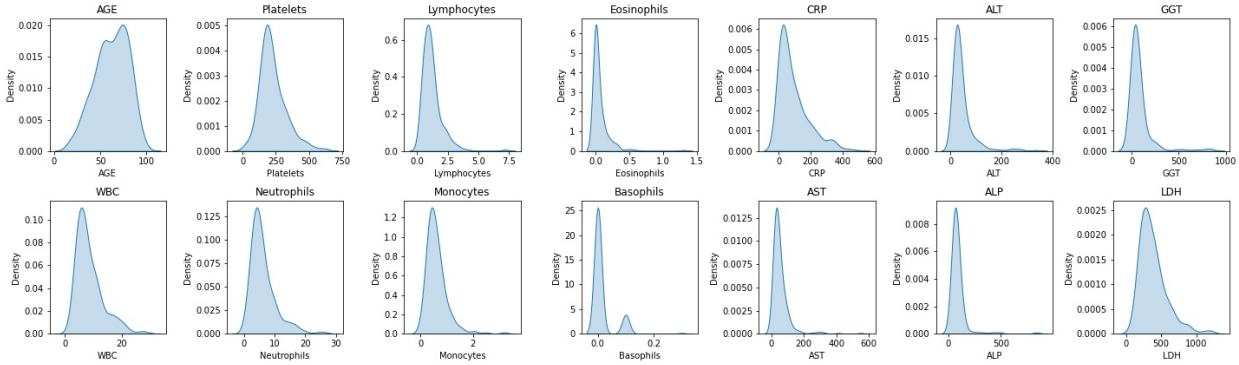
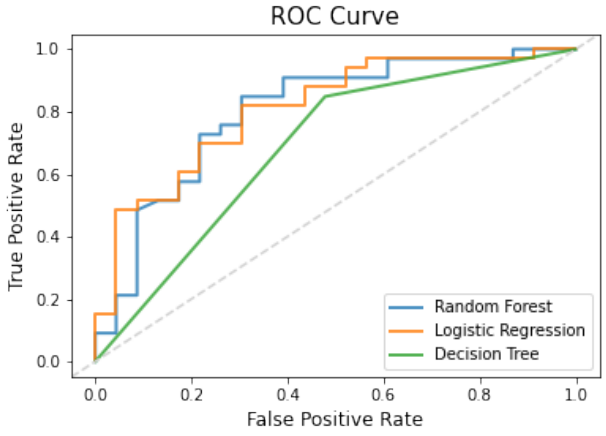


Figure B.1: Kernel density plots of the numerical features of the data set

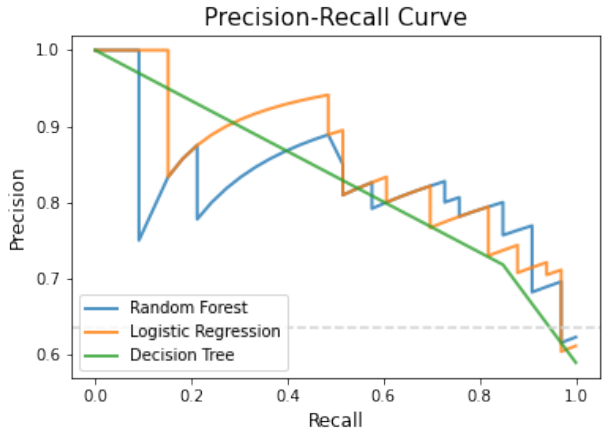
Nested k-fold Cross-Validation

1. Define set of hyper-parameter combinations, ***C***, for current model. If model has no hyper-parameters, ***C*** is the empty set.
2. Divide data into *K* folds with approximately equal distribution of cases and controls
3. (**outer loop**) For fold k_i in the *K* folds:
    1. Set fold k_i as the test set
    2. Perform automated feature selection on the remaining *K-1* folds
    3. For parameter combination c in ***C***:
        1. (**inner loop**) For fold k_j in the remaining *K-1* folds:
            1. Set fold k_j as the validation set
            2. Train model on remaining *K-2* folds
            3. Evaluate model performance on fold k_j
        2. Calculate average performance over *K-2* folds for parameter combination *c*
    4. Train model on *K-1* folds using hyper-parameter combination that yielded best average performance over all steps of the **inner loop**
    5. Evaluate model performance on fold k_i.
4. Calculate average performance over *K* folds

Figure B.2: Pseudo-code for K-fold nested cross validation [25]

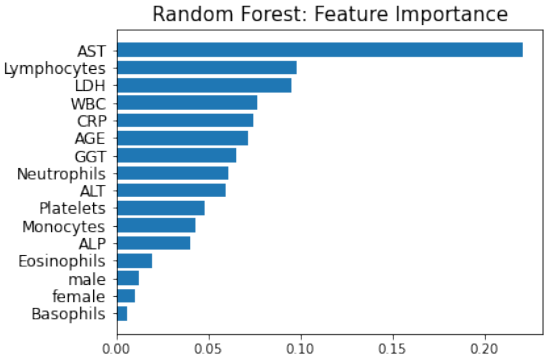


(a) Receiver Operating Characterisitcs (ROC) curve

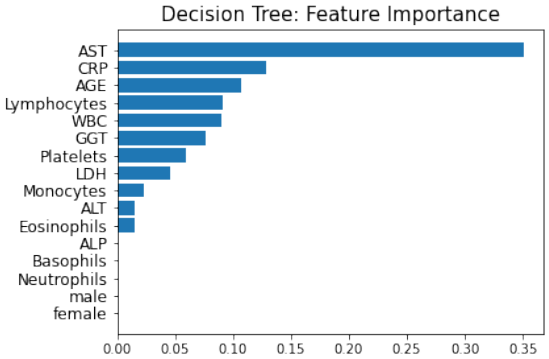


(b) Precision-Recall Curve

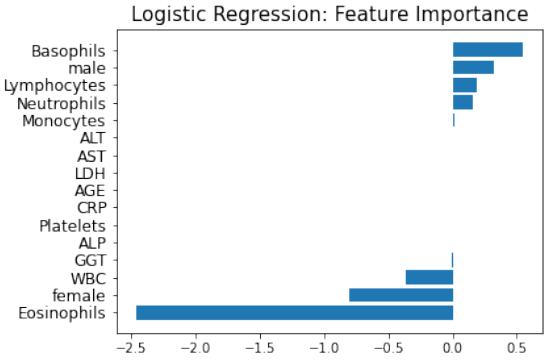
Figure B.3: ROC and Precision-Recall curve



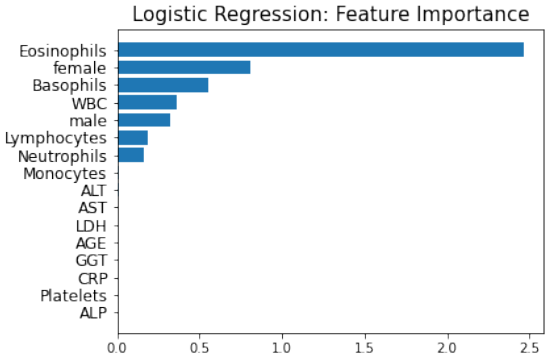
(a) Random Forest



(b) Decision Tree



(c) Logistic Regression (log space)



(d) Logistic Regression (absolute log space)

Figure B.4: Feature importance plots

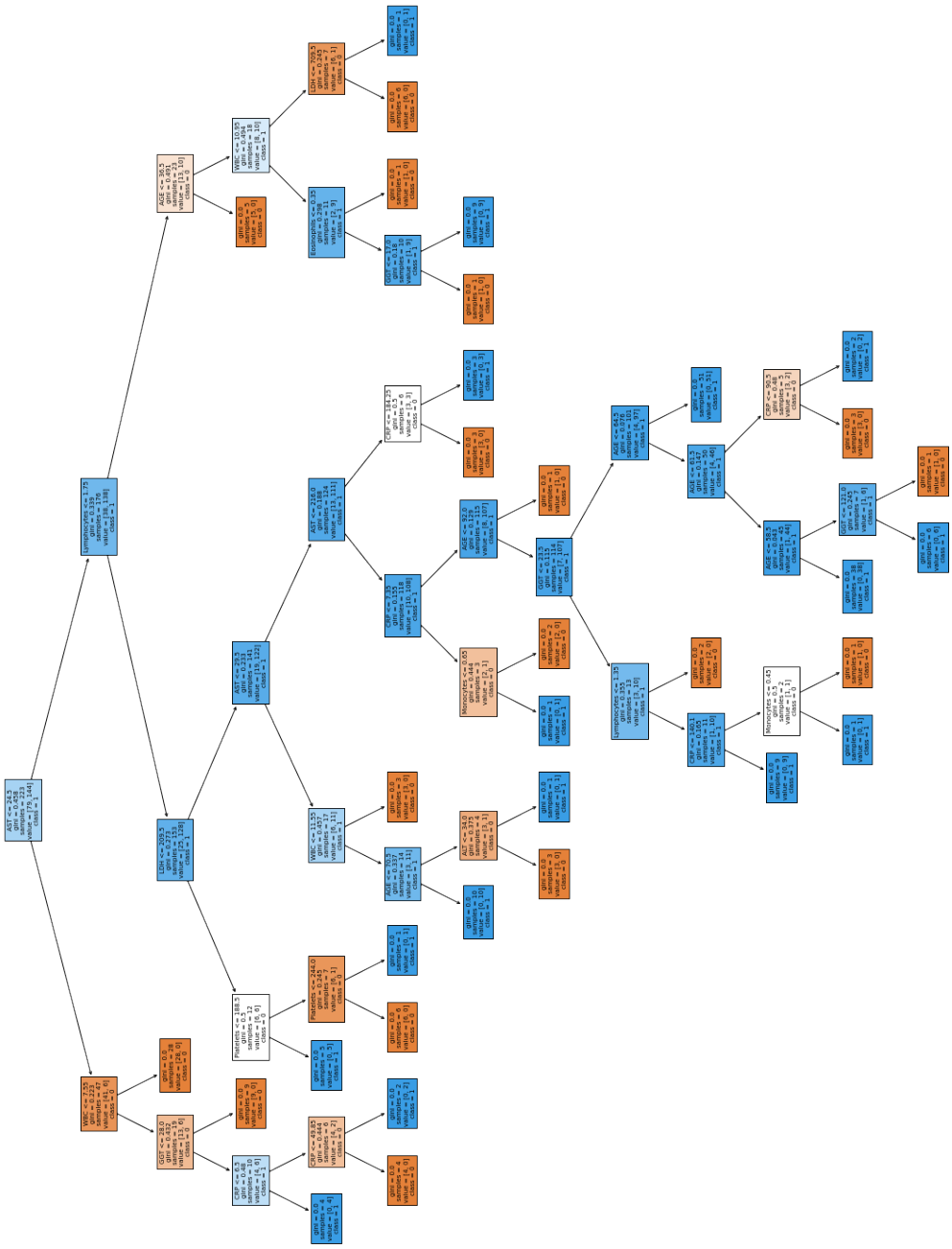


Figure B.5: Visualization of the Decision Tree produced during the validation phase

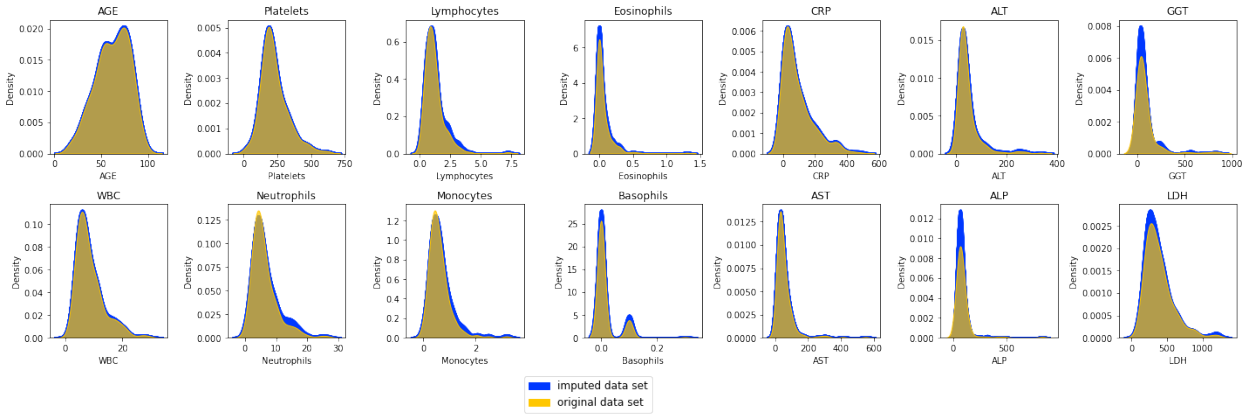


Figure B.6: Kernel density plots for the original (not imputed) data set (yellow) and the imputed data sets (blue)

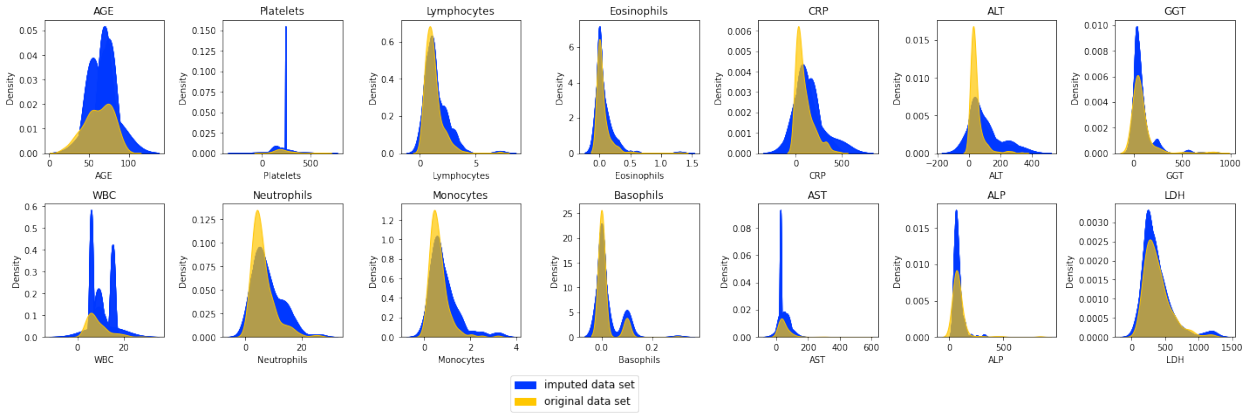


Figure B.7: Kernel density plots for the original (not imputed) data set (yellow and the imputed data sets where only the imputed samples are plotted (blue)

Bibliography

[1] A. K. Akobeng. "Understanding diagnostic tests 1: sensitivity, specificity and predictive values". In: *Acta Paediatr* 96.3 (2007), pp. 338–41. ISSN: 0803-5253 (Print) 0803-5253 (Linking). DOI: 10.1111/j.1651-2227.2006.00180.x. URL: https://www.ncbi.nlm.nih.gov/pubmed/17407452.

[2] P. Allison. *Imputation by Predictive Mean Matching: Promise and Peril — Statistical Horizons*. 2015. URL: https://statisticalhorizons.com/predictive-mean-matching.

[3] *An Introduction to Statistical Learning*. Springer Texts in Statistics. 2013. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

[4] M. J. Azur et al. "Multiple imputation by chained equations: what is it and how does it work?" In: *Int J Methods Psychiatr Res* 20.1 (2011), pp. 40–9. ISSN: 1557-0657 (Electronic) 1049-8931 (Linking). DOI: 10.1002/mpr.329. URL: https://www.ncbi.nlm.nih.gov/pubmed/21499542,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/pdf/MPR-20-40.pdf.

[5] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/a:1010933404324. URL: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf.

[6] D. Brinati et al. "Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study". In: *J Med Syst* 44.8 (2020), p. 135. ISSN: 1573-689X (Electronic) 0148-5598 (Linking). DOI: 10.1007/s10916-020-01597-4. URL: https://www.ncbi.nlm.nih.gov/pubmed/32607737,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7326624/pdf/10916_2020_Article_1597%20.pdf,https://zenodo.org/record/3886927#.X3xhPO1CRPY.

[7] Stef van Buuren. *Flexible Imputation of Missing Data, Second Edition*. 2nd ed. Boca Raton, Fla: CRC Press, 2018. ISBN: 978-0-429-96034-5. URL: https://stefvanbuuren.name/fimd/.

[8] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations inR". In: *Journal of Statistical Software* 45.3 (2011). ISSN: 1548-7660. DOI: 10.18637/jss.v045.i03.

[9] F. Cabitza and J. D. Zeitoun. "The proof of the pudding: in praise of a culture of real-world validation for medical artificial intelligence". In: *Ann Transl Med* 7.8 (2019), p. 161. ISSN: 2305-5839 (Print) 2305-5839 (Linking). DOI: 10.21037/atm.2019.04.07. URL: https://www.ncbi.nlm.nih.gov/pubmed/31168442,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6526255/pdf/atm-07-08-161.pdf.

[10] F. Cabitza et al. "Development, evaluation, and validation of machine learning models for COVID-19 detection based on routine blood tests". In: *Clin Chem Lab Med* (2020). ISSN: 1437-4331 (Electronic) 1434-6621 (Linking). DOI: 10.1515/cclm-2020-1294. URL: https://www.ncbi.nlm.nih.gov/pubmed/33079698,%20https://www.degruyter.com/downloadpdf/journals/cclm/ahead-of-print/article-10.15%2015-cclm-2020-1294/article-10.1515-cclm-2020-1294.pdf.

[11] P. C. Chen, Y. Liu, and L. Peng. "How to develop machine learning models for healthcare". In: *Nat Mater* 18.5 (2019), pp. 410–414. ISSN: 1476-1122 (Print) 1476-1122 (Linking). DOI: 10.1038/s41563-019-0345-0. URL: https://www.ncbi.nlm.nih.gov/pubmed/31000806,%20https://www.nature.com/articles/s41563-019-0345-0.pdf.

[12] M. P. Cheng et al. "Diagnostic Testing for Severe Acute Respiratory Syndrome-Related Coronavirus 2: A Narrative Review". In: *Ann Intern Med* 172.11 (2020), pp. 726–734. ISSN: 1539-3704 (Electronic) 0003-4819 (Linking). DOI: 10.7326/M20-1301. URL: https://www.ncbi.nlm.nih.gov/pubmed/32282894,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7170415/pdf/aim-olf-M201301.pdf.

[13] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Journal* (2006), pp. 233–240. ISSN: ISSN. DOI: 10.1145/1143844.1143874. URL: https://dl.acm.org/doi/pdf/10.1145/1143844.1143874.

[14] D. K. Djakpo et al. "Blood routine test in mild and common 2019 coronavirus (COVID-19) patients". In: *Biosci Rep* 40.8 (2020). ISSN: 1573-4935 (Electronic) 0144-8463 (Linking). DOI: 10.1042/BSR20200817. URL: https://www.ncbi.nlm.nih.gov/pubmed/32725148,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7414516/pdf/bsr-40-bsr20200817.pdf.

[15] Ensheng Dong, Hongru Du, and Lauren Gardner. "An interactive web-based dashboard to track COVID-19 in real time". In: *The Lancet Infectious Diseases* 20.5 (2020), pp. 533–534. ISSN: 14733099. DOI: 10.1016/s1473-3099(20)30120-1.

[16] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010. URL: https://www.sciencedirect.com/science/article/abs/pii/S016786550500303X?via%3Dih%20ub.

[17] D. Ferrari et al. "Routine blood tests as a potential diagnostic tool for COVID-19". In: *Clin Chem Lab Med* 58.7 (2020), pp. 1095–1099. ISSN: 1437-4331 (Electronic) 1434-6621 (Linking). DOI: 10.1515/cclm-2020-0398. URL: https://www.ncbi.nlm.nih.gov/pubmed/32301746,%20https://www.degruyter.com/downloadpdf/journals/cclm/58/7/article-p1095.pdf.

[18] Randy Goebel et al. "Explainable AI: The New 42?" In: *Machine Learning and Knowledge Extraction*. Lecture Notes in Computer Science. 2018. Chap. Chapter 21, pp. 295–303. ISBN: 978-3-319-99739-1 978-3-319-99740-7. DOI: 10.1007/978-3-319-99740-7_21. URL: https://link.springer.com/content/pdf/10.1007%2F978-3-319-99740-7_21.pdf.

[19] F. Gravetter and L. Wallnau. *Statistics for the behavioral sciences*. 9th ed. Belmont, CA: Wadsworth Cengage Learning, 2013. ISBN: 978-1-111-83099-1.

[20] Shachar Kaufman, Saharon Rosset, and Claudia Perlich. "Leakage in data mining". In: *Journal* (2011). ISSN: ISSN. DOI: 10.1145/2020408.2020496.

[21] John D. Kelleher, Aoife DArcy, and Brian Mac Namee. *Fundamentals of Machine Learning for Predictive Data Analytics: algorithms, worked examples, and case studies*. 1st ed. Camebridge, MA: The MIT Press, 2015. URL: https://mitpress.mit.edu/books/fundamentals-machine-learning-predictive-data-ana%20lytics.

[22] G. Lippi and M. Plebani. "Laboratory abnormalities in patients with COVID-2019 infection". In: *Clin Chem Lab Med* 58.7 (2020), pp. 1131–1134. ISSN: 1437-4331 (Electronic) 1434-6621 (Linking). DOI: 10.1515/cclm-2020-0198. URL: https://www.ncbi.nlm.nih.gov/pubmed/32119647,%20https://www.degruyter.com/downloadpdf/journals/cclm/58/7/article-p1131.pdf.

[23] Gilles Louppe. "Understanding Random Forests: From Theory to Practice". Thesis. 2014. DOI: http://dx.doi.org/10.13140/2.1.1570.5928.

[24] G. Lu and J. Wang. "Dynamic changes in routine blood parameters of a severe COVID-19 case". In: *Clin Chim Acta* 508 (2020), pp. 98–102. ISSN: 1873-3492 (Electronic) 0009-8981 (Linking). DOI: 10.1016/j.cca.2020.04.034. URL: https://www.ncbi.nlm.nih.gov/pubmed/32405079,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7217800/pdf/main.pdf.

[25] A. J. Masino et al. "Machine learning models for early sepsis recognition in the neonatal intensive care unit using readily available electronic health record data". In: *PLoS One* 14.2 (2019), e0212665. ISSN: 1932-6203 (Electronic) 1932-6203 (Linking). DOI: 10.1371/journal.pone.0212665. URL: https://www.ncbi.nlm.nih.gov/pubmed/30794638.

[26] The Association for Clinical Biochemistry Medicine and Laboratory. *Aspartate Aminotransferase (AST) — Lab Tests Online*. 2021. URL: https://labtestsonline.org/tests/aspartate-aminotransferase-ast (visited on 01/21/2021).

[27] The Association for Clinical Biochemistry Medicine and Laboratory. *C-Reactive Protein - Understand the Test*. 2020. URL: https://labtestsonline.org.uk/tests/c-reactive-protein.

[28] The Association for Clinical Biochemistry Medicine and Laboratory. *Full Blood Count - Understand the Test*. 2020. URL: https://labtestsonline.org.uk/tests/full-blood-count-fbc (visited on 10/25/2020).

[29] The Association for Clinical Biochemistry Medicine and Laboratory. *Lactate Dehydrogenase (LDH) - Understand the Test*. 2021. URL: https://labtestsonline.org/tests/lactate-dehydrogenase-ld (visited on 01/21/2021).

[30] Christoph Molnar. *Interpretable Machine Learning*. Lulu.com, 2020. ISBN: 978-0-244-76852-2. URL: https://christophm.github.io/interpretable-ml-book/logistic.html.

[31] T. P. Morris, I. R. White, and P. Royston. "Tuning multiple imputation by predictive mean matching and local residual draws". In: *BMC Med Res Methodol* 14 (2014), p. 75. ISSN: 1471-2288 (Electronic) 1471-2288 (Linking). DOI: 10.1186/1471-2288-14-75. URL: https://www.ncbi.nlm.nih.gov/pubmed/24903709,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4051964/pdf/1471-2288-14-75.pdf.

[32] D. P. Oran and E. J. Topol. "Prevalence of Asymptomatic SARS-CoV-2 Infection : A Narrative Review". In: *Ann Intern Med* 173.5 (2020), pp. 362–367. ISSN: 1539-3704 (Electronic) 0003-4819 (Linking). DOI: 10.7326/M20-3012. URL: https://www.ncbi.nlm.nih.gov/pubmed/32491919.

[33] R. B. Parikh, Z. Obermeyer, and A. S. Navathe. "Regulation of predictive analytics in medicine". In: *Science* 363.6429 (2019), pp. 810–812. ISSN: 1095-9203 (Electronic) 0036-8075 (Linking). DOI: 10.1126/science.aaw0029. URL: https://www.ncbi.nlm.nih.gov/pubmed/30792287.

[34] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[35] P. Poletti et al. "Age-specific SARS-CoV-2 infection fatality ratio and associated risk factors, Italy, February to April 2020". In: *Euro Surveill* 25.31 (2020). ISSN: 1560-7917 (Electronic) 1025-496X (Linking). DOI: 10.2807/1560-7917.ES.2020.25.31.2001383. URL: https://www.ncbi.nlm.nih.gov/pubmed/32762797,%20https://air.unimi.it/retrieve/handle/2434/763526/1557654/poletti%20eurosurv-25-3%201-1.pdf.

[36] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1993. ISBN: 9781558602380.

[37] J. R. Quinlan. "Improved Use of Continuous Attributes in C4.5". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 77–90. ISSN: 1076-9757. DOI: 10.1613/jair.279. URL: https://www.jair.org/index.php/jair/article/download/10157/24078.

[38] J. R. Quinlan. "Induction of decision trees". In: *Machine Learning* 1.1 (1986), pp. 81–106. ISSN: 0885-6125 1573-0565. DOI: 10.1007/bf00116251. URL: https://link.springer.com/content/pdf/10.1007/BF00116251.pdf.

[39] Dagmar Reiche. *Roche Lexikon Medizin*. 5th ed. Munich: Elsevier Urban und Fischer, 2003. ISBN: 978-3-437-15156-9.

[40] L. Rokach and O. Maimon. "Top-Down Induction of Decision Trees Classifiers—A Survey". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 35.4 (2005), pp. 476–487. ISSN: 1094-6977. DOI: 10.1109/tsmcc.2004.843247. URL: https://ieeexplore.ieee.org/document/1522531/.

[41] T. Saito and M. Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PLoS One* 10.3 (2015), e0118432. ISSN: 1932-6203 (Electronic) 1932-6203 (Linking). DOI: 10.1371/journal.pone.0118432. URL: https://www.ncbi.nlm.nih.gov/pubmed/25738806,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4349800/pdf/pone.0118432.pdf.

[42] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning*. 2014. ISBN: 9781107298019. DOI: 10.1017/cbo9781107298019.

[43] S. S. Shapiro and M. B. Wilk. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3-4 (1965), pp. 591–611. ISSN: 0006-3444 1464-3510. DOI: 10.1093/biomet/52.3-4.591.

[44] S. Sun et al. "Abnormalities of peripheral blood system in patients with COVID-19 in Wenzhou, China". In: *Clin Chim Acta* 507 (2020), pp. 174–180. ISSN: 1873-3492 (Electronic) 0009-8981 (Linking). DOI: 10.1016/j.cca.2020.04.024. URL: https://www.ncbi.nlm.nih.gov/pubmed/32339487,%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7194694/pdf/main.pdf.

[45] C. B. F. Vogels et al. "Analytical sensitivity and efficiency comparisons of SARS-CoV-2 RT-qPCR primer-probe sets". In: *Nat Microbiol* 5.10 (2020), pp. 1299–1305. ISSN: 2058-5276 (Electronic) 2058-5276 (Linking). DOI: 10.1038/s41564-020-0761-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/32651556,%20https://www.nature.com/articles/s41564-020-0761-6.pdf.

[46] D. Wang et al. "Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China". In: *JAMA* 323.11 (2020), pp. 1061–1069. ISSN: 1538-3598 (Electronic) 0098-7484 (Linking). DOI: 10.1001/jama.2020.1585. URL: https://www.ncbi.nlm.nih.gov/pubmed/32031570,%20https://jamanetwork.com/journals/jama/articlepdf/2761044/jama_wang_2020_oi_20001%209.pdf.

[47] M. Weinstock et al. "Chest X-ray findings in 636 ambulatory patients with COVID-19 presenting to an urgent care center: a normal chest X-ray is no guarantee". In: *JUCM* 10 (2020), pp. 13–18. URL: https://www.jucm.com/documents/jucm-covid-19-studyepub-april-2020.pdf/.

[48] I. R. White, P. Royston, and A. M. Wood. "Multiple imputation using chained equations: Issues and guidance for practice". In: *Stat Med* 30.4 (2011), pp. 377–99. ISSN: 1097-0258 (Electronic) 0277-6715 (Linking). DOI: 10.1002/sim.4067. URL: https://www.ncbi.nlm.nih.gov/pubmed/21225900,%20https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4067.

[49] L. Wynants et al. "Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal". In: *BMJ* 369 (2020), p. m1328. ISSN: 1756-1833 (Electronic) 0959-8138 (Linking). DOI: 10.1136/bmj.m1328. URL: https://www.ncbi.nlm.nih.gov/pubmed/32265220,%20https://www.bmj.com/content/bmj/369/bmj.m1328.full.pdf.