Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science

Quantifying the effects of biomarkers and comorbidities in predicting SARS Cov-2 associated mortality in hospitalized patients in Mexico

---

THESIS to obtain the **DEGREE** of
MASTER OF DATA SCIENCE

A thesis presented by:
Claudia Soto Alvarez

Thesis Advisors:
Dr. Diana Paola Montoya Escobar

Tlaquepaque, Jalisco, December, 2021

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science Approval Form

Thesis Title: **Quantifying the effects of biomarkers and comorbidities in predicting SARS Cov-2 associated mortality in hospitalized patients in Mexico**

Author: **Claudia Soto Alvarez**

Thesis Approved to complete all degree requirements for the Master of Science Degree in Data Science.

Thesis Advisor, **Dr. Diana Paola Montoya Escobar**

Thesis Reader, **Dr. Riemann Ruiz Cruz**

Thesis Reader, **Dr. Esteban Rodríguez Jímenez**

Academic Advisor, **Mc. Juan Carlos Martínez Alvarado**

Tlaquepaque, Jalisco, December, 2021

Quantifying the effects of biomarkers and comorbidities in predicting SARS Cov-2 associated mortality in hospitalized patients in Mexico

Claudia Soto Alvarez

Abstract

In this retrospective quasi-experimental, cohort study, the biomarkers, demographics, and clinical characteristics of the adult inpatients with laboratory-confirmed COVID-19 from Hospital Regional 110 (Guadalajara, Mexico) were analyzed who were hospitalized over the year 2020, between April 15 (i.e. when the first patient was admitted) to December 31 and had a definite outcome (discharged or dead), to establish the most important variables for the models.

In this study, 5 different Classifiers were used: Random Forest, Support Vector Machine, XGBoost, Naïves Bayes, and Symbolic Classifier to classify the outcome of the patients and also to quantify the effect of biomarkers and comorbidities in predicting SARS-CoV-2 positive associated mortality in hospitalized patients. Also, the Symbolic Transofmer was implemented to try to improve the performance of our model. As the dataset includes a big percentage of missing values, we proposed two models, one excluding the missing values and the other including all the missing values.

The Random Forest was implemented to obtain the variable importance, and also to the capacity of the model to handle the missing values.

The metrics ROC AUC and Accuracy were used to train the models, along with the Bayesian Optimization to tune the hyperparameters and to measure the performance.

Keywords- Biomarkers, Random Forest, Covid-19

Acknowledgment

First and foremost I am extremely grateful to my thesis advisor Dr. Paola Montoya who guided me throughout this project, and her continous support and patience. I am deeply grateful to doctor Rafael Soto for his contribution to this study.

I would also like to thank to all my teachers for their lectures, feedbacks and guidance during my Master's. My gratitude extends to the Instituto Tecnológico y de Estudios Superiores de Occidente for the scholarship.

Lastly, I wish to show my appreciation to the Hospital Regional 110, especially to doctor Fernando Morales for trusting in this project.

Contents

List of Figures

List of Tables

This thesis is dedicated to my parents and my brother.

1 Introduction

Contents

In late December 2019, a new coronavirus was identified in Wuhan (China), causing severe respiratory disease, including pneumonia. It was initially named Novel Coronavirus, and The World Health Organization (WHO) advised the following language associated with the virus. The virus causing the infection has been named - severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).

The disease caused as a result of infection is named - coronavirus disease (COVID-19). COVID-19 has been categorized as an airborne High Consequence of Infections Disease. SARS-CoV-2 has been spreading among people globally, causing a pandemic all over the world. As a viral infection, antibiotics are not an effective treatment. Nowadays there are vaccinations available.

At the end of 2020, after one year when the coronavirus pandemic began, in Mexico has been reported 104,00[1] deaths and in Jalisco state 4,732 deaths[2]. Studies about the Covid 19 in Mexican population during 2020 in the healthcare are few, as well the application of machine learning models.

Therefore, the implementation of a machine learning model could help to understand the behavior of the disease.

[1] WHO | World Health Organization, c. URL https://www.who.int/

[2] COVID-19 Tablero México, d. URL https://datos.covid-19.conacyt.mx/index.php

1.1 Background

COVID-19 is caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). Both, the new virus and the disease

were unknown before the outbreak in Wuhan in December 2019. It produces flu-like symptoms, including fever, cough, dyspnea, myalgia, and fatigue. Sudden loss of smell and taste (without mucus being the cause) has also been observed. In severe cases, it is characterized by pneumonia, acute respiratory distress syndrome, sepsis, and septic shock, leading to death.[3]

Numerous laboratory findings have been linked to COVID-19 disease to date. Hematological variables were the first findings reported in patients with SARS-CoV-2 who presented biochemical alterations. Leukocytosis with neutrophilia and lymphopenia were the most frequent findings in patients with the severe form of the infection. In addition, patients who manifest this stage of the disease present a dysregulation of the immune response that is responsible for the outcome of a systemic inflammatory response of great magnitude that will be harmful to the host.[4]

Platelets, also known as thrombocytes, are blood cells. They form in the bone marrow, sponge-like tissue in your bones. Platelets play an important role in blood clotting. In patients with COVID-19 what is commonly observed is the alteration of the coagulation mechanisms, a relatively moderate decrease in the platelet count is observed.[5]

Lymphocytes is a type of immune cell made in the bone marrow; it is found in blood and lymphatic tissue. Lymphocytes play an important role in maintaining the immune system. In SARS-CoV-2 infection, studies show marked lymphopenia. Lymphopenias, a higher neutrophil/lymphocyte ratio, fewer monocytes, eosinophils, and basophils have been observed compared to patients without symptoms of the disease.[6]

D-Dimer is the end product of fibrin degradation that occurs through plasmin and helps the formation and production of thrombi. Different studies have found significantly elevated D-Dimer levels in patients with severe COVID-19, compared with those whose symptoms were milder and with healthy subjects.[7]

The presence of acute kidney damage has been reported in up to 15% of positive COVID-19 patients, with a mortality rate that varies from 60-90% depending on the author and population studied, of this group of patients 35% had disease chronicle.[8]

In a study carried out in New York, United States in positive COVID-19 patients, it was identified that those with Chronic Kidney Disease as well as atrial fibrillation or heart failure had higher mortality.[9]

As a result of muscle metabolic processes and through creatine and creatine phosphate, creatinine is produced endogenously. Serum creatinine serves as a marker to estimate renal glomerular filtration since it is eliminated through the kidney. It serves as an estimator of chronic

[3] Manuel Ramón Pérez Abreu, Jairo Jesús Gómez Tejeda, and Ronny Alejandro Dieguez Guach. Características clínico-epidemiológicas de la COVID-19. 19(2). ISSN 1729-519X. URL http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1729-519X2020000200005&lng=es&nrm=iso&tlng=es

[4] M. Salazar, J. Barochiner, W. Espeche, and I. Ennis. Covid-19, hipertensión y enfermedad cardiovascular. 37 (4):176–180. ISSN 18891837. DOI: 10.1016/j.hipert.2020.06.003. URL https://linkinghub.elsevier.com/retrieve/pii/S1889183720300659

[5] Luis Edgardo López and María Daniela Mazzucco. Alteraciones de parámetros de laboratorio en pacientes con sars-cov-2. page 15

[6] Luis Edgardo López and María Daniela Mazzucco. Alteraciones de parámetros de laboratorio en pacientes con sars-cov-2. page 15

[7] Erika Fabiola Saquinga Jame. Título: Dímero d, tiempo de protrombina y plaquetas en la valoración del paciente con covid-19. page 57

[8] Sreedhar Adapa, Avantika Chenna, Mamtha Balla, Ganesh Prasad Merugu, Narayana Murty Koduri, Subba Rao Daggubati, Vijay Gayam, Srikanth Naramala, and Venu Madhav Konala. COVID-19 Pandemic Causing Acute Kidney Injury and Impact on Patients With Chronic Kidney Disease and Renal Transplantation. 12(6):352–361. ISSN 1918-3003. DOI: 10.14740/jocmr4200. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7295554/

[9] Samira S. Farouk, Enrico Fiaccadori, Paolo Cravedi, and Kirk N. Campbell. COVID-19 and the kidney: What we think we know so far and what we don't. 33(6):1213–1218. ISSN 1121-8428, 1724-6059. DOI: 10.1007/s40620-020-00789-y. URL https://link.springer.com/10.1007/s40620-020-00789-y

kidney disease or acute kidney injury. Elevated serum creatinine has been associated as an independent risk factor for mortality in patients with COVID-19. [10]

Symptomatic infection can range from mild to severe clinical picture. The mild clinical picture presents a clinical evolution of two weeks to three weeks from the onset of symptoms to recovery, this being the most common presentation reported in around 81% of COVID-19 patients. [11]

The severe picture of COVID-19 develops in the first 12 days of the disease with a 20% prevalence. During the evolution of the condition, a percentage of around 12 to 24% require invasive mechanical ventilation, which is a procedure that consists of securing the patient's airway through endotracheal intubation or tracheostomy, and partially or totally supplying the effort Ventilation, as well as gas exchange in patients with acute respiratory failure, this last complication can occur in patients with severe symptoms on day 15 of the event and 88% of these patients die around day 18.[12](Figure 1.1)

[10] Yichun Cheng, Ran Luo, Kun Wang, Meng Zhang, Zhixiang Wang, Lei Dong, Junhua Li, Ying Yao, Shuwang Ge, and Gang Xu. Kidney disease is associated with in-hospital death of patients with COVID-19. 97(5):829–838. ISSN 0085-2538. DOI: 10.1016/j.kint.2020.03.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7110296/

[11] Pieter A. Cohen, Lara E. Hall, Janice N. John, and Alison B. Rapoport. The Early Natural History of SARS-CoV-2 Infection. 95(6):1124–1126. ISSN 00256196. DOI: 10.1016/j.mayocp.2020.04.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0025619620303797; and Zunyou Wu and Jennifer M. McGoogan. Characteristics of and Important Lessons From the Coronavirus Disease 2019 (COVID-19) Outbreak in China: Summary of a Report of 72 314 Cases From the Chinese Center for Disease Control and Prevention. 323 (13):1239–1242. ISSN 0098-7484. DOI: 10.1001/jama.2020.2648. URL https://doi.org/10.1001/jama.2020.2648

[12] Safiya Richardson, Jamie S. Hirsch, Mangala Narasimhan, James M. Craw-

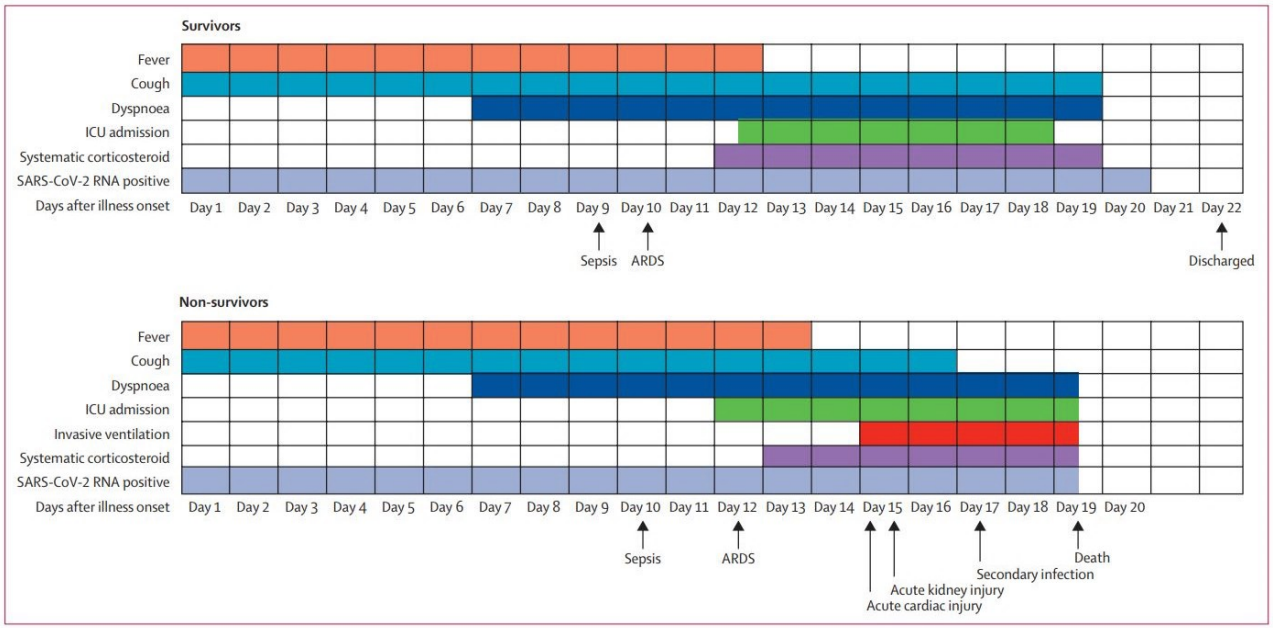

Figure 1.1: Clinical courses of major symptoms in patients hospitalised with COVID-19 *Reprinted and adapted from Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China Zhou*

The reason why some patients with severe COVID-19 persist and present a torpid evolution may be due to the elevated and increasing severe inflammatory response since they continue to increase markers of inflammation and pro-inflammatory cytokines compared to patients with a mild condition.[13]

COVID-19 is highly contagious, therefore, has sparse quickly; as of November 2021, Mexico has 3,867,976 cumulative confirmed cases and 292,850 deaths caused by COVID-19. In the case of Jalisco, there are more than 163,000 cumulative confirmed cases and 17,500 deaths, Figure 1.2.

Jalisco has had two main steps in the pandemic (October 2020-January 2021 and April 2021-July 2021). In these two phases of the pandemic, some hospitals have been saturated, as they have a limited number of beds assigned to patients with COVID-19. In addition, the number of deaths caused by COVIDs increased drastically.

Some patients required intubation and intensive care those arriving at the hospital. During their stay in the hospital, different clinical analyses were performed to determine, according to medical tests, the patient's state of health and their evolution during their stay in the hospital.

[13] Changsong Wang, Kai Kang, Yan Gao, Ming Ye, Xiuwen Lan, Xueting Li, Mingyan Zhao, and Kaijiang Yu. Cytokine Levels in the Body Fluids of a Patient With COVID-19 and Acute Respiratory Distress Syndrome: A Case Report. 173(6):499–501. ISSN 0003-4819. DOI: 10.7326/L20-0354. URL https://www.acpjournals.org/doi/full/10.7326/L20-0354

1.2 *Related work*

Several machine learning algorithms have been applied successfully in the context of medicine, to be able to make a medical diagnostics (i.e. medical imaging),such system can be seen as a classification task.

In recent times, the application of computational or machine intelligence in medical diagnostics has become quite common. While various statistical techniques may be applied in medical data classification, the major drawback of these approaches is that they depend on some assumptions (e.g., related to the properties of the relevant data) for their successful application. To know the properties of the dataset is a difficult task and sometimes it is not feasible. On the other hand, soft computing based approaches are less dependent on such knowledge. [14]

With the design of algorithms that are able to generalize from observed evidences, and to make predictions about unseen data, machine learning can be applied in many fields such as computer aided diagnosis, detection and segmentation[15]. In the last decade, random forests became a popular ensemble learning algorithm, as they achieve state-of-the-art performance in numerous computer vision tasks.

Table 1.1 shows a summary of related work using machine learning applied in medicine and some of them using biomarkers.

[14] Olivier Pauly. Random Forests for Medical Applications. page 204

[15] Md. Zahangir Alam, M. Saifur Rahman, and M. Sohel Rahman. A Random Forest based predictor for medical data classification using feature ranking. 15:100180. ISSN 2352-9148. DOI: 10.1016/j.imu.2019.100180. URL https://www.sciencedirect.com/science/article/pii/S235291481930019X

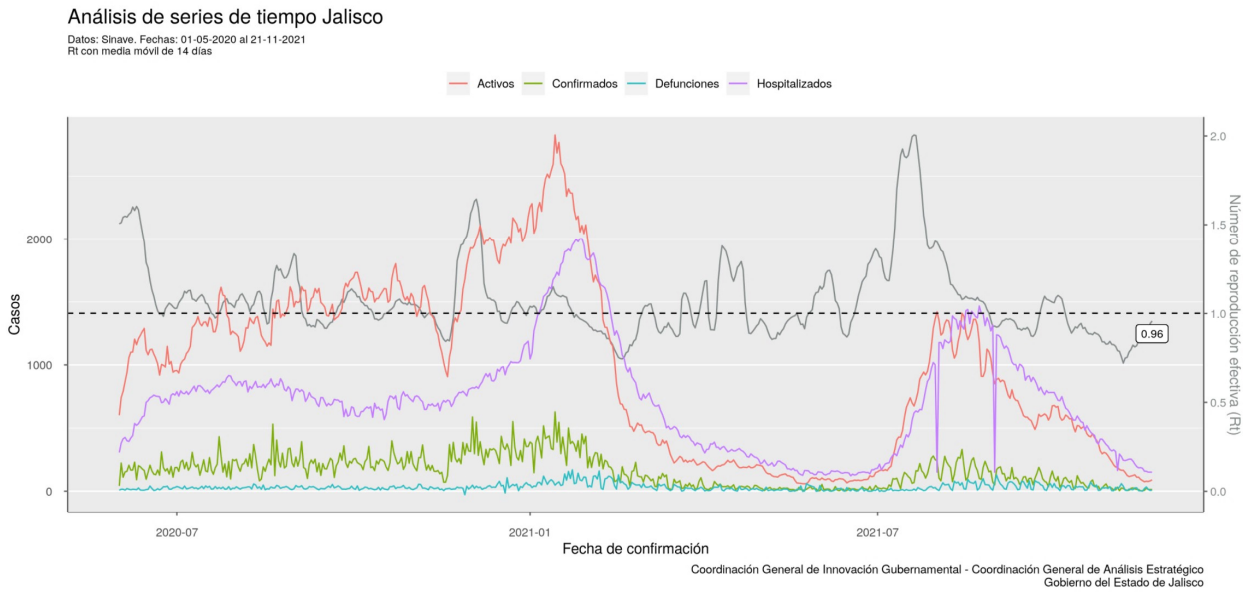

Figure 1.2: Jalisco Covid-19 trend *Official data from Jalisco government*

1.3 *Justification*

Since COVID-19 was considered a pandemic by the World Health Organization, world governments have coordinated information flows and issued guidelines to contain the overwhelming effects of this disease. At the same time, the scientific community is continually seeking information about transmission mechanisms, the clinical spectrum of the disease, new diagnoses, and strategies for prevention and treatment. One of the challenges is understanding the behavior of the laboratory tests results of biomarkers in hospitalized patients with COVID-19. Whose technique adopted for detecting the genetic material of COVID-19 requires equipment and specialized human resources, making it an expensive procedure.

We hypothesize that machine learning techniques can be used to determine the propensity to survive (or not) of hospitalized patients for COVID-19 through the joint analysis of popular laboratory tests clinical parameters. Machine learning techniques, such as Random Forest, and Support Vector Machines, enable the creation of disease

Table 1.1: Related Work summary.

Techniques	Objective	Author
Dynamic Bayesian Network	Detect probabilistic relationships among clinical variables (demographic variables, hematic biometry biomarkers, etc.) and identify risk factors related to survival and loss of vital functions.	Zandonà et al.[2019]
Random Forest	General methodology to classified diseases. Ranked and select the features to construct the predictor (applied in 10 diseases).	Alam et al.[2019]
XGBoost	The predictive power of biomarkers data to enhance the diagnosis of depression cases.	Sharma and Verbeke[2020]
KNN, SVM, Random Forest, Neural Network, Naïves Bayes AdaBoost	Predict the early lung tumor based on the metabolomic biomarkers features.	Xie et al.[2021]
Random Forest	Organ localization, segmentation, lesion detection and image categorization in medical imaging.	Pauly[2012]

prediction models and artificial intelligence techniques to analyze clinical parameters. Thus, we evaluated the existing correlations between laboratory parameters and the hospitalization days and developed classification models.

Studies during 2020, when the pandemic started, in the Mexican population are few, and the applications of mathematical models and machine learning in the health area are scarce in our country.

The implementation of a Machine Learning model could help to understand the behavior of the disease through a joint vision between the doctor and the use of technological tools, in a practical way and being able to serve for the taking of decisions, to focus them from the beginning to the people with greater risk, and therefore, less consumption of economic resources both for the country and for the Institute.

1.4 *Problem statement*

Covid-19 represents one of the greatest challenges in the recent history of public health, it has spread throughout the world, affecting more than 200 countries. So far it has not been possible to describe the behavior and evolution of the disease in the Mexican population.

Therefore, the implementation of a Machine Learning model could provide a better understanding of the behavior of this disease in the

body and speed up the response of medical personnel, since Machine Learning models in the health area in Mexico scarce.

Which has led us to generate the following research question: Can a Machine Learning Model quantify the effect of biomarkers and comorbidities in predicting SARS-CoV-2 positive associated mortality in hospitalized patients?

1.5 Objectives

1.5.1 General objective

To quantify with a Machine Learning model the effects of biomarkers and comorbidities in predicting SAR Cov-2 associated mortality in hospitalized patients in Mexico.

1.5.2 Specific objectives

- To explore the data and identify the effects of COVID-19 in the biomarkers and comorbidities.
- To obtain the best model to predict the discharge of the patient using multiple classifiers and optimize the hyperparameters.
- To apply a model using all the datasets including the missing values and optimize the hyperparameters.

2 Theoretical Framework

Contents

In this work, we consider five types of classification algorithms: Random Forest, XGBoost, Support Vector Machine, Naïves Bayes and the Symbolic Classifier. The Bayesian Optimization approach was used to tune the hyperparameters for each model. Also, the Symbolic Transformer was implemented to try to improve the results. Three of the Classifiers used are described below.

2.1 Random Forest

First introduced in 1960's, the decision trees have been widely used in several disciplines given that they are a very powerful algorithm, simple and also efficient for extracting knowledge from data.The extracted knowledge can be easily understood, interpreted in the form of a readable decision trees. They are excellent tools for choosing between several course of action.[1]

Decision trees are represented as acyclic graphs (Figure 2.1) with a root node and successive child nodes connected by directional branches (edges). Each node of the tree is associated with a decision and the leaf nodes are generally associated with an outcome or class label.In the case of a binary decision tree, each node gives the statement of the decision to be taken or the comparison to be made. There are two outgoing edges from the nodes.[2]

Common usages of decision tree models include the following:

[1] Wolfgang Ertel. *Introduction to Artificial Intelligence*. Undergraduate Topics in Computer Science. Springer International Publishing. ISBN 978-3-319-58486-7 978-3-319-58487-4. DOI: 10.1007/978-3-319-58487-4. URL http://link.springer.com/10.1007/978-3-319-58487-4

[2] M. Narasimha Murty and V. Susheela Devi. *Pattern Recognition*, volume 0 of *Undergraduate Topics in Computer Science*. Springer London. ISBN 978-0-85729-494-4 978-0-85729-495-1. DOI: 10.1007/978-0-85729-495-1. URL http://link.springer.com/10.1007/978-0-85729-495-1

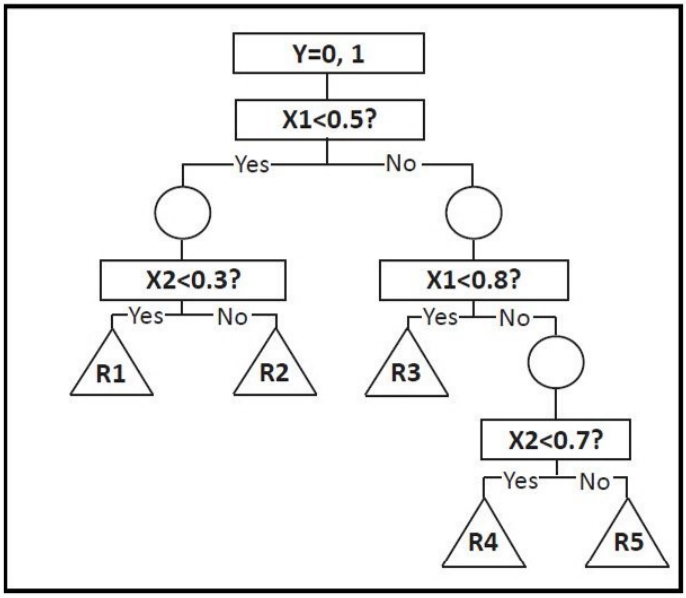

Figure 2.1: Simple decision tree model based on binary classification *Reprinted and adapted from Decision tree methods: applications for classification and prediction* SONG and LU

- **Variable selection:** Decision tree methods can be used to select the most relevant input variables that should be used form decision tree models.
- **Assessing the relative importance of variables:** Variable importance is computed based on the reduction of model accuracy when the variable is removed.
- **Handling of missing values:** Decision tree analysis can deal with missing data without needing to resort to imputation, it can classify missing values as a separate category that can be analyzed with the other categories.

Some of the limitations of the decision tree is that it can be subject to over-fitting and under-fitting, the strong correlation between input variables and the design time could be large. [3]

Random forest are an ensemble learning algorithm proposed by Breiman (2001), that constructs *ntrees* number of randomized decision trees during the training phase that are used to obtain *ntrees* predictions, these predictions are average to give the forest's prediction.

The random forest approach is a bagging method where deep trees, fitted on bootstrap samples, are combined to produce an output with lower variance. Generating bootstrap samples introduces a random component into the tree building process and reduce the variance of the prediction.

[3] Yan-yan SONG and Ying LU. Decision tree methods: Applications for classification and prediction. 27 (2):130–135. ISSN 1002-0829. DOI: 10.11919/j.issn.1002-0829.215044. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/

Despite taking bootstrap samples, also it sample over the predictors and keep only a random subset of them to build the tree. The idea behind randomly sampling predictors during training is to decorrelate the tree in the forest. Since the algorithm randomly selects predictors at each split, tree correlation will necessarily be lessened. [4] Figure 2.2 shows the random fores flowchart from ensemble methods: bagging, boosting and stacking.

[4] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York. ISBN 978-1-4614-6848-6 978-1-4614-6849-3. DOI: 10.1007/978-1-4614-6849-3. URL http://link.springer.com/10.1007/978-1-4614-6849-3

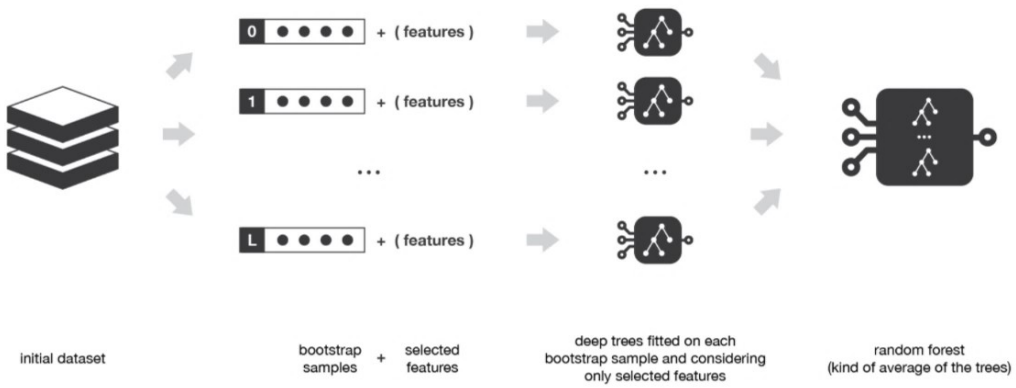

Figure 2.2: Random forest method *Reprinted and adapted from ensemble methods Rocca*

Aside from being simple to use, the method is generally recognized for its accuracy and its ability to deal with small sample sizes, high-dimensional feature spaces and complex data structures.[5]

One of the main advantages of random forest is that it is that has just a few parameters to tune, the hyperparameters tuned in this study were:

- **The number of trees to grow for the forest (*ntrees*):** Random forest is protected from over-fitting, therefore, the model will not be adversely affected if a large number of trees are built for the forest. But, the cost is primarily computational time and only if the number of predictors and number of samples are large, do computational burdens become an issue.

- **Number of predictors (*mtries*):** Chooses the number of predictors in each partition of the tree would seem to be a key tuning

[5] Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. 43(4). ISSN 0090-5364. DOI: 10.1214/15-AOS1321. URL https://projecteuclid.org/journals/annals-of-statistics/volume-43/issue-4/Consistency-of-random-forests/10.1214/15-AOS1321.full

parameter that should affect how well random forests performs. For classification problems, the recommendation is starting to set *mtries* to the square root of the number of P (number of predictors): $mtries \approx \sqrt{P}$. Then, trying a few more or a few less as well can be instructive. But a lot depends on the number of predictors and how strongly they are related. If the correlations are substantial, it can be useful to reduce the number of predictors sampled for each partitioning decision.

- **Maximum tree depth (*max_depth*):** In random forests, all trees are created independently, each tree is created to have maximum depth, and each tree contributes equally to the final model. Higher values will make the model more complex.
- **Minimum number of observations for a leaf (*min_rows*):** The minimum number of samples in each tree's leaf. The splitting process of the tree continues within each newly created partition until the stopping criteria is met, the minimum number of samples in a node or the maximum tree depth.[6] [7]

Random Forest also can measure variable importance. If we change a single predictor's input value and reclassify the record, we can determine that the predictor's importance is based on the new classification. This is done using OOB (Ouf of Bag) data.[8]

In this study, the Random Forest of the library H2O[9] was used.

2.2 *Support Vector Machine*

Many real-world problems involve prediction over two classes. An Support Vector Machine (SVM) is an abstract learning machine that will learn from a training data set and attempt to generalize and make correct predictions on new data.

In training a classifier, usually, we try to maximize classification performance for the training data. However, if the classifier is too fit for the training data, leas to overfitting.

For a binary problem, a SVM is trained so that the direct decision function maximizes the generalization ability, namely, the m-dimensional input space x is mapped into the l-dimensional $l \geq m$ feature space z. Then in z, the quadratic programming problem is solved to separate two classes by the optimal separating hyperplane, the points closest to this separation hyperplane are the vectors of support[10] (Figure 2.3).

Let us define a linear model of the form:

$$y(x) = w^T \varphi(x) + b$$

[6] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York. ISBN 978-1-4614-6848-6 978-1-4614-6849-3. DOI: 10.1007/978-1-4614-6849-3. URL http://link.springer.com/10.1007/978-1-4614-6849-3

[7] Richard A. Berk. *Statistical Learning from a Regression Perspective*. Springer Texts in Statistics. Springer International Publishing. ISBN 978-3-319-44047-7 978-3-319-44048-4. DOI: 10.1007/978-3-319-44048-4. URL http://link.springer.com/10.1007/978-3-319-44048-4

[8] Barrett E Lowe. The Random Forest Algorithm with Application to Multispectral Image Analysis. page 79

[9] Distributed Random Forest (DRF) — H2O 3.34.0.3 documentation, a. URL https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html

[10] Shigeo Abe. *Support Vector Machines for Pattern Classification*. Advances in Pattern Recognition. Springer London. ISBN 978-1-84996-097-7 978-1-84996-098-4. DOI: 10.1007/978-1-84996-098-4. URL http://link.springer.com/10.1007/978-1-84996-098-4

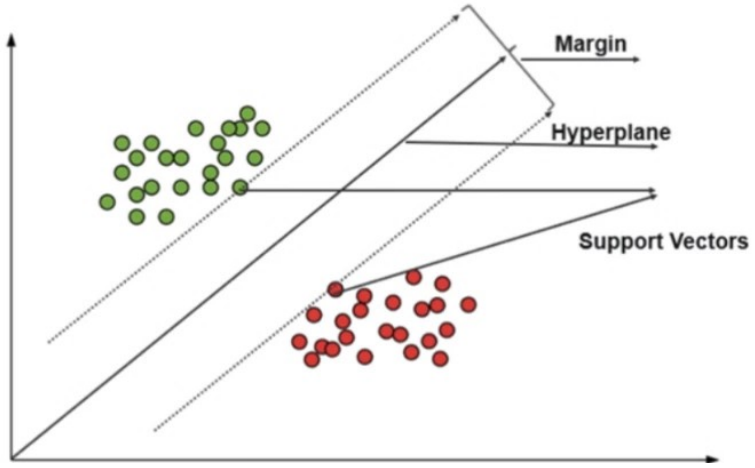

Figure 2.3: Support Vector Machine *Reprinted and adapted from Analytics Vidhya: The A-Z guide to Support Vector Machine* [202]

Where $\varphi(x)$ denotes a fixed feature-space transformation, b is the bias or ofsset of the hyperplane from the origin in input space. The training data set comprises N input vectors $x_1, ..., x_N$ with corresponding target values $y_1, ..., y_N$ where $y_k \in \{-1, 1\}$ and new data points x are classified according to the sign of $y(x)$.

Most real-life datasets contain noise, and an SVM can fit this noise leading to poor generalization. The effects of outliers and noise can be reduced by introducing a soft margin.

Optimization Problem (Lagrangian):

$$L(w, b, \xi; \alpha, \lambda) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k - \sum_{k=1}^{N} \alpha_k \{y_k [w^T \varphi(x) + b] - 1 + \xi_k\} - \sum_{k=1}^{N} \lambda_k \xi_k$$

where $\alpha_k, \lambda_k \geq 0$ since the inequality contraints.

The dual problem is stated as:

$$\max D(\alpha) = -\frac{1}{2} \sum_{k,l=1}^{N} y_k y_l k(x_k, x_l) \alpha_k \alpha_l + \sum_{k,l=1}^{N} \alpha_k$$

$$s.t. \sum_{k=1}^{N} \alpha_k y_k = 0$$

$$0 \leq \alpha_k \leq c, k = 1, ..., N$$

The SVM methodology can be summarized as follows: If it is required to classify a set of data (represented in an m-dimensional plane) no linearly separable, said data set is mapped to a larger space where linear separation is possible (this is done using functions called Kernel).

In this new plane, we are looking for a hyperplane that is capable of separate the input data into two classes; the plane must have the most considerable possible distance to the points of both classes. [11]

[11] Colin Campbell and Yiming Ying. Learning with Support Vector Machines. 5(1):1–95. ISSN 1939-4608, 1939-4616. DOI: 10.2200/S00324ED1V01Y201102AIM010. URL http://www.morganclaypool.com/doi/abs/10.2200/S00324ED1V01Y201102AIM010; and Johan A. K. Suykens, editor. *Least Squares Support Vector Machines*. World Scientific. ISBN 978-981-238-151-4

2.3 Genetic Programming

The generational Genetic Programming algorithm gives us a method to evolve candidate programs using the principles of natural selection.

There are 3 main steps involved in the algorithm.[12]

[12] Vimarsh Sathia, Venkataramana Ganesh, and Shankara Rao Thejaswi Nanditale. Accelerating Genetic Programming using GPUs. URL http://arxiv.org/abs/2110.11226

- **Selection:** It is the step where individual candidates from a given population are chosen for evolution into, We decide on a set of programs to evolve into the next generation, using a selection criterion. One of the selection schemes it the *Tournament selection*, winning programs are determined by selecting the best programs from a subset of the whole population. Multiple tournaments are held until we have enough programs selected for the next generation.

- **Mutation:** Before promoting the programs selected in the previous step to the next generation, we perform some genetic operations on them. The winning programs after selection are not directly carried forward into the next generation. Rather, mutations or genetic operations are applied on the selected programs, in order to produce new offspring.

- **Evaluation:** Once the population for the next generation is decided after selection and mutation, the fitness of all programs in the new generation is recomputed.

In this document we applied two Genetic Programming, The Symbolic Classifier and the Symbolic Transformer from the gplearn[13] library, the two of them are described below.

[13] API reference — gplearn 0.4.1 documentation, b. URL https://gplearn.readthedocs.io/en/stable/reference.html#symbolic-classifier

2.3.1 Symbolic Classifier

The goal in symbolic classification is to find a model that estimates the target class value (discrete) from the value of input variables (discrete, continuous).

This algorithm begins by building a population of naive random formulas to represent a relationship. The formulas are represented as tree-like structures with mathematical functions being recursively applied to variables and constants. Each successive generation of programs is then evolved from the one that came before it by selecting the fittest individuals from the population to undergo genetic operations such as crossover, mutation or reproduction.

The output of the program is transformed through a sigmoid function in order to transform the numeric output into probabilities of each class. In essence this means that a negative output of a function means that the program is predicting one class, and a positive output predicts the other.[14]

[14] Trevor Stephens. Gplearn Documentation. page 57

2.3.2 *Symbolic Transformer*

The Symbolic Transformer works slightly differently to Symbolic Classifier. The Symbolic Transformer is a supervised transformer that begins by building a population of naive random formulas to represent a relationship. The formulas are represented as tree-like structures with mathematical functions being recursively applied to variables and constants. Each successive generation of programs is then evolved from the one that came before it by selecting the fittest individuals from the population to undergo genetic operations such as crossover, mutation or reproduction. The final population is searched for the fittest individuals with the least correlation to one another.

The transformer seeks an indirect relationship that can then be exploited by a second estimator. Essentially, this is automated feature engineering and can create powerful non-linear interactions that may be difficult to discover in conventional methods. The transformer looks to maximize the correlation between the predicted value and the target. This is done through either the Pearson product-moment correlation coefficient or the Spearman rank-order correlation coefficient. In both cases the absolute value of the correlation is maximized in order to accept strongly negatively correlated programs.

The Spearman correlation is appropriate if your next estimator is going to be tree-based, such as a Random Forest or Gradient Boosting Machine. If you plan to send the new transformed variables into a linear model, it is probably better to stick with the default Pearson correlation. The Symbolic Transformer looks at the final generation of the evolution and picks the best programs to evaluate. The number of programs it will look at is controlled by the hall_of_fame parameter. From the hall of fame, it will then whittle down the best programs to the least correlated amongst them as controlled by the n_components parameter. The correlation between individuals within the hall of fame uses the same correlation method, Pearson or Spearman, as used by the evolution process.[15]

[15] Trevor Stephens. Gplearn Documentation. page 57

2.4 *Bayesian Optimization*

Bayesian optimization is an approach to optimizing objective functions that take a long time to evaluate. The ability to

optimize expensive black-box derivative-free functions makes Bayes Optimization extremely versatile.

The algorithm works by constructing a statistical model for modeling the objective function, this model is Gaussian Process, provides a posterior probability distribution that best describes the function ($f(x)$) that we want to optimize. Each time we observe f at a new point, this posterior distribution is updated. After evaluating the objective according to an initial space-filling experimental design, this points are used iteratively to allocate the remainder of a budget of N function evaluations, as shown in Algorithm 1. [16]

[16] Peter I. Frazier. A Tutorial on Bayesian Optimization. URL http://arxiv.org/abs/1807.02811

Algorithm 1 Basic pseudo-code for Bayesian optimization

Place a Gaussina process prior on f
Observe f at n_0 points according to an initial space-filling experimental design. Set $n = n_0$
while $n \leq N$ **do**
  Update the posterior probability distribution on f using all available data
  Let x_n be a maximizer of the acquistion function over x, where the acquisition function is computed using the current posterior distribution
  Observe $y_n = f(x_n)$
  Increment n
end while
Return a solution: either the point evaluated with the largest $f(x)$ or the point with the largest posterior mean.

As the number of observations grows, the posterior distribution improves, and the algorithm becomes more certain of which regions in parameter space are worth exploring and which are not, as seen in the Figure 2.4 below.

This process is designed to minimize the number of steps required to find a combination of parameters that are close to the optimal combination. To do so, this method uses a proxy optimization problem (finding the maximum of the acquisition function) that, albeit still a hard problem, is cheaper (in the computational sense) and common tools can be employed.

There are many parameters you can pass to maximize, nonetheless, the most important ones are:

- **n_inter:** How many steps of bayesian optimization you want to perform. The more steps the more likely to find a good maximum you are.

- **init_points:** How many steps of random exploration you want to perform. Random exploration can help by diversifying the exploration space.

The library that was used in this paper is bayesian-optimization.[17]

[17] fernando. Bayesian Optimization. URL https://github.com/fmfn/BayesianOptimization

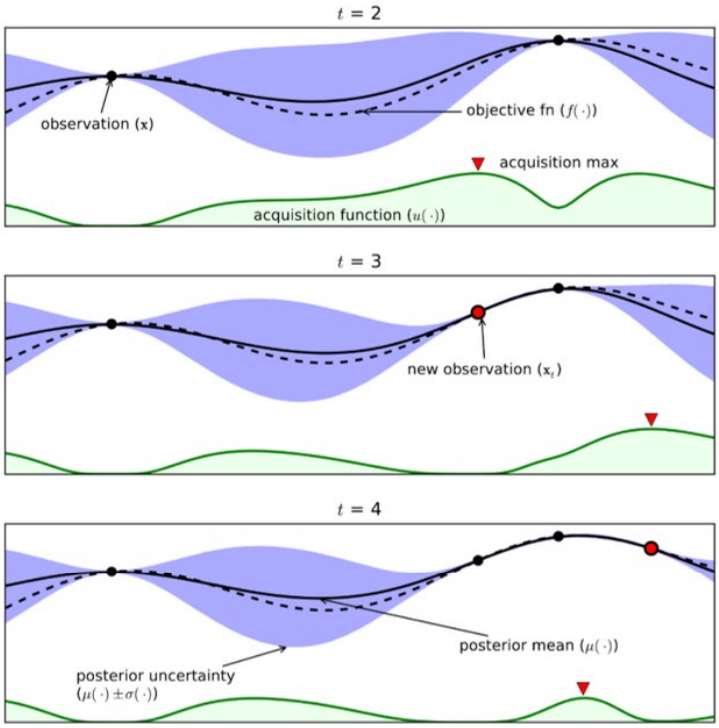

Figure 2.4: An example of using Bayesina otpimization. *Reprinted and adapted from A tutorial on Bayesian Optimization of Expensive Cost Function, with Application to Active User Modeling and Hierarchical Reinforcement Learning* Brochu et al.

2.5 Metrics

In a binary classification problem, evaluation metrics are calculated from True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). The evaluation of the models designed to solve classification problems is normally based on the confusion matrix (Table 2.1).

Table 2.1: Confusion Matrix

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

For the purposes of the present study, two main metrics were used to evaluate the models and to tune the hyperparameters.

Accuracy is the most common and simplest evaluation metric in classification modeling. This metric measures the percentage of the correct predictions made by the model, considering both classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

It's recommended to use this metric when the data are balanced.

ROC AUC Before defining this metric, first, we need to define the other two metrics:

- **Sensitivity:** This metric is used to measure the fraction of positive classes that are correctly classified.

$$Sensitivity = \frac{TP}{TP + FN}$$

- **Specificity:** Contrary to sensitivity, this metric is used to measure the fraction of negative classes that are correctly classified.

$$Specificity = \frac{TN}{TN + FP}$$

In a binary classification, it is generally considered the decision threshold at the probability of 0.5, which means,elements with probability 0.51 to 1 will be classified in the same class, while elements with probabilities between 0 to 0.49 will be in the same class.

Setting different thresholds for classifying positive class for data points will inadvertently change the Sensitivity and Specificity of the model. And one of these thresholds will probably give a better result than the others. The metrics change with the changing threshold values.

The **Receiver Operator Characteristic (ROC)** curve plots **Sensitivity** against the complement of the **Specificity** at various threshold values. The **ROC** is useful in determining the appropriate threshold to maximize the relationship between sensitivity and specificity (Figure 2.5).

In a **ROC curve**, a higher X-axis value indicates a higher number of False positives than True negatives. While a higher Y-axis value indicates a higher number of True positives than False negatives. So, the choice of the threshold depends on the ability to balance between False positives and False negatives. Therefore **ROC AUC** represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting negative classes as negative and positive classes as positive.[18]

Also, three other metrics were used for reference in this study.

- **Precision:** This metric is the percentage of positive classes that are correctly estimated.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** This metric is the proportion or real positive classes that were correctly estimated by the model.

$$Recall = \frac{TP}{TP + FN}$$

[18] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York. ISBN 978-1-4614-6848-6 978-1-4614-6849-3. DOI: 10.1007/978-1-4614-6849-3. URL http://link.springer.com/10.1007/978-1-4614-6849-3

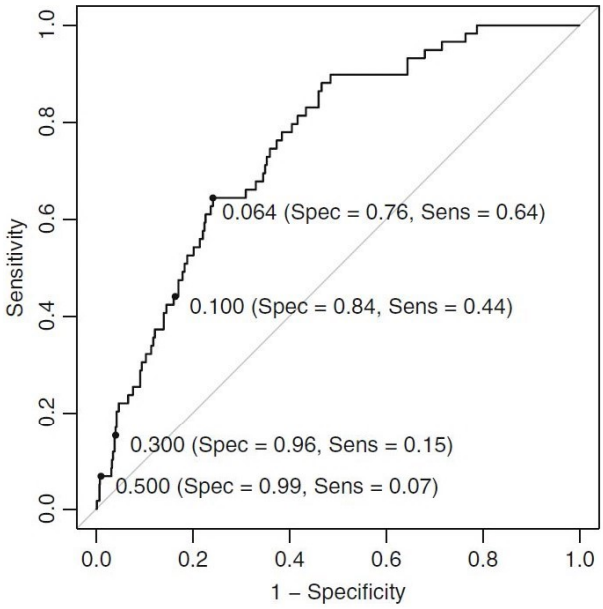

Figure 2.5: ROC curve for predicting the classes using the evaluation set. *Reprinted and adapted from Applied Predictive Modeling* Kuhn and Johnson

- **F1:** This metric is a weighting of the metrics precision and the recall. The F1 value assumes that precision and recall are equally important to us.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

3 Methods

Contents

In this retrospective quasi-experimental, cohort study, we include the adult inpatients with laboratory-confirmed COVID-19 from Hospital Regional 110 (Guadalajara, Mexico) (Institutional registration Number R-2021-1303-016), who were hospitalized over the year 2020, between April 15 (i.e. when the first patient was admitted) to December 31 and had a definite outcome (discharged or dead). Since some of the patients were discharged and a few days later were admitted again to the hospital for COVID-19, the database only includes the data for the last hospitalization. All the data is confidential and anonymous.

The methodology that was followed in this study was taken from Azure Architecture Center: [1]

- **Understand the problem.**
- **Data Acquisition and Understanding (exploring).**
- **Modeling.**
- **Deployment.**

For a better understanding, Figure 3.1 shows the methodology that was implemented.

[1] Kent Sharkey. What is the Team Data Science Process? - Azure Architecture Center. URL https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview

3.1 Dataset

The database includes records of more than 1,400 patients. Each record contains at least one hematic biometry and an arterial blood gases

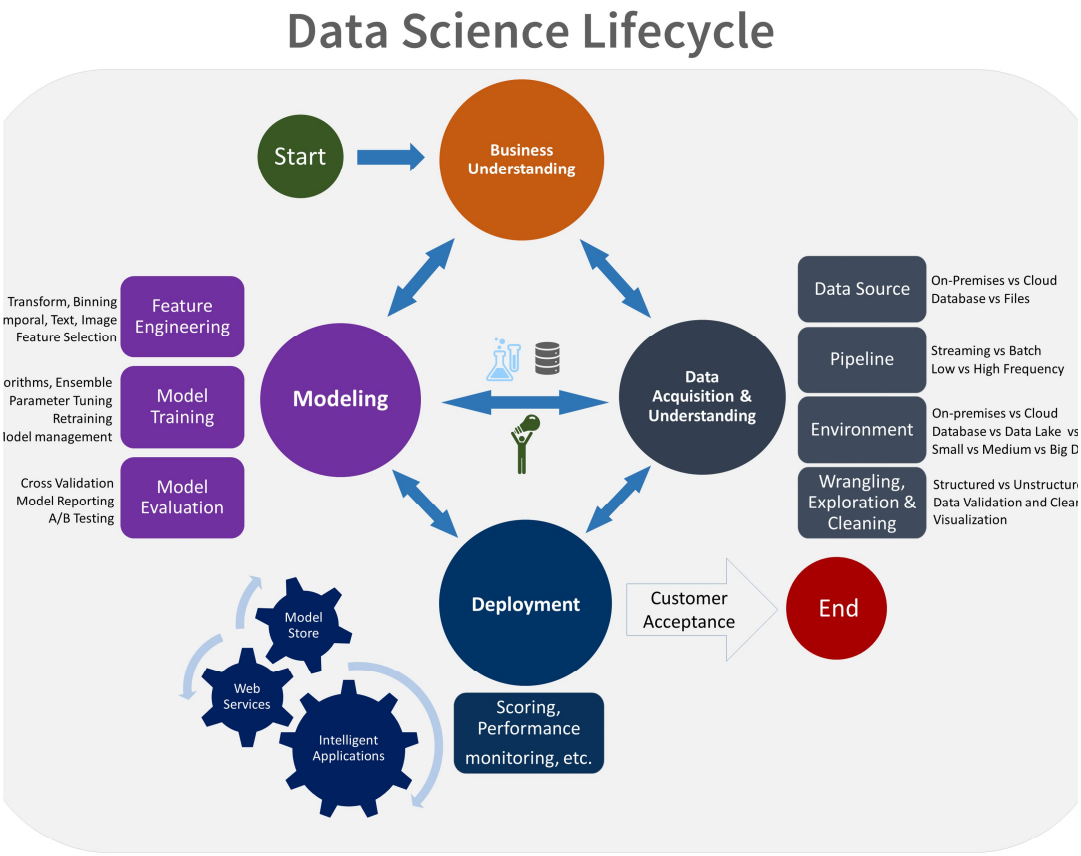

Figure 3.1: Visual representation of the Team Data Science Process lifecycle. *Reprinted from Azure Architecture Center by* Sharkey

test, the frequency of examinations was determined by the treating physician. Totaling more than 11,300 records.

The data contains 47 hematic biometry biomarkers, 7 arterial blood gases biomarkers, 15 variables of demographics and clinical characteristics of the patient (e. g. age, sex, comorbidities), 3 time-dependent variables (first and last day hospitalized, day of the examination), a variable that indicates if the patient was intubated and a definite outcome (death or discharge).

3.2 *Preprocessing*

3.2.1 *Data Cleaning*

For the data preprocessing, firstly, as hematic biometry test and arterial blood gases test, are two different examinations, they have distinct variables, each patient could have two exams ID with the same date, these tests were merged to only have one examination by day for each patient, also, if the patient had the same test on the same day, we only

kept the last test that was taken for that day.

The patients under the age of 40 were filtered due to not have enough records and we excluded biomarkers that were missing for more than 45% on the records, and those that from a medical point of view were irrelevant to this illness. With the arterial blood gases and D-dimer, we decided to keep the biomarkers even though of the missing values, between 45% and 34%, owing to their relevance in the illness.

Therefore, 33 hematic biometry biomarkers were excluded. The remained number of biomarkers reduces to 14 hematic biometry biomarkers and 7 arterial blood gases, in total 21 biomarkers.

Despite all the preprocessing, the missing data in all the biomarkers was still high. Since a change in a biomarker from one day to another is minimum, we decided to fill the missing data, which means that if the patient had a few biomarkers on one day and the next or the previous day those biomarkers were missing, the biomarkers were passed to the following or the previous day. However, if the row could not be filled and the missing data was more than 33%, this is, the row had more than 7 empty biomarkers, that row was deleted.

Table 3.1 shows the percentage of missing values after the data cleaning. Only the biomarkers with missing values are shown.

Variable	Missing values (%)
Lymphocytes	0.08
Monocytes	0.08
Neutrophils	0.08
Creatinine	2.09
Urea	2.16
Glucose	2.27
Sodium	5.3
Potassium	5.3
D-dimer	21.14
PH	21.76
Base excess	22.04
PO2	22.17
PCO2	22.22
HCO3	22.32
CO2	22.37
O2 Saturation	22.37

Table 3.1: Percentage of missing values.

In the case of the time-dependent variables, we have the first day and last day of the hospitalization in a date format, so the date was changed to the total day of hospitalization (the difference between the first day and the last day), but this variable is only for data exploration,

and it was excluded from the models. Then, the examination day was converted to day number that is the difference between the first day and the day of the examination.

On average, the total days hospitalized is 10 days, moreover, they are just a few patients that were hospitalized more than 25 days (less than 5%). Thus we are considering the last 25 days of examination for each patient.

Table 3.2: Demographics and clinical characteristics.

	Total (n=1,298)	Survivor (n=386)	Non-survivor (n=912)
Age, years (avg)	65	61	66
Sex			
Male	65.25%	67.10%	64.47%
Female	34.75%	32.90%	35.53%
Comorbidity	58.94%	49.74%	62.83%
Cancer	1.77%	0.78%	2.19%
Heart Disease	5.62%	5.70%	5.59%
Diabetes	31.97%	28.24%	33.55%
Hypertension	22.88%	22.54%	23.03%
Hypothyroidism	2.47%	2.07%	2.63%
Pneumopathy	4.01%	2.59%	4.61%
Chronic Kidney	24.19%	16.84%	27.30%
Kidney Acute	31.36%	24.09%	34.43%
AKI 1	20.42%	21.76%	19.85%
AKI 2	3.24%	1.04%	4.17%
AKI 3	7.70%	1.30%	10.42%
Intubation	45.22%	11.66%	59.43%
Total of days hospitalized (avg)	10	13	9

All the categorical data were converted into numeric features.In summary, the dimension of the final dataset is 1,298 patients, 3,884 records, and totaling 39 variables. In Table 3.3 are described all the variables.

3.2.2 *Data Exploration*

Following the data cleaning, we have 1,298 patients, each record represents a test in a different day, with different values in the biomarkers, therefore in this study each record represents a patient, thus we have 3,884 patients.

Figure 3.2 shows the age distribution, as we have mentioned before, the minimum age was set on 40 years, on average the survivors are younger (61 years old) than the non-survivors (66 years old).

Table 3.3: Variables description.

Variable	Description
ID*	Patient Identification.
Day Number	Day of the laboratory test after the hospitalization.
Total Days*	Total days hospitalized (excluded).
Sex	Gender (Male = 0, Female = 1).
Age	Age at hospitalized onset (age $\geq$ 40).
Class	Clinical outcomes (Dead = 0, Live = 1).
Erythrocytes	Red blood cell Hemoglobin-carrying.
Hemoglobin	Amount of oxygen-carrying protein in the erythrocytes.
Hematocrit	The percentage of red blood cells in a given volume of whole blood.
Platelets	Blood cells that play an important role in blood clotting.
Leukocytes	White blood cell responsible for maintaining the immune system.
Lymphocytes	Leukocyte family.
Monocytes	Leukocyte family.
Neutrophils	Leukocyte family.
Glucose	Blood sugar.
Urea	Waste products filtered out of the blood by the kidneys.
Creatinine	Waste products filtered out of the blood by the kidneys.
Sodium	Electrolyte.
Potassium	Electrolyte.
D-dimer	Protein fragment from the break-down of a blood clot.
PH	Measure of acid-base.
PCO2	Blood gas (carbon dioxide).
PO2	Blood gas (oxygen).
CO2	Carbon dioxide in blood.
HCO3	Electrolyte, associated also with acid-base (pH) imbalance.
Base Excess	Amount of acid required to normalize pH.
O2 Sat	Oxygen saturation in blood.
Cancer	Comorbidities (No = 0, Yes = 1).
Heart Disease	Comorbidities (No = 0, Yes = 1).
Diabetes	Comorbidities (No = 0, Yes = 1).
Hypertension	Comorbidities (No = 0, Yes = 1).
Hypothyroidism	Comorbidities (No = 0, Yes = 1).
Pneumopathy	Comorbidities (No = 0, Yes = 1).
Chronic Kidney	Comorbidities (No = 0, Yes = 1).
Kidney Acute	Disease (No = 0, Yes = 1).
AKI 1	Grade of acute kidney injury (No = 0, Yes = 1).
AKI 2	Grade of acute kidney injury (No = 0, Yes = 1).
AKI 3	Grade of acute kidney injury (No = 0, Yes = 1).
Intubation	Intubation (No = 0, Yes = 1).

* These variables were only used for exploration, therefore, there were excluded from the models

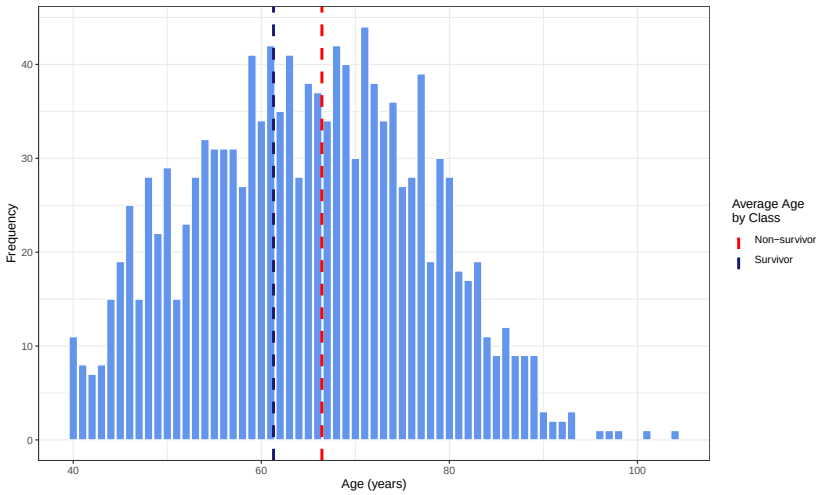

Figure 3.2: The age distribution of the patients.

On the other hand, Figure 3.3 shows the total number of days that a patient was hospitalized. As we have already pointed out, less than 5% patients were hospitalized more than 25 days, therefore only the last 25 days of examination for each patient are considered. The average hospitalization days for the survivors were longer (13 days) due to their recovery.

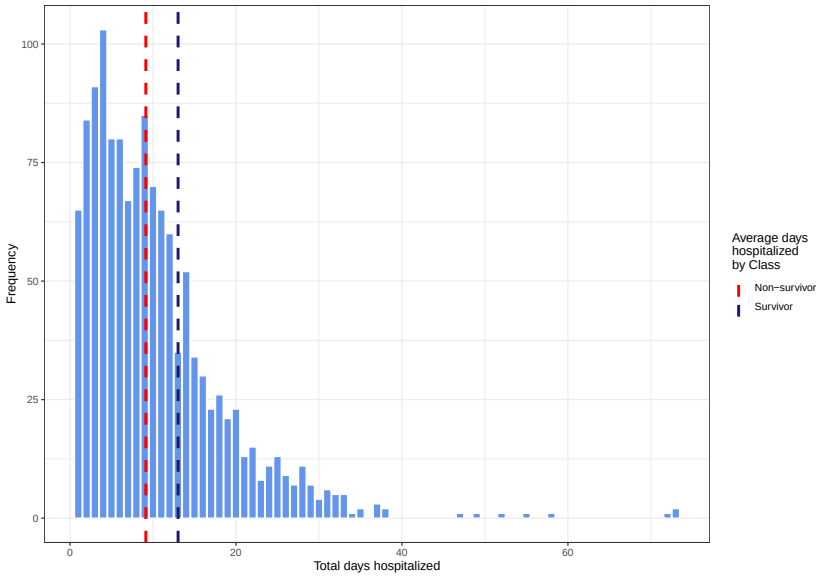

Figure 3.3: Hospitalized total days distribution by patients.

Taking the day 25 as the last day hospitalized, we can easy see in Figure 3.4 that 50% of the records are between the day 19 and the last day hospitalized.

There is big difference between Survivors and Non-survivors, more

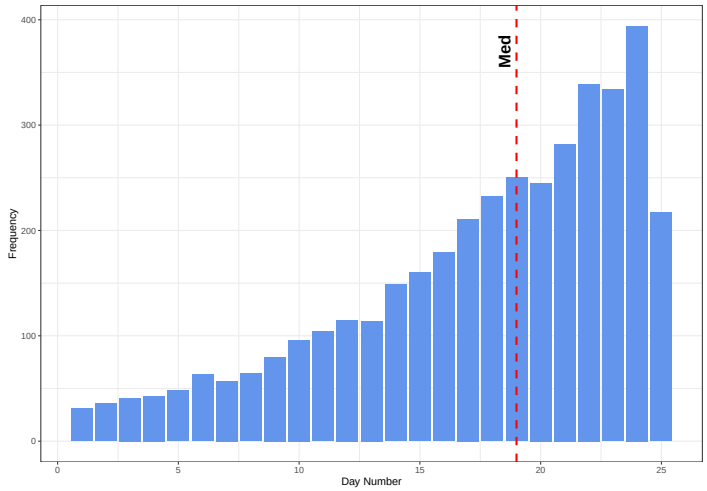

Figure 3.4: Distribution of the day of the laboratory test after the hospitalization.

than 59% of patients that died were intubated, whereas only 11.66% of the survivors were intubated (Figure 3.5). Intubation has been one of the most important variables but also can add noise into our model due to this variable being affected by some others factors.

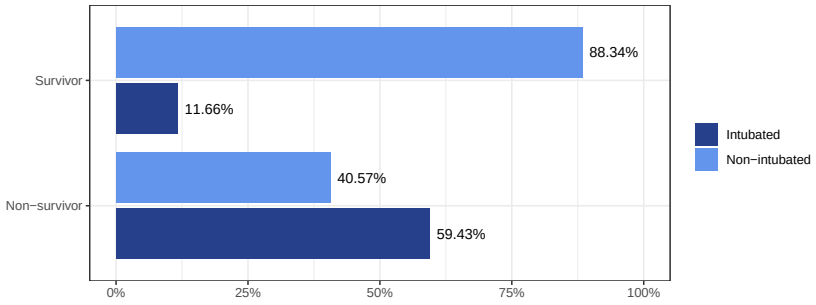

Figure 3.5: Percentage of patient intubated by Class.

Figure 3.6 displays the temporal changes in some of the most important biometry biomarkers that have shown significant differences between Non-survivors and survivors over time. On the left side, values are presented as median, error bars indicate the interquartile range, and, on the right error bars indicate ± 1 SE of the mean and in the background the scatter plot of all the records. As Figure 3.4 indicates, there are more records from the 15th onwards, owing to the average number of days hospitalized being 10 days, which means that more patients were admitted to the hospital around day 15. In Urea (A,B) (Figure 3.6) the patients Non-survivors were hospitalized with extremely higher levels, which indicates that the kidney is failing, in the last days the levels of the survivors tend to down, that is to say, the kidney starts to normalize, on the contrary of the Non-survivors that suggest an acute kidney injury (AKI1, AK2 or AKI3).

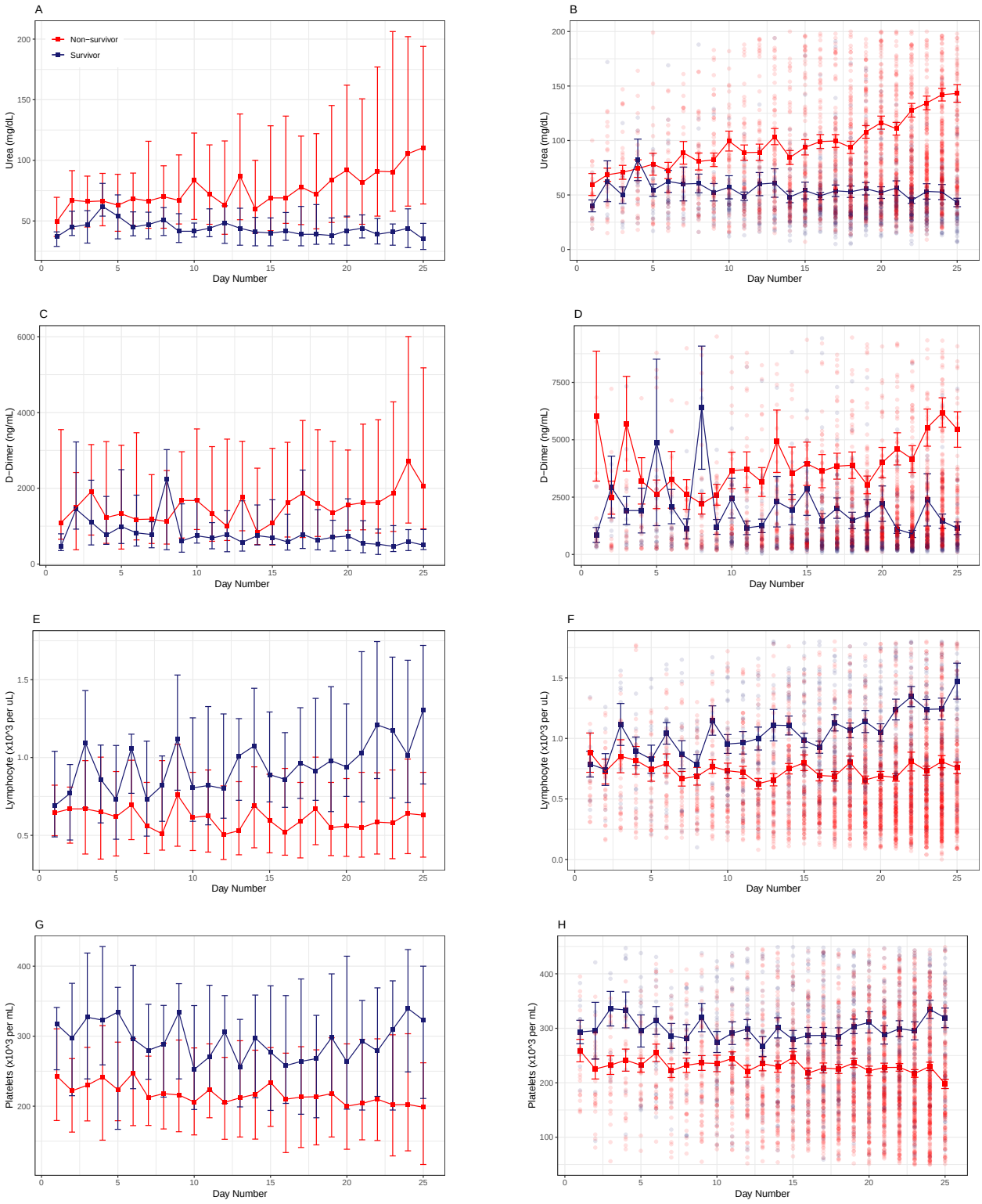

Figure 3.6: Temporal changes in hematic biometry biomarkers since the first hospitalization day until the last day hospitalized.

D-dimer (C,D) (Figure 3.6) in Non-survivor patients were increasing, which demonstrates hypercoagulability moreover an increased inflammation, their immune system response was higher, in other words, out of control. On the other hand, the Lymphocyte (E,F) and Platelets(G,H) decreased over time, while the levels of the Survivor patients increased and started being on more normal parameters.

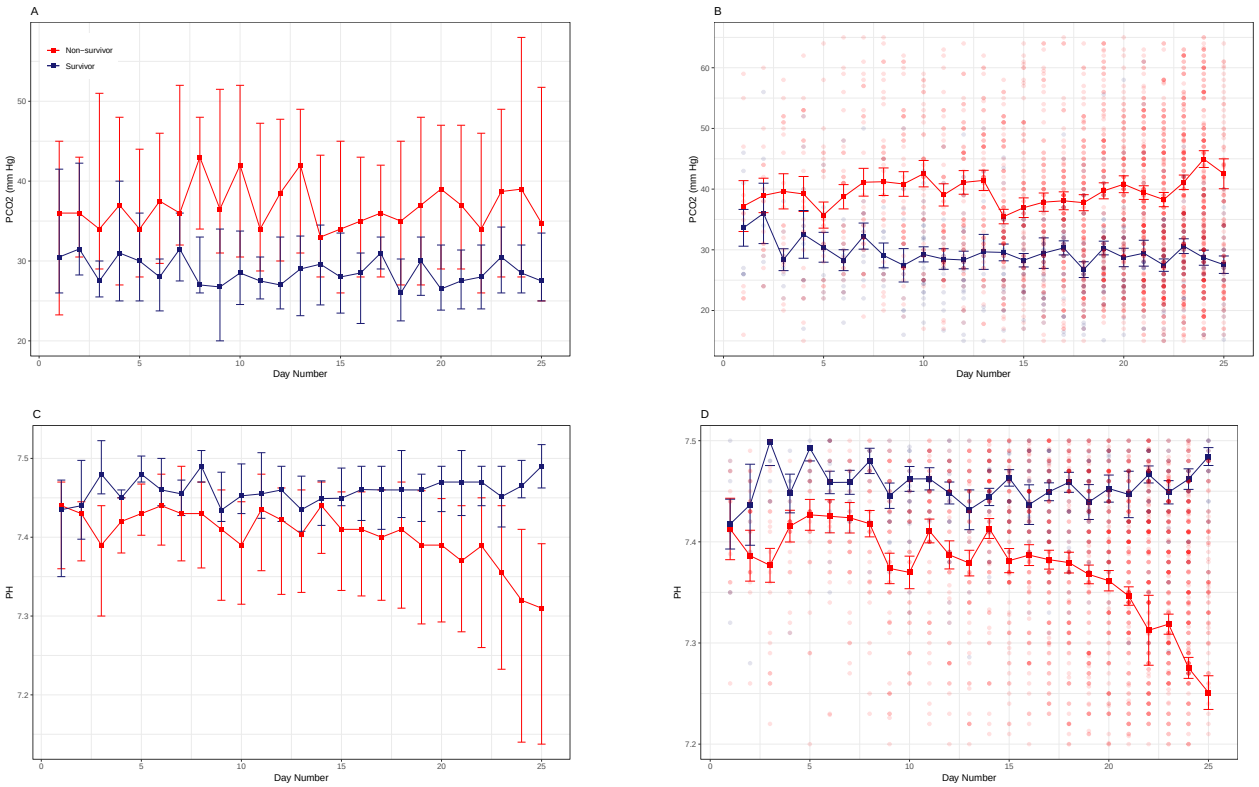

Figure 3.7: Temporal changes in arterial blood gases biomarkers since the first hospitalization day until the last day hospitalized.

Arterial blood gases biomarkers also expose variation between Survivor and Non-survivor patient. In Figure 3.7, PCO_2 (A, B) shows that the Non-survivors started accumulate carbon dioxide due to the lung failure, since PH (C,D) is also related to PCO_2, the blood started being more acid (i.e. acidosi).

3.2.3 *Data Balancing*

As shown in Table 3.2, the data is very imbalanced with only 30% of survivors, this can influence the algorithms, ignoring the minority class, therefore, the dataset needs to be balanced.

Two common techniques, to adjust the class distribution, are oversampling and undersampling. The first one generates new samples

to the minority class (the survivors), but it can cause overfitting and it may result in additional noise as well. Whereas undersampling, removes some random observation of the majority class (non-survivors) until the dataset is balanced, the drawback of it is that may discard potentially useful information.

In this study, we are going to use the second one, undersampling, due to the lack of information and missing values, besides to not add noise to the dataset.

3.3 *Modeling and Optimization*

The experiment was run in Intel (R) Core ^{TM} i7 2.80 GHz PC with 16 GB RAM, Windows 10 operating system and the coding environment is Python 3.8.5. The packages that were used are: H2O (version 3.32.1.7), scikit-learn (version 0.24.2), xgboost (version 1.4.1), gplearn (0.4.1) and bayesian-optimization (1.2.0) (Table 3.4).

Classifier	Library
Random Forest	h2o.estimators.H2ORandomForestEstimator
SVM	sklearn.svm.SVC
XGBoost	xgboost.XGBClassifier
Naïves Bayes	h2o.estimators.H2ONaiveBayesEstimator
Symbolic Classifier	gplearn.genetic.SymbolicClassifier
Symbolic Transformer	gplearn.genetic.SymbolicTransformer
Bayesian Optimization	bayes_opt.BayesianOptimization

Table 3.4: Classifier utilized in the present work and their corresponding implementations using H2O, scikit-learn, xgboost, gplearn and bayesian-optimization Python Libraries.

The dataset still contains variables with a strong percentage of missing values, for this reason we decided to create two models, the first one, **Model 1 (without missing values)**, a dataset that all the variables with more than 5.3% of missing values were excluded (Table 3.1) (i.e. all the arterial blood gases biomarkers, D-dimer, Sodium and Potassium were excluded), for the remaining variables, all the records that have a missing value were dropped, hence in the first dataset there were 3,765 records left. For the second, **Model 2 (with missing values)**, we keep all the variables and records (3,884).

For the **Model 1 (without missing values)** the classifiers that were implemented and compared were Random Forest, SVM, XGBoost, Naïves Bayes and Symbolic Classifier, to get the best model for each one the Table 3.5 shows the hyperparameters that were optimized with Bayesian Optimization and the bounded region. All the optimization process was conducted using 2-fold cross-validations, 5 steps of random

explorations (init_points), and with a maximum of 30 iterations (n_iter), except with Symbolic Classifier due computational cost. For Symbolic Classifier was set 2 init_points and only 5 iterations.

Hyperparameter	Default value	Bounded region
Random Forest		
ntrees	50	5 to 200
max_depth	10	0 to 20*
min_rows	1	1 to 50
mtries	-1	-2 to 25**
SVM		
C	1	0.1 to 100
sigma***	*scale*****	10^{-4} to 10
XGBoost		
n_estimators	100	50 to 200
max_depth	6	1 to 20
min_child_weight	1	1 to 50
learning_rate	0.3	0.001 to 0.1
gamma	0	10^{-10} to 10^{-3}
Naïves Bayes		
eps_sdev	0	0.1 to 1
Symbolic Classifier		
population_size	500	500 to 550
generations	10	2000 to 4000
tournament_size	20	10 to 20

* Setting this value to 0 specifies no limit.
** If -2 is specified, all the predictors of the Random Forest are used, when the value is -1, for classification, the number of variables is $\sqrt{P}$, where P is the number of predictors.
*** Sigma is also know as gamma ($1/2\text{sigma}^2$)
**** scale = n_features X.var()/2

Table 3.5: **Model 1** Hyperparameter Tuning: Default values and bounded region of hyperparameter space for the bayesian optimization for Random Forest, SVM, XGBoost, Naïves Bayes and Symbolic Classifier.

On the other hand, for the **Model 2 (with missing values)**, because of Random Forest can deal with missing values, as mentioned previously. To find the best model Table 3.6 shows the hyperameters and the bounded region, and as with **Model 1**, the bayesian optimization process was conducted using 2-fold cross-validations, 5 init_points and 30 iterations.

Hyperparameter	Default value	Bounded region
Random Forest		
ntrees	50	5 to 200
max_depth	10	0 to 20*
min_rows	1	1 to 50
mtries	-1	-2 to 36**

* Setting this value to 0 specifies no limit.
** If -2 is specified, all the predictors of the Random Forest are used, when the value is -1, for classification, the number of variables is $\sqrt{P}$, where P is the number of predictors.

Table 3.6: **Model 2** Hyperparameter Tuning: Default values and bounded region of hyperparameter space for the bayesian optimization for Random Forest.

The Symbolic Transformer has been used to improve the performance of some models so that it could help us to improve the model. It only

can be applied in data without missing values, in this case, is the **Model 1**. We followed the same methodology as the Symbolic Classifier, due to computational cost the hyperparameter tuning was with 2 init_points and 5 iterations (Table 3.7).

Hyperparameter	Default value	Bounded region
Symbolic Transformer		
population_size	1000	500 to 550
generations	20	2000 to 4000
tournament_size	20	10 to 20

Table 3.7: Hyperparameter Tuning: Default values and bounded region of hyperparameter space for the bayesian optimization for Symbolic Transformer .

The flowchart that was followed for each model is shown in Figure 3.8: After transforming all the categorical variables to factor we balance data, we split the data into train and test, if the model is the SVM we applied a data transformation (Normalization or Standarization), the model fitting is performed using the training dataset and the best hyperparameter values of each model were obtain using the Bayesian Optimization Algorithm, the best model is evaluated on the test dataset.

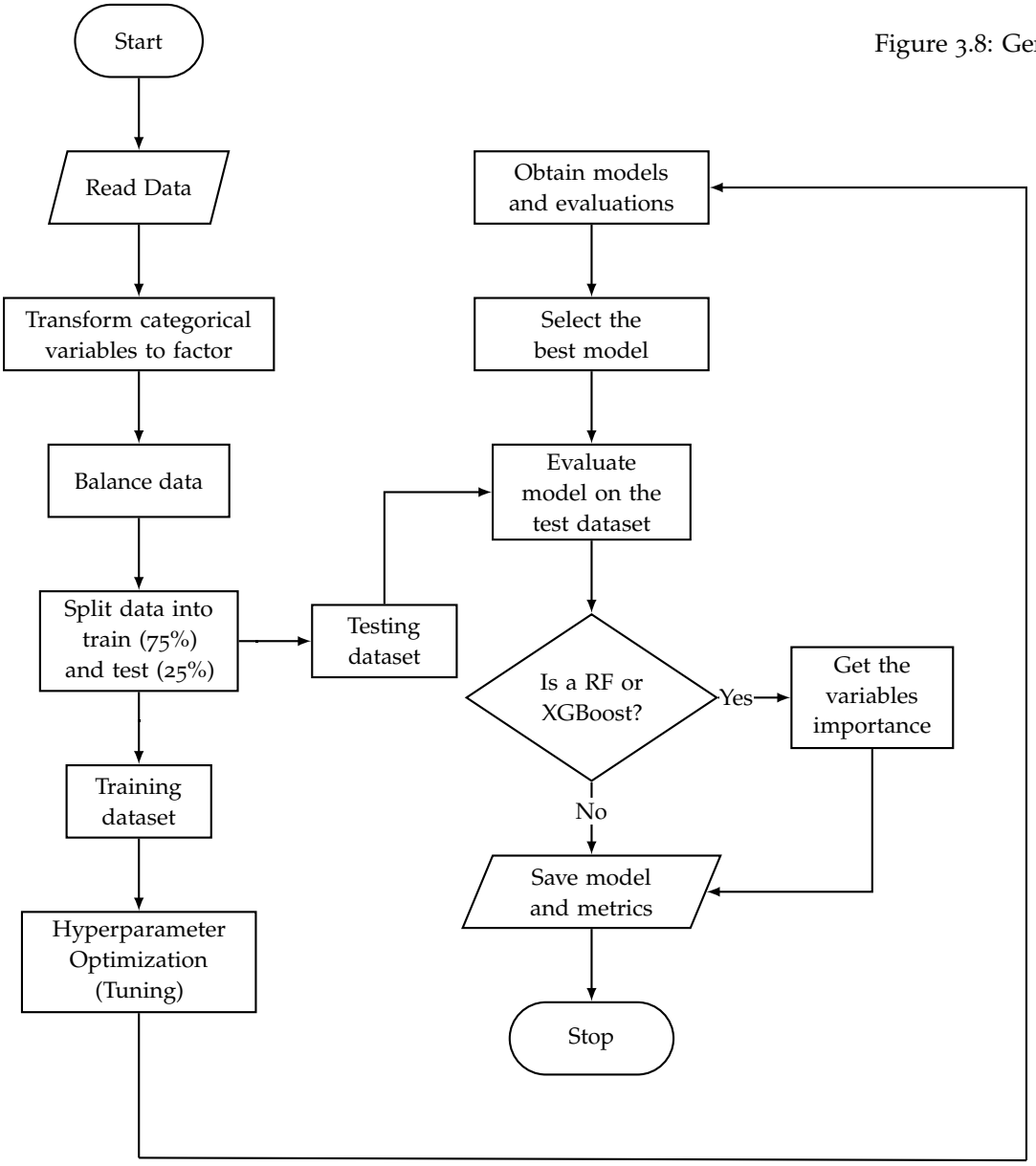

Figure 3.8: General flowchart

4 Results

Contents

As we mentioned in Section 3.2.2 and Section 3.3 we have two models, **Model 1 (without missing values)** and **Model 2 (with missing values)**. For each Classifier we followed Fig 3.8. Below, all the results are shown.

4.1 *Model 1 (without missing values)*

For **Model 1** the Table 3.5 shows the hyperparameters tuning applying Bayesian Optimization. The SVM is the only classfier that needs a data transformation, for the rest of the Classifiers that were used, none of them need a data transformation.

Therefore, for the SVM, first, we need to apply a transformation before tuning the hyperparameters, some of the variables follow a Gaussian distribution, therefore we decided to Standardized the data, and also to use the Min-Max transformation, due to this transformation doesn't affect the distribution of the data.

We tried all the Kernels with each data transformation (Standardized and Min-Max) and we kept the Kernel with the best performance for each one. Using the RBF Kernel we notices that the Standardized data performance better, and for the Min-Max data, we use the Polynomial Kernel.

Table 4.1 shows the results tuning the hyperparameters using the accuracy metric to evaluate the model. In our experience we have seen that using ROC AUC could improve the model, for this reason, we decided to use also the ROC AUC to tune the hyperparameters, Table 4.2 show the results using this metric.

Using both tuning metrics, XGBoost obtain 100% in all the training metrics, denoting that the classifier was overfitting, and then, drop on the testing dataset.

	ROC AUC	Accuracy	Precision	Recall	F1
SVM RBF					
train	0.929	0.9288	0.9176	0.9386	0.928
test	**0.8938**	**0.8933**	**0.9113**	**0.8863**	**0.8986**
SVM Polynomial					
train	0.9404	0.9399	0.9217	0.9586	0.9398
test	**0.8986**	**0.8996**	**0.8996**	**0.9137**	**0.9066**
Random Forest					
train	0.9416	0.8813	0.846	0.9257	0.884
test	**0.8868**	**0.887**	**0.8972**	**0.8902**	**0.8937**
XGBoost					
train	1	1	1	1	1
test	**0.8728**	**0.8724**	**0.8911**	**0.8667**	**0.8787**
Naïves Bayes					
train	0.8628	0.7926	0.758	0.8457	0.7995
test	**0.8055**	**0.8054**	**0.8266**	**0.8039**	**0.8151**
Symbolic Classifier					
train	0.8078	0.8073	0.7873	0.83	0.8081
test	**0.7935**	**0.7929**	**0.8197**	**0.7843**	**0.8016**

Table 4.1: Model 1 Results hyperparameter tuning metric Accuracy.

	ROC AUC	Accuracy	Precision	Recall	F1
SVM RBF					
train	0.9903	0.9902	0.9845	0.9957	0.9901
test	**0.8829**	**0.8828**	**0.8964**	**0.8824**	**0.8893**
SVM Polynomial					
train	0.9278	0.9274	0.9071	0.9486	0.9274
test	**0.8624**	**0.864**	**0.8626**	**0.8863**	**0.8743**
Random Forest					
train	0.946	0.8834	0.8686	0.8971	0.8826
test	**0.8599**	**0.8598**	**0.876**	**0.8588**	**0.8673**
XGBoost					
train	1	1	1	1	1
test	**0.8669**	**0.8682**	**0.8692**	**0.8863**	**0.8777**
Naïves Bayes					
train	0.8699	0.7933	0.7722	0.8186	0.7947
test	**0.7694**	**0.7699**	**0.7888**	**0.7765**	**0.7826**
Symbolic Classifier					
train	0.8123	0.8128	0.8224	0.7871	0.8044
test	**0.7865**	**0.7824**	**0.8447**	**0.7255**	**0.7806**

Table 4.2: Model 1 Results hyperparameter tuning metric ROC AUC.

Figure 4.1 shows the Classifiers comparison. For all the Classifiers that were optimized, in this case, the best tuning metric was the

Accuracy. Evaluation metrics suggest that in both cases, using accuracy or ROC AUC as a tuning metric, the SVM, (SVM Polynomial and SVM RBF respectively), outperforms all other classifiers.

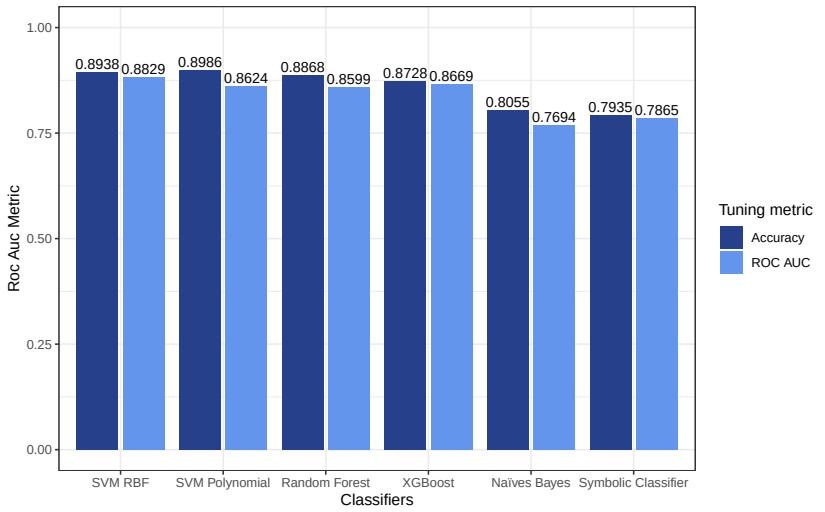

Figure 4.1: Classifiers comparison

Since the intubation variable is not reliable, due to diverse external factors, we decided to exclude this variable and see the performance for each Classifier excluding this variable.

We followed the same steps above, we use the accuracy and ROC AUC as tuning metrics. Table4.3 and Table4.4 show the corresponding results. We notice that when Intubation is excluded, the performance of the Classifiers decrease, but in both cases, the best Classifiers is a SVM, but now the best Classifier to the tuning Accuracy metric is the SVM RBF, and SVM Polynomial is the best using the ROC AUC as a tuning metric.

In Figure 4.2 we can easily see that all the classifiers have a better performance using the tuning accuracy metric except for the SVM Polynomial.

The best model including intubation and excluding intubation is shown in Table 4.5. But even though the SVM is the best model in both cases, we are not able to have the variable importance, we take the Random Forest to be able to have the variable importance, Table 4.6 displays the Random Forest obtained.

	ROC AUC	Accuracy	Precision	Recall	F1
SVM RBF					
train	0.973	0.9728	0.9635	0.9814	0.9724
test	**0.8647**	**0.864**	**0.8862**	**0.8549**	**0.8703**
SVM Polynomial					
train	0.9487	0.9483	0.9323	0.9643	0.948
test	**0.7731**	**0.7678**	**0.8429**	**0.6941**	**0.7613**
Random Forest					
train	0.9179	0.845	0.8018	0.9071	0.8512
test	**0.8179**	**0.817**	**0.8036**	**0.8721**	**0.8364**
XGBoost					
train	0.9379	0.9378	0.9321	0.9414	0.9367
test	**0.8302**	**0.8264**	**0.8874**	**0.7725**	**0.826**
Naïves Bayes					
train	0.8289	0.757	0.712	0.8443	0.7725
test	**0.7554**	**0.7573**	**0.7663**	**0.7843**	**0.7752**
Symbolic Classifier					
train	0.7542	0.7979	0.6657	0.7259	0.7523
test	**0.7113**	**0.8128**	**0.5961**	**0.6878**	**0.7196**

Table 4.3: Model 1 Results hyperparameter tuning metric Accuracy excluding intubation.

	ROC AUC	Accuracy	Precision	Recall	F1
SVM RBF					
train	0.8955	0.8953	0.883	0.9057	0.8942
test	**0.8142**	**0.8138**	**0.8374**	**0.8078**	**0.8224**
SVM Polynomial					
train	0.895	0.8946	0.8765	0.9129	0.8943
test	**0.8159**	**0.8138**	**0.8547**	**0.7843**	**0.818**
Random Forest					
train	0.8998	0.8303	0.7971	0.8757	0.8346
test	**0.803**	**0.8033**	**0.8207**	**0.8078**	**0.8142**
XGBoost					
train	1	1	1	1	1
test	**0.8058**	**0.8054**	**0.8293**	**0.8**	**0.8144**
Naïves Bayes					
train	0.8428	0.7723	0.7404	0.8229	0.7794
test	**0.7282**	**0.7301**	**0.7423**	**0.7569**	**0.7495**
Symbolic Classifier					
train	0.7731	0.7723	0.7474	0.8071	0.7761
test	**0.7047**	**0.7071**	**0.7186**	**0.7412**	**0.7297**

Table 4.4: Model 1 Results hyperparameter tuning metric ROC AUC excluding intubation.

Hyperparameter	Value
SVM Polynomial including intubation	
C	35.8210
Sigma	1.5645
SVM RBF excluding intubation	
C	13.9808
Sigma	3.1268

Table 4.5: Best Model 1 hyperparameters.

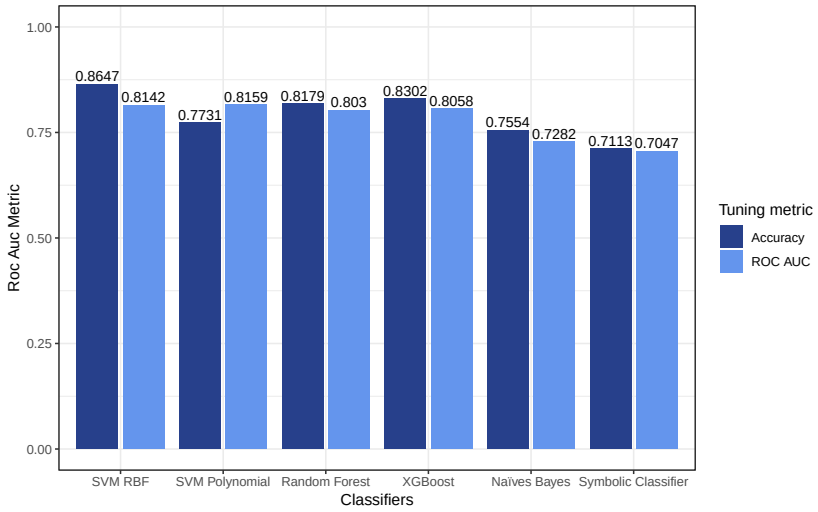

Figure 4.2: Classifiers excluding intubation comparison

Hyperparameter	Value
Random Forest including intubation	
ntrees	192
max_depth	19
min_rows	2
mtries	-1*
Random Forest excluding intubation	
ntrees	139
max_depth	0**
min_rows	1
mtries	4

* When the value is -1, for classification, the number of variables is $\sqrt{P}$, where P is the number of predictors.
** Setting this value to 0 specifies no limit.

Table 4.6: Random Forest best Model 1 hyperparameters.

4.1.1 *Symbolic Transformer*

Symbolic Transformer can help us to improve the model, despite that the it increase the number of variables, we tried to improve the best **Model 1** that was obtained, the SVM Polynomial with the tuning Accuracy metric.

Due to the computational cost, the model was not able to iterate more times, it took more than 5 hours to tune the hyperparameter just for 7 iteration. We can see in Table 4.7 and Table 4.8 the results and the model.

	ROC AUC	Accuracy	Precision	Recall	F1
SVM Polynomial					
train	0.9453	0.9448	0.9248	0.9657	0.9448
test	**0.8649**	**0.8682**	**0.8504**	**0.9137**	**0.8809**

Table 4.7: Model 1 SVM Symbolic Transformer.

Hyperparameter	Value
SVM Polynomial	
C	35.8210
Sigma	1.5645
Symbolic Transformer	
population_size	525
generations	3,098
tournament_size	10

Table 4.8: Symbolic Transformer hyperparmeters.

In the heuristics models, increasing the generation and letting the model explore could improve the models, but for our problem, the model did not make any improvement (Fig 4.3).

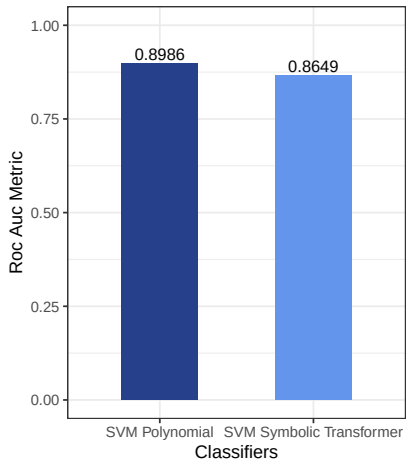

Figure 4.3: SVM Polynomial with Symbolic Transformer comparison.

4.2 *Model 2 (with missing values)*

For **Model 2** we took all the variable and kept the missing values. The models that can handle missing values are the Random Forest by H2O library, and like we did in Sec 4.1 we use the Accuracy and ROC AUC to tune the hyperparameters and we did a model including intubation and another model excluding intubation.

Table 4.13 and Table 4.10 show the results. We notice like with the model 1, when intubation is excluded the Classifiers performances decrease, but contrary to model 1, now the tuning ROC AUC Classifiers are better (Fig 4.4).

Despite that **Model 2** includes missing values, we are able to obtain good models with similar metrics as we obtained with the Random Forest **Model 1**. Table 4.11 present the best **Models 2**.

	ROC AUC	Accuracy	Precision	Recall	F1
RF including Intubation					
train	0.941	0.8876	0.8507	0.9422	0.8941
test	**0.8779**	**0.8763**	**0.8242**	**0.9494**	**0.8824**
RF excluding Intubation					
train	0.9114	0.847	0.8136	0.9032	0.8561
test	**0.8139**	**0.8124**	**0.7684**	**0.8819**	**0.8212**

Table 4.9: Model 2 Random Forest hyperparameter tuning metric Accuracy

	ROC AUC	Accuracy	Precision	Recall	F1
RF including Intubation					
train	0.9418	0.8842	0.8515	0.9328	0.8903
test	**0.8821**	**0.8804**	**0.8255**	**0.9578**	**0.8867**
RF excluding Intubation					
train	0.9156	0.8389	0.7942	0.918	0.8516
test	**0.8266**	**0.8247**	**0.7734**	**0.9072**	**0.835**

Table 4.10: Model 2 Random Forest hyperparameter tuning metric ROC AUC

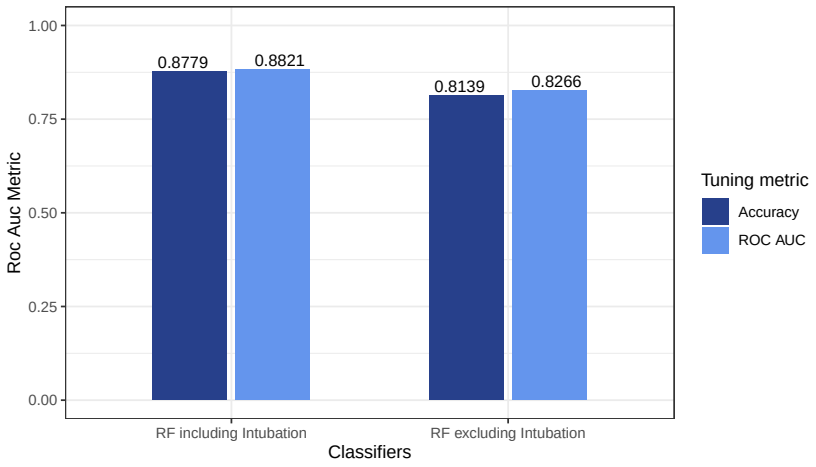

Figure 4.4: Model 2 Classifiers comparison

4.2.1 *Model 2 by days*

Keeping the missing values we are able to split the dataset and see how the variable importance change over the days. Days 1 to 19 represents 50% and the remaining days the other 50% of the data as Figure 3.4 has shown. We follow the same methodology.

The ROC AUC tuning metric in both cases outperforms accuracy tuning metric classifiers, and we were able to obtain good evaluation metrics doing a model splited by day (Figure 4.5). Table 4.14 show the respective hyperparameters.

Hyperparameter	Value
Random Forest including intubation	
ntrees	165
max_depth	20
min_rows	1
mtries	19
Random Forest excluding intubation	
ntrees	189
max_depth	20
min_rows	1
mtries	14

Table 4.11: Random Forest best Model 2 hyperparameters.

	ROC AUC	Accuracy	Precision	Recall	F1
RF days 1 to 19					
train	0.934	0.8746	0.8439	0.9419	0.8902
test	**0.8611**	**0.8676**	**0.8343**	**0.9419**	**0.8848**
RF days 20 to 25					
train	0.9441	0.8947	0.8646	0.9291	0.8957
test	**0.8274**	**0.827**	**0.8765**	**0.7634**	**0.8161**

Table 4.12: Model 2 Random Forest by days hyperparameter tuning metric Accuracy

	ROC AUC	Accuracy	Precision	Recall	F1
RF days 1 to 19					
train	0.9547	0.9068	0.9038	0.929	0.9162
test	**0.8709**	**0.8746**	**0.8606**	**0.9161**	**0.8875**
RF days 20 to 25					
train	0.9469	0.8966	0.8702	0.9254	0.8969
test	**0.8543**	**0.8541**	**0.8837**	**0.8172**	**0.8492**

Table 4.13: Model 2 Random Forest by days hyperparameter tuning metric ROC AUC

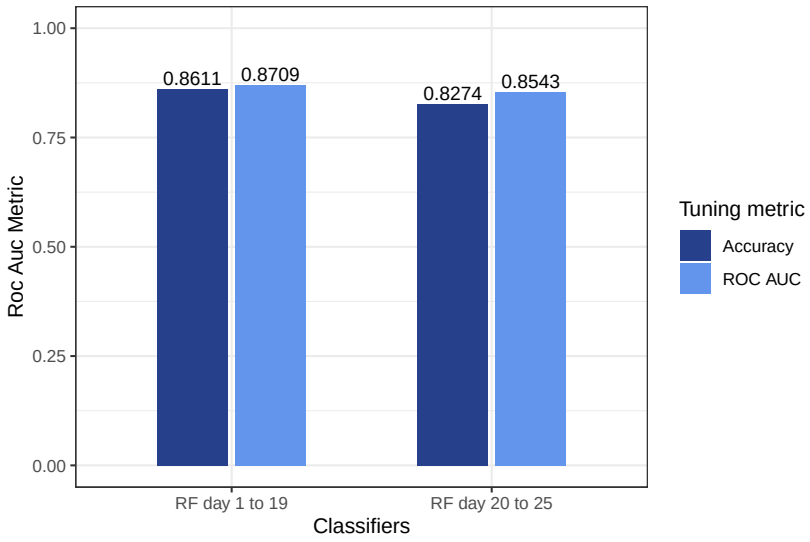

Figure 4.5: Model 2 Classifiers comparison

Hyperparameter	Value
RF day 1 to 19	
ntrees	119
max_depth	20
min_rows	1
mtries	2
RF day 20 to 25	
ntrees	162
max_depth	16
min_rows	1
mtries	15

Table 4.14: Random Forest by day best model hyperparameters.

4.3 *Variable Importance*

Taking the Random Forest Classifiers that we obtained in the previous section, we are able to obtain the Variable importance for each model.

Variable importance **Model 1** and **Model 2** including intubation are display Figure 4.6. In both models the Intubation variable is the most importance variable, but in the **Model 2** that variable takes more importance, advising that with missing values the variables Intubation and Age have a higher impact on the model.

In **Model 1** the urea and lymphocytes follow after intubation, as we have seen in Figure3.6 show the differentiation between Non-survivors and Survivor. On the other hand **Model 2** PH and PCO2 that were excluded due to the missing values in **Model 1**, with the urea, follow after intubation and age, showing the variation that we have already seen.

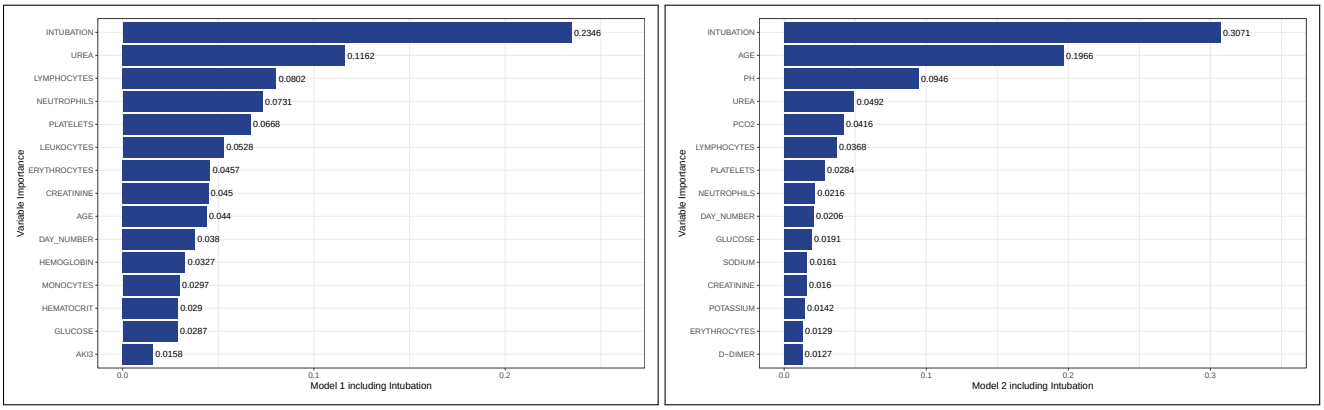

Figure 4.6: Model 1 and Model 2 including intubation variable importance.

Figure 4.7 presents the variable importance of **Model 1** and **Model 2** but excluding intubation. We can see that in the case of **Model 1** the order of the variables are extremely similar, just excluding the intubation, but also the values are similar. Whereas **Model 2** also shows similar variable order but it could be a little bit more unstable because of the missing values.

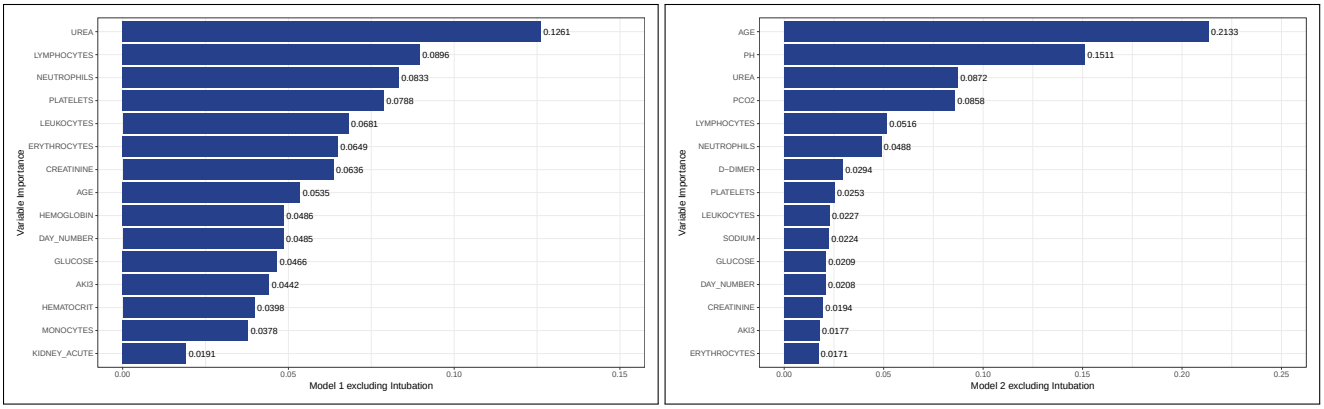

Figure 4.7: Model 1 and Model 2 excluding intubation variable importance.

On the **Model 2 by days** we were able to see how the variable importance change over time. As we have seen Figure 3.6 and Figure 3.7 the biomarkes show remarkable temporal changes differences between Non-Surivors and Survivor.

Several Covid-19 studies found that this virus strongly affects the kidney. Figure 4.8 shows that on days 1 to 19, when several patients just have been hospitalized, the two most important variables are Intubation and Age, on the other model, those variables do not have the same importance, this importance is been reduced drastically, and it seems that the some biomarkers increased their importance.

Creatinine is more important on days 20 to 25 with the Urea, demonstrating the kidney damage, presumably, the acute kidney injury was not healed. PH importance increased over the time, that means that the acid in the blood is increasing. In section 3.2.2 we have seen that Lymphocytes decreased, (i.e. lymphopenia), due that they are trying to attack the illness, in the Non-survivors the response is incommensurate.

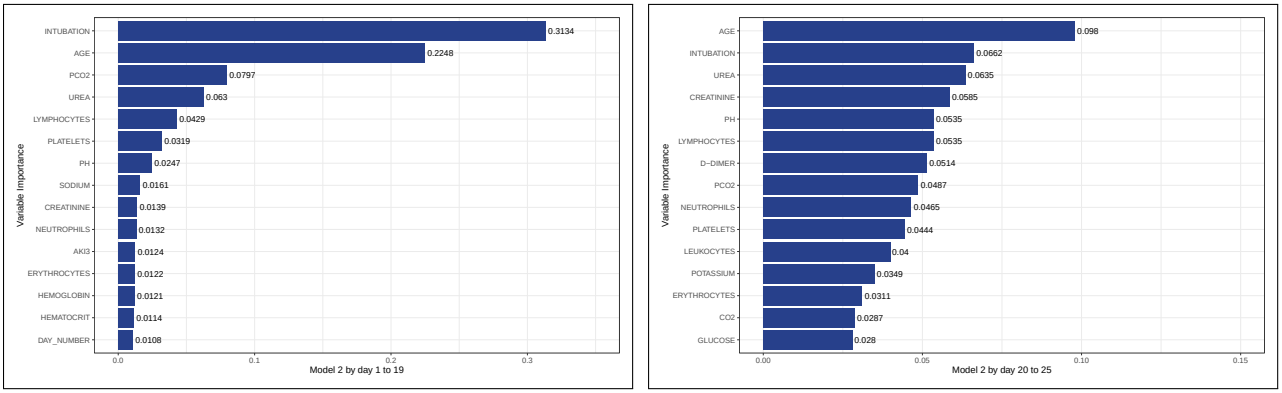

Figure 4.8: Model 2 by days variable importance.

4.4 Discussion

In this Chapter, we have seen all the methods that were implemented to try to obtain the best Classifier. Figure 4.9 shows**Model 1 (without missing values)** best Classfiers. SVM Polynomial including intubation was the best of all but the Random Forest, including this variable, obtain also a similar performance, but when Intubation was excluded the model performance decrease and the Random Forest was more affected than the SVM.

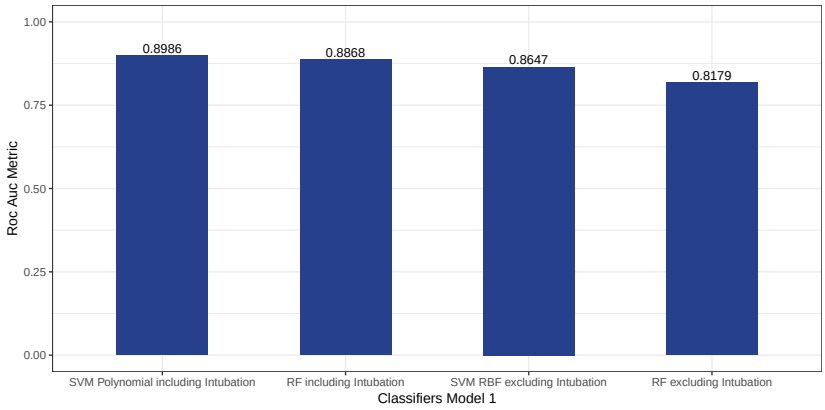

Figure 4.9: Best Model 1 Classifiers comparison

Using the Genetics programming, in the case of the Symbolic Classifier the performance was one of the worst, and due to the cost computational, we could not iterate more times to find the best model (only 7 times). And when we tried to increase the performance of the SVM Polynomial using the Symbolic Transformer, the Classifier performance did not improve, and the cost computational was high as the Symbolic Classifier, in both cases with only 7 iterations with the Bayesian Optimization, it could take more than 5 hours.

The main objective of this study is being able to obtain the variable that are more important for the model, as well as a Classifier that were able to handle the missing values. For this reason we choose the Random Forest, due to the capacity of this Classifier. Figure 4.10 displays **Model 2 (with missing values)** best models, with similar performance as the Random Forest in **Model 1**.

When we compared the variable importance we were able to appreciated that including the missing values the variable importance as the Intubation and Age increase.

Splitting the data by days, shows how the variable importance change, as we have seen in Section 3.2.2.

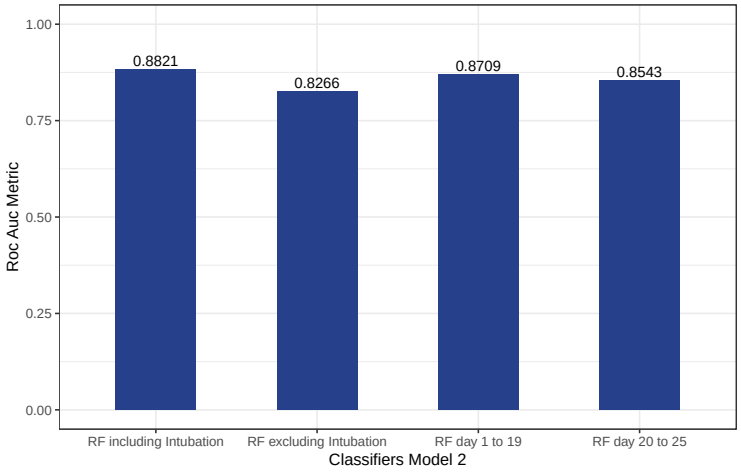

Figure 4.10: Model 2 Classifiers comparison

5 Conclusion

In the biomakers, we were able to find several difference between Survivors and Non-survivors. In Urea the Non-survivor showed higher levels and never starts to normalize, suggesting a kidney injury. The Lymphocyte decreased over time in the Non-survivors. The Non-survivors showed inflammation and hypercoagulability according to D-dimer. The blood gases biomarkers indicate a lung failure.

Several Classifiers have been applied in this study, creating two types of models due to the missing values in the real data: **Model 1 (without missing values)**, **Model 2 (with missing values)**, and implemented several methods and approaches to improve the prediction and obtain the best Classifier. Even tough that SVM Polynomial was the best of all, the Random Forest, obtain also a similar performance, and has the advantage that it can deal with missig values.

One approach was to exclude intubation due to the external factors that can affect this variable, When we include this variable all the Classifiers performance were upper than 79% of ROC AUC, and we almost obtained the 90% in the SVM Polynomial . On the other hand, when this variable was exclude, all the model performance decrease and we got an ROC AUC between 71% and 86%.

With missing values, **Model 2**, including and excluding intubation we got 88% and 82% ROC AUC respectively. and with the model by days we get similar metrics.

Using the Genetics programming, in the case of the Symbolic Classifier the performance was one of the worst and applying Symbolic Tranformer to the SVM Polynomial to try to increase the performance we could not improve this Classifier. The cost computational of this algorithm is high, we could not iterate more times to find the best model (only 7 times) and it could take more than 5 hours.

The main objective of this study is being able to obtain the variable that are more important for the model, as well as a Classifier that were able to handle the missing values. For this reason we choose the Random Forest, due to the capacity of this Classifier.

In **Model 1** and **Model 2** the Intubation variable was the most importance variable, but in the **Model 2** that variable takes more

importance, advising that with missing values the variables Intubation and Age have a higher impact on the model.

Several Covid-19 studies found that this virus strongly affects the kidney, and it can be seen in the levels of the Urea and Creatinine, also other biomarkers are affected like Lymphocytes, Platelets and D-Dimer, and the biomarkers related to the arterial blood gases. In our models, we obtained, in the variable importance, that urea, lymphocytes, PH, PCO2, platelets, etc., being reasonable with other investigations and the data exploration.

The Roc Auc for each classifier was closed to 0.88, and with the variable importance we were able to obtain the best Machine Learning model that can quantify the effects of biomarkers and comorbidities in predicting SARS Cov-2 associated mortality in Hospitalized Patients in Mexico.

For future work we will apply the model to the patients that were vaccinated, imputing missing values, with a deeper analysis and use other techniques to obtain the impact of the biomarkers.

Bibliography

Support Vector Machine | Beginners Guide to Support Vector Machine. URL https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/.

Distributed Random Forest (DRF) — H2O 3.34.0.3 documentation, a. URL https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html.

API reference — gplearn 0.4.1 documentation, b. URL https://gplearn.readthedocs.io/en/stable/reference.html#symbolic-classifier.

WHO | World Health Organization, c. URL https://www.who.int/.

COVID-19 Tablero México, d. URL https://datos.covid-19.conacyt.mx/index.php.

Shigeo Abe. *Support Vector Machines for Pattern Classification*. Advances in Pattern Recognition. Springer London. ISBN 978-1-84996-097-7 978-1-84996-098-4. DOI: 10.1007/978-1-84996-098-4. URL http://link.springer.com/10.1007/978-1-84996-098-4.

Sreedhar Adapa, Avantika Chenna, Mamtha Balla, Ganesh Prasad Merugu, Narayana Murty Koduri, Subba Rao Daggubati, Vijay Gayam, Srikanth Naramala, and Venu Madhav Konala. COVID-19 Pandemic Causing Acute Kidney Injury and Impact on Patients With Chronic Kidney Disease and Renal Transplantation. 12(6): 352–361. ISSN 1918-3003. DOI: 10.14740/jocmr4200. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7295554/.

Md. Zahangir Alam, M. Saifur Rahman, and M. Sohel Rahman. A Random Forest based predictor for medical data classification using feature ranking. 15:100180. ISSN 2352-9148. DOI: 10.1016/j.imu.2019.100180. URL https://www.sciencedirect.com/science/article/pii/S235291481930019X.

Richard A. Berk. *Statistical Learning from a Regression Perspective*. Springer Texts in Statistics. Springer International Publishing. ISBN

978-3-319-44047-7 978-3-319-44048-4. DOI: 10.1007/978-3-319-44048-4. URL http://link.springer.com/10.1007/978-3-319-44048-4.

Eric Brochu, Vlad M. Cora, and prefix=de useprefix=true family=Freitas, given=Nando. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. URL http://arxiv.org/abs/1012.2599.

Colin Campbell and Yiming Ying. Learning with Support Vector Machines. 5(1):1–95. ISSN 1939-4608, 1939-4616. DOI: 10.2200/S00324ED1V01Y201102AIM010. URL http://www.morganclaypool.com/doi/abs/10.2200/S00324ED1V01Y201102AIM010.

Yichun Cheng, Ran Luo, Kun Wang, Meng Zhang, Zhixiang Wang, Lei Dong, Junhua Li, Ying Yao, Shuwang Ge, and Gang Xu. Kidney disease is associated with in-hospital death of patients with COVID-19. 97(5):829–838. ISSN 0085-2538. DOI: 10.1016/j.kint.2020.03.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7110296/.

Pieter A. Cohen, Lara E. Hall, Janice N. John, and Alison B. Rapoport. The Early Natural History of SARS-CoV-2 Infection. 95(6):1124–1126. ISSN 00256196. DOI: 10.1016/j.mayocp.2020.04.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0025619620303797.

Wolfgang Ertel. *Introduction to Artificial Intelligence*. Undergraduate Topics in Computer Science. Springer International Publishing. ISBN 978-3-319-58486-7 978-3-319-58487-4. DOI: 10.1007/978-3-319-58487-4. URL http://link.springer.com/10.1007/978-3-319-58487-4.

Samira S. Farouk, Enrico Fiaccadori, Paolo Cravedi, and Kirk N. Campbell. COVID-19 and the kidney: What we think we know so far and what we don't. 33(6):1213–1218. ISSN 1121-8428, 1724-6059. DOI: 10.1007/s40620-020-00789-y. URL https://link.springer.com/10.1007/s40620-020-00789-y.

fernando. Bayesian Optimization. URL https://github.com/fmfn/BayesianOptimization.

Peter I. Frazier. A Tutorial on Bayesian Optimization. URL http://arxiv.org/abs/1807.02811.

Erika Fabiola Saquinga Jame. Título: Dímero d, tiempo de protrombina y plaquetas en la valoración del paciente con covid-19. page 57.

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York. ISBN 978-1-4614-6848-6 978-1-4614-6849-3. DOI:

10.1007/978-1-4614-6849-3. URL http://link.springer.com/10.1007/978-1-4614-6849-3.

Barrett E Lowe. The Random Forest Algorithm with Application to Multispectral Image Analysis. page 79.

Luis Edgardo López and María Daniela Mazzucco. Alteraciones de parámetros de laboratorio en pacientes con sars-cov-2. page 15.

M. Narasimha Murty and V. Susheela Devi. *Pattern Recognition*, volume 0 of *Undergraduate Topics in Computer Science*. Springer London. ISBN 978-0-85729-494-4 978-0-85729-495-1. DOI: 10.1007/978-0-85729-495-1. URL http://link.springer.com/10.1007/978-0-85729-495-1.

Olivier Pauly. Random Forests for Medical Applications. page 204.

Manuel Ramón Pérez Abreu, Jairo Jesús Gómez Tejeda, and Ronny Alejandro Dieguez Guach. Características clínico-epidemiológicas de la COVID-19. 19(2). ISSN 1729-519X. URL http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1729-519X2020000200005&lng=es&nrm=iso&tlng=es.

Safiya Richardson, Jamie S. Hirsch, Mangala Narasimhan, James M. Crawford, Thomas McGinn, Karina W. Davidson, and and the Northwell COVID-19 Research Consortium. Presenting Characteristics, Comorbidities, and Outcomes Among 5700 Patients Hospitalized With COVID-19 in the New York City Area. 323(20): 2052–2059. ISSN 0098-7484. DOI: 10.1001/jama.2020.6775. URL https://doi.org/10.1001/jama.2020.6775.

Joseph Rocca. Ensemble methods: Bagging, boosting and stacking. URL https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

M. Salazar, J. Barochiner, W. Espeche, and I. Ennis. Covid-19, hipertensión y enfermedad cardiovascular. 37(4):176–180. ISSN 18891837. DOI: 10.1016/j.hipert.2020.06.003. URL https://linkinghub.elsevier.com/retrieve/pii/S1889183720300659.

Vimarsh Sathia, Venkataramana Ganesh, and Shankara Rao Thejaswi Nanditale. Accelerating Genetic Programming using GPUs. URL http://arxiv.org/abs/2110.11226.

Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. 43(4). ISSN 0090-5364. DOI: 10.1214/15-AOS1321. URL https://projecteuclid.org/journals/annals-of-statistics/volume-43/issue-4/Consistency-of-random-forests/10.1214/15-AOS1321.full.

Kent Sharkey. What is the Team Data Science Process? - Azure Architecture Center. URL https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview.

Amita Sharma and Willem J. M. I. Verbeke. Improving Diagnosis of Depression With XGBOOST Machine Learning Model and a Large Biomarkers Dutch Dataset (n = 11,081). 3:15. ISSN 2624-909X. DOI: 10.3389/fdata.2020.00015. URL https://www.frontiersin.org/article/10.3389/fdata.2020.00015.

Yan-yan SONG and Ying LU. Decision tree methods: Applications for classification and prediction. 27(2):130–135. ISSN 1002-0829. DOI: 10.11919/j.issn.1002-0829.215044. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/.

Trevor Stephens. Gplearn Documentation. page 57.

Johan A. K. Suykens, editor. *Least Squares Support Vector Machines*. World Scientific. ISBN 978-981-238-151-4.

Changsong Wang, Kai Kang, Yan Gao, Ming Ye, Xiuwen Lan, Xueting Li, Mingyan Zhao, and Kaijiang Yu. Cytokine Levels in the Body Fluids of a Patient With COVID-19 and Acute Respiratory Distress Syndrome: A Case Report. 173(6):499–501. ISSN 0003-4819. DOI: 10.7326/L20-0354. URL https://www.acpjournals.org/doi/full/10.7326/L20-0354.

Zunyou Wu and Jennifer M. McGoogan. Characteristics of and Important Lessons From the Coronavirus Disease 2019 (COVID-19) Outbreak in China: Summary of a Report of 72 314 Cases From the Chinese Center for Disease Control and Prevention. 323(13): 1239–1242. ISSN 0098-7484. DOI: 10.1001/jama.2020.2648. URL https://doi.org/10.1001/jama.2020.2648.

Ying Xie, Wei-Yu Meng, Run-Ze Li, Yu-Wei Wang, Xin Qian, Chang Chan, Zhi-Fang Yu, Xing-Xing Fan, Hu-Dan Pan, Chun Xie, Qi-Biao Wu, Pei-Yu Yan, Liang Liu, Yi-Jun Tang, Xiao-Jun Yao, Mei-Fang Wang, and Elaine Lai-Han Leung. Early lung cancer diagnostic biomarker discovery by machine learning methods. 14(1):100907. ISSN 1936-5233. DOI: 10.1016/j.tranon.2020.100907. URL https://www.sciencedirect.com/science/article/pii/S1936523320303995.

Alessandro Zandonà, Rosario Vasta, Adriano Chiò, and Barbara Di Camillo. A Dynamic Bayesian Network model for the simulation of Amyotrophic Lateral Sclerosis progression. 20(S4):118. ISSN 1471-2105. DOI: 10.1186/s12859-019-2692-x. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2692-x.

Fei Zhou, Ting Yu, Ronghui Du, Guohui Fan, Ying Liu, Zhibo Liu, Jie Xiang, Yeming Wang, Bin Song, Xiaoying Gu, Lulu Guan, Yuan Wei, Hui Li, Xudong Wu, Jiuyang Xu, Shengjin Tu, Yi Zhang, Hua Chen, and Bin Cao. Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: A retrospective cohort study. 395(10229):1054–1062. ISSN 01406736. DOI: 10.1016/S0140-6736(20)30566-3. URL https://linkinghub.elsevier.com/retrieve/pii/S0140673620305663.