

**BABEȘ -BOLYAI UNIVERSITY CLUJ-NAPOCA FACULTY OF
MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION MATHEMATICS AND COMPUTER SCIENCE IN
ENGLISH**

DIPLOMA THESIS

Covid-19 exploration and prediction using machine learning

Author

Cristina MIERLĂ

Supervisors

Lecturer Professor Adriana M. COROIU

Teaching Assist. PhD. Horea-Bogdan MUREȘAN

Cluj-Napoca

2022

**BABEȘ -BOLYAI UNIVERSITATEA CLUJ-NAPOCA FACULTY DE
MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA MATEMATICĂ ȘI INFORMATICĂ ÎN LIMBA
ENGLEZĂ**

LUCRARE DE LICENȚĂ

Covid-19 explorare și predicție folosind învățarea automată

Autor

Cristina MIERLĂ

Supervizori

Lecturer Professor Adriana M. COROIU

Teaching Assist. PhD. Horea-Bogdan MUREȘAN

Cluj-Napoca

2022

Contents

ABSTRACT	2
CHAPTER 1: INTRODUCTION	3
Motivation.....	3
The need for innovation.....	3
Connection with medicine.....	3
Context	4
Objective.....	5
CHAPTER 2: THEORETICAL ASPECTS	6
From mystery to science.....	6
Artificial Intelligence (AI)	6
Data quantity and quality	9
Prediction	9
Machine Learning (ML).....	10
Artificial Neural Networks (ANN).....	11
Medical knowledge	14
Covid-19.....	14
Medical diagnosis and prediction	15
CT Scans in Covid-19 Detection	16
Tabular Data Analysis.....	18
Observations used to construct the model	22
Distribution and data visualization	22
How can we interpret the data?	26
The shortcomings of the data	31
CHAPTER 3: STATE OF THE ART	32
Comparison with related work.....	32
The innovation factors.....	33
CHAPTER 4: APPLICATION DEVELOPMENT.....	34
Scope	34
General details and requirements	34
Technologies used	35
Development process	36
Backend server	36
User interface	40
Model training and testing.....	48
CHAPTER 5: CONCLUSION AND FUTURE IMPROVEMENTS	55
Comparing results and findings	55
Further improvements	56
Conclusion	57
TABLE OF FIGURES	58
BIBLIOGRAPHY	60

Abstract

This study focuses on discovering correlations and similarities between patients with Covid-19 and developing an application that could serve as an aid for the medical staff in forecasting the state of a patient with this disease. It also concentrates on ways to combat unbalanced data, by using synthetically enhanced records and changing the structure of the distribution. An artificial neural network was used in predicting the outcome of a patient hospitalized with Covid, based on the initial dataset. This model is evaluated using accuracy, f1-score and by analyzing the confusion matrix created.

Chapter 1: Introduction

Motivation

The need for innovation

In this day and age, innovation is the driver for every new discovery. Artificial intelligence is one of the trendiest topics right now, besides blockchain, but that is another discussion. Because of its mysterious and not yet fully discovered nature, this technology attracts many engineers and mathematicians in finding new algorithms and optimizations problems. The Data Science field constitutes of many study subjects such as statistics, business, economics, mathematics, marketing, and many others. This is the reason why everybody is interested in artificial intelligence, more exactly machine learning.

Connection with medicine

We are so privileged to live today's age, where we have sterile vaccines and specialized medical staff that can perform complex operation. The need to always progress will encourage civilizations to growth and advancement. The medical department will always need improvements, so machines and smart algorithm are one path to prosperity. One of our biggest problems is us, the people; most failures are caused by human error or biases. Because it's impossible to construct the perfect human, machines are the next best thing. Another big advantage is the speed and storage capacities that are incomparable, a computer being able to perform simple tasks, such as multiplication, much quicker. With all this in mind, artificial intelligence will be a great aid in the near future of health care, it being already in use.

These domains come together to assist in a worldwide pandemic, the SARS COV-2 virus, or COVID-19. By using a smart system that would predict the outcome and treatment for a patient, the work would be diminished in every hospital or medical facility. In this moment artificial intelligence can only make suggestions and cluster data, the label of "diagnostic" is not yet fit for any machine. Decision making would be much easier with the help of a good machine learning algorithm, leading to a decrease in human error.

Context

In December of 2019 there was a virus discovered in China, that soon enough took over the whole planet. It created the newest pandemic the world encountered. In these moments it stands just above the majority of worldwide pandemics, in terms of death toll. In 2022 it still affects thousands of lives every day, but it's impact slowly decreased through these years. This disease affects the pulmonary system, heart and in small proportion other organs such as liver, brain, and muscles. As most illnesses, it greatly affected elderly and the people with autoimmune disorders. A way that this virus spread so rapidly was through air born particles, so coughs and sneezes are its best friend. One factor that also contributed to its rapid spread was the fact that it could have no symptoms on some hosts. In the beginning, the most common symptoms were high fever, cough, muscle soreness, loss of taste and smell. This list changed from time to time because of different mutations and variants of the virus. The medical system in most countries was overwhelmed and most hospitals had to turn into Covid hospitals. Because of this, we needed a solution to help the medical staff better manage this situation.

Cases		Deaths	
408M		5.8M	
+2.39M		+12,109	
Location	Cases ↓		Deaths
 Romania	2.53M		61,363
 United States	77.6M		918K
	+170K		+2,807
 India	42.6M		508K
 Brazil	27.3M		637K
	+164K		+1,129

Figure 1: Covid-19 statistics from 12.02.2022

Objective

Following the establish context, one of the most important aspects that needs to be understood from the beginning is the reason, what are we doing and why? Scientific relevance comes from smart questioning and an easy-to-understand purpose.

As we discovered, nowadays hospitals are filled with patients suffering from Covid-19. This overcrowding comes at a time when no one is prepared, and the implemented systems become overwhelmed. Machine learning comes as a quick answer in helping doctors and other medical staff in managing patients and supply distribution.

Due to the large number of cases, this moment is the perfect time to better study the optimization of diagnosis using artificial intelligence. For statistical purposes, we need to keep track of hospitalized people and their characteristics, such as gender, age, past conditions that may lead to the current situation. With all this gathered data we should assess how different groups of people are affected by this virus, such as different genders or age groups. This could be achieved by plotting the data based on various features.

In this moment, the only way to establish the severity of a patient is a human assessment or through lung CT scans. Another big problem that was discussed many times in the news was supply shortage. To better manage and prepare a patient and the whole hospital, it would be great if we could predict the outcome of a certain patient with a likelihood degree. This could be achieved by using artificial intelligence to learn the patterns and layers of the data to predict how people with similar values would progress.

To summarize our objectives:

- being able to perform basic operations on the data
- visualizing statistics about different features
- predicting the outcome of a new patient with a high amount of certainty

Chapter 2: Theoretical aspects

From mystery to science

Artificial Intelligence (AI)

Artificial intelligence can come in many forms and shapes, such as robots that know how to solve a simple puzzle, to computer simulators that can win popular games, such as Chess or Go, against the best players in the world. To better understand what artificial intelligence is, we need to start by establishing what is intelligent behavior. Are toasters intelligent because they don't burn toast, or is a security system more intelligent because it keeps us safe? I think both are. Further on, we will try to make this boundary less ambiguous.

We first need to examine the term intelligence, in the human context, defining it as a person's ability to understand, adapt based on their surroundings, retain information, and use knowledge to manipulate one's environment [1]. In more general terms, intelligence is defined as the ability to perceive and process data, transform data into information and use this knowledge towards goal-directed behavior [1]. Intelligence is composed from selecting stimuli and outside processes, filtering the information and build upon existing one. So, we can say that learning means forming an internal model of the external world [2]. For a machine to best emulate human behavior, it needs to follow the same steps its makers took while solving the same problem. This requires a design extremely similar to the one that already exists in our brain, neurons [3].

Other theories done by Knipers believed that at the root of intelligence lays creativity. This belief stands because people often perceive creative children to be intelligent [4]. Further on, anyone can attest to the fact that, if intelligence is not a result of creative things, intelligence and creativity coexist and depend on one another. This reasoning may at least be partly consistent because great things are achieved only by using both of them.

Researcher Doeben-Heisch, backed by Vogt in 2007, believed that intelligence is made from shared knowledge, such as language and culture [4]. Mehler thought that language is critical to human survival, as are humans critical to language evolution and distribution [4]. Social interactions are required to achieve evolution. Besides that, we are social creatures, so this part is not only obligatory but vital for a healthy and well-lived life.

Research done by Restino proves that the sociological aspects of our lives are crucial to our intelligence. Respective to this, he believes that a SOCIO agent, that learns new behaviors from others and gains knowledge through social interaction with humans, should be the future that we all aspire to [4]. Parisi and Mirolli in their paper "Language as a cognitive tool" [5] came to the conclusion that robots can do better in social situations if they can discern the difference between external and internal changes. By this difference, we mean how the environment can have other effects on their behavior compared to past actions, memories and learned

performances. Adding to this, Bittencourt and Marchi [6] tried to implement this type of internal-external difference. They believed that environmental stimuli provided "experience flux" and emotional values are decided based on the quality of information, good or bad [4].

MacLennan [4] thought that humans don't need to access consciousness for every ordinary task, such as walking, eating, or breathing. To add to this, any function can be automated if enough repetition has been done. Therefore, we cannot equate human-like behavior to consciousness [4]. We should not forget about the unconscious mind. It helps the brain retain, process, and tie all information received through the conscious phase. So, the mind being in a conscious state memorizes and makes low processes on all information received and the unconscious mind further refine, what was retained and ties all memories to past events. In the journey of building a human-like artificial intelligence, we should consider both, not only what the mind does while solving the problem but more important stages that happen before and after.

Another very interesting view I want to point out is that from Friedlander and Franklin [7]. They believed that we attribute mental states to other people based on our own subjective mental state. We build models or hypotheses in our hypothetical environments, that are unique to us and cannot be replicated. After interacting with these outside stimuli, that we perceive in our own particular way, we use this model to decide how we behave and react to certain situations [4]. Nilsson [4] tried to prove that this theoretical (abstract) model can significantly improve their learning rate. Just by giving a robot a hypothetical environment, where he can test different outcomes and see how they play out and the consequences of his actions, the robot showed results faster than robots that didn't use this kind of algorithm.

By 1980, researchers such as Brooks discovered that the best way to tackle building an artificial intelligence model is by creating individual models. These algorithms were made with respect to different aspects of the human brain, such as a planning module, a memory module, an arithmetic module, a language module, etc. Together, all these modules can form an intelligent being [4].

On the other hand, there is extensive proof that a machine doesn't need to exactly replicate the human brain behavior to arrive at the same outcomes as a human can [3]. Despite these similarities, artificial intelligence algorithms are not built with a brain-like structure in mind, but by using mathematical formulas and statistics [8]. We need to not forget the fact we don't yet fully grasp the full concept on how our brain works its mysterious magic. Not even the most advanced psychological principles could unfold the deep connections we have in our brain, at least not all of them [9]. There is also the well-discussed question: nature vs nurture. It is wrong to assume that we are born with no prior knowledge, in that sense many people believe that the best answer is both, we are born with all the tools for building our mind map. In this point in time, we don't have so much in common with artificial intelligence as others would like to believe. We cannot even define the concept of a "perfect brain" to study and try to replicate. Are blind people defect? Are children with learning difficulties

not a good reference in studies? Even if we could exactly define this notion of the brain, artificial intelligence lacks many capabilities that most human automatically inherit: incapability to recognize deeper essences of objects and the absence of the ability to question its beliefs, even if they are not logically wrong. It seems very intuitive to think that to achieve human-like behaviors we need to build a human-like machine. History and great achievements show us that this doesn't necessarily need to be the case. For example, planes were designed from research done on birds, but we accomplished a flying machine that doesn't really resemble or relate to natural flight [8]. Maybe we should invest more time in discovering the general nature of our intelligence [9]. One day, we may just hope to create a machine that will approach our intellect so close that the difference between the two is almost indistinguishable [3].

Even though this concept has a much older history, we can set the beginning in 1956 at a conference in Dartmouth College, when the phrase "Artificial intelligence" was first used [4]. Nonetheless, even though artificial intelligence exists for many years already, there are still many problems with it, such as data quality, accountability of the results, but also issues regarding the users, transparency, bias safety, and security. There have been many studies that show a clear racial bias in algorithms such as image recognition or language comprehension in automatically reviewing CVs and Resumes [10].

Artificial intelligence exists because we need automatic methods of simulating complex and time-consuming human behaviors, that apply knowledge and reason [3]. These technologies can be referred as a simulation of human intelligence with the help of statistics and logic, that can learn, reason, perceive, analyze, and process natural language [10]. In the grand scheme of things, artificial intelligence can help humans better interact with each other and with manmade technologies [10].

Nowadays, artificial intelligence is capable to solve problems from a diverse range of domains, while also further adapting the solution [3]. We can find applications of artificial intelligence everywhere in our day to day lives, but most notably in healthcare diagnosis, targeted treatments, service robots, fraud detection and illness recognition or classification in medical scans [10].

Over the years, artificial intelligence didn't evolve with the help of huge discoveries in the industry, but rather because of the enormous amount of data collected through the rise and extensive use of the world wide web. Because there are over 3 billion users connected simultaneously, there is more data being distributed than it can be collected. This data availability is combined with an immense computing ability given by powerful machines [10].

One big challenge in building AI algorithms is transparency because the majority of users are not literate in this field, and people are scared alarmed because of "scare media" [10]. A problem that nonprogrammers face is the "why?"; for us, the important question is "how?". On the other hand, disclosing much information about the data used in training the model can be a huge problem because of security, safety reasons and

vulnerability exploitation. On a higher level, if an algorithm makes a wrong prediction or classification, should the developer be held accountable for this mistake? Who can take responsibility? When working with any artificial intelligence model we should always keep this issue in mind, and not give this system higher stakes than humans. For now, machines should be an aid in our survival, not a guide. In the future, this may change because the training data has such a big part in the results, in contrast with how the program was developed. So, while artificial intelligence is sure to give the humanity great advantages and effortless life, it will come with the cost of entrusting our money, politics, life and much more to an artificial machine. With that being said, the designer should account for an agent's behavior and provide information about his decisions [10]. To ensure users' trust, it is best that we provide them with the necessary tools for understanding how the algorithm got to a certain conclusion.

Data quantity and quality

The story is much longer than simply "data is everything". We need to talk about the type of data, how the data is handled, how big is it and how it was acquired. Regardless of the task, we can agree that the data must be good, meaning correctly labelled and readable by a machine. It is also very important that the data is recent, or at least up to date with the problem we want to solve [11]. In a survey done in 2016, 38% percent of respondents said that the positive impacts of the algorithms will outweigh the negatives. On the other hand, 37% percent believe that negatives will outweigh positives [12].

The problem of information processing can be divided into two stages. The first part deals with the basic understanding of the problem, extracting the "what" and the "why", we can refer to this stage as the "theory of computation". The second part consists of choosing and implementing the appropriate model, so we need to answer the "how". Because the world is full of possibilities, we may be sure that the best method for solving the problem will come eventually [9].

A problem that is still debated these past years is privacy and data security. People became increasingly aware and scared that their personal information could become public and be used in malicious ways. Because of this, many laws such as GDPR were imposed in most countries. This makes research done on significantly important data almost impossible, putting harvesting personal data into a grey legal zone. Nowadays collecting data about people is extremely hard and it must be encrypted before being analyzed. Most people choose to look at fabricated data because it requires less resources and time to be gathered.

Prediction

The steps in guessing a future outcome can consist in transforming the initial data, applying an algorithm, evaluating the effectiveness of the output, modifying the data modelling, reapply the algorithm,

evaluating, and so forth, until it converges to a useful result. This type of informal deduction goes by many names in the artificial intelligence world: adaptive, self-organizing, learning [3].

Machine Learning (ML)

What makes machine learning algorithms different from any other type of system is that the programmer doesn't need to implement every step that the program needs to do [10]. This is a remarkable discovery in the world of automatization because some tasks can now be resolved without them being manually programmed. Through training with a large amount of data the algorithm generates a set of rules that are updated continuously with the help of retraining.

To better introduce the idea of machine learning in this discussion, let's talk about babies. How do babies learn almost anything? Through 2 very important processes: observation and reasoning of past or outside behaviors, trial, and error. At the core, the simplest idea of learning is choosing the hypotheses that best fits out brain model. We can think of machine learning in the same way; babies see how their parents walk or talk; artificial machines are feed big amounts of data that act in the exact same way. All information received by the target is filtered, interpreted, and analyzed. In the early stages of childhood, humans make connections between visual objects and auditory signals, like words. They also recognize patterns, for example when a human repeatedly points to a butterfly saying "butterfly", they will remember this and will get very confused if the next time the adult calls it a "watermelon". Above this, neurons make tighter connections when we make associations between 2 different, well-established, notions. For example, learning that the equality of the mean and median in a dataset can indicate a normal distribution of the data, 2 individual concepts that can be correlated. All in all, machine learning is the concept of learning something without being given the recipe, only the ingredients. Through trial and error and retraining, supported by an external actor, this system can deliver the desired outcome. These "ingredients" are the key component of this algorithm, meaning that the quality of the data will reflect how well it will deliver. We already touched upon this subject in the previous chapter. After this argument, we can say for sure that machines are not babies and they need so much more help from humans, weight, or bias adjustments, epoch modifications, modifications, and many others, depending on the complexity of the algorithm. Learning is more based on statistics, patterns, logic, imposed restrictions, than strengthening connections between neurons, like in the case of human brains. The comparison with babies helps us understand that data is the most important key in every artificial intelligence algorithm, but especially in machine learning. Many researchers attested to the fact that a healthy development of children needs human interactions. Kids that were put in placement homes, earlier than one year of age, develop language and understanding problems in later stages of evolution. To this sum up, this point, to make machines learn, they need the same care and help as children.

Artificial Neural Networks (ANN)

The first neural network system that led to this remarkable discovery was proposed by neuropsychiatrist Warren McCulloch and logician Walter Pitts in 1943 and was known as the mathematical idealization of the nervous system [13].

Once the media caught onto artificial neural networks, everyone began to have a very big interest in "thinking" machines. While the cybernetic movement arises, Frank Rosenblatt achieved an algorithm that could learn by approximating a function and after his discovery, the world accomplished the next best revelation,

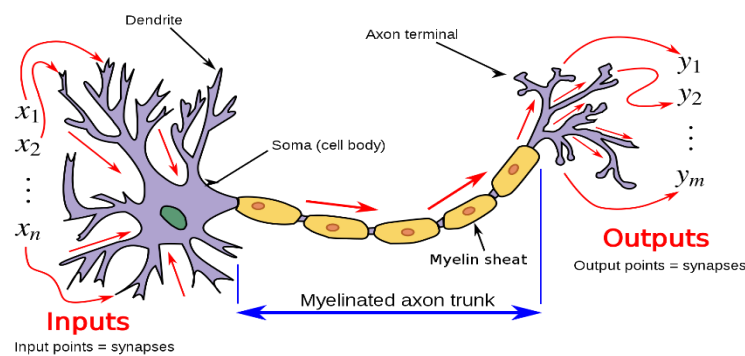


Figure 2: Human neuron depiction reflecting how an artificial neural was constructed [66]

backpropagation [13]. The McCulloch-Pitts [14] model remained one of most referenced artificial neural networks because of their logic, mathematical notations, and computations, and also their philosophical view. This discovery provided the first steps of building an artificial machine based on the human brain, using mathematics [13]. After this, there were no huge sightings, just new iterations of the existing technologies and advanced ways to transform logic and mathematical reasoning into neurological language [13].

A new era began in 1958, when Frank Rosenblatt made significant work on the idea of "perception". He believed that this was the core logic on how a machine can actually "learn". He believed that the nervous system is not a collection of logical connections, as it was believed in past works, but it uses probability and statistics. He suggested a new artificial neural network model that used statistical regression, one that had its roots in Legendre [13].

The next historical discovery in the world of neural networks was made by Cybenko in 1989, when he stated that these types of algorithms can even work with multiple layers. The catch was that they require more generalized statistics to train. Around 2006, scientists discovered that deep networks gave much better performances than shallow systems when applied complex patterns. With that, there was the rediscovery of backpropagation, now much more complex and necessary [13]. This method allowed systems to be trained using the regression method. At this time, everything was already discovered in a way or another, even before

someone acclaimed the concepts. History doesn't remember the people that do something the best way, but the first ones that do it right enough.



We should switch to a more practical view of this topic and talk more about how artificial networks are build these days. A neural network model will go through the following stages: creation, compilation, fitting, evaluation, and prediction. Compilation and fitting could be shortened by using the term: training. After the evaluation the model can be retrained till the error becomes small enough. This mechanism could be triggered automatically, or the iteration can be done manually using other parameters. In this part, we are interested in covering the general concepts of training a neural network model. In a later chapter we will talk more about the proposed solution.

In theory, a network contains some key components that can be furthermore broken down into more granular pieces. The main components are:

- input layer
- hidden layer(s)
- output layer

The first one is responsible for gathering data and configuring the layout of the network. This is the entry point from the real-world information to computer understood code. In the middle we have one or more hidden layers that are connected through weights.

A layer can have any number of nodes that each decides what information should pass through, based on some activation functions, such as Sigmoid, Softmax, ReLU, Linear, MaxOut, Binary step, and so on. The role of these nodes is to parse the data and search for any connections between the features and the target data. These decisions are made by changing the values of the weights and biases. Usually, these operations are done in the background, by the machine over many iterations. As programmers, we can control a few parameters such as the learning rate, the batch size, the number of epochs, number of layers and the activation functions for each one. As we mentioned before, backpropagation was one of the grates discoveries in the AI field of study, and nowadays is an industry standard. We should talk more about the notions just introduced:

-  learning rate - the amount or the frequency that the weights are changed during training; if we have a model with a high learning rate, its weights will change quickly and sporadically, making the error of the prediction jump up and down.
-  batch size - the size of the sample from the training data, that is iterated once; we can think of batch sizing as “how many times do we want to iterate over the whole set”; a small size brings many iterations but my include sample biases and small precision due to the low amount of data; whereas a big size of

batches (maybe a single batch equal to the training data size) could lack the power to overcorrect over multiple iterations.

🧠 epochs - the number of times that the algorithm is executed over the whole training set; a common example is an epoch that has a batch equal to the whole training set, this is called a gradient descent learning algorithm; usually this number is chosen to be big so that the algorithm would have the chance to learn every particularity about the dataset, however it affects time and space complexity.

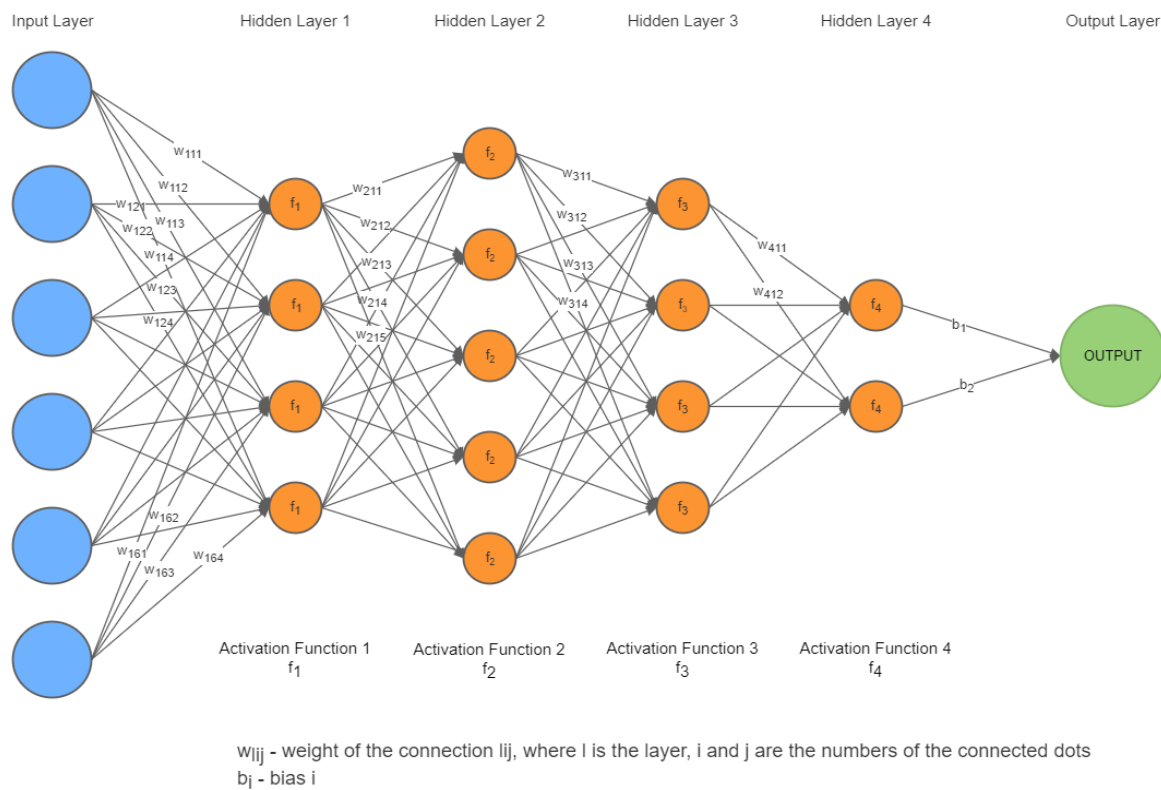


Figure 3: Diagram depicting the construction of a neural network

Medical knowledge

Covid-19

Covid-19 is a new emerging virus that originated from China in December 2019. This disease grew quickly by air born particle spread, it being derived from the Sars strain. It affected the body by attacking the lungs and the respiratory system [15]. The most common symptoms were cough, high fever, muscle soreness, loss of smell and taste.

In the first month of the pandemic, the World Health Organization (WHO) reported that 81% of confirmed patients were classified as mild or moderate, 14% were severe and lastly 5% of cases were said to be critical (classification done on 72314 patients) [16].

Because of the enormous number of Covid-19 cases, a large number of them suffer from post-Covid symptoms. This group of people was named in the literature [17] the “long-haulers”. The majority of symptoms include neurocognitive issues (dizziness, confusion), autonomic heart difficulties (tachycardia, palpitations), gastrointestinal (vomiting), respiratory problems (cough, general fatigue), musculoskeletal pain (arthralgias, myalgias) and psychological related trauma (post-traumatic stress disorder, anxiety, addiction, insomnia) [17]. This paper [17] claims that 70% of hospitalized patients show several post-Covid symptoms up to 3 months. Whereas 50-75% of non-hospitalized patients present no signs of symptoms after only one month free of Covid-19.

We define 2 methods from which we can conclude that someone suffers from past Covid-19 symptoms or as some may say, “long Covid”:

- if there is a clear relationship in time between the Covid-19 disease and symptoms related to it
- the symptoms preceding the virus infection should come after the disease has passed and they must be new or worsened for a certain patient

These circumstances [17] brought the question: when is a patient fully recovered from the disease, so when do we draw the line between Covid symptoms and post-Covid symptoms?

We can divide the patients in 3 categories: non-hospitalized, hospitalized, asymptomatic. Taking into consideration other aspects, such as intrinsic factors (age, gender, pre-existing comorbidities), extrinsic factors (biological, psychological, social) but also hospital factors (days in hospital, intensive care unit (ICU) admission, prolonged bedding). The authors from [17] created 4 distinct phases of the post-Covid symptoms:

- transition phase 4-5 weeks
- acute post-Covid 5-12 weeks

- long post-Covid 12-24 weeks
- persistent post-Covid >24 weeks

This study [18] is trying to categorize the most common risk factors for severe cases of the Covid-19 disease and death caused by it. As it has been discovered, this virus affects mainly the lungs, so from this we can easily say that people that already have any (acute) respiratory disease indicated severe diseases if infected with Covid-19. Also, hypertension and cardiovascular disease may lead to severe cases of Covid-19. Other studies discuss that obesity and smoking were associated with increased risk, but this correlation may also be caused due to the manner of these conditions, obesity and smoking being firstly related to heart and lung conditions, presented before. Adding to that, older age, cardiovascular disease, diabetes, chronic respiratory disease, hypertension, cancer, and other autoimmune disorders are highly linked to severe cases of Covid-19 and increased risk of death.

Because many people live their life without being diagnosed of their supposed diseases, it is still not very clear how important these comorbidities and preconditions affect the outcomes following the Covid-19 disease, in comparison to more quantitative features like age, gender, days spent in hospitalization or intensive care unit (ICU), or oxygen levels. Since these findings [18] are concentrated on extreme or rare comorbidities of people that end up in the hospital, we cannot apply these results on the population mass.

Medical diagnosis and prediction

The first step is to establish what do we mean by prognosis. This notion refers to predicting the risks of a specific patient based on his or her profile and background. Usually, by outcomes we refer to specific events, such as death or progression percentages [19]. Generally, doctors are interested in predicting the course of a certain illness, not the conclusion. This practice is utilized since the beginning of time by using tests and risk profiles, and a popular example would be FREENOME [20, 21, 22].

We can define 3 steps in this process:

- research (collecting and synthesizing information)
- detection (disease diagnosis by pattern recognition using symptoms data and medical images)
- prevention (prediction by calculating a person's probability of infection)

These AI methods facilitate a better decision-making process in diagnosis and charting a patient in a personalized way, and they may guide doctors to comprise the best after care routines for their patients [23]. A great advantage that is brought by intelligent systems is the reliability of the healthcare anywhere in the world, comprised and adjusted by specialists and engineering experts all over the planet.

We can even expand the idea of risk prediction to patients' background, by creating an electronic medical record. By syncing live information about patients with a predictive algorithm we can achieve early diagnosis of multiple diseases. This type of algorithm can also announce the primary caretaker of the patients of any high risk regarding their conditions [24].

The need of early diagnosis of this disease is extremely high due to its great rate of spreading (contagion), so that the affected people would be quarantined and treated appropriately [4]. Furthermore, as a result of the uniqueness of this disease, there is little to no data, also poor quality of the known cases [23].

Due to national shortage of rooms and supplies, a system that could predict the future risks of patients from a hospital is a powerful tool. Recently, the news reported that in Romania there is a higher demand in oxygen tubes for ventilators. If they could be prepared in advance for what is to come the shortage problem would be almost avoidable. This kind of studies help us better understand the disease and how it can be managed.

At the peak of the Covid-19 pandemic in 2020 there were a plethora of research done on tracking the spread and predicting the risk of this disease. In this paper [25], the author chose to analyze 729 of the papers, from which the majority were published in China and the USA. Some very interesting things I would like to emphasize are the main AI research focuses, related to the pandemic: prediction, classification, and diagnosis. Other extremely important piece of information we can get from this paper [25] is about the top 5 technologies used or specified in those papers: deep learning machine learning, artificial intelligence, convolutional neural networks, and transfer learning.

CT Scans in Covid-19 Detection

Artificial intelligence was used in image screening; the algorithm receives as input a computed tomography (CT) of a patient's lungs and it gives an overlay of the infected area [26]. This technique is very useful in illness prediction, risk assessment and diagnosis. CT imaging and risk prediction algorithms are a great tool when used simultaneously with the real-time polymerase chain reaction (RT-PCR) because this test still has a high rate of false-negative results, and it takes too much time for the response (24-48 hours) [27]. This test alone is not enough in pandemic times, especially when the number of daily cases increases rapidly, because of its low sensitivity and high loss of time. Using CT scans along sides the PCR tests, gives the doctors an advantage in early diagnosing the disease [28].

By using the newest technologies available we can achieve new results in the medical sector, which are more effective, more convenient, and more personalized. This type of testing was very adequate in Covid-19 diagnosis because this virus was prone to greatly affect the lung capacity and respiratory system. Some studies [29] showed that the sensitivity of chest CT scans to detecting Covid-19 is higher than the standard RT-PCR

tests (98% for the CT scans over 71% for the PCR tests). Other related research [28] used Gaussian filtering to pre-process the raw images and to calculate and choose the weights based on the type of the Gaussian function. This smoothing filter was very efficient in eliminating the noises. This shows that this type of algorithm can be of great use and help to better establish the way the disease affected the body, during the “positive” state of the illness, also after the symptoms have disappeared.

Even though many studies support the above claims, evaluations based solely on CT scans or clinical symptoms are not enough to achieve clinical standards. Therefore, many researchers that work in collaboration with hospitals try to combine all these results, and more, so that the model gives the best performance.

The National Health Commission of China approved that CT scans can be used in detecting the spread of Covid-19 in the lungs [27]. Because this analysis requires a trained radiologist to read and interpret them, there is a necessity to develop a deep-learning algorithm that could automatically predict the area and percentage of the virus in the lungs, but also the risk of it being Covid-19 and not any other pulmonary disease.

The paper [30] presents a new view by asking the question: What happens till a patient is confirmed with Covid? Should its symptoms and conditions be taken in consideration? What happens to patients that have false negative tests? Regardless the dataset that this research is based on, we do not know if these things were taken in consideration, if the days spent in hospitalization are considered from the beginning or from the positive Covid test.

This paper [29] compiled some of the most used supervised learning algorithm in image-based detection and classification: convolutional neural networks (CNN), support vector machine (SVM), logistic regression (LR), linear discriminant analysis (LDA) and random forest (RT). Let’s analyze some of the results presented here, in regards of the models used in the first 4-5 months of 2020:

Model	Accuracy	Sensitivity	Specificity
CNN	96.78%	98.66%	96.46%
SVM	98.71%	97.56%	99.68%
LR	97.30%	96.70%	98.30%
LDA	93.80%	90.00%	84.70%
RF	87.90%	90.70%	83.30%

Table 1: Comparison between multiple models in [29]

In [27] there were compiled multiple results of Covid-19 prediction algorithms based on CT scans:

- Xu [31] → 86.7%
- Wang [32] → 89.5%
- Sethy [33] → 95.38%
- Narim [34] → 98%

In [24] the author concentrated on multiple results and techniques and showed that a hybrid system composed of many kinds of models performs much better, having a higher accuracy, than a simple one.

Tabular Data Analysis

In [23] the authors attempted to analyze the effects of Covid-19 using a recurrent neural network system.

Because the mortality rises rapidly with age, less than 0.2% for people under 60 and 9.3% for individuals over 80 years old, this paper [18] concluded that age is far more important than past comorbidities, younger people with the same illness being five times less likely to develop severe forms of Coronavirus.

This observation about age is supported by [35, 36], where they analyzed the effects of age in the Covid pandemic situation. From 611583 patients that tested positive with Covid-19, the average age was 61.3 years, where 23.2% were over 80 years and less than 1% were younger than 50. In the figure below we can clearly see an exponential growth in the risk in patients over 60 years. Also, patients in the group >80 years had 60% higher risk compared to the group 70-79 [35].

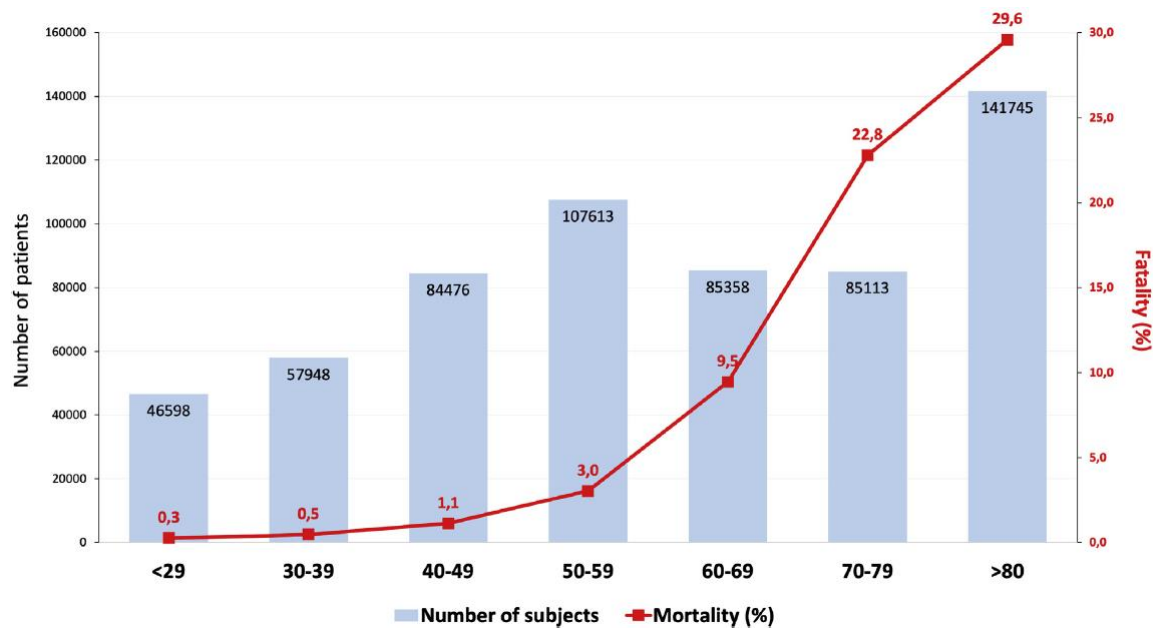


Figure 4: Table depicting high mortality rate correlated to higher age groups [35]

These results are also endorsed by other studies, such as [16] where the authors said that higher risk of death associated with Covid-19 is resulted from older age, male gender, comorbidity, elevated D-dimer, and organ disfunction. This study [16] suggests that patients that were discharged as critically or severely ill display significantly higher C-reactive protein, neutrophil count, serum potassium, cardiac troponin 1 (CTN1), PCT, brain natriuretic peptide (BNP), D-dimer, but hemoglobin and serum albumin were significantly lower. In the critically ill group, there were observed higher levels of platelet count, fibrinogen and serum calcium compared to the severe category.

Variable	All (n = 598)	Illness Severity			P value
		Moderate (n = 400)	Severe (n = 85)	Critical (n = 113)	
Age (years)	57 (42–66)	52 (39–64)*†	61 (49.5–67)*	65.5 (58.25–72)	<0.0001
Sex					0.067
Male	347/598 (58.03)	219/400 (54.75)	54/85 (63.53)	74/113 (65.49)	
Female	251/598 (41.97)	181/400 (45.25)	31/85 (36.47)	39/113 (34.51)	
Smoking					0.883
Never	525/584 (89.90)	355/399 (88.97)	74/85 (87.06)	91/100 (91.00)	
Current	44/584 (7.53)	33/399 (8.27)	4/85 (4.71)	7/100 (7.00)	
Former	15/584 (2.57)	11/399 (2.76)	2/85 (2.35)	2/100 (2.00)	
Symptoms					
Fever (temperature $\geq 37.3^{\circ}\text{C}$)	454/559 (81.22)	324/400 (81.00)	70/85 (82.35)	60/74 (81.08)	0.958
Dry cough	188/559 (33.63)	120/400 (30.00)*	43/85 (50.59)	25/74 (33.78)	0.001
Sputum production	172/559 (30.77)	120/400 (30.00)	22/85 (25.88)	30/74 (40.54)	0.112
Pharyngodynia	38/559 (6.80)	31/400 (7.75)	6/85 (7.06)	1/74 (1.35)	0.132
Chest pain	28/559 (5.01)	18/400 (4.50)	5/85 (5.88)	5/74 (6.76)	0.581
Shortness of breath	149/559 (26.65)	59/400 (14.75)*†	59/85 (69.41)*	31/74 (41.89)	<0.0001
Any comorbidity	301/584 (51.54)	167/399 (41.85)	40/85 (47.06)	94/100 (94.00)	<0.0001
Hypertension	198/584 (33.90)	91/399 (22.81)*†	31/85 (36.47)*	76/100 (76.00)	<0.0001
Cardiovascular disease	38/584 (6.51)	25/399 (6.27)	4/85 (4.71)	9/100 (9.00)	0.469
Diabetes	77/584 (13.18)	42/399 (10.53)*	7/85 (8.24)*	28/100 (28.00)	<0.0001
Carcinoma	21/584 (3.60)	11/399 (2.76)*	2/85 (2.35)	8/100 (8.00)	0.048
Cerebrovascular disease	19/584 (3.25)	13/399 (3.26)	0/85 (0.00)	6/100 (6.00)	0.052
COPD	11/584 (1.88)	7/399 (1.75)	2/85 (2.35)	2/100 (2.00)	0.809
Others	99/584 (16.95)	86/399 (21.55)*†	6/85 (7.06)	7/100 (7.00)	<0.0001
Days from illness onset to diagnosis confirmed	4 (2–7)	4 (2–6)	8 (5–13)*	7 (4–11)	<0.0001
Days from illness onset to admission	7 (3–11)	6 (3–10)*†	10 (6–14)	10 (0–14)	<0.0001

Figure 5: [16]

Other research [8] has shown that there exists a correlation between age, sex, certain comorbidities, ethnicity, and obesity in Covid-19. A study [25] that analyzed only 3 features, using a machine learning system, made use of the lactic dehydrogenase, lymphocyte count and high-sensitivity C-reactive protein and achieved almost 90% accuracy.

This paper [37] presents a study conducted on 155 people (82 men and 73 women) that were tested positive for Covid-19 using that PCR test. Above that, the patients were admitted and treated at the same hospital. The average age for this study is 64, with the range being 59.5–81. Upon admission, every patient received a set of standardized tests, physical examinations, and chest scans. The CT scans were analyzed regarding the severity level and diagnosing inflammatory changes.

The results from this research [37] may bring significant interest for my study. The results concluded that the medical analysis and tests are highly correlated to the degree of the severity. This evidence suggested that some laboratory parameters increase from a mild case of Covid-19 to severe, and the others are inversely proportional. We can divide the findings into 3 categories:

- values that increase from patients with mild symptoms of Covid-19 to severe cases
- values that decrease from patients with mild symptoms of Covid-19 to severe cases
- data without significant differences

Comparative analysis of laboratory parameters in patients upon admission to the hospital, depending on the severity of COVID-19					
Parameter	Clinical severity 1	Clinical severity 2	Clinical severity 3	Clinical severity 4	p^*
Leukocytes	6.3 ± 3.6	7.67 ± 3.9	8.52 ± 4.7	9.47 ± 4.8	0.007
Neutrophils	4.46 ± 3.2	5.55 ± 3.7	6.90 ± 4.5	8.10 ± 4.8	0.02
Total protein	65.3 ± 6.8	64.3 ± 5.2	63.2 ± 5.1	58.4 ± 4.6	0.02
Albumin	36.4 ± 4.1	33.9 ± 5.6	32.6 ± 5.4	29.1 ± 2.6	0.001
Urea	7.3 ± 2.4	6.9 ± 2.2	10.7 ± 3.8	15.3 ± 4.1	0.002
Creatinine	126.5 ± 31.2	114.8 ± 30.1	116.5 ± 29.6	176.1 ± 32.6	0.005
Bilirubin	13.5 ± 4.1	11.3 ± 3.9	13.4 ± 3.6	21.9 ± 5.2	0.007
AST	59.2 ± 26.2	51.9 ± 23.3	91.0 ± 59.3	102.8 ± 65.5	0.01
Calcium	0.98 ± 0.35	0.93 ± 0.24	0.85 ± 0.29	0.49 ± 0.16	0.002
Glucose	6.99 ± 2.7	7.46 ± 2.8	8.28 ± 3.2	11.9 ± 3.8	0.02

Figure 6: [18]

After examining Figure 6: , we can complete the categories with the corresponded evaluations:

- leukocytes, neutrophils, urea, creatine, glucose, aspartate aminotransferase (AST), bilirubin
- total protein, albumin, calcium
- sodium, potassium, chlorine, iron

Comparative analysis of laboratory parameters in patients with COVID-19 upon admission to the hospital, depending on the degree of changes detected by CT of the chest organs (CT0–CT4), $M \pm SD$					
Parameter	CT1	CT2	CT3	CT4	p^*
Neutrophils	4.78 ± 3.3	5.98 ± 3.7	5.73 ± 3.6	7.46 ± 3.9	0.01
Leukocytes	1.66 ± 0.62	1.18 ± 0.71	1.29 ± 0.9	0.94 ± 0.32	0.001
Albumin	36.9 ± 4.6	33.3 ± 3.7	33.6 ± 5.2	30.4 ± 2.8	0.001
Urea	6.3 ± 5.2	8.1 ± 3.6	8.5 ± 4.7	12.1 ± 8.3	0.001
Creatinine	123.4 ± 78.1	130.3 ± 65.4	103.5 ± 56.9	152.1 ± 90.0	0.01
AST	43.6 ± 12.5	62.3 ± 24.9	75.2 ± 28.7	77.1 ± 30.8	0.02
LDH	563.4 ± 102.5	826.2 ± 259.4	866.9 ± 234.9	$1,103.0 \pm 522.4$	0.003
Calcium	1.06 ± 0.46	0.94 ± 0.21	0.84 ± 0.32	0.703 ± 0.18	0.01
Glucose	6.49 ± 2.3	7.41 ± 2.7	8.12 ± 2.5	10.0 ± 4.2	0.001
D-dimer	583.2 ± 132.4	$1,780.9 \pm 1,446.9$	$1,663.6 \pm 1,165.4$	$1,750.3 \pm 1,240.8$	0.001
CRP	48.4 ± 23.7	125.8 ± 82.2	127.4 ± 73.5	171.0 ± 90.4	0.001

Figure 7: [18]

These results were assessed [18] using a second group that was separated in illness severity using CT scans results. Some measures were backed up and confirmed these findings; patients with mild traces in the lungs have lower levels of neutrophils, urea, creatine, AST and blood glucose and a higher level of calcium. Additionally, other laboratory makers were set in place: lymphocytes, eosinophils, lactate dehydrogenase (LDH), D-dimer and C-reactive protein (CRP). By compelling changes among the different severity groups, it appears that patients with mild cases of Covid-19 have higher levels of lymphocytes and lower levels of LDH, D-dimer and CRP, compared to the sever cases.

So, from all the above, the authors of [18] conducted that there is a strong positive correlation between the clinical and radiological severity and the levels of neutrophils, albumin, creatine, urea, and calcium; a strong negative correlation with lymphocyte counts and a moderate positive correlation between the level of D-dimer, glucose, and CRP and the radiological severity.

By further analyzing lethal cases using postmortem examination, forensics found that 97.5% of patients died from complications due to acute respiratory distress syndrome (ARDS).

Distribution of comorbidities in patients with different outcomes of COVID-19		
Parameter	Discharged patients, $n = 73$	Patients with the lethal outcome, $n = 82$
Arterial hypertension	38 (52%)	67 (81.7%)
Ischemic heart disease	33 (45%)	52 (63.4%)
Chronic heart insufficiency	9 (12%)	44 (54%)
Diabetes	29 (40%)	29 (35.4%)
Malignancy (in the medical history)	44 (60%)	13 (16%)
Chronic lung diseases	47 (64%)	26 (32%)
No comorbidity	26 (36%)	12 (14.6%)

Figure 8: [18]

Another study [30] chose to consider the following features about a given patient: demographics, comorbidities, symptoms, exposure history, vital signs, laboratory results, chest radiograph and chest computed tomographic imaginary results, disposition, and treatments.

Nominal data was processed by transforming them into frequencies or just keeping the original values. For continuous data, the mean with the standard deviation or the median with the interquartile ranges, were used. To distinguish between different groups, the Fisher's and Mann-Whitney U tests were performed.

If we want to take into consideration the medication that the patient received, the study [16] presented that 61.07% of the patients received lopinavir/ritonavir antivirals within 2 days of hospitalization; 70.83% received antibiotics and 23.08% corticosteroids. From this dataset, they concluded that 60.18% of patients assigned in the critical group died (68 out of 113), 5.88% of the severe cases (5 out of 85) and only 1.50% in the moderate category (6 out of 400).

Observations used to construct the model

Distribution and data visualization

In this chapter we will talk more about the dataset managed by the application and used in training the algorithm. We will start by establishing how it was obtained and what does it contain. At the end of the chapter, we will progress by explaining how it was processed and “translated” to be understood by the prediction model.

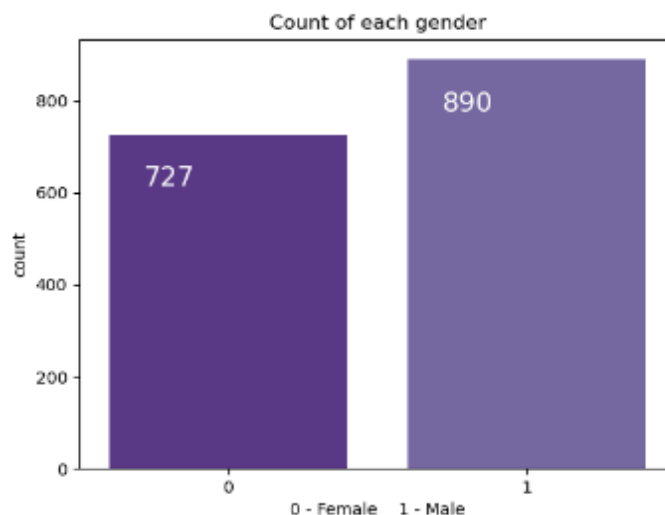


Figure 9: Distribution of gender in the dataset

The dataset used to achieve this model was received from Clujana Hospital and made up from data collected during the Covid-19 patients in 2020 and early 2021. This pandemic revealed a lot of shortcomings in every hospital, this signaled the negligence present in these highly regarded institutes. This hospital gathered data about hospitalized patients with a positive RT-PCR (real time polymerase chain reaction) test. Besides that, most patients received pulmonary CT scans to reveal the severity and damage of the virus in the lungs, the TSS (total severity score). The information included in the data set was: age, gender, initial diagnosis, release diagnosis, number of days spent in hospitalization, number of days spent in ICU (intensive care unit), list of comorbidities, medication received, hospital test results, respiratory aid procedures, CT procedures, human written description of the CT result, severity of the illness, release date and type of release. From all these features we found that the information about the CT results was almost irrelevant or out of the scope of this project, also the respiratory aid procedures were not used because they were too similar along all received patients, meaning that this information became almost irrelevant.

This dataset contains 1617 patients, from which 727 females and 890 males. The age has a skewed distribution plot, leaning more towards older patients, most being present in the 66-73 years group. This grouping was done by using the frequency of each age, and by creating the groups through Sturges' rule.

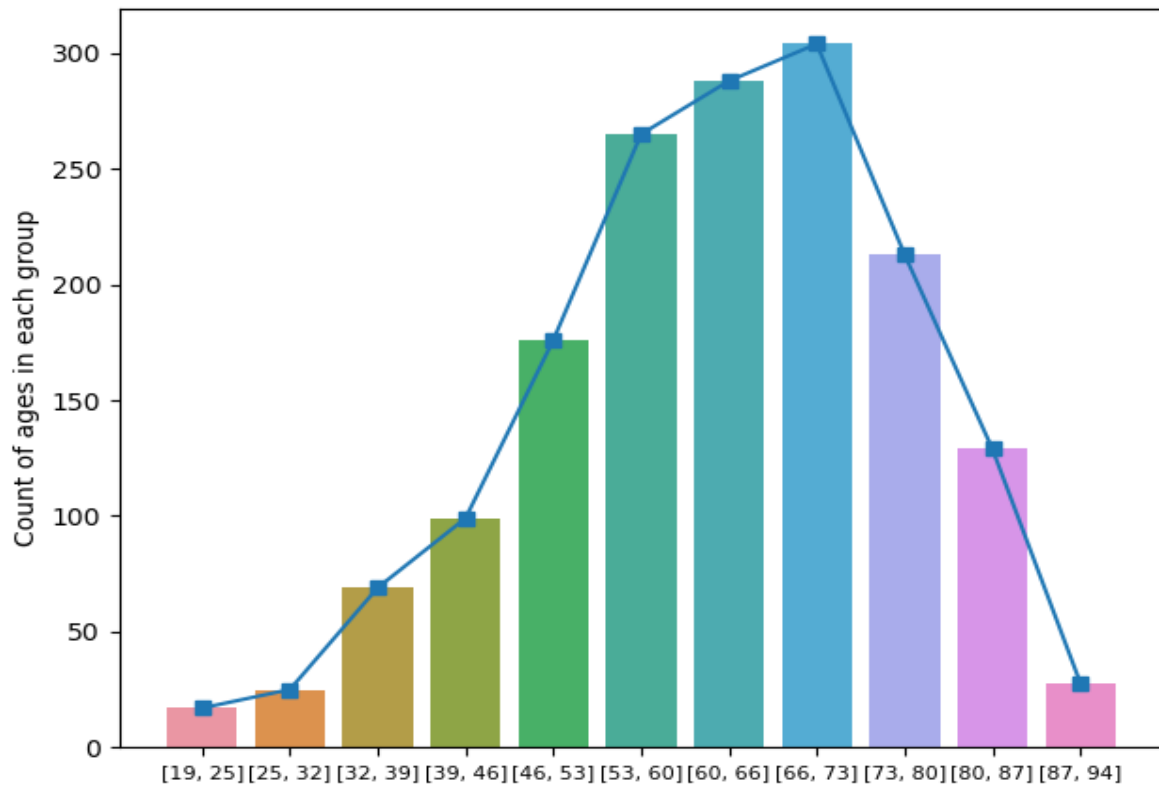


Figure 10: Distribution of age groups in the dataset

For easy interpretation, the hospital release states were converted into numbers: cured - 0, improved - 1, stationary - 2, worsened - 3 and deceased - 4. One major discovery in this dataset was unbalanced data. The number of patients assigned to each label was not equally distributed, there were 101 cured patients, 1224 improved, 52 stationary, 7 worsened and 233 deceased. From a total of 1617 patients, it was extremely clear that this is going to be a problem, because this column represented the prediction target. In a future chapter we will discuss the solution proposed.

This dataset also contains a column depicting the type of release: mild, moderate, severe. These results were also unbalanced and because of the limited nature of the group (only 3 types) it became irrelevant for the model. This column was used only for data comparison and endorsement (review).

Another important information included was the number of days spent in hospitalization and ICU (intensive care unit). This tells us that the more days a patient spends in the hospital, the more serious the illness becomes. Besides that, the ICU staying increased the severity of the disease exponentially. This thinking led us to the conclusion that the number of days spent in hospitalization greatly impacted the outcome of the patient.

By comparing these results (days spent in hospitalization and ICU, release state and disease type) it was possible to see how they relate to each other. Because of the unbalanced nature of the raw data, these results don't seem to correlate as much as we expected, not only that, but they did not bring any new findings.

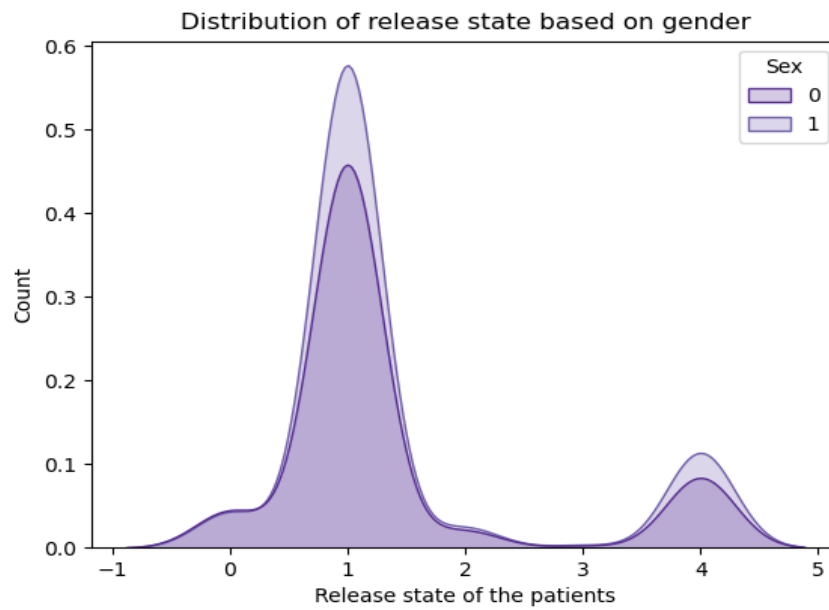


Figure 11: Distribution of release state based on gender

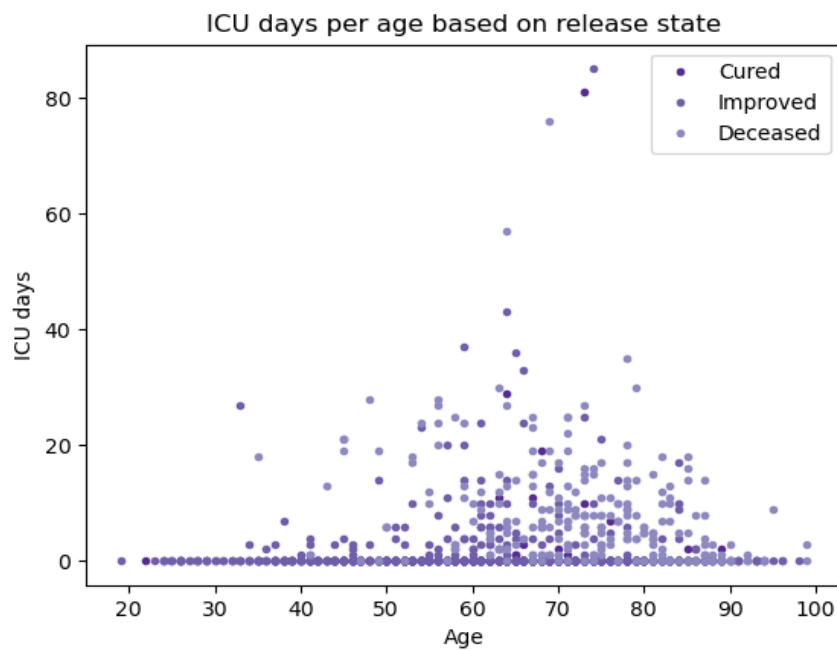


Figure 12: Distribution of days based on the number of ICU days and release state

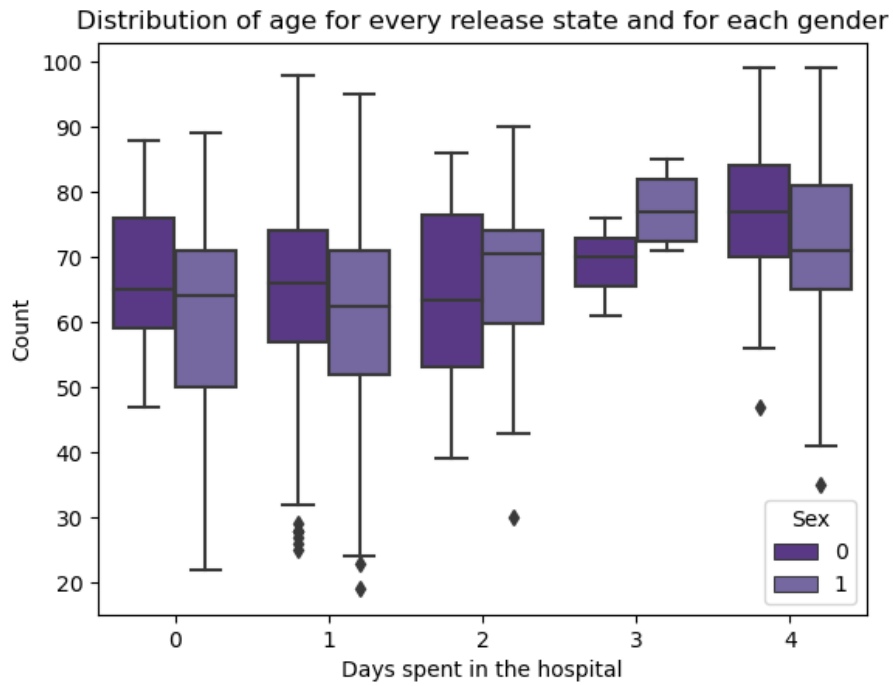


Figure 13: Distribution of ages for every release state category and every gender

This dataset also contained 2 columns for medication received and one for investigations done in the hospital. The first one did not seem so significant for the outcome we wanted to achieve but it was included in the final model. The ladder was treated with great importance because of the existing literature, a thorough comparison being done in a later chapter. The most common investigations included tests for red blood cells, leukocytes, and hemoglobin (roughly on 1600 patients).

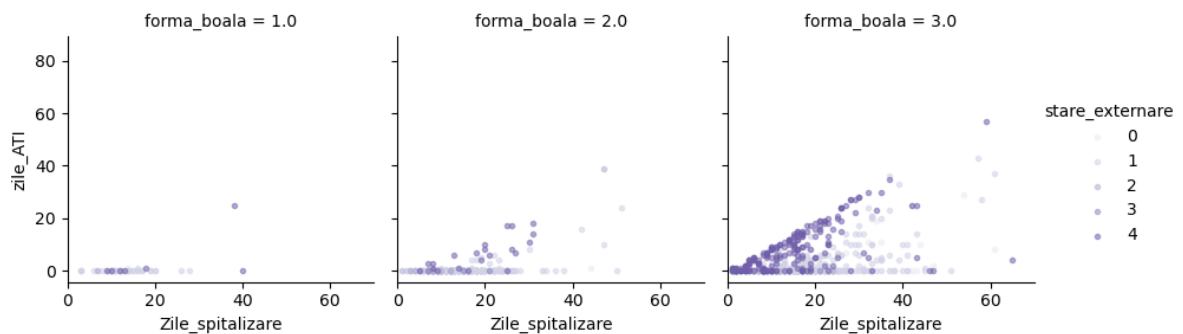


Figure 14: Correlation between days spent in hospitalization and ICU for each release state and type (1-mild, 2-moderate, 3-severe)

In the comorbidities column, the medical staff recorded past illnesses and conditions that could be related to Covid-19. This data seemed to be of great importance because of its strong relation with

heart and pulmonary disease. Because this column contains only the names of the illnesses that a certain patient possesses it becomes very difficult for an artificial machine to interpret such information. We as humans can easily discern from a patient A that is 25 years, no past illnesses only mild headaches, and patient B with 54 years, with pulmonary cancer; the second one is most likely to have a harder time going through Covid-19. On that list there are over 1300 unique diseases among which most patients present at the hospital had essential (primary) hypertension and acute respiratory failure. The last two columns, initial diagnosis, and release diagnosis, contains illnesses from the same list as the one above but the difference is that there is only one diagnosis per patient.

Diag_pr_int	Diag_pr_ext	Zile_spitalizare	zile_ATI	Sex	Varsta	Comorbiditati	Medicatie	Analize_prim_set
J18.9	J17.8*	14	0	F	86	E87.6 Hipopota	algocalmin s	UREA - 34 mg/dl) (1:
NULL	J12.8	20	2	M	86	E11.65 Diabet r	algocalmin s	UREA - 117 mg/dl) (:

Figure 15: Example of raw data from the dataset

How can we interpret the data?

The categorical variables such as gender and release state can be easily interpreted using a numerical relationship (female - 0, male - 1). For numerical data the story is also simple because we can just use the given values (for example when we are talking about the age).

The medication column contains human written data about the patient, which is nearly impossible to interpret for a computer.

Medicatie
algocalmin sol inj 1 g/2 ml azitrox (r) 500 cefort 1 g clorura de sodiu 0,9% pl 500 ml clorura de sodiu 0.9%-250 ml coc anxiar 1 mg apa pentru preparate injectabile b. braun 500 ml aspenter 100 mg aspenter 75 mg candesartan atb 16 mg algocalmin sol inj 1 g/2 ml algozone 500 mg azitrox (r) 500 bisotens (bisoprolol) 10 mg candesartan atb 16 mg cefort

Figure 16: Medication column

We can solve this problem by using a technique named One Hot Encoding [38] which transforms a column that contains multiple types or categories into multiple columns that only contains a True/False value. This algorithm was custom implemented for this dataset in particular, so that it would better fit the initial values of the dataset. In this moment there are multiple columns and for each patient there is a 0 or 1 for whether he took this type of medication.

algocalminsolinj1g/2ml	cloruradesodiu0,9%pl500ml	cloruradesodiu0.9%-250ml	codeinafosforicaeel15mg
1	1	1	1
0	1	0	0
1	0	1	0
1	1	0	0

Figure 17: One hot encoded medication column

By doing that, the model can easily interpret this information about patients. Unfortunately, one big drawback of this technique is that it makes the dataset bigger than it already is, and in most cases very large data drags down the model. The same solution was also proposed for the investigations column. The big difference in this case is that the new columns are not Boolean anymore, because for every test we are also given the result for that given test. So instead of using 1 or 0 we can complete the new column with the given value.

Analyze_prim_set
UREA - 26 mg/dl) (13 - 43) GLICEMIE - 90 mg/dl) (70 - 110) ASAT/GOT - 30 U/l) (0 - 31) ALAT/GPT - 19 U/L
UREA - 84 mg/dl) (13 - 43) GLICEMIE - 242 mg/dl) (70 - 110) ASAT/GOT - 64 U/l) (0 - 35) ALAT/GPT - 74 U/L
UREA - 34 mg/dl) (13 - 43) GLICEMIE - 118 mg/dl) (70 - 110) ASAT/GOT - 45 U/l) (0 - 31) ALAT/GPT - 17 U/L

Figure 18: Investigations and results

UREA	GLICEMIE	ASAT/GOT	ALAT/GPT	Creatinina	CK-MB*	LDH	CK
26	90	30	19	0	15	392	22
84	242	64	74	1	18	825	58
34	118	45	17	0	0	728	0

Figure 19: One hot encoded investigation column

One other problem that arises is the patients that did not need this types of investigation performed. In this moment we completed empty spaces with 0 but this can lead to a bad and unusable model in 2 cases: 1. if a certain unperformed test would had have a bigger number than the expected the result, and 0 is an outlier value for that group, but it does not match the characteristics for that patient; 2. if a test has results around the 0 value and the patient would have scored different if he would have had the chance to have this analysis done, and now he is considered in the “average” or “normal” pile, when it should not be the case. These 2 possibilities can be shortly rephrased in Type 1 error and Type 2 error [39].

As for the comorbidities column, the story takes an interesting turn. Firstly, we need to introduce more data that was gathered about the patients. For each illness we counted how many patients

had it and count how many ended up in each group. For a better understanding, we will take some examples:

Boala	Count	Vindecata	Ameliorata	Stationara	Agravata	Decedata
I05.0 Stenoza mitrala	6	0	3	0	0	3
I10 Hipertensiunea esentiala (pri	1010	69	743	25	6	167
I25.9 Cardiopatie ischemica cron	247	14	159	7	1	66
I34.0 Insuficienta mitrala (valva)	204	15	139	3	1	46
I35.1 Insuficienta (valva) aortica	62	4	41	1	0	16
I48 Fibrilatia atriala si flutter	220	14	137	6	0	63
I50.0 Insuficienta cardiaca conge	119	5	79	5	0	30

Figure 20: Computed counts for every comorbidity, based on how many patients end up in each release state

In the first column there is the name and the code of the illness we want to analyze. In the Count column there is the total number of patients that have this disease. The rest of the columns represent the number of patients that had each of the 5 given outcomes. This information matters because this can reveal the mortality percentage for every disease in this dataset. This is important in establishing how probable is for a patient, with some characteristics, to end up in one of the given 5 release states. These results bring with them all the problems this dataset has, mainly unbalanced data. Just in the example above, we can clearly see that most of the values lie in the Improved (Ameliorat) column. This is the next problem we want to solve. How can we tell the computer that not every comorbidity should lead to Improvement?

We also counted how many overall patients lie in each group (101 - Cured, 1224 - Improved, 52 - Stationary, 7 - Worsened, 233 - Deceased => Total = 1617). If we suppose that each group had “fair odds”, we could say that the probability to end up in any of them would be:

$$\frac{\text{total number of patients in that category (favourable)}}{\text{total number of patients (total)}}$$

Because we know this is not the case, we cannot rely on this simple equation. Each fraction will be proper or sub unitary (below 1, the unit value). Because of that, we may be able to “reverse the ratio” just by subtracting the result from 1 (the unit), the inverse of the probability. What are we trying to do and why? The intention is to create a way in which we transform a big value into a small one and the other way around. In other words, we know that only 6% of the patients in the dataset will have a positive outcome ($\frac{101}{1617} = 0.062$) and 75% of the patients were included in the improved group ($\frac{1224}{1617} =$

0.756), we can “turn around” this ratio by subtracting from 1 ($1 - 0.062 = 0.938 \Rightarrow 93\%$ and $1 - 0.756 = 0.244 \Rightarrow 24\%$). Further we try to define some set names for these constants:

$$\text{cured: } 1 - \frac{101}{1617} = 0.938 \Rightarrow 93\%$$

$$\text{improved: } 1 - \frac{1224}{1617} = 0.244 \Rightarrow 24\%$$

$$\text{stationary: } 1 - \frac{52}{1617} = 0.967 \Rightarrow 96\%$$

$$\text{worsened: } 1 - \frac{7}{1617} = 0.995 \Rightarrow 99\%$$

$$\text{deceased: } 1 - \frac{233}{1617} = 0.855 \Rightarrow 85\%$$

We can also notice they sum (almost, because of the decimal approximation) up to 400, which is exactly 5 (release states) $\times 100$ (percent) $- 100 = 400$. The goal was attained, the dataset was turned “upside-down”.

When we are talking about a certain comorbidity, we can gain a new information, the number of patients in that release state group. We already establish that because of the overall imbalanced data of the set, the number do not truly reflect how many patients should be in each category. If we multiply the number of patients from an illness group having a given release state with the above constants, we could actually “balance out” the numbers.

Example: I25.9:	Cured: $14 \times 0.938 = 13.132$
	Improved: $159 \times 0.244 = 38.796$
	Stationary: $7 \times 0.967 = 6.769$
	Worsened: $1 \times 0.995 = 0.995$
	Deceased: $66 \times 0.855 = 56.43$

This is a big improvement from the big spike that we had from the improved section, but there is more to do. How can we tell the machine that these release states “line on a spectrum”, from best to worst? We can just tell him that by creating a relation from cured to -1 and from deceased to 1 . The middle values will be chosen in the following way: improved: -0.5 , stationary: 0.25 and worsened: 0.5 . The inclusion of 0 would not have been a great idea because it would have caused the data to negate in the wrong places and knowing that the “left side of the scale” (where the improved section is) is

“heavier”, the value was placed in the right side. Let’s work further by changing the values in the above example according to the added rules:

Example: I25.9: Cured: $13.132 * (-1) = -13.132$

Improved: $38.796 * (-0.5) = -19.398$

Stationary: $6.769 * 0.25 = 1.692$

Worsened: $0.995 * 0.5 = 0.497$

Deceased: $56.43 * 1 = 56.43$

In this moment we computed a distinct value for each of the 5 target categories in connection to one of the given comorbidities. To measure a custom “severity rate” for each illness we try to sum up the values that we got. This will help us in establishing “how deadly an illness” can be, in relation with Coronavirus. For the above example, by summing up the results we the value: 26.089, which corresponds to the one computed by the algorithm (for simplicity in this example the numbers were rounded to 3 digits in the fractional part).

I05.0	2.203154
I10	-3.00819
I25.9	26.23392
I34.0	9.640847
I35.1	5.203927
I48	25.59988
I50.0	12.5991

Figure 21: Final weight of every comorbidity

How can we interpret these numbers? In a few words, as the value becomes bigger the chances of it to lead to a severe outcome rise. These results are much more straightforward to be understood and interpreted by a machine.

The next question that we come across is how can we relate these numbers to each patient? We know that the comorbidity column contains a list of conditions, and we also have a specific value for each comorbidity. To compute the likelihood of a severe outcome for a certain patient we tried to sum up all the corresponding values for his comorbidities. Right now, every column has a numeric value that can be read and learned by an artificial algorithm.

The shortcomings of the data

Even though this dataset is virtually perfect for being filled with data about real patients, we cannot ignore the obstacles we are facing. To begin with, we need to be aware that this dataset has a human bias because the records were manually imputed into another application and exported as an Excel file. This makes the columns containing descriptions unreliable and unpredictable. These results are not double checked with any other hospital or medical unit. Other institution might have discovered some similar results as the dataset revealed, but this is just an assumption.

As was iterated many times, the number of patients in each of the 5 target categories was not equal, not even close, making the dataset unbalanced. This may lead to overfitting the model [39]. We will come back to this problem in a later chapter, where the used solution is explained.

The comorbidities column brings with it many problems; because its values are computed based on the given dataset the data is strictly relevant to this scenario and cannot be generalized to other diseases or hospitals. This is caused because “outside data” is not taken into consideration and what is in the data set is seen as absolute truth. In reality, we will never know if a patient with a specific condition was just an exception, and in some other “normal or regular” settings he would have a totally different outcome. For a better understanding we can take a simple example: let’s imagine we have patient X with a severe type of lung cancer. They magically survive though Covid-19 while in hospital and end up in the “stationary” category. Because an artificial “brain” is incapable of knowing that this was just a miracle, it will put together that “lung cancer does not affect mortally at all”, which any human could easily contradict. These type of unknown exceptions and patterns can make illnesses fall in the wrong category, giving future patients a wrong diagnosis. Sure, this information could be relearned and adjusted whenever a new patient would be assessed, but how many patients would it take for this system to be right enough? These restrictions can break the algorithm and give an erroneous prediction.

Lastly, this disease is rapidly changing and the medications and treatment for it are improving constantly. Because of this, it is very unlikely and unrealistic to think that 2022 findings related to the pandemic will correlate to information from 2020. In this short amount of time, this virus has mutated many times already and new vaccines were created that influenced the outcome of infection. These new unknown changes may influence treatment and hospitalization decisions, so it is considered a crucial key in solving this problem. This does not mean that this research and data are now completely null and irrelevant, constantly looking for new questions will certainly improve later outbreaks. We are also interested in comparing the data from this hospital with other results from around the world.

Chapter 3: State of the art

Comparison with related work

Compared to the cited papers in the theoretical part of this thesis, the used dataset also contains similar details about the patients such as age, gender, and the number of days since the hospital admission. In addition, many papers [8, 16, 18] mention information about preconditions linked to worsened state cases of Covid-19. Some works contain a set list of important and notable diseases, using Boolean values for each patient. The given dataset treats this information very differently, in total being approximately 1300 unique illnesses it was impossible to use Boolean values in every column with a given illness, for each patient. By keeping this information in one column, the size of the dataset was not changed. In this column the list of conditions was replaced with a numeric value, calculated using the formula presented in the chapter 2, subchapter Observations used to construct the model, how can we interpret the data.

An important finding noted in [37] is how test results differ based on the patient outcome. These tests were chosen to correspond to the ones mentioned in the cited paper, to be precise, number of blood leukocytes, UREA, level of creatine protein and glucose. By plotting the data that is also named in the paper we can see how it is related to other findings. Data correlation comes from a linear relationship, forming a diagonal line like the graph of the equation: $x = y$, for a positive correlation, and a line similar to the graph of: $x = -y$, for a negative correlation. Because of how the results were displayed, it is inconclusive if there is a relation between the release state and the outcome of these 4 hospital tests. It was observed that in the used dataset the 4 common tests performed do not correlate as strongly as presented in [37]. This could mean that the data presented is not correct or is biased because of human error. On the other hand, because not many papers presented these associations, it is very difficult to be sure that there must exist a real connection.

A positive aspect in comparison to other papers is the number of records in the data set. Before pre-processing, when the data is raw, there are 1613 patients without null values while in [37] there are 155 unique patients and in [16] are 589, whilst in [35] there are 611583 cases studied. This indicates that in this research there is a significant number of participants, but the results would not be fairly compared to other papers, where there is such a high difference as the one mentioned.

The innovation factors

In the plethora of papers, a simple bachelor's thesis must have something very special so that it is well received and highly viewed. But what makes something innovative? In the journey of answering this question we begin to see that uniqueness is not only characterized by the solution, but also how we got there. The mysteries that were not discovered today become tomorrow's questions.

The unique approach of considering every precondition of the patients and taking into consideration how they affect the outcome of the illness is a factor that was not specified in other papers that covered similar issues. The formula used in calculated for each comorbidity was not seen in any other published work and this means that its validity cannot be confirmed or denied, but for sure it is something that adds something unique to this paper.

Chapter 4: Application development

Scope

In this chapter we will cover the application and how it was developed. We will begin by establishing the role and goal that was reinforced by earlier chapters. The scope of this application is to support the medical staff, to better manage patients in the Covid-19 unit of any hospital and to forecast the future outcome of a new patients. The latter would serve greatly to better administer supplies such as oxygen tanks, ventilators and medications, and commodities like rooms and beds. This system aims to organize and help the medical staff and prevent any supply shortages if they may appear. By any means these predictions should be considered diagnostics, but only a guidance for anyone responsible for the Covid-19 patients. This system is designed to be used by medical professionals throughout the stay of any patient.

General details and requirements

The intended users of this application should be able to visualize all available and past patients in the Covid-19 section of the hospital, in addition it must be possible to select any patient and examine their digital record. This digital record contains general data about the patient such as age, gender, and number of days spent in hospital, together with complex data, like past illnesses and conditions and the medication received.

The most important feature of this application is the prediction of new data based on past observations. This system is required to guide and aid a medical staff in making important decisions about the future of a patient and how to manage supplies. The prediction must be at a medical standard to be relevant, and it should provide insightful information that comes as assistance to everybody that uses it.

Technologies used

Because of the libraries and capabilities offered, python was the first choice for building the model. Besides the model, the server needed to handle crud and statistical operations; these was also implemented using python, in the same application. In the python server, data is receiver from the client using the Flask framework [40]. The main model used for prediction is built from the TensorFlow library [41, 42]. This was used for building, and evaluating the model, and testing other classification models for comparison. The data was contained in a csv file, so there was no need for a database. For a simple and modern user interface, the front end was developed using the Angular framework [43, 44]. Complex elements, such as tables, fields navigation bars and many others were taken from the Angular Material library [45]. These components were further customized using scss files.

The frontend part of this application was written in Visual Studio Code from Microsoft, while the server side was implemented in PyCharm offered by JetBrains. This software offered intelligent senses and a variety of plugins that would facilitate the making of the application.

Backend: Python, Flask, TensorFlow

Frontend: Angular

Version control system: GitHub

Development process

Backend server

This server was written using Python 3.6 on PyCharm, a platform provided by JetBrains. The project was divided into 5 layers: data analysis, data processing, prediction model, service, main/controller. These layers were written and connected using python classes. We will go through every module and analyze its usages and the logic behind the code. For every stage the code was tested by running the individual scrips from each class using:

```
if __name__ == '__main__':  
    d = da.DataAnalysis("csv_dataset.csv")  
    pr = DataProcessing(d.getDataset())
```

Code segment 1: Running individual classes

In the first part, the data analysis, we used the *matplotlib* and *seaborn* libraries to display characteristics about the data. This enabled us to make a broader idea about the dataset and how we should tackle the next steps in the journey. Some of the plots were included in the chapter that talked about the dataset.

```
fig = plt.figure(figsize=(9, 6))  
sns.histplot(data_decedat, x=data_decedat["Varsta"],  
             y=data_decedat["Zile_spitalizare"], cmap='OrRd',  
             kde=True, label='Deceased', bins=45)  
sns.histplot(data_vindecat, x=data_vindecat["Varsta"],  
             y=data_vindecat["Zile_spitalizare"], cmap='BuGn',  
             kde=True, label='Cured', bins=45, alpha=0.5)  
plt.ylabel("Hospitalization days")  
plt.xlabel("Age")  
patch1 = mpatches.Patch(color='green', label='Cured')  
patch2 = mpatches.Patch(color='orange', label='Deceased')  
plt.legend(handles=[patch1, patch2])  
plt.title("Hospitalization days per age based on release state")  
plt.show()
```

Code segment 2: Data analysis class

The data analyzation consists of transforming the values into numbers, so that the model could easily understand them. Another important step is storing the data, the processed dataset is stored in a separate file to increase the time and memory of the program, by not changing the values every time the application is run. Any other additional information that aids in preparing the data is stored in separate files to speed up the operations. This process starts with replacing the null values with the default Not a Number variable from Python. The simple text categories that can be categorized without any other pre-processing are also replaced with numbers starting from 0. After that, other more complex operations are performed, which were better explained and exemplified in the chapter referenced before.

```

self.df.replace("NULL", np.NAN, inplace=True)
self.df.replace("", np.NAN, inplace=True)
self.df.replace("_", np.NAN, inplace=True)

```

Code segment 3: Replacing empty or null fields

For some columns, the best approach was to use One Hot Encoding [39]; because of the irregular nature of the data, the logic was implemented, not used from any external library. The principle was fairly easy to replicate and compared to using existing functions, custom methods tend to be more precise and fit better around the given data. By doing this, we were able to manipulate the values however we needed to.

```

def changeMedicatie(self):
    """
    One Hot Encoding for the "Medicatie" column.
    :return: self.medicatie: dictionary()
    """
    dm = {}
    indx = 0
    self.medicatie = dict()
    for record in self.df.Medicatie:
        med_list = str(record).split("|| ")
        self.medicatie[indx] = med_list
        for med in med_list:
            med = med.replace(" ", "")
            try:
                self.df[med][indx] = 1
            except:
                self.df[med] = np.zeros(self.df.shape[0], dtype=int)
                self.df[med][indx] = 1
                pd.to_numeric(self.df[med])
            dm[med] = 1
        indx += 1
    for key, value in dm.items():
        if self.df[key].sum() <= self.df.shape[0] * 0.2:
            self.df.drop([key], inplace=True, axis='columns')
            self.df.drop(['Medicatie'], inplace=True, axis='columns')
    csv_file = "csv_medicatie.csv"
    try:
        with open(csv_file, 'w') as f:
            for key in self.medicatie.keys():
                f.write("%s,%s\n" % (key, self.medicatie[key]))
    except IOError:
        print("I/O error")
    return self.medicatie

```

Code segment 4: One Hot Encoding

All the computations for the comorbidities column were also custom written, without using any external function. The steps and the motivation were already established, now is the moment to dive deeper into the implementation. We will divide the algorithm into multiple steps followed by explanations of each part in detail and the implementation.

- Step 1. Count how many patients with a given comorbidity end up in each of our five release states → use this logic for every unique comorbidity

These computations were done in a separate function. The names of all comorbidities were previously saved in a text file: “text-comorbiditati.txt”.

```
def comorbidityCountsDataset(self):
    """
    Creates a dictionary with every illness and how many people had each type of
    severity.
    :returns: DataFrame
    """
    try:
        self.com_ext = self.df[["Comorbiditati", "stare_externare"]]
        comorbidityCountMatrix = {}
        comorbidityFile = open("text-comorbiditati.txt", "r")
        comorbidityNames = csv.reader(comorbidityFile)

        # create a dictionary with key: comorbidity code and value: matrix with
        6 values
        for row in comorbidityNames:
            # count, vindecata, ameliorata, stationara, agravata, decedata
            identifierUniqueCode = row[0].split(" ", 1)[0]
            comorbidityCountMatrix[identifierUniqueCode] = [0, 0, 0, 0, 0, 0]
            for comorbidityColumn, outcome in self.com_ext.itertuples(index=False):

                # if there exists a comorbidity list in the column of the dataset and
                the code we are looking for is that list
                if type(comorbidityColumn) is str and row[0] in comorbidityColumn:
                    comorbidityCountMatrix[identifierUniqueCode][0] =
                    comorbidityCountMatrix[identifierUniqueCode][0] + 1
                    comorbidityCountMatrix[identifierUniqueCode][int(outcome) + 1] =
                    comorbidityCountMatrix[identifierUniqueCode][int(outcome) + 1] + 1
                    dictr = {}
                    for key, value in comorbidityCountMatrix.items():
                        dis = key.split(" ", 1)
                        dictr[dis[0]] = value[5]
                    comorbidityFile.close()

            self.comorbidityCounts = pd.DataFrame(comorbidityCountMatrix)
            self.comorbidityCounts = self.comorbidityCounts.transpose()
            self.comorbidityCounts.rename(
                columns={0: 'Count', 1: 'Vindecata', 2: 'Ameliorata', 3: 'Stationara', 4:
                'Agravata', 5: 'Decedata'},
                inplace=True, errors="raise")
        except IOError:
            self.comorbidityCounts = pd.DataFrame()

    return self.comorbidityCounts
```

Code segment 5: Create dictionary of illnesses

Step 2. Create the weights that would balance the results, based on the number of people in each group and a custom weight that would place the values on a linear axis; the result will be multiplied by the numbers computed at Step 1 to give us the formula for each comorbidity


```

weight = {0: 0, 1: 0, 2: 0, 3: 0, 4: 0}

forma_weight = {0: -1, 1: -0.5, 2: 0.25, 3: 0.5, 4: 1}
total_count = self.df.stare_externare.value_counts().sum()
count_forma =
pd.DataFrame(self.df.stare_externare.value_counts()).to_dict()['stare_externare']
for i in range(0, 5):
    weight[i] = forma_weight[i] * (1 - count_forma[i] / total_count)

col_names = self.comorbidityCounts.index
forma = {0: 'Vindecat', 1: 'Ameliorat', 2: 'Stationar', 3: 'Agravat', 4: 'Decedat'}
comorbidityWeights = {}
for names in col_names:
    comorbidityWeights[names] = 0
    for i in range(0, 5):
        comorbidityWeights[names] += self.comorbidityCounts[forma[i]][names] * weight[i]

```

Code segment 6: Impose data bias for balancing

For each outcome in the dataset, we count how many times it appears in all patients we sum them. We try to overcome the unbalancing from the data by subtracting the result from 1. The reason for doing this is that the proportion for any comorbidity is a proper fraction. After that, each result is multiplied by a set value so that the cases that resulted in a positive outcome may lean more towards negative values and illnesses with high mortality rates would give a big positive value. These constant values are multiplied by how many people from each outcome group had the given comorbidity.

Step 3. Split the comorbidity column into multiple names of illnesses and sum up each weighted formula; this will give us a value for each patient

```

indx = 0
self.comorb = dict()
for row in self.df.Comorbiditati:
    if row is not np.NaN:
        comb_list = row.split(',')
        regspt = re.sub(r'([A-Z])', r'\2', row)
        regspt = re.sub(' ', '', regspt)
        regspt = re.split('@', regspt)
        comb_weight = 0
        self.comorb[indx] = regspt
        for comb in comb_list:
            comb = comb.split(" ", 1)[0]
            if comb in comorbidityweights:
                comb_weight += comorbidityweights[comb]
        self.df["Comorbiditati"][indx] = float(comb_weight)
    else:
        self.df["Comorbiditati"][indx] = 0
    indx += 1

```

Code segment 7: Splitting the comorbidity column

In the comorbidity column, every patient will have the sum of weights of all illnesses that he has, that are present in this column. Firstly, the initial value must be split by the delimiters present in the list, which is “,”. If the comorbidity is not in the list or the initial value of the column is null, the result will be defaulted to 0.

The computations done to transform this column into a numeric one might be the most original part of this piece. In most cited papers, the used datasets were composed by only numerical values, most work and research being done on the used predictive model. Most of the time spent in writing this thesis was allocated for describing and processing the data.

The most interesting part of the application is the machine learning model. Because of the complexity and the advanced steps taken in making the model, it is best to be discussed in its separate chapter.

The next module consists of the Service layer. Here lies the implementation of the methods requested by the client level. The data received is sent to the right class to be parsed and processed.

The most upper layer was built using the Flask library [40]. It handles requests from the web client, sends and receives data. The functions are only responsible to get data from the request and send it further to the service. This framework was chosen because it was easy to use and it was very similar to other languages, such as handling HTTP requests using Java.

User interface

When building a modern website, we stumble upon a simple question: React vs Angular? Why was Angular our answer? This framework is mainly used for building complex single-page applications [43]. It also offered a wide variate of component libraries, from Angular Material [45], Clarity design system, NG Bootstrap, Ignite and many others. For a beginner, Angular Material seemed simple enough, and had an abundance of features and components.

Let's establish the flow of the application: a user enters the application and sees the home page with the list of all the patients from the Covid-19 hospital area. In the navigation bar there are 3 buttons (Home, Asses a new patient and a dropdown with the name Statistics). The home button brings the user back to the initial page, "Asses a new patient" redirects the user to a form in which he can input data related to a new patient and receive a prediction about its release state. The last one, the "Statistics" dropdown brings multiple types of statistics about the whole dataset or just a subset. Further one, we will present in more detail the pages.

The list of patients includes some details about the patients, such as the unique identification code, age, gender, and the release state. The last column of the table has a button that redirects us to more details about the specific patient.

<div> Home Asses a new patient Statistics </div>					
List of admitted patients Covid-19 department				<div> <input type="text"/> <input type="button" value="Search"/> </div>	
Patient Id	Age	Gender	Outcome	Raport	
19904	72	Female	improved	<input type="button" value="view"/>	
20157	62	Male	improved	<input type="button" value="view"/>	
20159	86	Female	improved	<input type="button" value="view"/>	
20160	86	Male	cured	<input type="button" value="view"/>	
20161	69	Male	improved	<input type="button" value="view"/>	
20163	52	Male	improved	<input type="button" value="view"/>	
20164	75	Male	improved	<input type="button" value="view"/>	
20165	80	Female	improved	<input type="button" value="view"/>	
20170	56	Male	improved	<input type="button" value="view"/>	
20171	87	Female	improved	<input type="button" value="view"/>	
20172	70	Male	deceased	<input type="button" value="view"/>	
20175	67	Female	deceased	<input type="button" value="view"/>	
20176	76	Female	cured	<input type="button" value="view"/>	
20177	61	Female	improved	<input type="button" value="view"/>	

Figure 22: Home page with patient list

The list of patients is also paginated using a configurable angular plugin. This pagination helps in better scrolling through the records. In the upper right corner, there is a search input that filters all patients with a given Id. It is configured to match the input characters not only from the beginning, but on any position in the string.

The details page contains the rest of the information about the patient, days spent in hospitalization and ICU, comorbidity list, tests run during their stay, and medication received. This information was parsed and displayed from the initial dataset. The role of this page is to keep a digital record for every patient, to better establish patterns and connections between different patients.

Home
Asses a new patient
Statistics

Patient Id: 20182
This is the unique identifier of the patient.

Age: 64
Gender: Male

Days spent in hospitalization: 31
Days spent in ICU: 7

Comorbidities list:

I05.0 Stenoză mitrală
I10 Hipertensiunea esențială (primară)
I25.9 Cardiopatie ischemică cronică nespecificată
I34.0 Insuficiență mitrală (valvă)
I35.1 Insuficiență (valvă) aortică
I48 Fibrilație atrială și flutter
I50.0 Insuficiență cardiacă congestivă
K44.9 Hernia diafragmatică fără obstrucție sau gangrenă
U75.0 Pacientează activă în cadrul medicamentelor

Figure 23: General details of a patient

Home	Assess a new patient	Statistics
------	----------------------	------------

K44.9 Hernia diafragmatica fara obstructie sau gangrena
K76.0 Degenerescenta grasoasa a ficatului neclasificata altundeva
M47.99 Spondiloza nespecificata localizare nespecificata
U07.1 COVID-19 cu virus identificat

Tests run during their stay:

UREA - 26 mg/dl (13 - 43)
GLUCEMIE - 90 mg/dl (70 - 110)
ASAT/GOT - 30 U/l (0 - 31)
ALAT/GPT - 19 U/L (0 - 34)
Creatinina - 0.7 mg/dl (0.51 - 0.95)
CK-MB * - 15 U/L (0 - 25)
LDH - 392 U/L (0 - 450)
CK - 22 U/L (0 - 145)
Proteina C reactiva - 43 mg/L (0 - 10)
Sodiu - 134 mmol/l (134 - 150)
Potasiu - 4.2 mmol/l (3.4 - 5.1)

*Values also normalized to creatinine

Figure 24: General details of a patient - routine tests

The next important panel in the application is about statistical information about the dataset. All the plots displayed on this panel were done by using the python library, matplotlib [46]. This part is divided into 4 pages: Distribution of age groups, Distribution of age based on release state, Clustering with PCA and K-Means, Clustering based on age and gender. Each page is loaded live by unique requests to the server. In most cases these requests are made when the page is opened. Let's talk about each one separately:

Distribution of age groups: Graph describing how the release state differs in different age groups. The grouping was done manually using Sturges' rule [47]. From this plot there can be deduced that elderly people may suffer more severe types of Covid because more people are treated in the hospital. There is a not a linear rising as the ages increases, but a normal skewed distribution, that may be caused by the overall higher number of people that live in these age groups. This graph also shows the minimum and maximum values.

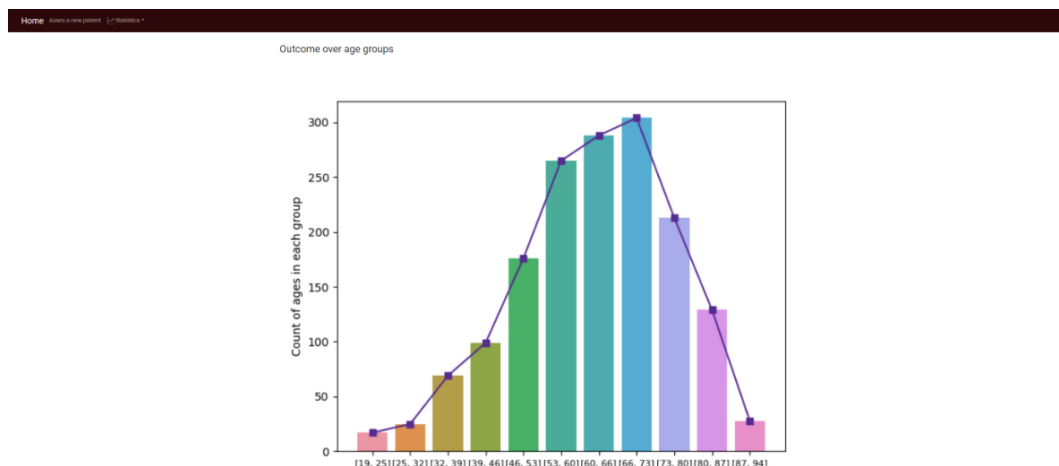


Figure 25: Distribution of age groups

Distribution of age based on release state: This page shows the relation between a patients age and the number of days spent in the ICU. This can be relevant to determine if there exists a relevant relations ship between the 2, meaning that there may also be a correlation between them. This figure also displays information about the outcome of the patient, lighter dots representing a deadlier outcome while darker dots are cured patients. There is a clear ‘line’ at 0 ICU days, maybe because most people did not stay in the ICU; also in the middle of the graph there is a clear bundle of data that rises with age. By looking closer to *Figure 25: Distribution of age groups* and *Figure 27: Distribution of age based on outcome*, we may see a that they have almost the same shape. Is it that happened by accident or a clear relationship? Hard answer, but where there are many data points (as in groups [60, 66] and [66, 73]), there are more chances for bigger variations in the data.

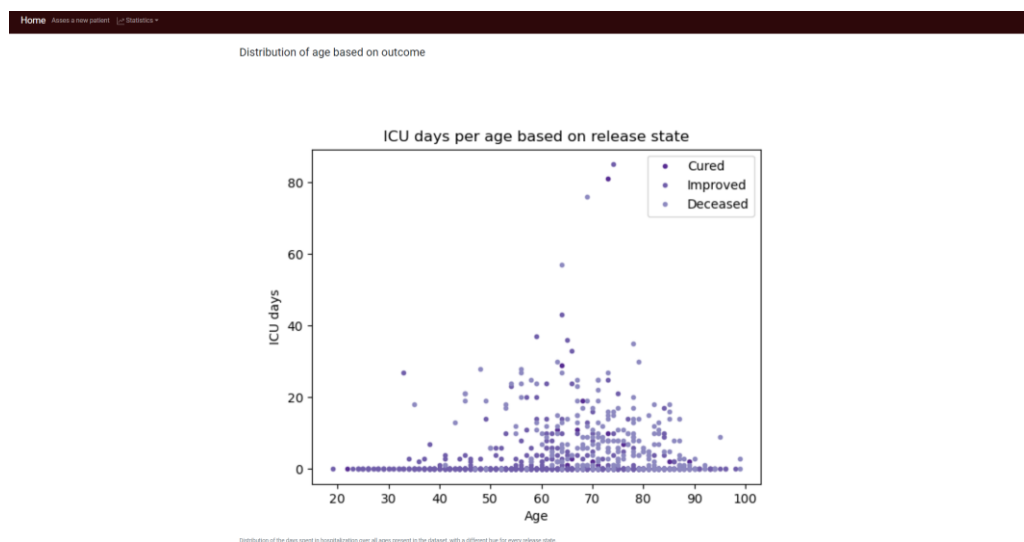


Figure 27: Distribution of age based on outcome

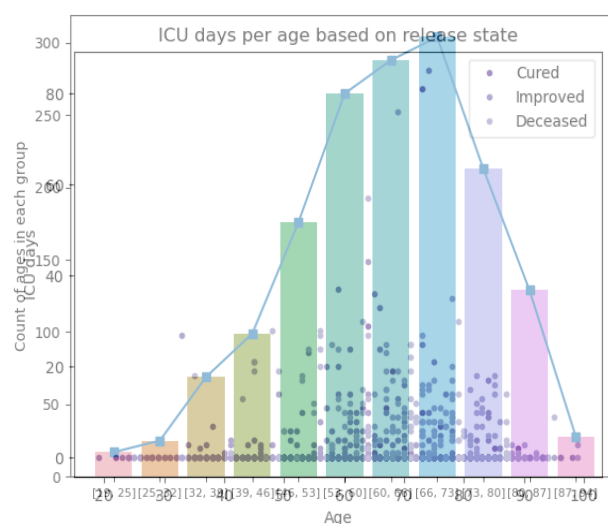


Figure 26: Relation between figures 25 and 27

Clustering with PCA and K-Means: this page consists of 2 graphs displaying the data by using 2 dimensional points. This information is useful to better show how many outliers are there in the dataset and how much variation is in the data.

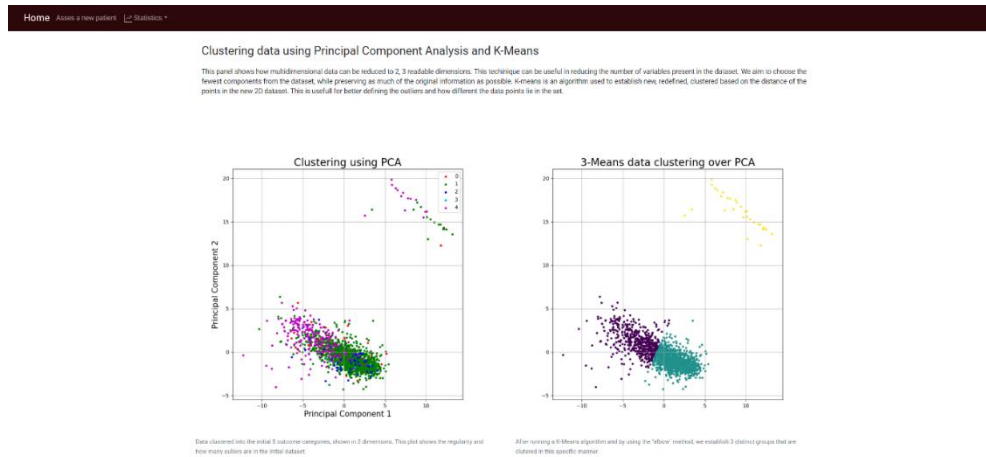


Figure 28: Clustering with PCA and K-Means

PCA stands from Principal Component Analysis [48] and is a technique used in data reduction and visualization. It is very useful in observing clusters, outliers and overall, how different types of points lie in the dataset. When working with multivariable tables, visualization becomes very hard, almost impossible; a tool like PCA could make this task a lot simpler. Data reduction could also be needed when a dataset has many lines and many columns, in our case [6120 rows x 108 columns]. *Figure 29: Representation of PCA* shows how PCA works.

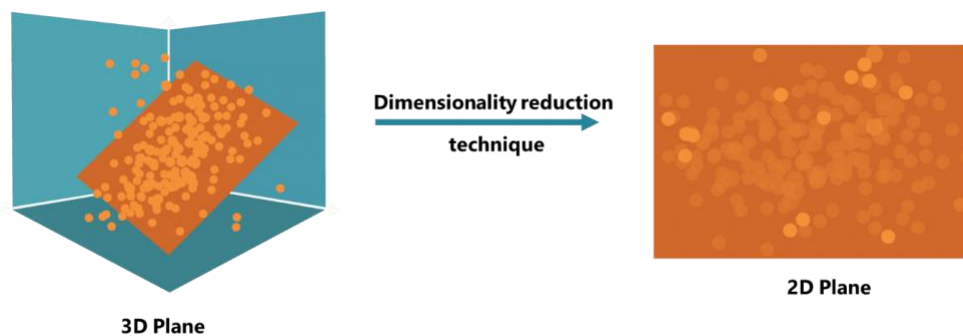


Figure 29: Representation of PCA [48]

Combined with this technique, a K-Means algorithm [49] was used on the reduced data to create new clusters. This algorithm determines how the dataset is divided into K clusters given K centroid points. So, it helps to check if the data is clustered in chunks corresponding to the outcome classes. The unsupervised learning algorithm classifies data by position, this information is used to observe how much the data varies between every release state. In combination with PCA, the data is reduced,

therefore information was highlighted, and the noise is lowered. This will improve the performance of the K-Means algorithm [50, 51]. In this case, the K was chosen using the elbow method [52], the graph generated provides that 3 is the right number of clusters. The elbow method refers to the way the number

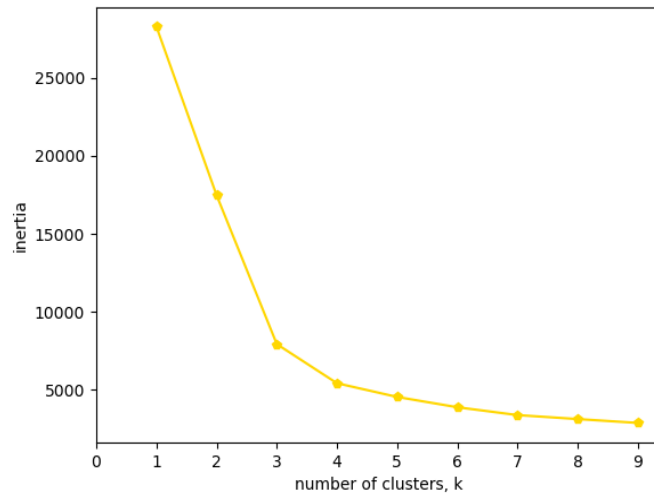


Figure 30: Elbow method from the K-Means algorithm

of optimal features is chosen. The point in which the graph starts to flatten, parallel to the Ox axis. To be more exact, the minimum number of features with the least amount of inertia. In the above plot we can observe that in the point “3” the algorithm starts to approach the minimum value, from which it does not improve significantly.

Clustering based on age and gender: this page slices the initial dataset based on a given age and gender. The user is presented with an input box and toggle button for the 2 genders. After the submit

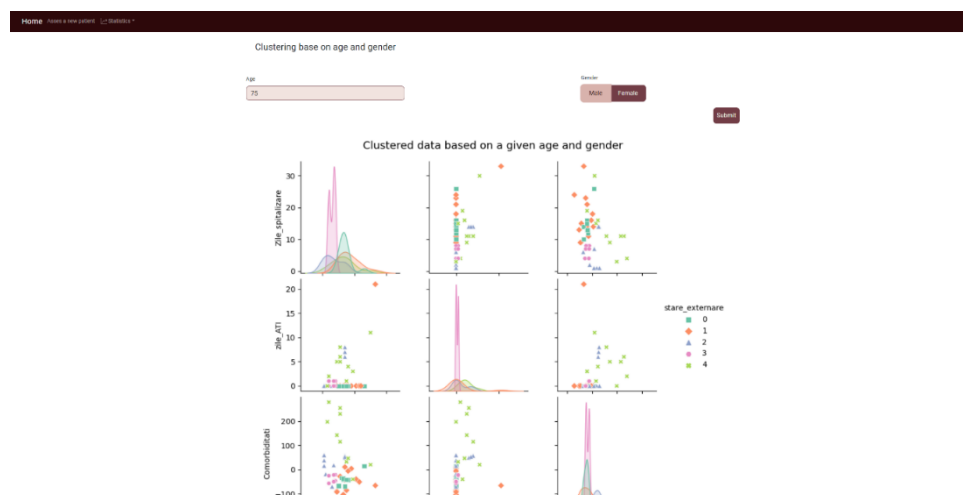


Figure 31: Clustering based on age and gender

button is pressed, a request is sent to the server, that computes a new subset with only patients that have the wanted values. The result is represented into a Pair plot that contains information about days spent in hospitalization, days spent in the ICU and the computed comorbidities value. It is all classified by the release state in 1-5 colors. This information is important in establishing relationships and correlations between these columns.

Home Asses a new patient [Statistics](#)

Patient Id

This is the unique identifier of the patient.

Age

Gender

☐ Male ☐ Female

Days spent in hospitalization

Days spent in ICU

Tests run during their stay

Tests and analysis

Example: URCA - 20 mg/dl (13 - 43) || QUCCMC - 121 mg/dl (70 - 110) || ASAT/DO7 - 79 U/L (0 - 35) || ALAT/DO7 - 61 U/L (0 - 40) || Creatinine - 1.07 mg/dl (0.67 - 1.17) || LDH - 750 U/L (0 - 450) || Prothrombin C-reactive - 61 mg/L (0 - 10) || Time de prothrombine - 17.8 sec (13 - 16) || AP (%) - 85.0 % (80 - 100) || INR - 1.12 (0.8 - 1.2) || APTT (sec) - 29.8 sec (20 - 30) || APTT-ratio - 0.94 (0.7 - 1.3) || FIB (mg/dl) - 671.7 mg/dl (180 - 450) || D-dimér - 86.1 ng/ml (0 - 500) || Troponine T - HS - -440 ng/L (0 - 14) || Lactate - 4.88 *10³µL (4 - 11) || BASP - 0.01 *10³µL (0.00 - 0.13) || BAST - 0.2 % (0 - 1) || NH₄ - 4.79 *10³µL (1.8 - 7.4) || NH₄ - 8.7 % (0 - 75) || FCSH - 0.00 *10³µL (0.00 - 0.67) || FCSH - 0.0 % (0 - 5) || YW - 0.37 *10³µL (1.5 - 3.5) || YW - 7.5 % (0 - 40) || MCR - 0.17 *10³µL (0.2 - 1) || MCR - 4.4 % (0 - 10) || Hemat - 4.67 *10³µL (4.5 - 6.0) || Hemoglobine - 17.9 g/dl (13.5 - 17) || Sodium-électrolyte-méde - 19.8 % (0 - 100) || Hemoglobine-électrolyte-méde - 26.5 pg/ml (27 - 37) || Conc.-méde-de-hémoglobine-électrolyte-méde - 33.2 g/dl (32 - 36) || HEM-CV - 14.2 % (0 - 15) || HEM-CV - 41.6 % (30.2 - 49.2) || Hematocrit - 38.9 % (40 - 54) || Tromboctes - 203 *10³µL (150 - 400) || MPV - 8.1 fL (8.1 - 12) || PDW - 16.3 fL (11 - 18) || PCT - 0.165 fL (0.15 - 0.5)

Medication they received during their stay

Medication received

Example: algocalmin sol inj 1 g/2 ml || actron 10 500 || calbion 1 g || chlorure de sodiu 0.9% pt 500 ml || chlorure de sodiu 0.6% 250 ml || codéine fosforice ae 15 mg || dexaméthazone ompharm 8 mg/ml || digoxin 0.25 g || enterolactis plus 3 g || faspargine 5700 ie anti factor xii/0.6 ml || furosemid 40mg cpr || furosemid sol inj 20 mg/2 ml || hydroxychloroquine sulphate tablets, 200 mg || lopineur and ritonavir tablets 200 mg/50mg || parolparol sun 40 mg || paracetamol b braun 10 mg/ml || plaquavir 200mg || quamatol 20mg/5ml f || sinitron 4mg cpr || spironolactone biocel 25 mg || vitamine b1 sol inj 100 mg/2 ml || vitamine b6 sol inj 50 mg/2 ml

Initial diagnosis

Choose one comorbidity for the initial diagnosis

List of any relevant conditions and illnesses

Select all comorbidities that apply

Submit

Figure 32: Asses a new patient

The last and the most important page is “Asses a new patient”, which is a form where a medical professional can input information about a new patient. The purpose of this page is to better determine the current state of a patient, given its condition and stay time. This is done by an intelligent algorithm that can help the doctors in making decisions. This page is bound to the server by an API request that sends the new, unseen data to it and then makes all the preprocessing needed. After that, the prediction is made by the chosen algorithm. The result is then sent back to the user and then nicely displayed in a modal.

After pressing submit, a panel that displays the result appears, presenting the predicted label, and the confidence level for each of the five labels. These results help in optimizing the diagnostic and treatment of a patient at a certain time of its stay in the hospital.

The screenshot displays a medical prediction interface. At the top, there's a navigation bar with 'Home', 'Assess a new patient', and 'Statistics'. The main form is divided into several sections:

- Days spent in hospitalization:** A text input field containing the value '14'.
- Days spent in ICU:** A text input field containing the value '7'.
- Tests run during their stay:** A large text area containing a list of laboratory tests and their results, such as 'UREA - 20 mg/dl (13 - 43)', 'GLUCOSE - 121 mg/dl (70 - 110)', 'ASAT/GOT - 79 U/l (0 - 35)', etc.
- Medication they received during their stay:** A text area containing a list of medications, such as 'aligocalmis sol inj 1 g/2 ml', 'azithro (r) 500', 'cefot 1 g', etc.
- Initial diagnosis:** A dropdown menu with the selected option 'U07.1 COVID-19 cu virus identificat'.
- List of any relevant conditions and illnesses:** A section with a 'Select all comorbidities that apply' checkbox and several radio button options like 'I35.1 Insuficienta (valva) aortica', 'R74.8 Nivelul anormal al altor enzime serice', etc.

A 'Prediction Result' modal is open in the center, displaying the following information:

- Prediction Result**
- The patient has a high chance to be release as: Deceased
- Cured -> 19.0%
- Improved -> 14.0%
- Stationary -> 65.0%
- Worsened -> 6.0%
- Deceased -> 100.0%

At the bottom of the form, there is a 'Submit' button.

Figure 33: Prediction result

A new prediction can be made just by closing the modal and changing the values.

This system stands out due to all the available statistics that are calculated with the real time patients and the capability of forecasting some assumptions about the release state of a patient, all in the same place. Applications or models that tried to predict the wellbeing of Covid-19 patients, were already developed, but most of them use CT images along with tabular information, and they do not display so much general information about the population.

This study uses many algorithms and statistical techniques that try to make as much sense of the data and the results as they can.

Model training and testing

The first approach was to evaluate built-in algorithms, which were specified in the literature about AI in medicine, especially solving Covid-19 problems. The models that were tested and compared were: Logistic regression, Random Forest classifier, K-nearest Neighbors classifier, Support Vector Machine, Decision tree classifier, Gaussian process classifier, MLP classifier. Before this step, the data was over sampled to combat the unbiased data problem.

To talk a little more about imbalanced data, we should establish what is it, why is it such a big problem and how can we solve it. This notion is referring to skewed information, events or outcomes that do not have the same or similar probability of happening. As shown before, in *Figure 11: Distribution of release state based on gender*, the outcomes in the dataset do not the same likelihood, to be precise, “Cured” has a 0.06 chance, “Improved” has 0.75, “Stationary” only 0.03 probability, “Worsened” 0.004 chances only 7 patients being in this category out of 1617 and lastly “Deceased” with 0.144. Along this project, there were many ways in which this data bias was treated, such as imposing custom bias variables that would bring these probabilities to the same number.

A well-known technique is SMOTE, Synthetic Minority Over-sampling Technique [53, 54]. This method constructs new synthetic data to fool the algorithm that there exist more records for each outcome. The algorithm works in a similar manner to the K-Means method, meaning that, it builds clusters around points and creates a set amount of new data around the centroids. This ensures that the new samples are closely related to existing values, but the downside to this method is the low exploratory rate of this algorithm, meaning that the labels may be constricted to the said values, comparing to real world cases where the data may be more spread out. Along SMOTE, a more particularized method would be ADASYN, adaptive synthetic oversampling method, where the oversampling takes into consideration the distribution of the initial dataset. After testing and comparing these approaches, the base algorithm performed better on the same configuration of the model:

Over sampling method	Precision	Recall	F1	Accuracy
ADASYN	0.828	0.740	0.782	0.799
SMOTE	0.824	0.724	0.771	0.797

Table 2: Comparison between oversampling methods, SMOTE and ADASYN

The above table reveals that ADASYN is a better fit for this dataset.

After multiple tests, there could be observed that the new dataset is still almost equally grouped, even after using the method that took the distribution into account. Following that, the best decision was to manually readjust the size of each label, by deciding the number of inputs from each label. The largest group, improved patients, should remain the same and should still represent the biggest class of the dataset. Starting from a perfectly balanced dataset, with 20% of each label, the optimum value for the improved category could be 30%. From some simple calculations, based on the last percentage we arrive to the following new distribution:

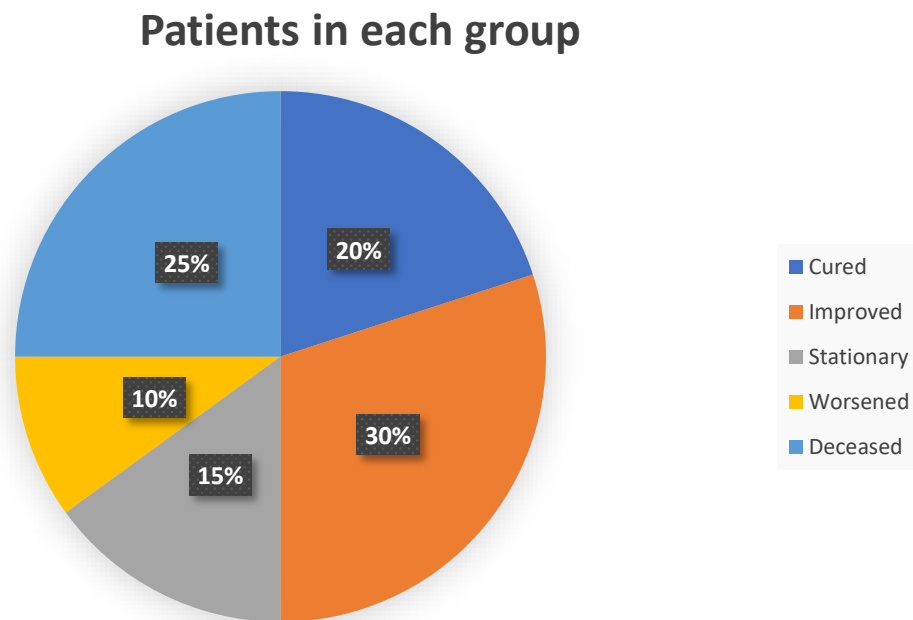


Figure 34: Distribution of the new dataset, after oversampling

To be more exact: Cured now has 816 inputs, Improved has the same 1224 real values, Stationary contains 612 values, Worsened, with the least amount of data, has 408 and Deceased has 1020 records. This method ensured that the new dataset does not interfere with reality, and how often could some types be encountered.

To come back to the original idea, the starting base for the algorithm are the models that were presented above. The processed data returns the following results when the simple models were tested:

Model	Precision	Recall	Accuracy
Logistic regression	0.82	0.84	0.83
Random Forest	0.98	0.98	0.98
K nearest Neighbors	0.83	0.86	0.81

Support vector Machine	0.83	0.84	0.83
Decision tree	0.89	0.89	0.88
Gaussian Naive Bayes	0.54	0.59	0.47
MLP	0.98	0.98	0.98

Table 3: Results of simple classification models

After evaluating these models, the next step is building a neural network, which would make a more custom model that could solve this problem with a higher accuracy. A popular library that is used for creating any layer type model is TensorFlow [55, 56]. This is an open-source library that defines machine learning models and testing functions [57]. The usage of this library will allow us to create and build a complex network following some small easy steps.

As explained before, the concept of neural networks is just a complex way of implementing a classification based on error reduction through different functions. The Tensor sequential [56] model is made by adding multiple hidden layers, specifying the number of nodes and the activation functions. These, plus the number of layers, are the hyperparameters that must be adjusted so that the model fits the problem in the best way.

```
model.add(tf.keras.layers.Dense(units=n[i],
                                activation=activation_fct[i],
                                kernel_initializer
                                    =tf.keras.initializers.GlorotNormal(),
                                kernel_regularizer=tf.keras.regularizers.l2(0.01),
                                bias_initializer=tf.keras.initializers.Zeros(),
                                bias_regularizer=tf.keras.regularizers.l2(0.01),
                                name=name))
model.add(tf.keras.layers.GaussianNoise(0.3))
```

Code segment 8: How to add a new hidden layer

The model contains 3 hidden layers, and it was trained and tested using built in functions. Because of the much talked about bias and overfitting [58], after each dense layer is added a normalization layer, that has a Dropout purpose. Dropout [59] means stopping a given percent of the weights to be adjusted so that the outcome does not differ as much in every iteration. The chosen dropout function was Gaussian Noise, as it gave much better results than other functions, and because it was designed specifically to prevent overfitting the model.

After creating the model, it needs to compute some metrics at every step so that we can compare how different hyperparameters affect learning. We can characterize them in 4 categories: error metrics (regression metrics), accuracy metrics, probabilistic metrics and classification metrics based on

True/False positive & negative. The loss function helps us plot a regression line for each epoch that can help us interpret how the model improves at every step. A “healthy” loss function graph should slowly and steadily descend until it reaches a point where the line flattens and follows a trajectory parallel to the Ox axis. The straight path at the end indicates that the model learned as much as it could, from that moment on it will probably maintain its state without getting better or worse. Accuracy metrics are used to calculate how often predictions correspond with the target label. Probabilistic metrics compute different probabilistic measures on the prediction output. And lastly, classification metrics are computed using true/false positive and negative values, such as precision, recall, specificity, and sensitivity. These values are very useful in determining if the model overfits or underfits the results. To add to this, these metrics are important in the medical field because they reveal how suitable this algorithm is.

In the implementation of the model only 5 metrics were taken into consideration: Mean squared error, Recall, Precision, Categorical Accuracy and Categorical Cross entropy. The first one helps us in deciding whether a model has a bigger error margin than another one, in other words, how wrong are the predictions given by the algorithm. Accuracy is one of the most known measures for any predictive algorithms, because of this it is also very easy to understand. Because the dataset is unbalanced, for a better understanding of the model, we need to compare precision and recall [60].

The algorithm needs to calculate its metrics in relation to the outcome. Because we are trying to predict a multi-valued label, we need to one hot encode every prediction result and the product of the machine is an array containing the confidence level for every label. This concludes that the metrics should also be categorical, for this reason we will consider the categorical accuracy and the categorical cross entropy.

At every step of learning, when the algorithm performs backpropagation [61], a dynamic learning rate optimizer is used to better adjust the model to the problem dataset [62]. This function is used to modify the learning rate after every epoch so that the algorithm learns at a steady rate. This is done by an observer that punishes the algorithm if it learns too fast or not enough. In this model stochastic gradient descent was chosen because it gave the best results.

After multiple runs, the algorithms shown the best performance with a batch of 100 records. This model was then tested using all available keras activation functions (relu, elu, selu, tanh, sigmoid, softmax, softplus, softsign, exponential). Because the model has 3 dense hidden layers, each with its own activation layer, there are 729 possibilities, from which just some result in considerable values for f1-scores and accuracy. Subsequently, after 1000 epochs, the first 10 models, based on f1-score, recall and accuracy, in this order, stood out with the following specifications:

Model	Loss	Mean square error	Accuracy	Precision	Recall	F1-score
exponential, selu, softplus	0.5965	0.0347	0.8722	0.8819	0.8722	0.8770
exponential, elu, elu	0.6605	0.0348	0.8795	0.8795	0.8795	0.8795
exponential, relu, relu	0.6256	0.0367	0.8795	0.8884	0.8722	0.8802
exponential, tanh, elu	0.5856	0.0343	0.8832	0.8921	0.8759	0.8839
exponential, softplus, tanh	0.6270	0.0352	0.8862	0.8999	0.8868	0.8933
exponential, selu, elu	0.5861	0.0304	0.9014	0.9040	0.8941	0.8990
exponential, elu, selu	0.5635	0.0276	0.9124	0.9117	0.9051	0.9084
exponential, selu, tanh	0.5299	0.0251	0.9197	0.9225	0.9124	0.9174
exponential, selu, selu	0.5281	0.0273	0.9197	0.9194	0.9160	0.9177
exponential, tanh, selu	0.5612	0.0271	0.9197	0.9230	0.9197	0.9213

Table 4: Results of ANN model

It is clear from the table above that the best results were attained when the first hidden layer has the activation function is ‘exponential’. Other important remark is that the functions Rectified Linear Units (ReLU), Exponential Linear Units (ELU) and Scaled Exponential Linear Units (SELU), appear the most frequent. It should also be noted that the second to last model reaches the smallest loss value, so these results must be checked multiple times for the most accurate results from each model. Respectively, the lowest mean square error is reached in the third to last model.

In medical context, the recall value, this being the sensitivity of the model, was the most important metric, because it may be better to over “predict” than to miss some cases. This does not mean that other metrics, such as accuracy and precision are irrelevant, but when it comes to a choosing between similar models, the recall score may the tie breaker. The formulas for precision, recall and f1-score are as follows:

Precision	Recall	F1-score	where: tp - true positive tn - true negative fp - false positive fn - false negative
$\frac{tp}{tp + fp}$	$\frac{tp}{tp + fn}$	$\frac{tp}{tp + \frac{1}{2}(fp + fn)}$	

Table 5: Precision, Recall, F1-score formulas

To establish the consistency of the results of the models, at least the best 5 should be ran multiple times, and the final score is calculated by computing the mean from each run. These models were chosen using p-value tests [63, 64] and by comparing calculated measures, highest f1-score and accuracy, lowest loss and mean square error.

No.	Model	Loss	Mean square error	Accuracy	F1-score
1	exponential, selu, elu	0.5614	0.0351	0.8817	0.8841
2	exponential, elu, selu	0.6218	0.0376	0.8751	0.8775
3	exponential, selu, tanh	0.6163	0.0390	0.8751	0.8728
4	exponential, selu, selu	0.5137	0.0266	0.9124	0.9026
5	exponential, tanh, selu	0.5892	0.0351	0.8868	0.8859

Table 6: Average performance for the best 5 models

To ensure that the model does not overfit the given dataset, we need to also take in consideration the loss function, calculated during the training of the model. If the graphic of the function does not converge to a tangent to the Ox plane, then the model did not finish the training process. The learning curve should also be observed and judged.

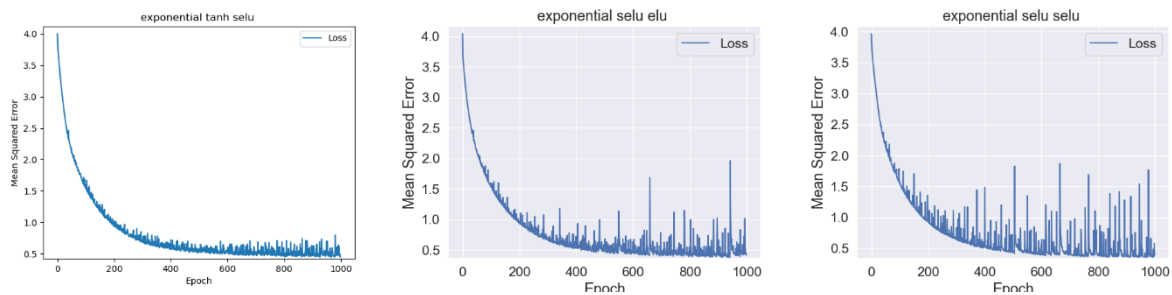


Figure 35: Loss function plots of the top 3 models

For a better visualization of the above metrics, we can look at the confusion matrix and compare how different models predicts 20% of the values of the dataset.

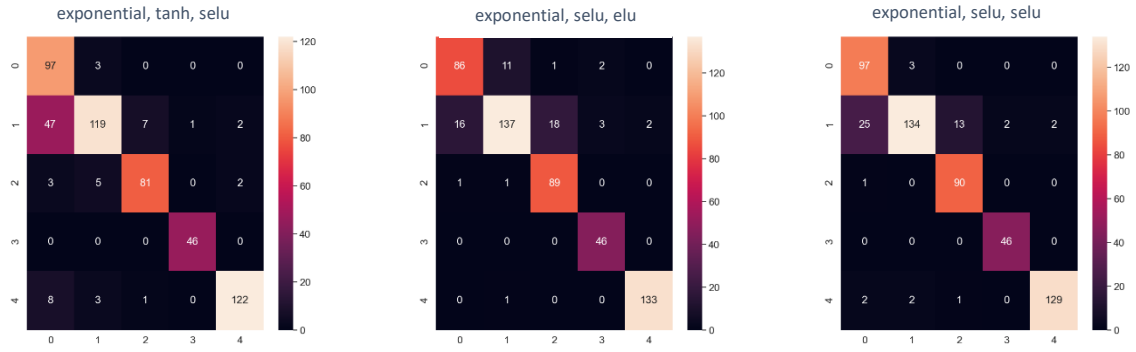


Figure 36: Confusion matrix of the top 3 models

From the confusion matrices we can deduce that all models perform generally well, however, model number 4, shown in the last picture from Figure 36: *Confusion matrix of the top 3 models*, has an overall lower number on the incorrect classes. But these results may be misleading, which brings us to the last step, analyzing the confusion matrix computed over the whole available dataset. This leads us to a new result, where **model 1** has the most amount of values on the first diagonal.

After examining the results and the plots, we can confidently say that **model 1** had the best performance. The chosen model has the activation functions of the hidden layers exponential, selu and elu, with 100, 70 and 20 nodes for each mentioned layer. This model has an accuracy of 88% and a f1-score of 88%. After evaluating the model against 20% of the data, the results were plotted against the expected outcome to see if this algorithm is overfitting the predictions. If the predicted values were spread out through out every label, this means that the model did not fit or learned an outcome more than the rest.

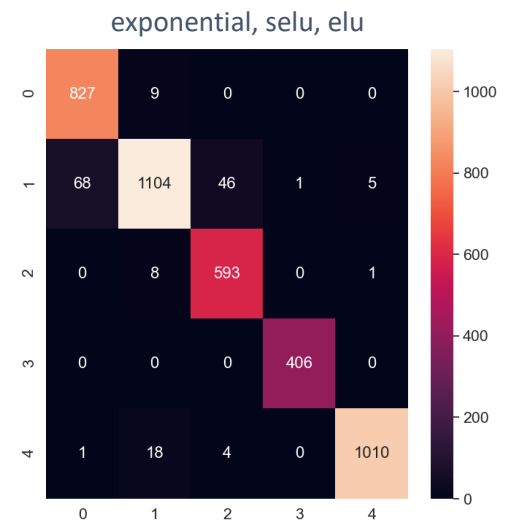


Figure 37: Confusion matrix over the whole dataset of the best model

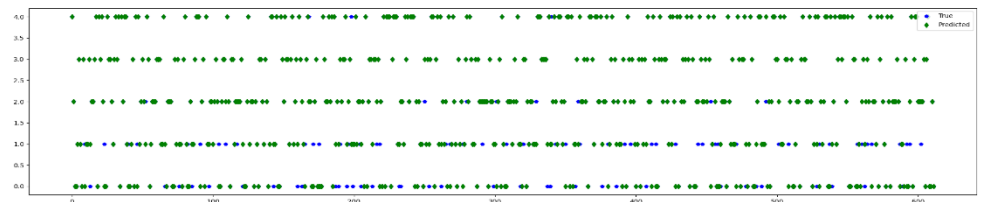


Figure 38: Evaluating the model

Chapter 5: Conclusion and future improvements

Comparing results and findings

Because of the medical context of the cited papers, regarding similar topics, the simple models, KNN, Linear Regression, Random Forests, Decision tree or others, cannot be adjusted to the given dataset in the same manner as a neural network can be manipulated. These tests are relevant for comparing the results with scientific literature and establish the type of the solution that works best for this problem, in a short amount of time. The fine tuning of the hyperparameters gives the model more “knowledge” and makes the tests more effective. There are many more ways and methods to change the ANN, comparing to the other mentioned models.

There were little to no studies that compiled tabular data about Covid patients, let alone, datasets that present as many details as the given one. Along these papers, there are also studies done on CT scans of the lungs or chests of Covid-19 patients. Even though the comparison to tabular data is not fair, the results should be noted and discussed.

In [29] were presented some models implemented based on CT scans of Covid-19 patients; the results from these 2 individual studies are compared bellow:

Model	Recall [29]	Recall	Accuracy [29]	Accuracy
Logistic regression	0.96	0.84	0.97	0.83
Random Forest	0.90	0.98	0.87	0.98
Support vector Machine	0.97	0.84	0.98	0.83

Table 7: Comparison between simple models results

It should be noted that the resulted dataset has a comparable performance with one used in the study cited above. Even though the type of inputs used was not the same, a parallel should be drawn between the results. Other studies, [31], [32], [33] and [34], also based on CT scans, had similar performances on accuracy levels, 0.86, 0.89, 0.95 and 0.98 respectively.

Most studies done on tabular data present more information about the patients and what characteristics most stood out in each individual group. Little to no studies that were done to this date talked about machine learning models used in data exploration. One study that also used a simple neural network over a dataset of over 4 million records [65] presented results with an accuracy of 93.15%. A study [25] analyzed a dataset containing only 3 features using a machine learning algorithm, and it produces almost 90% accuracy. The solution presented has an accuracy of 88% and a f1-score of 88%.

Further improvements

In a later development stage, an algorithm that better chooses the best model should be put in place, with the help of genetic algorithms. Compared to choosing the most appropriate model just by using statistical means and observations, this method brings a more exploratory approach, that tries to avoid such greedy techniques. This algorithm also brings “chaos” and knows how to take advantage of the best features of a model to exploit them. A big drawback of this process is very time consuming, to make an estimation: we ran the chosen model for 1000 epochs, thus, for a population of 50 models, the genetic algorithm should be run for at least 100 epochs, this comes to 5,000,000 iterations just for calculating the fitness of initial population, disregarding any offspring to simplify the calculations.

Other than that, because of the unbalanced data, it would be a great idea to try to relabel the dataset using an unsupervised clustering algorithm, then confronting them with the original one and with a specialist that could determine the validity of the new labels. This approach could potentially greatly increase the performance of the algorithm. An issue with this technique is to alter the data, removing the naturality, so basically destroying the purity of the dataset. The aim of this study is not to label or to find clusters among the given patients, because they were already categorized, presumably, the correct way.

The last procedure that could be tried is optimizing the feature selection. This would also boost the performance of the algorithm, just by creating other features or filtering them. Through many trials it has been discovered that not using all the columns could potentially return a better model. An algorithm that would be worth taking a look at is the Decision Tree model, that provides information about the importance of each feature. In the end, the goal is to use as many optimization techniques as we know so that we can produce the best algorithm.

Conclusion

At the start of this study, we tackled information about machine learning and AI models used to solve problems in medicine. Due to the fact that this disease is extremely new, there were little to no published studies, that were backed by similar data. Most papers approached a dataset composed by lung CT scans and tried to correlate the images to the severity of the disease or just recognize the existence of Covid-19 from the scans.

After analyzing the dataset, some features needed to be pre-processed, following a One Hot Encoding procedure for 2 columns, and creating a new numeric representation, by using a mathematical formula, based on frequency and the corresponding label. In the final analysis, the dataset revealed an unbalanced distribution. By using various methods, such as ADASYN, the labels in the final dataset have a closer proportion. This was a crucial step in making an accurate prediction. After that, the values are scaled so that every value is in the interval $[0, 1]$. This is done so that the values in every feature are not so spread around.

The model was created through multiple tests, where the final algorithm was composed of 3 hidden dense layers, with 3 dropout layers in between. The determined configuration was created with combating overfitting and having a proportional precision and recall in mind. The concluding model has a performance of over 88% accuracy and 88% f1-score.

Table of Figures

Figure 1: Covid-19 statistics from 12.02.2022.....	4
Figure 2: Human neuron depiction reflecting how an artificial neural was constructed [65]	11
Figure 3: Diagram depicting the construction of a neural network	13
Figure 4: Table depicting high mortality rate correlated to higher age groups [35]	18
Figure 5: [16].....	19
Figure 6: [18].....	20
Figure 7: [18].....	20
Figure 8: [18].....	21
Figure 9: Distribution of gender in the dataset	22
Figure 10: Distribution of age groups in the dataset.....	23
Figure 11: Distribution of release state based on gender.....	24
Figure 12: Distribution of days based on the number of ICU days and release state.....	24
Figure 13: Distribution of ages for every release state category and every gender	25
Figure 14: Correlation between days spent in hospitalization and ICU for each release state and type (1-mild, 2-moderate, 3-severe).....	25
Figure 15: Example of raw data from the dataset.....	26
Figure 16: Medication column.....	26
Figure 17: One hot encoded medication column.....	27
Figure 18: Investigations and results.....	27
Figure 19: One hot encoded investigation column	27
Figure 20: Computed counts for every comorbidity, based on how many patients end up in each release state.....	28
Figure 21: Final weight of every comorbidity.....	30
Figure 22: Home page with patient list	41
Figure 23: General details of a patient	41
Figure 24: General details of a patient - routine tests.....	42
Figure 25: Distribution of age groups	42
Figure 27: Distribution of age based on outcome.....	43
Figure 26: Relation between figures 25 and 27	43
Figure 28: Clustering with PCA and K-Means	44
Figure 29: Representation of PCA [48].....	44
Figure 30: Elbow method from the K-Means algorithm	45
Figure 31: Clustering based on age and gender.....	45
Figure 32: Asses a new patient	46

Figure 33: Prediction result.....	47
Figure 34: Distribution of the new dataset, after oversampling.....	49
Figure 35: Loss function plots of the top 3 models	53
Figure 36: Confusion matrix of the top 3 models.....	54
Figure 37: Confusion matrix over the whole dataset of the best model.....	54
Figure 38: Evaluating the model	54
Table 1: Comparison between multiple models in [29].....	17
Table 2: Comparison between oversampling methods, SMOTE and ADASYN.....	48
Table 3: Results of simple classification models.....	50
Table 4: Results of ANN model.....	52
Table 5: Precision, Recall, F1-score formulas.....	52
Table 6: Average performance for the best 5 models	53
Table 7: Comparison between simple models results.....	55
Code segment 1: Running individual classes	36
Code segment 2: Data analysis class.....	36
Code segment 3: Replacing empty or null fields.....	37
Code segment 4: One Hot Encoding	37
Code segment 5: Create dictionary of illnesses.....	38
Code segment 6: Impose data bias for balancing	39
Code segment 7: Splitting the comorbidity column	39
Code segment 8: How to add a new hidden layer	50

Bibliography

- [1] J. & K. J. & K. T. Paschen, "Artificial intelligence (AI) and its implications for market knowledge in B2B marketing," *Journal of Business & Industrial Marketing*, 2019.
- [2] S. Dehaene, *How We Learn: The New Science of Education and the Brain*, Penguin Books Ltd, 2021.
- [3] B. Peter, "The Emergence of Artificial Intelligence: Learning to Learn," *AI Magazine*, 1985.
- [4] E. S. Brunette, R. C. Flemmer and C. L. Flemmer, "A Review of Artificial Intelligence," *School of Engineering and Advanced Technology*, 2009.
- [5] M. & P. D. Mirolli, "Language as a cognitive tool," *Minds and Machines*, 2009.
- [6] Bittencourt, Guilherm, Marchi and Jerusa, "An embodied logical model for cognition," *Artificial Cognition Systems*, 2006.
- [7] D. Friedlander and S. Franklin, "LIDA and a theory of mind," *Proceeding of Artificial General Intelligence*, 2008.
- [8] "Is the Brain a Useful Model for Artificial Intelligence?," [Online]. Available: <https://www.wired.com/story/brain-model-artificial-intelligence/>.
- [9] D. Marr, "Artificial Intelligence - A Personal View," *Massachusetts Institute of Technology*, 1977.
- [10] "Artificial Intelligence and Machine Learning: Policy Paper," 2017. [Online]. Available: <https://www.internetsociety.org/resources/doc/2017/artificial-intelligence-and-machine-learning-policy-paper/>.
- [11] "The Importance of Data in Machine Learning," 2020. [Online]. Available: <https://serengetitech.com/tech/the-importance-of-data-in-machine-learning/>.
- [12] J. A. Lee Rainie, "About this canvassing of experts," [Online]. Available: <https://www.pewresearch.org/internet/2017/02/08/algorithms-about-this-canvassing-of-experts/>.
- [13] K. Browne, "Artificial neural networks," 2019.
- [14] W. & P. McCulloch, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, 1943.
- [15] V. C. B. C. G. S. R. R. D. H. G. S. Wynants L, "Prediction models for diagnosis and prognosis of covid-19: systematic review and critical popraisal," 2020.
- [16] K. Z. M. Y. D. L. Y. L. K. B. Xu, "Application of ordinal logistic regression analysis to identify the determinants of illness severity of COVID-19 in China," *Epidemiology and Infection*, 2020.

- [17] F.-d.-L.-P. C, P.-C. D, G.-M. V, C. ML and F. LL, "Defining Post-COVID Symptoms (Post-Acute COVID, Long COVID, Persistent Post-COVID): An Integrative Classification," *Int J Environ Res Public Health*, 2021.
- [18] . K. Jordan R E, "Covid-19: risk factors for severe disease and death," 2020.
- [19] R. P. V. Y. G. D. A. D. Moons KG, "Prognosis and prognostic research: what, why, and how?".
- [20] "Freenome," [Online]. Available: <https://www.freenome.com>.
- [21] N. W. D. L. T. Wan, "Machine learning enables detection of early-stage colorectal cancer by whole-genome sequencing of plasma cell-free DNA," *BMC Cancer*, 2019.
- [22] L. YH, B. H and K. DJ, "How to Establish Clinical Prediction Models," *Endocrinol Metab*, vol. 31, no. 1, 2016.
- [23] Y. S. S. B. S. Malik, "How artificial intelligence may help the Covid-19 pandemic: Pitfalls and lessons for the future," *Rev Med*, 2021.
- [24] H. S, Y. J, F. S and Z. Q, "Artificial intelligence in the diagnosis of COVID-19: challenges and perspectives," 2021.
- [25] M. Islam, T. Poly, B. Alsinglawi, F. Lin, C. S, J. Liu and J. W., "Application of artificial intelligence in covid-19 pandemic: Bibliometric analysis," *Healthcare (Switzerland)*, vol. 9, no. 4, 2021.
- [26] Y. H. A. & K. Mohamadou, "A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19," *Appl Intell*, 2020.
- [27] S. D, K. V, Vaishali and K. M, "Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks," *Eur J Clin Microbiol Infect*, 2020.
- [28] Le, DN., Parvathy, V.S. and Gupta, "IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification," *Int. J. Mach. Learn. & Cyber*, 2021.
- [29] F. Y, "Sensitivity of chest CT for COVID-19: comparison to RT-PCR," *Radiology*, 2020.
- [30] J. S. Adam, J. M. Eric, K. Mayers, R. Fernandes, A. L. Rowe, P. Viccellio, H. C. Thode, A. Bracey and M. C. Henry, "Cohort of Four Thousand Four Hundred Four Persons Under Investigation for COVID-19 in a New York Hospital and Predictors of ICU Care and Ventilation," *Annals of Emergency Medicine*, vol. 76, no. 4, 2020.
- [31] J. X. C. P. L. L. S. L. C. Y. S. J. L. L. Y. Z. H. X. K. R. ., Xu X, "Deep learning system to screen coronavirus disease 2019 pneumonia," 2020.

- [32] K. B. M. J. Z. X. X. M. G. J. M. Y. J. L. Y. M. X. X. B. Wang S, "A deep learning algorithm using CT images to screen for corona virus disease (COVID-19)," *medRxiv preprint*, 2020.
- [33] B. S. Sethy PK, "Detection of coronavirus disease (COVID-19) based on deep features," 2020.
- [34] K. C. P. Z. Narin A, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep," 2020.
- [35] C. Bonanad, S. García-Blas, F. Tarazona-Santabalbina, S. J, B.-G. V, F. L, A. A, J. Núñez and A. Cordero, "The Effect of Age on Mortality in Patients With COVID-19: A Meta-Analysis With 611,583 Subjects," *Med Dir Assoc.*, 2020.
- [36] M. L. R. L. Soares RCM, "Risk Factors for Hospitalization and Mortality due to COVID-19 in Espírito Santo State, Brazil," 2020.
- [37] T. S. A. E. M. N. V. I. S. A. T. I. M.A., "Biomarkers of clinical and radiological severity of a new coronavirus infection caused by SARS-CoV-2 virus, and their association with a severe variant of its course," *Bulletin of Siberian Medicine*, 2021.
- [38] K. & P. T. & P. C. Potdar, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. International Journal of Computer Applications," 2017.
- [39] X. Ying, "An Overview of Overfitting and its Solutions," 2020.
- [40] M. Grinbel, Flask Web Development, United States of America: O'Reilley Media, 2018.
- [41] P. Goldsborough, "A Tour of TensorFlow," *Proseminar Data Mining*, 2016.
- [42] M. Abadi, "TensorFlow: learning functions at scale," *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, 2016.
- [43] B. M. Hamerník, Development of Modern User Interfaces in Angular Framework, Masaryk University Faculty of Informatics, 2020.
- [44] С. статей, ТЕОРИЯ И ПРАКТИКА МОДЕРНИЗАЦИИ НАУЧНОЙ ДЕЯТЕЛЬНОСТИ В УСЛОВИЯХ ЦИФРОВИЗАЦИИ, Omega Science, 2020.
- [45] K. V.K., Introduction to Angular Material, Berkeley, CA: Material Design implementation with AngularJS, 2016.
- [46] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, pp. pp. 90-95, May/June 2007.
- [47] A. L. Philip, L. R. Paul, M. David and E. M. James, "Improving Accuracy and Efficiency of Registration by Mutual Information using Sturges' Histogram Rule".*School of Computer Science, Cardiff University; School of Optometry, Cardiff University.*

- [48] E. Kaloyanova, "What Is Principal Components Analysis?," 12 December 2019. [Online]. Available: <https://365datascience.com/tutorials/python-tutorials/principal-components-analysis/>. [Accessed 5 March 2022].
- [49] E. Kaloyanova, "What Is K-means Clustering?," 23 January 2020. [Online]. Available: <https://365datascience.com/tutorials/python-tutorials/k-means-clustering/>. [Accessed 5 March 2022].
- [50] D. Chris and H. Xiaofeng, "K-means Clustering via Principal Component Analysis".
- [51] E. Kaloyanova, "How to Combine PCA and K-means Clustering in Python?," 10 March 2020. [Online]. Available: <https://365datascience.com/tutorials/python-tutorials/pca-k-means/>. [Accessed 5 March 2022].
- [52] U. Edy, E. S. Jadmiko and G. S. K. Vincensius, "K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median," *Departement of Information System, Post Graduated School Diponegoro University*, 2019.
- [53] V. C. Nitesh, W. B. Kevin, O. H. Lawrence and P. K. W., "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, 2002.
- [54] F. Alberto, G. Salvador, H. Francisco and V. C. Nitesh, "SMOTE for Learning from Imbalanced Data: Progress and Challenges," *Journal of Artificial Intelligence Research*, 20 Apr 2018.
- [55] P. Bo, N. Erik and N. Ying, "Deep Learning With TensorFlow: A Review," *Journal of Educational and Behavioral Statistics*, 10 September 2019.
- [56] M. L. w. TensorFlow, Nishant Shukla, vol. Chapter 10, Manning Publications, 2018.
- [57] P. Goldsborough, "A Tour of TensorFlow," 2016.
- [58] Y. Xue, "An Overview of Overfitting and its Solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, 2019.
- [59] B. Pierre and J. S. Peter, "Understanding Dropout," *Advances in Neural Information Processing*, 2013.
- [60] E. B. L. R. a. D. V. Jianping Zhang, "Learning rules from highly unbalanced data sets," *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 571-574, 2004.
- [61] R. HECHT-NIELSEN, "III.3 - Theory of the Backpropagation Neural Network**Based on "nonindent" by Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on Neural Networks 1, 593–611, June 1989. © 1989 IEEE.,," *Academic Press*, pp. 65-93, 1992.
- [62] G.-A. C. a. S.-X. C. Xiao-Hu Yu, "Dynamic learning rate optimization of the backpropagation algorithm," *Transactions on Neural Networks*, pp. 669-677, 1995.

- [63] J. Brownlee, "How to Use Statistical Significance Tests to Interpret Machine Learning Results," 3 May 2017. [Online]. Available: <https://machinelearningmastery.com/use-statistical-significance-tests-interpret-machine-learning-results/>. [Accessed 6 April 2022].
- [64] D. Curtis, "Comparison of artificial neural network analysis with other multimarker methods for detecting genetic association," *BMC Genet*, 2007.
- [65] T.-G. A. H.-U. I. L.-M. R. U. A. Quiroz-Juárez MA, "Identification of high-risk COVID-19 patients using machine learning," 2021.
- [66] "Artificial neuron," [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neuron.
- [67] A. R. A. H. I. M. Naseem M, "Exploring the Potential of Artificial Intelligence and Machine Learning to Combat COVID-19 and Existing Opportunities," *LMIC: A Scoping Review. J Prim Care Community Health*, 2020.
- [68] Kolozsvári, L. Róbert, Bérczes, Tamás, A. Hajdu, R. Gesztelyi, A. Tiba, I. Varga, A. B. Al-Tammemi, G. J. Szöllősi, Garbóczy, S. H. Szabolcs and J. Zsuga, "Predicting the Epidemic Curve of the Coronavirus (SARS-CoV-2) Disease (COVID-19) Using Artificial Intelligence," 2020.
- [69] A. Marchisio, A. M. Hanif, M. Shafique and M. Martina, "A Methodology for Automatic Selection of Activation Functions to Design Hybrid Deep Neural Networks," 2018.