Data-Driven Inference of COVID-19 Clinical Outcome

Joaquín Salas[a], Dagoberto Pulido[a], Omar Montoya[a] and Isaac Ruiz[b,*]

[a] *Instituto Politécnico Nacional*
[b] *Instituto Mexicano del Seguro Social*

ARTICLE INFO

Keywords:
COVID-19 Final Diagnosis
Important Feature Selection
COVID-19 outcome classifier

ABSTRACT

At the final stage of a COVID-19 infection, patients either improve or die. Knowing the outcome could offer some guidelines for tracking the evolution of patients, improving attention, making arrangements, and assigning resources. Given the available wealth of data, an alternative is to perform the analysis of existing clinical information using data-driven methods. Provided a clinical record, we aim to extract the characteristics that distinguish between those COVID-19 patients that improve and those who die. In our approach, we select the relevant features using the algorithm of Boruta, a feature selector wrapper method. Boruta generates an importance score that allows ranking the descriptors. Using the extracted features, we train classifiers to analyze new cases. We assess the performance of the classifiers using *Precision-Recall* and ROC curves and establish the ranges at which risk assessment permits effective decision-making. The Boruta algorithm obtained 21 features deemed important out of 61 initial ones. Our best performing classifier resulted in an area under the ROC and Precision-Recall curves of $0.959 \pm 2.4 \times 10^{-3}$ and $0.934 \pm 6.4 \times 10^{-3}$, respectively, at one standard deviation.

1. Introduction

COVID-19 is an infectious disease caused by the SARS-CoV-2 virus. The most frequent symptoms at the beginning are fever, cough, and fatigue. Other signs that appear afterward include sputum production, myalgia, headache, hemoptysis, diarrhea, dyspnea, and lymphopenia [1]. Clinical diagnosis allows advancing in the detection of suspicious cases. However, the definitive diagnosis of COVID-19 is made by quantitative reverse-transcription polymerase chain reaction (qRT-PCR) assays, which are molecular techniques that consist of obtaining a large number of copies of a nucleic acid fragment. In principle, amplification makes it possible to identify or rule out SARS-CoV-2 in the sample with a high probability, requiring, on average, 5.2 copies of the envelope protein marker of the E genome and 3.8 copies of the RdRp genome polymerase for a probability detection rate of 95% [2]. Once a physician has declared a case confirmed positive, patient evolution will determine the outcome, whether the patient improves or dies. In this paper, we propose a data-driven approach to track symptomatology and assess the patient's chances for survival.

As the amount of information available to us increases, machine learning emerges as an alternative that provides supplemental information to infer outcomes based on predictors. Its response speed is its strength in the presence of exponential growth in the number of infections. Thus, the machine learning community is responding through methodologies that allow predicting the change in the rate of infection over time [3], drug discovery [4], genome classification [5], and the prediction of patient survival [6]. In this paper, we employ machine learning techniques for extracting the most important features defining survival. Then, we proceed to construct classifiers to predict a patient outcome based on its clinical record.

2. Feature Extraction

Given the database, in our study, we carry out an analysis to select the variables that best distinguish between improved and deceased COVID-19 patients. With this aim, we use the implementation of the `Boruta` algorithm [7] contained in **R**, a wrapper-type method that accepts variable importance measure methods as underlying classifiers. Many other selection methods consider all the characteristics at the same time, such as Mutual Information and Super Casual Correlation [8]. However, we selected Boruta's algorithm because it considers multiple associations simultaneously, without an exhaustive search.

Boruta's algorithm begins by defining shadow variables for each predictor. Then, the method shuffles the shadow variable's values over all the observations of the same predictor. Boruta's algorithm compares the relative importance of the original variables and the shadow variables. If the importance value of an original variable is statistically higher or lower than the maximum importance of the best-valued shadow variable, the method labels the predictor as important or not important. Then, Boruta discards the tagged variables and repeats the procedure until it tags all the variables.

The Random Forest algorithm naturally handles a measure of importance [9]. For a tree, a variable's importance is related to the increase in precision achieved with the partitions made in the nodes, typically measured in terms of the Gini index or the entropy/information gain. For forest, the method averages the importance of a predictor over all trees. However, due to the Random Forest algorithm's haphazard nature, each run may result in a different selection. To attack this problem, we run the Boruta algorithm k times, recording the characteristics and relative importance assigned to

[*] Corresponding author: Joaquín Salas. CICATA Querétaro. Instituto Politécnico Nacional. Cerro Blanco 141, Colinas del Cimatario. Querétaro. México. 76090. Tel.: +52-55-5729-6000 × 81015

✉ jsalasr@ipn.mx (J. Salas)
ORCID(s):

them. With each of the resulting subsets of attributes, one could construct a classifier, remaining to identify which offers the best generalization capacity.

3. Classifiers

Using the characteristics considered important, we built a set of classifiers using Random Forests, Support Vector Machine and Extreme Gradient Boosting.

Random Forests [9] are a special extension of Decision Trees where one trains a particular tree with a subset of the dataset available. The algorithm selects a random subset of the predictors at each split, and one creates a large number of trees, naturally constructing an ensemble of classifiers. Like other tree-based methods, Random Forests handle equally well regression and classification problems and are similarly well adapted to deal with numerical and categorical variables. Furthermore, the independence of the ensemble reduces the variance during voting for classification or averaging for regression.

Support Vector Machines [10] are constructed on the criterion of maximizing the distance between discriminatory hyperplanes and the classes. Once maximized the margin, the closer points at either side of the hyperplane are called support vectors. To allow for misclassification there is a cost budget C. In the general case, where there is no hyperplane dividing the classes, one project the data over non-linear, possibly higher-dimensional spaces, where the expectation is that the data will be linearly separable.

Extreme Gradient Boosting [11] combines the approach of creation a large number of decision trees in Random Forest with the construction of a cascade of weak classifiers typical of boosting approaches [12]. The approach incorporates sequential tree growing where, like in boosting, the weight is adjusted for the predictors wrongly classified by the previous tree. In turn, just like in Random Forest, each sequential tree is constructed with a random subsample of the data.

During the learning process, we split the data into training and validation sets. We repeat this operation l times for each classifier, in a process known as cross-validation, each time performing a partition with the same percentage, but a random subset of the data. Next, for each classifier, we evaluated performance using ROC curves and *Precision-Recall*. As a result of repeated performance evaluation, we have variations that we can summarize statistically.

4. Experimental Results

In our experiments, we aim to determine the ability of the feature selector to produce relevant descriptors and reliable classifiers.

4.1. Experimental Setup

For our study, we take 93,668 records from a database provided by the Secretariat of Health, corresponding to the government archives from January 6, 2020, to June 26, 2020, containing 24,716 confirmed positive cases of COVID-19 after a qRT-PCR test.

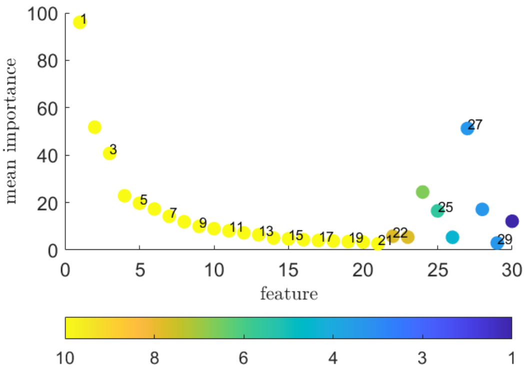


Figure 1: Feature Selection. Important characteristics in the process of discriminating between confirmed and discarded cases for COVID-19 using the Boruta algorithm (best seen in color, see Table 4.1 for further details).

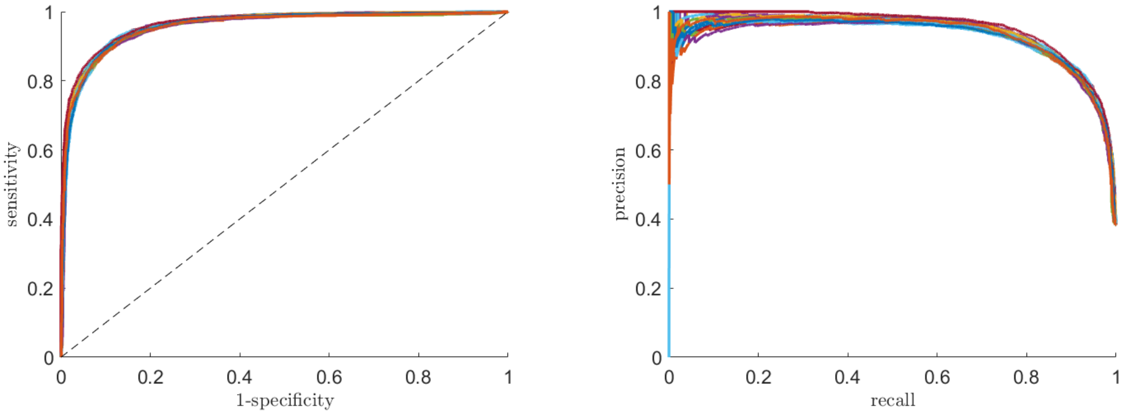
A database record contains 192 variables. Some of these variables are associated with administrative information, such as the registration number; post-diagnosis treatment of COVID-19, such as whether a hospital admitted the patient to the Intensive Care Unit (ICU); or personal information, such as name and address. As a pre-process, we remove these features to leave only those that would make up a prediagnostic clinical verification list. After removing the related administrative features, the number of predictors was 61. We further split the data, keeping only the records for which an outcome existed (16,300). We also split the data among those with improvement diagnoses (10,056, 61.7%) and those who deceased (6,244, 38.3%).

The computer we use to process the information runs on the Windows 8.1 operating system and consists of a 64-bit Intel i7-3770 processor at a clock frequency of 3.4Hz, and operating with 16MB of RAM. We developed our computer programs in **R** version 3.6.3.

4.2. Feature Selection and Classifier Construction

To assess the important features, we ran Boruta's algorithm 10 times leaving 21 features deemed important (see Figure 1 and Table 4.1). We noticed that these 21 features consistently appeared as a result of the feature selector, 9 features appeared in some of the iterations, while predictors 31 were always rejected as unimportant. By far, age is the most important feature, 96.1± 1.00 at one standard deviation.

With each of these features, one could construct classifiers, remaining to identify their performance measured as their generalization capacity. To do this, we first fine-tune the hyperparameters for each of the models. In the case of Random Forest, we refined the hyperparameters corresponding to the number of variables randomly sampled at each split and the number of trees in the random forest. We use ten-fold cross-validation, for the former, and grid search, for



(a) ROC, AUC $= 0.959 \pm 5.0 \times 10^{-3}$

(b) *Precision-Recall*, AUC $=0.934 \pm 6.4 \times 10^{-3}$

Figure 2: Evaluation of performance. The ROC and *Precision-Recall* curves summarize combinations of elements in the confusion table at different threshold values.

Table 1
Feature importance. The importance is the mean decrease in accuracy for the classifier when that variable is left out.

ID	Name	Importance	ID	Name	Importance	ID	Name	Importance
1	age (years)	96.1± 1.00	8	cyanosis	11.9± 2.72	15	endotraqueal intubing	4.7± 0.67
2	sex (male, female)	51.8± 0.39	9	polypnea	10.0± 2.13	16	hypertension	4.3± 0.64
3	cephalalgia	40.7± 13.63	10	chronic diseases	9.0± 1.31	17	cardiovascular	4.0± 0.46
4	myalgia	22.9± 2.21	11	COPD	8.1± 0.85	18	kidney	3.8± 0.44
5	arthralgia	19.7± 2.38	12	diabetes	7.2± 1.05	19	liver	3.5± 0.41
6	rhinorrhea	17.3± 1.45	13	asthma	6.4± 0.84	20	neurology	3.3± 0.29
7	dyspnoea	14.2± 2.51	14	vaccine	5.0± 0.66	21	cancer	2.6± 0.32

Table 2
Performance result for the RF, SVM and the XGBoost classifiers.

classifier	AUC *Precision-Recall*	AUC ROC
SVM	$0.887 \pm 4.4 \times 10^{-3}$	$0.827 \pm 5 \times 10^{-3}$
RF	$0.714 \pm 11.2 \times 10^{-3}$	$0.828 \pm 4.5 \times 10^{-3}$
XGBoost	**0.934** $\pm 6.4 \times 10^{-3}$	**0.959** $\pm 2.4 \times 10^{-3}$

the latter, resulting in 4 (four) for the number of variables and 1,500 for the number of trees. For the SVM, we fine-tuned the cost to 128 using 10-fold cross-validation and testing 10 (ten) different cost values. In the case of the XGBoost algorithm, it requires considerably more hyperparameters to tune. We fine-tuned using grid search on the learning parameter (0.1), the fraction of predictors used for splitting at each level (0.33), the maximum depth for the trees (10), the fraction of the data to sample (0.5), the minimum number of observations in a terminal node (1), the minimum reduction in the loss function to grow a new node in the tree (0.1) and the number of trees (500).

Using these hyperparameters and the characteristics considered significant, we built classifiers based on Random Forests, Support Vector Machine, and Extreme Gradient Boosting. For learning, we split the data into 50% for training and 50% for validation. We repeat this process 30 times, each performing a partition with the same percentage, but using uniform random sampling. Next, for the classifier, we evaluated its performance using ROC and *Precision-Recall* curves. As a result of repeated performance evaluation, we have variations that we summarize in Table 2. With $0.959 \pm 2.4 \times 10^{-3}$ and $0.934 \pm 6.4 \times 10^{-3}$ the XGBoost gives the largest values for the ROC Area under the Curve (AUC) and the *Precision-Recall* AUC. In these results, we express uncertainty at one standard deviation (see Figure 2).

An intriguing question corresponds to the limits of performance at which the classifier makes relatively mild or no mistakes. These values define the levels at which we are quite confident in the outcome. For instance, at a mean specificity of 0.99, we have a mean sensitivity of 0.565, while at a mean sensitivity of 0.99, we have a mean specificity of 0.498. On the other hand, at the mean recall of 0.99, we have a mean precision of 0.55, and at a mean precision of 0.97, we have a mean recall of 0.6[1].

Conclusion

Through a classifier, one establishes the mapping between input characteristics and an output class. Given a confirmed positive case to COVID-19, this document shows that it is feasible to construct a high-performance classifier to

[1] To facilitate the confirmation of our findings and improve on our framework, we are making our code available at `github.com/joaquinsalas/COVID19-DataDriven-Classifier`. Also, a live demonstration is available at `http://imagenes.cicataqro.ipn.mx`.

predict the patient's outcome. Although characteristics associated with COVID-19 are now widely known [6], our research quantifies the importance of each of them related to the final patient diagnosis. Likewise, the family of classifiers presented here offers the ability to assign a measure of certainty regarding the prediction. A decision-maker may set a threshold for assignment to one or the other class balancing risks and costs. Decision-makers may play out these trade-offs using the performance curves as they permit to know the decision thresholds that allow assigning an outcome to a patient with COVD-19. Our tool can thus have a critical complementary value for clinical diagnosis and follow up.

During a fast-paced epidemy, such as COVID-19, it is essential to have unbiased assessments for decision-making. In this research, we show that a data-driven approach may offer an alternative that is fast, inexpensive, and error-bound.

References

[1] Hussin Rothan and Siddappa Byrareddy. The Epidemiology and Pathogenesis of Coronavirus Disease (COVID-19) Outbreak. Journal of Autoimmunity, page 102433, 2020.

[2] Victor Corman, Olfert Landt, Marco Kaiser, Richard Molenkamp, Adam Meijer, Daniel Chu, Tobias Bleicker, Sebastian Brünink, Julia Schneider, and Marie Schmidt. Detection of 2019 Novel Coronavirus (2019-nCoV) by Real-Time RT-PCR. Eurosurveillance, 25(3), 2020.

[3] Samir Bandyopadhyay and Shawni Dutta. Machine Learning Approach for Confirmation of COVID-19 Cases: Positive, Negative, Death and Release. medRxiv, 2020.

[4] Yiyue Ge, Tingzhong Tian, Sulin Huang, Fangping Wan, Jingxin Li, Shuya Li, Hui Yang, Lixiang Hong, Nian Wu, Enming Yuan, et al. A Data-Driven Drug Repositioning Framework Discovered a Potential Therapeutic Agent Targeting COVID-19. bioRxiv, 2020.

[5] Gurjit Randhawa, Maximillian Soltysiak, Hadi El-Roz, Camila de Souza, Kathleen Hill, and Lila Kari. Machine Learning using Intrinsic Genomic Signatures for Rapid Classification of Novel Pathogens: COVID-19 Case Study. bioRxiv, 2020.

[6] Li Yan, Hai Zhang, Yang Xiao, Maolin Wang, Chuan Sun, Jing Liang, Shusheng Li, Mingyang Zhang, Yuqi Guo, and Ying Xiao. Prediction of Survival for Severe COVID-19 Patients with Three Clinical Features: Development of a Machine Learning-based Prognostic Model with Clinical Data in Wuhan. medRxiv, 2020.

[7] Miron Kursa and Witold Rudnicki. Feature Selection with the Boruta Package. Journal of Statistical Software, 36(11), 2010.

[8] E. Tuv, A. Borisov, G. Runger, and K. Torkkola. Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. Journal of Machine Learning Research, 10:1341–1366, 2009.

[9] Leo Breiman. Random Forests. Machine Learning, 45(1):5–32, 2001.

[10] Johan Suykens and Joos Vandewalle. Least Squares Support Vector Machine Classifiers. Neural Processing Letters, 9(3):293–300, 1999.

[11] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In International Conference on Knowledge Discovery and Data Mining, pages 785–794, 2016.

[12] Yoav Freund and Robert Schapire. Experiments with a New Boosting Algorithm. In ICML, volume 96, pages 148–156. Citeseer, 1996.