TU WIEN Informatics

Empirical comparison of two ordinal regression algorithms in the context of predicting the length of hospital stay for COVID-19 patients

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software and Information Engineering

by

Rastko Gajanin
Registration Number 11930500

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig
Assistance: René Baranyi
Richard Habenicht

Vienna, 27th June, 2022

Signature Author

Signature Advisor

Empirical comparison of two ordinal regression algorithms in the context of predicting the length of hospital stay for COVID-19 patients

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software and Information Engineering

by

Rastko Gajanin
Registration Number 11930500

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Advisor: Thomas Grechenig

Wien, 27th June, 2022

Erklärung zur Verfassung der Arbeit

Rastko Gajanin

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 27. Juni 2022

Rastko Gajanin

Acknowledgements

Firstly, I wish to express my sincerest gratitude to my advisors Dr. René Baranyi and Mr. Richard Habenicht, who guided me throughout this thesis with exceptionally valuable and prompt feedback and enabled me to finish the thesis in such a short period.

I would also like to thank my sister Teodora and my parents Vesna and Radoslav for enabling me to pursue my further education at the Technical University of Vienna, and for encouraging, consulting, and supporting me over the entirety of my studies so far.

Special thanks go to my girlfriend Ana for her patience, kindness, and words of encouragement that motivated me not only in the course of writing this thesis but also during all previous semesters.

Abstract

Due to the COVID-19 pandemic, hospitals on the global scale faced major issues and multiple severe capacity shortages, especially in the first year of the pandemic. One of the ways to disburden the hospital management staff and improve the quality of bed assignments in the intensive care unit (ICU) can be seen in the employment of machine learning (ML) models. These models can provide the ICU administrative staff with the information needed to improve the quality of bed allocations. Various disease and patient-related metrics can be predicted with ML models, including the length of hospital stay (LoS) of each patient. Predicting this metric is the first central point of this thesis. Along with that, the thesis aims to provide an empirical comparison of the two different ML model types that are tailored for solving ordinal regression problems. The first is a representative of the Random Forest (RF) models and the second of the Neural Network (NN) paradigm.

In order to achieve the stated goals, the central topics of the thesis will be introduced along with screening, analysis, and the presentation of state-of-the-art literature on these subjects. Based on the screened literature, eight models are then developed (four RF and four NN models). Half of them implement the naive methodology and the other half the optimized approaches to ordinal regression. Another differentiation criterion is whether four or two classes are being predicted. The eight models are then evaluated and compared, not only to one another but also to the models developed in state-of-the-art literature and their value for the medical staff is assessed.

It has been established that the naive binary classification RF model achieved the best performance across multiple evaluation metrics (with respect to the given dataset). The model is capable of correctly identifying 89% of the patients who stay in the hospital for longer than 50 days, which could be beneficial for the ICU administration personnel. The results further show that models which are specifically tailored for ordinal regression achieve comparable and, in some cases, a worse performance than their naive counterparts on the provided dataset. It can, further, be observed that the RF models generally produce better results than the NN models if there are four classes in the response. Regarding the binary response, both RF and NN models achieve comparable performance across all considered metrics.

Keywords: *COVID-19, ordinal regression, ordinal classification, machine learning, random forests, neural networks, ranking, length of stay*

Contents

CHAPTER 1

Introduction

This chapter is meant to provide the reader with an introduction to the central issues this thesis aims to address along with the underlying motivation, goals, and the methodology used to achieve them. As the final part of the introduction, the contents of the primary chapters of the thesis are outlined.

1.1 Problem Description

It has been more than two years since COVID-19 was declared a global pandemic. In these two years, the hospitals around the world had significant issues with capacity management, especially in the intensive care units.

To address this problem, researchers are already making efforts to reliably predict the length of hospital stay (LoS) for each hospitalized COVID patient, with the goal of enabling medical staff to manage and utilize their resources more efficiently. For the purpose of predicting, one can use different statistical tools and machine learning models. The focus of this thesis is on evaluating different machine learning-based approaches to solving this problem. Both statistical tools and, especially, machine learning models need *representative data* in order to fit the model correctly and attain reasonable results, which are applicable in the medical practice. Researchers and practitioners have already gathered an abundance of data and made it publicly available. One of the datasets collected by the researchers will be used in this thesis.

After the examination of the main dataset used in this thesis, it has been established that the response variable (LoS) is a discretized ordinal categorical variable, containing the duration of hospital stay (LoS) for each patient as an interval (e.g. 'between 20-30 days'). This would, therefore, represent a special type of a regression/classification problem, because it is not a pure regression nor a pure classification problem. The discussion and an empirical comparison of two Machine Learning approaches to solving this kind

of regression/classification problem will be the main topic of this thesis and the two main categories, which will be compared are *Ordinal regression with Random forests* and *Ordinal regression with Neural Networks*.

1.2 Motivation

Due to the growing necessity, induced by COVID-19 disease, for a solution that would assist the Intensive Care Unit (ICU) management in their planning and resource allocation, the Machine Learning prediction models are becoming increasingly popular choices to support the ICU management.

Numerous prediction models have been trained, which aim to predict various COVID-19 metrics, such as daily infections, daily ICU admissions, death rate, and length of hospital (ICU) stay. By predicting the length of ICU stays for each patient accurately, a significant increase in the efficiency of patient treatment in the ICU can be expected. This would be primarily caused by more optimal utilization of hospital beds, whose "idle"/"inactive" time would be minimized. Having that said, the majority of the models use relatively small datasets (<10k observations) but have the LoS variable in form of an integer, which allows the use of straightforward regression techniques. In this work, a larger dataset will be used ($\approx$ 318k observations) with the LoS variable being an ordinal categorical variable. The dimensions of the dataset seem very promising and will therefore be further examined in this thesis. An additional shortcoming of the majority of the prediction models already developed is that they contain a high risk of bias, caused by unclear reporting, unrepresentative selection of patients, etc.

The goal of the thesis is twofold: along with contributing to the COVID-19 research and developing a model for ICU management, an empirical comparison between the two popular ML approaches to ordinal regression will also be carried out. Both Random Forests and Neural Networks have been very well documented on their own even in the context of Ordinal Regression. However, to the author's knowledge, the current literature does not contain a direct comparison of the two paradigms in the context of Ordinal Regression in any way. These methods have typically been tested on the public toy datasets or simulated data. Conversely, this thesis will apply both methodologies to a real-world dataset, report the performance, and assess the results of each in a comparative manner.

1.3 Goals

The first aim of this thesis is to gain insight into the state-of-the-art by conducting a thorough literature review. Secondly, topic-related data which will be used in this thesis is a publicly available dataset [1] containing the "Length of hospital stay" (LoS) of COVID patients [2]. Moreover, the performance of two state-of-the-art algorithms for ordinal regression will be evaluated, when applied to the provided COVID-19 dataset, whose response variable is a discretized continuous variable. The goal of this thesis is to

obtain an objective comparison of the discussed approaches and gain an insight into their effectiveness when predicting the LoS of COVID-19 patients. Besides creating models which could be useful for the medical staff, the extent of improvement these algorithms exhibit over the naive solutions to the problem of ordinal regression is also of great interest.

In particular, the following two algorithms will be compared:

- **CORAL**, which is described in [3] and represents a novel modification of a classical neural network approach to the problem of ordinal regression
- **Ordinal forests**, which is a representative of the random forest approach to ordinal regression and was introduced by [4]

The following metrics will be considered during the evaluation:

- Accuracy
- Precision of the '>50 days' class
- Sensitivity of the '>50 days' class
- Mean Multi class AUC [5]
- unweighted Kappa (Cohen's Kappa) [6]
- linear weighted Kappa (suggested by [4])

Each sub-result will also contain a "side-by-side" visualization of each evaluation measure, where the graphical representation of the evaluation measures will be offered in order to make it more intuitive for the reader.

After implementing the practical part of the thesis (comparison of the two algorithms), domain experts (nurses/technicians) with the role of external stakeholders will be contacted for feedback. This feedback will contain the evaluation of used predictor variables as well as suggestions for variables that could be included in future models to increase the performance metrics. Based on this feedback, meaningful proposals for future research could be inferred.

1.4 Methodological Approach

In order to obtain the expected results, the following methodology will be used:

1. The relevant sources on the broader topic of predicting various COVID-19 metrics will be found, such as the prediction of the number of infections, mortality rate, number of hospitalizations, the length of hospital stay, or identification of the patients with predispositions for a more severe infection

2. The dataset is then selected, which satisfies certain criteria such as number of observations, reliability, and trustworthiness
3. Depending on the type of the prediction problem (regression/classification/time series prediction) the state-of-the-art literature will be consulted and two Machine learning approaches will be chosen for evaluation on the given dataset
4. Data will be pre-processed
5. The two approaches are then applied and evaluated on the pre-processed data according to the given metrics
6. General comparison of the models is carried out
7. Visualizations are then created for each of the considered metrics
8. Generated models are discussed and compared to other state-of-the-art solutions
9. Importance of the predictors for each model is extracted and discussed
10. Model which performs best across the majority of metrics is then chosen
11. Suggestions and requirements for future research will, finally, be formulated

1.5 Outline

Having presented the problem and the goals behind the thesis along with the motivation and the methodological approach, the relevant theoretical background is provided in the Chapter 2. This does not only include the basic information about the COVID-19 disease and Long COVID but also on the fundamental Machine Learning processes, popular algorithms, and their employment in solving the most common ML problem types. After establishing a foundation for further chapters, in Chapter 3, the current state of the research in the field of predicting the COVID-19 metrics is discussed, along with the advanced approaches to ordinal regression. Chapter 4 contains a detailed description of the procedure by which the results were obtained together with a comprehensive summary of the findings. Then, in Chapter 5 the obtained results are discussed and compared to the state-of-the-art solutions. The thesis is concluded with a summary of its main findings and recommendations for future research.

CHAPTER 2

Theoretical background

Through this chapter, the theoretical foundation for the rest of the thesis is provided. Since the author's main concern is predicting one of the metrics related to the COVID-19 patients, the reader will, firstly, be provided with the general facts about the COVID-19 disease. Furthermore, the consequences the virus can have on certain patients will be discussed, in particular the cases where the symptoms persist beyond their typical duration. The latter is also referred to as the "Long COVID". Having introduced the medical part of the theoretical foundations, the standard approach to developing a machine learning model is going to be described. Afterward, the primary classification of the machine learning problems/tasks will be discussed. This would bring the reader closer to the peculiarity of the main problem of this thesis, the *Ordinal Regression*, which will, consequently, be described. In order to obtain a better understanding of the approaches that will be discussed in the upcoming chapters, the main principles behind Random Forests and Neural Networks are to be outlined as well. To conclude this chapter some of the naive approaches to Ordinal Regression will be named and described.

2.1 The COVID-19 Disease

In the last two years nations worldwide have been confronted with one of the greatest pandemics of the twenty-first century. COVID-19 (Short for Coronavirus disease 2019, named on 11. February 2020 by WHO [7]) is a clinical disease caused by the **S**evere **A**cute **R**espiratory **S**yndrome **Co**rona**V**irus **2** (**SARS-CoV-2**) pathogen [8], which is usually characterized by pneumonia-like symptoms such as fever, dry cough and fatigue [9]. The first cases of infection have been reported in the city of Wuhan in Central China in late December 2019. [10]. Therefore, the virus has been partly named after the year of discovery, 2019. In a matter of months the virus has spread to almost all parts of the world, with the exception of only a few island nations [11, 8], which led to the declaration of COVID-19 as a pandemic on 11. March 2020 [12].

In the following subsections, the virological background of the disease and its epidemiology will first be described. This includes the facts about the relationship of SARS-CoV-2 with other viruses, the natural reservoir of the disease, its origin, ways of transmission, and symptoms. The methods used to diagnose, treat and prevent the disease will then be examined before proceeding to the topic of Long COVID.

2.1.1 Virology and Epidemiology

In the following, the virological and epidemiological aspects of the disease will be discussed. Firstly, the SARS-CoV-2 virus and its origin will be defined. The characteristics of the virus will then be named and compared to other viruses of the same family. Afterward, its structure is presented and the identified transmission routes are briefly summarized. Lastly, the findings regarding clinical manifestations of the disease are brought forward.

Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)

„*Coronavirus* is an enveloped, positive single-strand RNA virus belonging to the *Coronaviridae* family the order *Nidovirales*.“ ([10, 13]) The name stems from the crown-shaped spikes on the surface of the virus envelope. Generally, the coronaviruses can be divided into four main subclasses: α, β, γ and δ. [9] The β subclass would be of interest for the research at hand since the SARS-CoV-2 belongs to this subcategory.[9] Other human pathogens such as SARS-CoV-1 and MERS-CoV, which were responsible for mild outbreaks in 2002 and 2012 respectively[8], also belong to this subgroup.[9] Through gene sequencing it has even been proven that there is only 73% amino acid similarity between the SARS-CoV-1 and SARS-CoV-2, while SARS-CoV-2 is genetically very similar to other coronaviruses which are found in bats (even up to 96%). [14, p. 5] For this reason, scientists believe that the virus is zoonotic [15] and suspect that the virus has spread from bats to humans [10]. The study that supports this claim was carried out by the Chinese Center for Disease Control and Prevention (CDC). In the course of the study the researchers found out that 31/33 positive Wuhan specimens came from the same Seafood Market (Huanan) [9], which could imply that several zoonotic events happened there, that caused the SARS-CoV-2 to be transmitted from the host of unknown species to a human [16]. Three recent studies also back up this claim: [17, 18, 19].

In Figure 2.1 it can be observed that the virion has a roughly spherical shape and possesses an envelope [21]. Coronaviruses have four structural proteins: Spike (S) protein, envelope (E) protein, membrane (M) protein, and nucleocapsid (N) protein [13], all of which are illustrated in the figure. All four and especially Spike protein play a significant role when it comes to the host range and the virus's ability to bind to the host's receptors [13, 22].

Since RNA viruses are typically characterized by frequent mutations, this enables them to further evolve and develop different virus *variants*. Different variants can have distinct epidemiological and virological properties. These usually include higher transmissibility or mortality. Two of the most prominent variants that have emerged by now are the

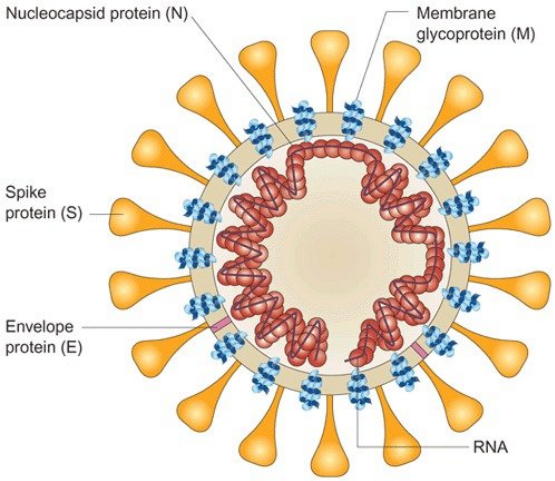

Figure 2.1: Schematic diagram of the SARS coronavirus virion structure [20]

Delta (B.1.617.2) and the *Omicron* (B.1.1.529) variant. As reported by [23] the Delta variant had a significant increase in infectivity namely by 1.5-2.5 accompanied by an increase in lethality by 1.5. Similarly, the Omicron variant exhibited very high values of transmissibility rates, with no clear evidence about its mortality [23].

Transmission routes

The speed at which the virus has already spread and is spreading among humans is explained by its high efficacy and infectivity. These two properties can be seen as consequences of the plurality of the transmission paths. It has been established that the virus is primarily transmitted through the respiratory route, most commonly via droplets. [24] Secondary routes of virus transmission found by researchers are the *aerosol* [24, 25] and *oral-fecal* route[1] [26]. The ability of the virus to survive on inanimate surfaces has been researched as well. In this sense, it has been discovered that the virions can be situated on surfaces like glass, metal, and plastic for up to 9 days [28]. This discovery played a significant role not only in expanding the common knowledge about the virus among the population but also in the shaping of public precautionary measures, for example, one such measure would be the increased frequency of disinfection of door handles and other commonly touched objects in public (e.g. restaurant tables, grab bars and grab rails in public transport...). Finally, the vertical transmission route has also been studied. There is a considerable amount of studies supporting the claim that the virus can be transmitted from the SARS-CoV-2 positive mother to the neonate during pregnancy, it is, however, very rare [22]. One interesting study concerning this matter

[1]The stool samples of the COVID-19 patients remained COVID positive on average 11 days after the patient tested negative (i.e. negative respiratory tract samples).[26] This led to speculations about further infections through sewers.[27]

is provided by [29], who observed a case where the neonate had an increased count of SARS-CoV-2 IgM and IL-6 antibodies after the mother's infection with the disease. The facts presented in this paragraph are partly summarised in Figure 2.2.

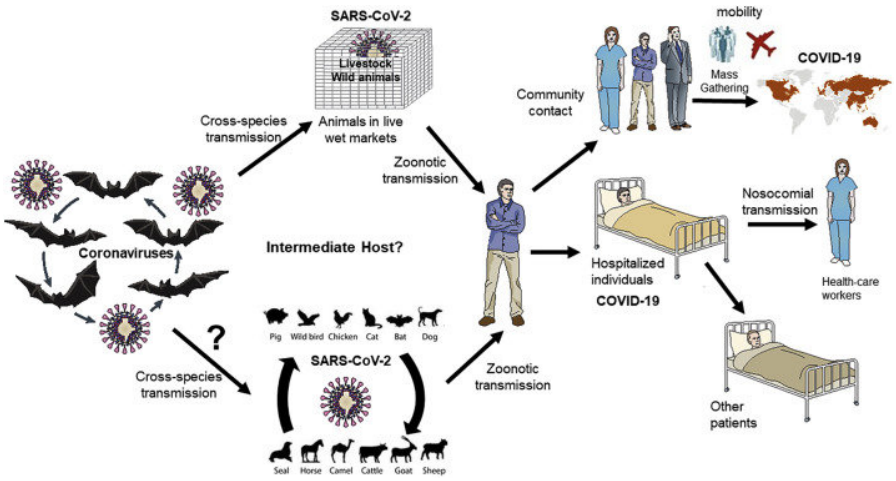

Figure 2.2: COVID-19 transmission paths [30]

Clinical manifestations

An infection with SARS-CoV-2 is usually followed by fever (in 98.6% of the reported cases), fatigue (69.6%) and dry cough (59-76%) [31]. Other, somewhat less common symptoms include anorexia (39.9%), myalgia (34.8%) and dyspnea (31.2%) [31, 32]. Further symptoms reported in less than 20% of the cases in the [31] study are nausea, headache, dizziness, diarrhea, and abdominal pain. It is to be concluded that not only the respiratory tract is affected by the disease, but also gastrointestinal and musculoskeletal systems as well [10]. Additionally, asymptomatic cases have also been reported [22]. If left untreated these symptoms tend to grow into mild to moderate types of pneumonia and hypoxia. Further development of untreated disease usually leads to acute respiratory disease syndrome (ARDS), systemic inflammatory response and multiorgan failure [33]. Another term used by researchers is *Cytokine storm*. This term refers to uncontrolled and rapid discharge of large amounts of cytokines into the bloodstream and represents another possible state the patients can reach [22, 34]. Significantly higher concentration of pro-inflammatory cytokines (and chemokines) in comparison to the anti-inflammatory ones can cause serious damage to multiple organs [22, 34]. Several research papers discovered (through CT scans) that the majority of the patients exhibited bilateral patchy shadows or ground-glass opacity in the lungs. [31, 22] Furthermore, histological analyses were used by [35] in order to establish that a higher viral load is present in the lower respiratory tract rather than in the upper. An observation made by [36] is that patients with chronic comorbidities such as diabetes, malignancies, and hypertension tend to develop more severe symptoms on average. As a consequence, these patient

cohorts have a greater predisposition to further complications in case of a SARS-CoV-2 infection.

2.1.2 Diagnosis

The most common methods to establish an infection are: via isolation of SARS-CoV-2 from the respiratory tract samples, sequencing of the viral genome, rapid antigen or polymerase chain reaction (**PCR**) tests of the viral nucleic acid from the upper respiratory tract samples [9, 27]. Examples of common upper respiratory tract samples are nasopharyngeal and oropharyngeal swabs, sputum, and saliva. [9]. It is important to note that the patient's epidemiological history is also to be taken into account. More specifically, information about traveling to areas with a high risk of infection, or information about close contacts who might also be positive (e.g. family or work colleagues) [9].

2.1.3 Treatments and Prevention

The treatment of the disease after the onset of symptoms is not explicitly determined and has challenged the medical staff globally since there is not that much information regarding anti-viral therapies and pharmaceuticals, which could directly target the disease [22]. Since all positive cases require urgent treatment, doctors usually resort to anti-viral, antibiotics, or oxygen therapy [22, 37].

As a preventive measure, multiple countries have engaged in research on the vaccine for SARS-CoV-2 . According to [38] the vaccine platforms for COVID-19 can be classified into classical and "next-generation platforms". The Classical platforms already have plenty of research behind them, and they were used for the development of vaccines for other diseases such as *Polio* or *Influenza*, while the same does not hold for the next-generation platforms [38]. The prominent representatives of classical vaccine platforms are [38]:

- Whole inactivated virus (Virus based)
- Live-attenuated virus (Virus based)
- Protein sub-unit (Protein based)
- Virus like particle (Virus based)

On the other hand, representatives of the novel vaccine platforms are [38]:

- Viral vector (Manufacturers: J&J, AstraZeneca, Gamaleya R.I.)
- DNA (Nucleic acid-based)
- (m)RNA (Nucleic acid-based) (Manufacturers: Pfizer-Biontech, Moderna TX)
- Antigen-presenting cells

The advantage the next-generation vaccine platforms offer over the classical ones is the lower cost of production and faster development pace. Therefore the majority of researchers who develop COVID-19 vaccines select one of the *novel* platforms [38, 39]. Additionally, in case a new virus variant appears, the mRNA vaccines can be adjusted very promptly in comparison to other platforms.

2.1.4 Remark

Unfortunately, covering all aspects of the COVID-19 disease is far beyond the scope of this section (and this thesis). Only the most relevant facts were selected regarding the topic with the awareness that there exist numerous important and valuable insights that had to be left out. Therefore, only a reference for further reading will be made: the comprehensive COVID-19 summary performed by Rana and Tripathi [22] is one of the most reliable and sound sources for further research on the given topic to the author's knowledge. Further references to relevant research in the field can be found in the given paper. Some of them have also been included in this thesis.

2.2 Long COVID

As mentioned beforehand, it was noticed that certain patients experience a significant delay in the development of their symptoms or their symptoms persist for prolonged periods. In the research community, this phenomenon is typically referred to as the "Long COVID" and will be the main topic of interest in this section. Long COVID will firstly be defined along with its sub-classifications. A few remarks will then be included regarding the current state of the research on the topic. Furthermore, the pathophysiological background of the disease is provided together with the prevalent symptoms of the illness. Finally, the treatments of the disease are named, and the subsequent problems related to the treatment methodologies are discussed.

2.2.1 Definition

As defined by National Institute for Health and Care Excellence (NICE) [40] *Long COVID*, *Long Haulers* or *Post Acute COVID Syndrome* denote the persistence or appearance of different COVID-19 symptoms in patients more than 4 weeks after being diagnosed with SARS-CoV-2 infection. [41]

Since this matter is a part of the ongoing research, it should be pointed out that the facts contained here mirror the state of the research in May 2022. Almost all research papers that were found on this topic contain a remark which emphasizes that the knowledge on the topic is incomplete, and the authors express the necessity for further research.

Multiple researchers have agreed that the subject is to be divided into two subcategories: *subacute COVID-19*, which includes the persistence or appearance of symptoms within 4-12 weeks after the acute COVID-19 phase, and *chronic or post-COVID-19 syndrome*

where symptoms stretch beyond the 12th week after the initial acute phase and cannot be explained by an alternative diagnosis [42].

Furthermore, [41] maintains that the disease could be classified by the predominant symptoms which are reported during the illness as follows:

1. Post-COVID cardio-respiratory syndrome
2. Post-COVID fatigue syndrome
3. Post-COVID neuro-psychiatric syndrome

The subdivisions of this type have proven to be very beneficial in the process of determining the etiology of any disease, not just COVID-19. This will be mentioned in the Subsection 2.2.4, where the typical treatments of the illness are discussed.

2.2.2 Pathophysiology

The *exact* pathophysiological background of the symptoms has not yet been established. However, the following mechanisms represent potential candidates: consequences of single or multiple organ damage, the difference in time needed for the recovery of each organ system, disruption of the immune system, and stimulation of the hyperinflammatory state [42, 41]. [42] argue that the similarities in the post-acute sequelae of the SARS-CoV-2 , SARS-CoV-1 and MERS could be attributed to their genetic similarities (as discussed in the previous section). It should be mentioned that the social and financial consequences of the disease represent potential contributors to the psychological symptoms expressed in the post-acute phase [41].

[42] provide a grouping of mechanisms that might affect the pathophysiology of the Long COVID:

1. virus-specific pathophysiologic alterations
2. inflammatory damage as a response to the acute infection and with that immunologic aberrations
3. sequelae of post-critical illness

A summary of the above-named pathophysiological processes that are potential explanations of Long COVID is provided in the Figure 2.3 and designed by [41].

2.2.3 Symptoms

The majority of the symptoms recorded in patients suffering from Long COVID can be classified into two subcategories [41]:

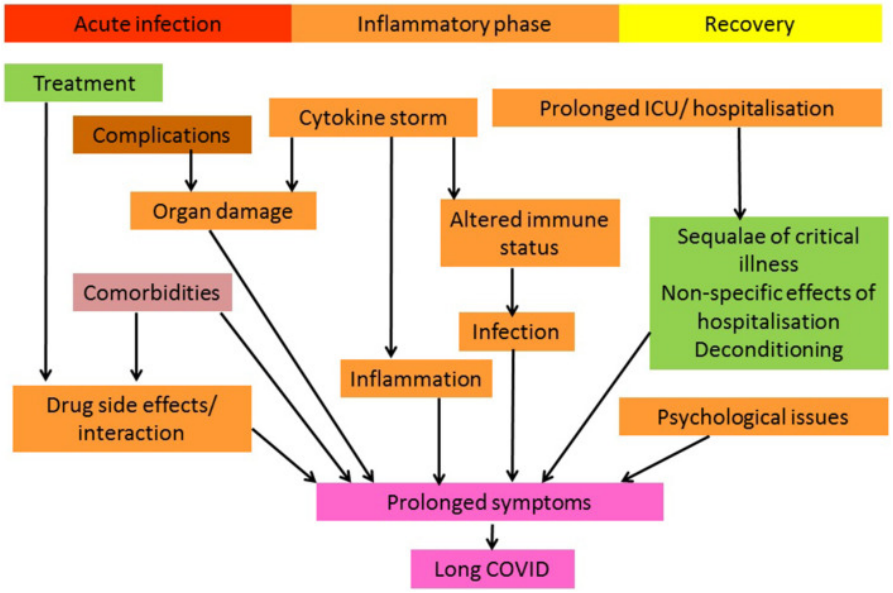

Figure 2.3: Different pathophysiological processes that might explain Long COVID [41]

1. Upper respiratory issues, headache, and fatigue

2. Multi-system complications (typically including gastrointestinal problems and febrility)

Fatigue is reported to be one of the most common symptoms that persist during recovery from acute SARS-CoV-2 infection regardless of the illness severity [41]. The post-viral fatigues can also be seen after infections with other viruses, for example, MERS-CoV, Ebola, or influenza [41]. Generally, it is caused by multiple factors and is compared with myalgic encephalomyelitis/chronic fatigue syndrome (ME/CFS) due to the resemblance between the two. [43]

The second most common symptom which persisted after the acute phase is *Dyspnea* or shortness of breath [43]. Similarly to fatigue, associations have not been found between disease severity and the presence of this symptom in the post-acute phase [43]. It is hypothesized that the chronic dyspnea is a result of residual lung involvement, which, in general, eventually gets resolved [41].

Further symptoms include cough, neuro-cognitive issues such as concentration problems and cognitive blunting ("brain fog"), insomnia, cardiovascular complications, myalgia, and others.

2.2.4 Treatment

As noted at the beginning of this section, the classification can significantly aid the process of establishing the correct treatment, because it offers general guidelines for differentiation not only of the disease and symptoms but of treatments as well. More specifically, a general treatment for the Long COVID is very difficult to find because of the diversity of the symptoms and the multitude of their localizations, hence developing a treatment for only one symptom group is considerably more efficient than developing a treatment for all of them at once. As pointed out by [41] and [43] the treatment should encompass a *multidisciplinary approach* along with different treatment types such as: *symptomatic treatment*, *physiotherapy* and *psychological assistance*. For example, symptomatic treatment of cough or myalgia is recommended by using paracetamol and cough suppressants. Psychological assistance is crucial for patients dealing with psychological issues caused by acute or post-acute COVID infection.

On the other hand, offering such extensive treatment can put an enormous burden on the hospitals and the medical staff. In particular, because the symptoms persist for prolonged periods and the ICU/hospital resources are kept occupied for this duration. In view of this, the guidelines for the frequency of medical examinations have been proposed [41]. On the other hand, a remedy might be situated in the field of Machine Learning.

2.3 Basics of Machine Learning modeling and forecasting projects

In this section a step-by-step explanation of the typical supervised machine learning workflow is presented, which stems from the industry standards such as *KDD*, *SEMMA* or *CRISP-DM* [44]. This consists of the following steps, which will be more exhaustively described in the following subsections:

1. Problem definition
2. Data Collection and Examination
3. Data Pre-processing
4. Choosing the appropriate model
5. Building and training the model
6. Evaluating the model

A visualization of these steps is presented in Figure 2.4. Moreover, the potential feedback loops (e.g. *model refinement*) are also visible in the workflow graph, along with some further (intermediate) steps. Furthermore, one can see the CRISP-DM phases of model development in Figure 2.5, where certain steps from the general workflow have been

renamed (Problem definition → *Business understanding*), or merged and one further step was added (*Deployment*).

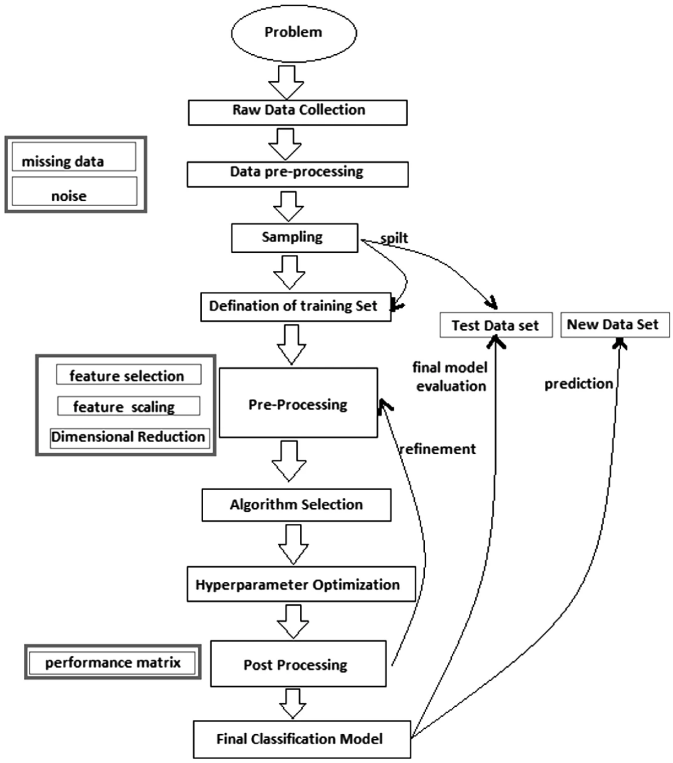

Figure 2.4: Typical Machine Learning Model Development Workflow [45]

2.3.1 Problem definition

The first phase of any Machine Learning development cycle is defining the problem statement and the task at hand. This step is called *Business understanding* in the CRISP-DM taxonomy. [46] In this phase the exact *problem definition* is formulated along with the *goals* the model should achieve. Related to the latter, one usually also defines the success criteria. These include, for example, the loss function (MSE/Cross-Entropy...) and performance metrics (Accuracy/Precision/R^2...). If a project of a larger scale is at hand, the majority of planning also happens in this phase. For example Requirement analysis, Risk Evaluation, resource distribution, and Costs are discussed. [46]

2.3.2 Gathering the appropriate data

In this phase, the data is collected and assessed with various metrics. The result of initial data collection is simply an abundance of raw data, which is then described, explored, and its quality is verified [46].

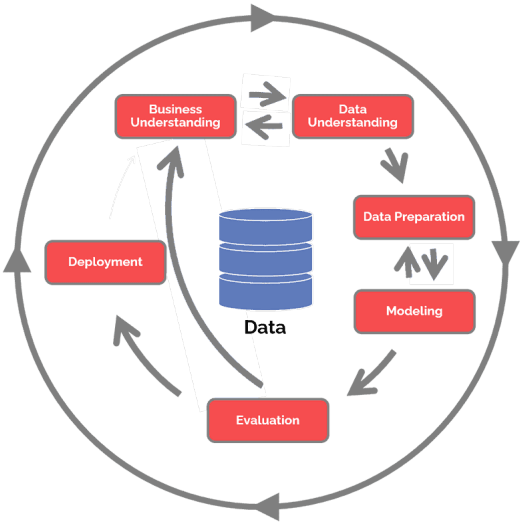

Figure 2.5: CRISP-DM Phases of development [46, 47]

Initial data collection

Firstly, there can be multiple data sources, each with varying data quality, which needs to be taken into account when passing the results into the next development phase. Secondly, an important criterion for data sources is their reliability and trustworthiness. A strong bias in the data can lead to serious complications and even the abandonment of the project entirely.

Data description

In this step, the relevant facts about the collected data are extracted. The usual metrics that are discussed here are: the format of the data, the size of the dataset(s), and whether the relevant variables are represented to a sufficient extent. If the dataset does not satisfy such constraints, another one is found.

Data exploration

This step contains a more thorough analysis of the data and its underlying structures. Various tools such as statistical data analysis and visualizations are used in order to obtain a deeper understanding of the data at hand and uncover significant patterns within. Some commonly used tools for statistical data exploration are Principal Component Analysis, Cluster Analysis, Discriminant Analysis, and Factor Analysis.

Verification of the data quality

The final step of this phase includes a critical assessment of the dataset's quality. The topics usually discussed here are the portion and localization of missing values and, consequently, whether imputation or removal of these missing values is preferred. One also performs outlier detection and tries to explain their origin.

2.3.3 Data Preprocessing

This stage encompasses all necessary steps of transforming the raw data into the model's input data. According to [46] this phase can be further divided into five steps:

1. Selection
2. Cleansing
3. Construction
4. Integration
5. Formatting

Each of the five steps will now be described in more detail:

Data Selection

In the first step of this phase, a decision is made for each variable (feature) if it should be kept or discarded. The decision is usually followed by a justification. The majority of data that is discarded is either noise or data irrelevant to the model (e.g. patient/customer ID). Removing data also decreases its dimensionality, which leads to simpler models, enables better generalization and reduces the risk of overfitting.

Data Cleaning

Having important features selected, the next task is to handle missing values and outliers. These actions are performed in accordance with the decisions made at the final step of the previous stage (Verification of data quality). Missing values are either imputed or removed and outliers can either be removed, adjusted, or left unchanged, depending on the goal and further constraints.

Data Construction and Integration

In data construction, new features are added through derivation with already existing features as a foundation (e.g. by multiplying two features to get a new one). This disburdens the machine learning model since it does not need to learn these relations. Additionally, one transforms certain features in this step to make them more appropriate/interpretable

for the ML model. Examples of these transformations are Label Encoding, One-hot Encoding, or Discretization.

In the second part, *Integration*, the data from different datasets are merged/aggregated into a singular dataset (in case there are multiple datasets provided of course).

Data Formatting

Lastly, the data can be adjusted for modeling purposes by modifying the contents of each record individually. Common methodologies applied here are *reordering* of records, centering and scaling certain features or elimination of forbidden values in strings [46].

2.3.4 Choosing the appropriate model type

The focus of this stage is deciding which model type is going to be used depending on the problem definition and the dataset provided. Support vector machines (SVMs), Neural Networks, Random Forests, or Simple Gradient Descent Classifiers/Regressors can serve as examples of commonly used model types. Each of these has its benefits and drawbacks. Hence, the selection of the appropriate type should be accompanied by a sound justification.

2.3.5 Training

In order to obtain an unbiased error approximation, the dataset is typically split into two subsets: *training set* and *test set*. The training set is fed to the model for its training/fitting, while the test set is only used for validation and measuring the model's performance on the "unseen" data.

Having the model type selected and the data prepared, the next step is to create the model by initializing the base model with the initial set of values and passing the training data through the model to decrease the loss function and with that increase, the performance metrics that were chosen. A further concern, which is to be addressed in this phase is the *hyperparameter tuning*[2].

2.3.6 Evaluating the models

The assessment of the model's performance takes place in this phase. Multiple error measures and evaluation metrics can be incorporated into the process in order to determine the characteristics of the model and its training. One can gain various insights into the model and its training performance based on the analyses performed in this stage. Moreover, for the purpose of interpreting the results or explaining the anomalies, the domain experts are usually involved in the evaluation as well. In case the validation errors are too large or the model does not satisfy the criteria established in the first phase, the model definition and training can be repeated. One useful tool for model

[2] A detailed elaboration on the topic of hyperparameter tuning can be found in Section 2.8.

evaluation is *visualization*, because it allows the analysts and stakeholders to obtain a tangible representation, that is easy to understand and interpret. Through the results of this phase, a better understanding of the original problem is obtained and the knowledge of the given topic is expanded.

The following metrics will be used to evaluate the performance of the models fitted in the course of this thesis:

- **Accuracy** - number of correctly classified observations divided by the total number of observations [48]
- **Sensitivity** (True positive rate or recall) - the proportion of correctly classified positive samples of a single class
- **Precision** - number of correctly classified positives over the total number of positive predicted observations [48]
- **ROC** (Receiver operating characteristics) - The ROC curve depicts the change of TPR and FPR with varying thresholds for classifying observations as positive (usually from $-\infty$ to ∞). In the multi-class response setting, the ROC curve can be calculated by employing either the OvR or the OvO methodology. OvR (One-vs-Rest) strategy builds the ROC for each response class separately, more specifically by taking one class as the positive class and the remaining classes as negative. In the OvO (One-vs-One) case, all combinations of classes are built and the ROC is calculated. [5]
- **AUC** (Area under the ROC curve) - The AUC metric calculates the area under the ROC curve. If the response contains more than two categories, the ROC curves are created either using the OvO or the OvR methodology, their AUCs are calculated separately and they are aggregated. The latter is typically done by computing their (weighted) mean. This metric provides us with the information about how well the model differentiates between the classes, however, it is sensitive to class imbalances [48].
- **Cohen's Kappa** [6] - This metric is a typical measure of interrater reliability (the extent of similarity of the predicted values that are predicted by two different raters/models). The unweighted Kappa is suitable for assessing the performance of nominal classifiers [6], while the weighted version is more appropriate for ordinal classifiers because the penalty is larger if the prediction is further away from the actual class [4]. Two examples of the weighted Kappa are linear and quadratic weighted Kappa [49]. As the names already suggest, the linear weighted Kappa penalizes the errors in terms of a linear function, whereas the quadratic Kappa utilizes a quadratic function for error penalization.

It should be noted that Sensitivity and Precision are only related to *one response class*, while the other metrics evaluate the performance across all categories.

2.4 The main supervised ML prediction problem types: Regression and Classification

The majority of the supervised machine learning tasks can be divided into two main categories: *Regression* and *Classification*.

Both regression and classification have found their applications in almost all areas of science, such as medicine, physics, astronomy, and numerous more. They are statistical and Machine Learning tools that enable researchers and experts from diverse fields of study to assess their results, gain valuable insights into the collected data, make predictions about future events and create new knowledge through inference.

The two methodologies will now be discussed separately in more detail. Their definitions will, firstly, be provided along with a few historic remarks related to their development. Then, the basic approaches for each methodology will be explained at greater length. Finally, concrete examples of their applications and more advanced approaches will be named. The discussion will be accompanied by figures in order to present the matter more intuitively.

2.4.1 Regression

Linear Regression describes a process of finding coefficients β_0 ... β_p : $\beta_i \in \mathbb{R} \forall i \in [1, p]$ s.t. the *continuous* response variable Y is being modeled as a linear combination of p feature variables X and the coefficients β_0 ... β_p (2.1). The coefficients are chosen under the constraint of minimizing the predefined objective function, which depends on the differences between the modeled and the actual values, also known as the *errors* or *residuals*. One popular and established approach to implementing this constraint is the *Least Squares* method (LS), which minimizes the sum of the squared residuals. This minimization can in simpler cases be performed analytically, but in more complex settings the heuristics, such as Stochastic Gradient Descent [50, 51], are proven to be more efficient.

The methodology of Least Squares Regression dates back to 1805 and was firstly introduced by Legendre [52]. It was further examined and expanded by Gauss in his work [53]. Numerous advancements on the topic of regression have been presented in the research community. Some of them include using novel objective functions or modifying the existing ones with the goal of simply improving its performance in general or making the process more robust against outliers (For the interested reader the author highly recommends the research carried out by Huber on this topic [54, 55]).

On the other hand, there are also non-linear regression methods. One example would be polynomial regression, which uses higher-order polynomials of the feature variable(s) to model the response. [56].

The central theoretical model of multiple linear regression can be represented with the

following equation:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon = \boldsymbol{\beta} \cdot \mathbf{1}\boldsymbol{x} + \epsilon \tag{2.1}$$

where y denotes the response variable, x_1 to x_p denote the p feature variables (summarized in the vector $\boldsymbol{x}$) and ϵ represents the residuals. Note: $\mathbf{1}$ marks that there is one element equal to 1 added at the beginning of the vector $\boldsymbol{x}$, which is needed for multiplication with the β_0 coefficient.

The aim of the least squares estimator is to minimize the following objective function:

$$\sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - ... - \beta_p x_{ip})^2 \tag{2.2}$$

where n marks the total number of $(\boldsymbol{x}, y)$ observations.

Figure 2.6: Regression

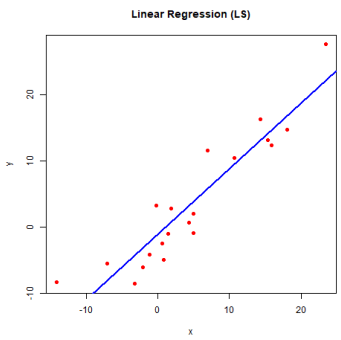

Figure 2.7: Linear Regression

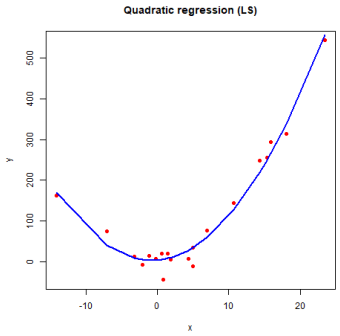

Figure 2.8: Quadratic (Polynomial) Regression

Figures 2.1 and 2.8 illustrate the process of finding a regression line on a set of observations (red points), where x axis denotes the feature/explanatory variable and y axis represents the response. The red points represent the ordered pairs (x_i, y_i) of the observations defined by these two variables. In Figure 2.7 a blue regression line can be seen, which was found with the Least Squares method. Here only one feature, one response variable, and a clear linear correlation between x and y are presented. On the other hand, in Figure 2.8, the response variable y can be modeled as a polynomial of the feature variable x.

Of course, multiple linear regression does not possess enough expressive power, which is necessary to model the complex relationships of real-world data. Hence, the research has been done on more sophisticated statistical and machine learning models, with the latter being of more interest to us. Some examples would be Support Vector Machines (SVMs), Random Forests (RFs) and Neural Networks (NNs) [57]. Both RFs and NNs will be further discussed in this chapter.

2.4.2 Classification

The second common variable type is categorical variables. In contrast to continuous variables, observations of categorical variables can take on only a finite number of values [58]. These can further be classified into binary, where only two categories exist (e.g. yes/no, present/absent), and multi-categorical. The criterion for further subdivision of multi-categorical variables is the presence of the ordering between the values. If the order between the values of the variable can be established, they are called *ordinal categorical variables*. Conversely, if the order between the values cannot be established, they are then referred to as *nominal categorical variables* [58].

Classification can generally be described as the process of assigning a class to an observation based on its input features. This assignment can further create a label-based grouping of observations. It can be defined more precisely as follows:

Definition 1 *Given a set of p input features (quantitative or qualitative variables) and a finite set of class values (labels) D_y, also referred to as the domain of the response variable, one defines classification as a process of assigning* ***exactly one*** *of the labels $l \in D_y$ to one observation i based on its input features $\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ denotes a p dimensional vector with input feature values. Observations with the same labels can additionally be summarized into groups, also called* ***clusters****. Classification can further be divided into two fields, one where the aim is to* ***predict the correct label*** *and the second where the aim is to* ***find*** *the groupings in the data.*

It should also be added that the terms *classification* and *labeling* are interchangeable and both can be found in the literature.

Two standard examples of classification from everyday life would be deciding if an E-Mail is spam or not based on its contents or determining what animal is shown in the picture. Both of the examples are quite established in the Machine Learning community and have attained remarkable results with respect to accuracy in the past decade.

It is important to note that the methodology discussed here primarily relates to *supervised* classification methods [59]. Supervised classification methods assume that the response labels (from D_y) are known in advance for a certain number of observations. Based on these observations the prediction models are then developed. On the other hand, *unsupervised* classification methods usually do not have any prior grouping information. Hence, they aim to *find* groupings/clusterings in the data, by exploring the similarities between observations [59]. A specialized research field dedicated to finding clusters in the data is called *Cluster Analysis* and has been thoroughly researched by J.A. Hartigan [60, 61].

One prominent statistical approach to the task of supervised classification is *Discriminant Analysis* [58]. The term Discriminant Analysis was firstly used by Fisher in his work [62] [63]. However, since this thesis primarily focuses on Machine Learning, some Machine Learning-oriented approaches to the task at hand will be presented in the following:

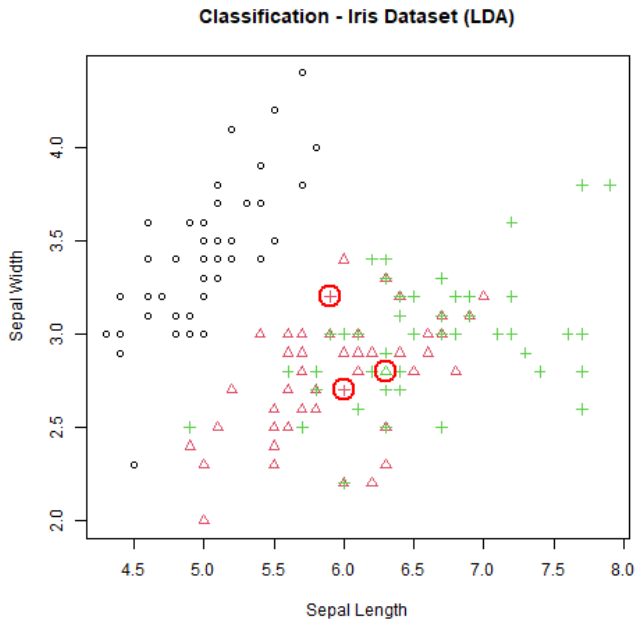

Figure 2.9: Classification with Linear Discriminant analysis on the Iris Dataset [64, 65]

In Figure 2.9 the results of the Linear Discriminant Analysis (LDA) applied on the Iris dataset [64, 65] are plotted. Sepal length is plotted against the Sepal Width of the flowers observed and their actual species are marked in different colors. LDA was used to predict their species from the same dataset again. The subsequently predicted classes are plotted with different shapes. The misclassified observations are additionally marked with bold red circles.

There also exist ways of reformulating the classification problem as a regression problem and vice-versa and this procedure can be advantageous in certain cases. However, it also carries a risk of losing or suppressing some properties of the data, which might be useful for the model. Hence, approaches that have been developed for both classification and regression aim to utilize the given data as efficiently as possible with the goal of attaining the best results.

One approach that exploits the categorical property of the response variable are *Decision Tree Classifiers* or *Classification Trees*, which were thoroughly researched by [66], [67] and [68]. These will be discussed in more detail in Section 2.6.1.

2.5 Ordinal Regression

Ordinal Regression (OR) can be seen as a combination of both classification and regression. Typical classification algorithms assume that the response variable y is a *nominal* categorical variable. In the setting of Ordinal Regression, however, the response variable

is an *ordinal* categorical variable, meaning there exists an ordering between the class labels. Since most of the classification algorithms do not utilize this ordering information of the class labels, applying them would certainly result in sub-par results. This already raises a few questions about which methods are to be applied when approaching these types of tasks.

In order to answer these questions, a formal definition of Ordinal Regression is firstly given along with its peculiarities w.r.t. to standard regression and classification. A recently introduced taxonomy of the OR methods proposed by [69] is demonstrated, which offers a classification of the Machine Learning OR approaches[3]. The final subsection is devoted to naming several common applications of OR in the scientific and industrial fields.

2.5.1 Definition

Ordinal Regression or Ordinal Classification, also referred to as "ranking" in the literature is defined as follows:

Definition 2 *Given are the vector of input features (quantitative or qualitative)* $\boldsymbol{x}$ *and an ordinal categorical response variable* y *with the domain* D_y *and labels* $l_1, l_2 ... l_Q \in D_y$ *with some ordering information* $l_1 \prec l_2 \prec ... \prec l_Q$*. Ordinal regression denotes the process of predicting the class label* $l_i \in D_y$ *of the new observation based on its input features. [69]*

Thanks to the ordering information, the elements of D_y can be compared, which would not be possible if y was a nominal categorical variable [69]. On the other hand, the operator $\prec$ does not fully correspond to the $<$ operator used in regression, since the labels of ordinal regression "do not carry metric information" [69].

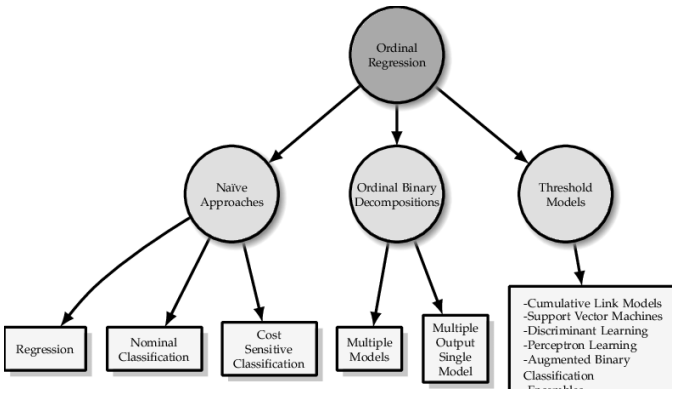

Figure 2.10: Taxonomy of ordinal regression approaches proposed by [69]

[3]There also exist statistical approaches to ordinal regression and they were developed before the ML methodologies, however these are not a topic of interest for this thesis. As a further reference for the reader on this topic, we recommend the works of P. McCullagh [70, 71], Anderson [72] and C. Winship [73]

2.5.2 Taxonomy

As stated by [69] a considerable amount of research has been performed on the subject of ordinal regression in the last two decades, however, the general taxonomy was absent. Therefore, they proposed a categorization of approaches to ordinal regression tasks, which is summarized in Figure 2.10. Providing the taxonomy plays an important role in further research and assessing the current state-of-the-art [69].

2.5.3 Applications

The problem of ordinal regression gained popularity due to the ever-growing demand in the field of information retrieval. [74]. According to [69] it finds its use in various research fields, such as medical research, age estimation [75], credit rating, image classification, social sciences, text classification, and more.

Modern approaches to building Ordinal Regression models will be discussed in depth in the Chapter 3, more specifically, in Sections 2.9, 3.2, 3.3.

2.6 Random Forests

Random Forest (RF) describes an approach to classification and regression, that is based on aggregating multiple decision trees in an *ensemble*. [76] Hence, in order to explain the Random Forest classifiers/regressors, their main components, *decision trees*, are to be explained first. Along with decision trees, the predecessor/base of the Random Forest algorithm, the *Bagging predictors*, will be described in detail before proceeding to extend this base methodology with additional characteristics to obtain the Random Forest predictors. Subsequently, two key terms related to Random Forests will be defined and clarified, namely, the *Out-of-bag Error (OOB)* and *Variable Importance*. Finally, the typical advantages and disadvantages of the RF predictors are presented.

It is also important to note that the first extensive research on the Random Forest paradigm has been conducted by Breiman [77, 78, 79]. His work will further be discussed in this chapter, after introducing several basic concepts.

2.6.1 Decision Trees

[80] define decision trees as sequential models, that join multiple simple tests in a sequence. Each of these tests assigns an observation to a sub-tree (successor) based on the comparison with a certain threshold value (for quantitative variables) or with a set of class labels (for qualitative variables) [80].

In other words, the new observation $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ is fed to the beginning of the tree (the root node). The root node can be observed as an inner (decision) node. A decision is then made based on the splitting criterion: If the splitting variable j is quantitative, the node compares the value of that variable in the feature vector (x_{ij}) to one (or more) threshold(s). Based on that comparison the observation is passed onto the root of a

designated sub-tree. In the case of qualitative splitting variables, the criterion compares the value x_{ij} to the subsets of that variable's domain and based on that comparison passes the observation onto the successor sub-tree. The process is repeated until terminal (leaf) nodes are reached and each of them usually indicates exactly one class label of the response variable.

It should be pointed out that there also exist approaches that use multiple splitting variables instead of one.

A simple decision tree is illustrated in the Figure 2.11. Here four input features are used and all four are quantitative (continuous) variables. It is furthermore assumed that there are two response class labels A and B.

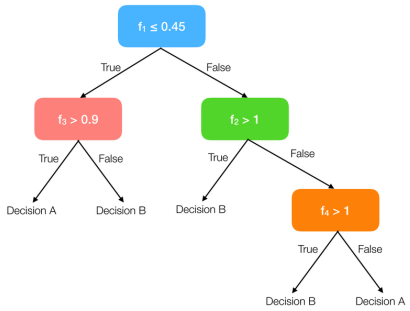

Figure 2.11: Basic structure of a decision tree with four input features [81]

Since Decision Trees have a finite number of terminal (leaf) nodes, it can be inferred that they are better suited for classification rather than regression, even though they are capable of addressing both. The main advantage of decision trees as claimed by [67] is their capability to decompose complex decision problems (usually in higher dimensions) into multiple simple ones. As stated by [80] the logical decisions of DTs are much more interpretable and understandable to the modelers and researchers than weights in neural networks.

2.6.2 Bagging Algorithms

Having the decision trees introduced, the next focal point of the discussion would be the foundation of the Random Forest methodology, the ***B**oosted **agg**regat**ing** ≈ **Bagging** algorithms*.

As previously noted, Random Forests consist of multiple Decision trees as their base learners (also called *weak learners*) forming an ensemble. The rationale behind the establishment of a decision-tree ensemble, instead of using decision trees on their own is due to their *high variance* [82]. To remedy this, *Bootstrap aggregation* or *Bagging* methods have been proposed by Breiman and have proven themselves to be very effective [82, 78].

Breiman defines Bagging as follows [78]: The main idea behind bagging is to increase the number of training sets, by repeated sampling (bootstrapping) the original training dataset. One builds B new training sets by bootstrapping and uses these new training sets to fit B different base learners, in our case decision trees. In order to finally obtain a low-variance (more accurate) learner one *averages* the prediction results of all B base learners (in case of regression) or decides based on *majority vote* (in case of classification).

As concluded in [78], some of the interpretability of single learners is sacrificed for the increase in accuracy.

2.6.3 Random Forests

According to [82], one common problem with the bagging trees is the fact that they might in some cases be correlated. This issue can be traced back to the process of fitting the individual trees, more precisely to the number of predictors used at each new split. The authors further state that in case there is one very 'strong' predictor [4] and several moderately 'strong' ones, the majority of decision trees constructed with bagging will tend to share the first split node, namely the strongest predictor. This would make them look very similar and therefore have a high correlation. (As formally stated in [77], the strength of the individual predictors and the correlation of the decision trees influence the generalization error of the whole ensemble.) Finally if the average of multiple similar trees is calculated, it does not yield a significant decrease in model variance. Random Forests can be seen as a modification of the bagging method, which addresses the above-stated problem. By limiting the number of predictors available for each split node of each decision tree to *random* m features (usually $m \approx \sqrt{p}$, where p is the total number of features), one reaches the desired diversity (low correlation) of the trees [82, 83].

Furthermore, it holds that due to the Law of Large Numbers, Random Forests **do not overfit** if additional trees are added, however, they do generate a lower bound for the generalization error [77].

2.6.4 Out-of-Bag Error (OOB)

During training, it is desirable to have an error estimate, usually of the generalization error [77]. The error estimation and comparison indicates if the model is making any progress. The majority of Machine Learning approaches either use a dedicated *test set* or *Cross-Validation* in order to obtain this estimate. However, since Bagging and Random Forests use Bootstrap samples and build multiple classifiers, it is possible to use a completely different error estimate, the *Out-of-Bag (OOB) error*.

To define OOB error, the OOB samples and classifiers need to be defined first. In his work, Breiman gives the following definition of Out-of-Bag samples and, consequently, of OOB classifiers [77]:

[4] The predictor, that possesses plenty of information useful for making a prediction.

> Assuming an ensemble learner is to be fitted, which consists of multiple weak learners (e.g. trees), randomly sampled observations are to be drawn from the training set $\boldsymbol{T}$ and used for fitting each of the weak learners. Let us denote the set of observations used for training the base learner k with T_k. In order to get the out-of-bag classification of the observation $(x, y) \in \boldsymbol{T}$, only the results of those base learners are used, which **did not** use this observation during their training, in other words, all learners l for which holds $(x, y) \notin T_l$.

Not only does the Out-of-bag principle provide a decent estimate of the generalization error (even though it tends to overestimate in some cases), but it also removes the need for dedicating a certain amount of observations for a test set [77]. Using the OOB classifier as an error estimation strategy is equivalent to using a test set of the same size as the training set according to [77].

2.6.5 Variable Importance

Another term frequently associated with random forests is the *Variable Importance*. As a researcher or a practitioner, it is useful to know which variables have a strong influence on the model. Conversely, if certain variables exhibit little to no influence on the model, they can be ignored with little to no change in model performance. Omitting insignificant variables from the model can be very useful because it makes the model easier to understand and generalize better since unnecessary noise is removed. In his original paper [77], Breiman suggested the permutation-based measure of variable importance for random forests, which is described in the following.

After fitting the forest, the values of the examined variable j are permuted. Then, the predictions are made on the OOB samples, each usually having a new value on its j-th position, due to the permutation performed in step one. The misclassification rate is recorded for the predictions obtained in this manner. Variable importance of variable j is then calculated as the percentage by which the misclassification rate increased compared to the misclassification rate of regular OOB error estimate (without permutations). This is performed for each variable $j \in \{1..p\}$

Breiman also notes that the dependency of the two variables certainly plays a role in their joint influence on the model (importance). For example, if two variables are highly correlated, they carry very similar information. Therefore, removing one of them from the model does not yield a significant decline in the model's performance. [77]

2.6.6 Advantages and Disadvantages

This section provides general strengths and weaknesses of the Random Forest models.

Strengths

- **Reduced chance of overfitting** - Since the outputs of multiple *uncorrelated* weak learners are aggregated, the chance of overfitting is significantly reduced.
- **Handling missing data** - According to [84], the RFs are capable of handling missing data by using the *proximity* of the datapoints.
- **Robustness** - Multiple studies have shown that RFs are more robust w.r.t. outliers and noise in comparison to other regression/classification methodologies [85]

Weaknesses

- **Long fitting times** - Large amounts of decision trees require prolonged training periods.
- **Low Explainability - "black box" nature** - Even though they offer the measure of variable importance, other aspects and still remain unclear due to the aggregation process.

2.7 Neural Networks

The goal of this section is to give a brief introduction to the Neural Network (NN) paradigm. Firstly, their historical development will be summarized. Then, the formal definition of a neuron, layer, and neural network is provided along with the basic characteristics that determine their behavior. Furthermore, the process of updating/training the model will be explained. After covering the essentials of the standard approach, a few extensions of the conventional NNs will be named and described. Lastly, the advantages and disadvantages of using NNs as a model type are presented.

2.7.1 Historical Background

Inspired by the inner workings and mechanisms of the human brain, researchers of the twentieth century began exploring the learning capabilities of artificially created neural structures. Throughout the past eighty years, this field of research faced quite turbulent changes. Ranging from the enthusiastic beginnings in the 1950s, over the pessimistic and dark period around the 1970s, to the era of innovation and vast development [86], which is present to this day, the neural network research is one of the most promising research fields of the modern science, showing continuous growth in the past decade. Further information on historical advancements in the field of neural networks can be found in [86].

2.7.2 Neurons, Layers and Neural Networks

The following definition of a neural network is partly inspired by [86]:

A neural network (NN) is a machine learning model consisting of multiple basic units and neurons. These neurons are typically organized in layers, s.t. the neurons of the same layer are not connected. Each neuron can have ingoing and outgoing edges (*synapses*), from the previous or towards the next layer respectively. Neurons without ingoing synapses are called input neurons (or *peripheral afferents* as defined by [87]). Each neuron is, furthermore, defined by the following four elements, which can also be seen in the Figure 2.12:

1. Input weights
2. Activation Function
3. Bias Term
4. Output

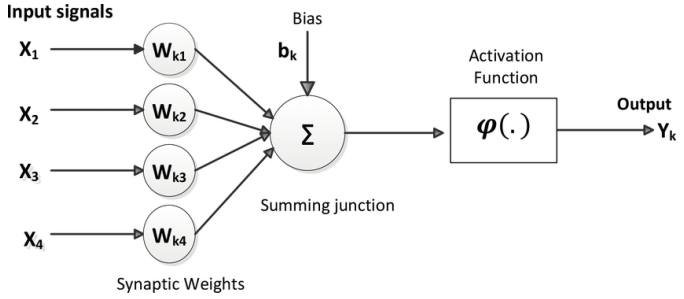

Figure 2.12: The McCulloch-Pitts model of a neuron [88]

In the Figure 2.12 four input signals $x_1 ... x_4$ are to be seen, each weighted with weigths $w_{k1} ... w_{k4}$. These are further summed up and the bias is added ($\sum_{i=1}^{4} x_i w_{ki} + b_k = s_k$). This sum is then fed to the activation function ϕ. The output y_k is then equal to the value $\phi(s_k)$.

Having the fundamental constituents of the network explained, the larger unit, the network itself, and its dependence on the singular neurons, are going to be discussed in the following. Namely, the behavior of the network depends on the following three parameters:

1. The Activation functions of the neurons
2. Learning Rule (Defines how the weights are updated)
3. Architecture (The pattern in which the neurons are connected)

The most widely used form of neural networks are feed-forward neural networks. These denote that the signal only flows "forward", that is, from the input nodes towards the output nodes. On the other hand, recurrent neural networks also allow the flow of data in opposite directions. These, however, have proven to be highly complex. Furthermore, *back-propagation* is the most frequently applied method for neural network learning, more specifically for updating the weights and biases. *Back-propagation* calculates the gradients (derivatives) of the error function from the output layer towards the input layer with respect to the weights and biases [89]. The calculations are performed in a back-to-front manner in order to increase the reusability of the intermediate results. Based on these gradients and the *learning rate*, the weights and biases are adjusted. Back-propagation is the foundation of more complex learning algorithms such as *Stochastic Gradient Descent (SGD)* or *Adaptive moment estimation (ADAM)*.

2.7.3 A layered architecture

As previously noted, neural networks can be seen as a stack of interconnected layers. In the common notation the layers are usually divided into the *input layer*, one or more *hidden layers* and the *output layer*. In the literature, the *single layer* perceptron denotes a neural network with no hidden layers, while a *multi layer perceptron (MLP)* describes a network with one or more hidden layers. It should also be mentioned that the perceptrons were firstly introduced by Rosenblatt [90].

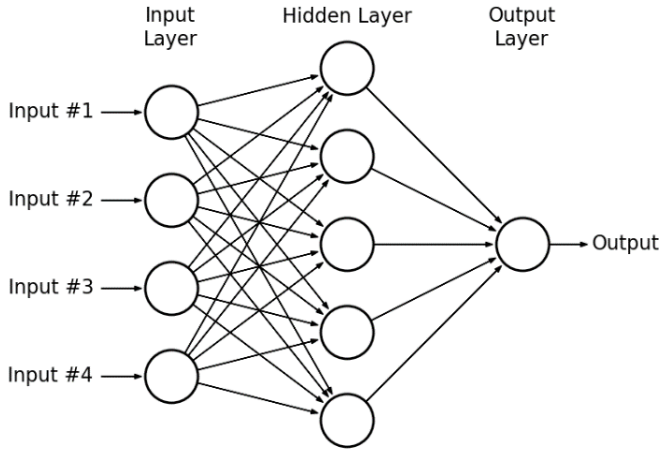

Figure 2.13: Single hidden layer Perceptron (Multilayer Perceptron) [91]

In Figure 2.13 four input signals can be seen feeding into the input layer with the same amount of nodes. Outputs of the input layer are *densely*[5] connected to the hidden layer, which contains five neurons. The outputs of the hidden layer are further fed into the

[5]Densely connected means that each node from the previous layer is connected with each node in the next layer.

single node of the output layer, which transforms them into a single output signal. Note: the explicit notation of bias terms is missing. For the simplicity of the figure, the authors probably decided to leave them out.

2.7.4 Extensions of the original approach

In the current state of the research, there are a plethora of modifications to the original neural network approach. Hence, only a few of the methodologies will be mentioned, that have gained a considerable amount of popularity in recent years. One extension of the perceptron is the *Recurrent Neural Network (RNN)*. RNNs allow the feedback connections, in other words, outputs of one layer can be used as input to one of the previous layers. Hopfield networks [92] and Self-organizing machines [93] can be brought forward as fair examples of recurrent neural networks [86]. These, additionally, allow researchers to create models for human memory retention [86].

Another modification of the original approach are *Convolutional Neural Networks*, firstly introduced by LeCun et al. in their paper [94]. Convolutional neural networks organize neurons as a grid, which usually corresponds to the pixels or groups of pixels on a visual field. Hence each neuron processes a different part of the image. Convolutional neural networks can also have multiple layers, which increases their power. As expected these are usually used for facial recognition and image classification as well as for speech recognition [94]. Hinton provided extraordinary research on *Deep Belief Networks* [95], which are based on *Restricted Boltzmann Machines (RBMs)* [96]. Deep Belief Networks find their use in Dimensionality reduction, Feature Extraction, Facial Recognition, and numerous other fields of Machine Learning and Data Science.

2.7.5 Advantages and Disadvantages of neural networks

In this section, the positive and negative characteristics of neural networks will be briefly summarized in a listing fashion. The facts presented here were partly inspired by [97] and [98].

Benefits

- **Ease of development / Shorter development times** - NNs are relatively easy to develop since nowadays even the inexperienced developers are capable of creating considerably performant NNs thanks to numerous high-level development frameworks (e.g. `PyTorch` and `Keras` libraries in Python are very popular choices)

- **High Expressiveness** - NNs are capable of expressing a wide range of both linear and complex non-linear dependencies between the predictors and the response variables

- **Noise resistence** - According to [99], Deep NNs are very resilient against a considerable amount of noise in the input data, allowing them to achieve very robust models that have excellent generalization capabilities.

Drawbacks

- **Overfitting** - Since the NNs are capable of modeling such complex relationships and they have a large number of parameters to be estimated, they often tend to overfit, in other words, "memorize" the training data.
- **Randomness** - The initial weights and biases are usually sampled from a given distribution (typically uniform), therefore in some cases the outcomes can differ substantially after repeated training.
- **Low Interpretability ("black box" nature)** - From a researcher's/developer's point of view, it is generally of interest to discover not only *which* covariates play a significant role in the model's prediction scheme but also *how* they influence the predictions made by the model. This task is, unfortunately, not quite straightforward when it comes to NNs, especially in cases where they have many layers.

Each of the drawbacks has already been discussed in the literature and the methodologies to remedy them have been proposed (a few of which yielded very beneficial results).

2.8 Hyperparameter Tuning

Learning rate, mentioned in the section above, can be seen as a hyperparameter of the model. A hyperparameter is a user-defined value, which usually does not change as the model learns. However, the model's learning performance is very dependent on these values, hence they are to be chosen with caution.

There are several issues when it comes to choosing the optimal hyperparameters (this process is also called *Hyperparameter tuning*). A few of them have been described in [100] and will be briefly summarized in the following:

Firstly, depending on the complexity of the model it can take a considerable amount of time to train a single instance of the model, with *only one* hyperparameter configuration. Additionally, certain hyperparameters (e.g. batch size in neural networks) can strongly influence the training time of neural networks.

Secondly, the randomness (stochasticity) induced in many machine learning algorithms can pose a concern regarding the optimality of the found solution. One example would be the initialization of neural networks, which plays a crucial role in their modeling. More specifically, the initial values of the weights and biases significantly influence further training and should be chosen carefully [98].

Lastly, the complexity of the search space and the dependence of certain hyperparameters on others can induce even greater training times. [100] state the number of hidden layers and the size of each hidden layer as an example of this concern since the number of hidden layers directly influences the second hyperparameter, the size of each layer.

As stated by [100], the metaheuristic optimization research can be very useful in the further development of automatized hyperparameter tuning. They also mention a few state-of-the-art approaches to this matter. Some of the popular methodologies are Bayesian and sequential model-based optimizations. Furthermore, the Random Search has proven to be a decent baseline for the comparison of different hyperparameter optimization algorithms. [100, 101].

Apart from the advancements in the scientific research on the topic, nowadays exist multiple well-established high- to low-level libraries/frameworks, which enable practitioners to perform hyperparameter tuning of their ML models efficiently. Most of them are implemented in the programming language Python. Some examples would be *Hyperopt* [102] and *Optuna* [103].

2.9 Naive approaches to ordinal regression

In this section two typical naive approaches to ordinal regression will be presented and their shortcomings will be discussed. These two naive methodologies are: 1) converting ordinal classes to real numbers and then performing classic regression and 2) ignoring the ordering information completely and performing standard classification.

2.9.1 Mapping classes to $\mathbb{R}$ + Classic Regression

In this approach, a mapping function is firstly developed s.t. $m : D_y \to \mathbb{R}$ where D_y denotes the domain of the response variable. The mapping function is frequently chosen to be $m(l_i) = i$ [69]. Hence, in order to decode the predicted values (which are in $\mathbb{R}$ they are rounded to the nearest integer ($m^{-1}(p) = \lceil p - 0.5 \rceil$). The response variable is then passed as an input of this function in order to obtain a pseudo-continuous variable. Having the response variable contain values from $\mathbb{R}$ the standard approaches to regression can be applied, e.g. by using Linear or Polynomial Regression models, Random Forests, Neural Networks, or SVMs.

However, there are several issues with this methodology, the primary one being that the distance between the classes remains unknown [69]. This creates difficulties with deciding on the appropriate decoding function (m^{-1}). In particular, if the distances between the classes are not homogenous (which is usually the case) this can lead to relatively complex decoding functions since each threshold between two numbers is to be set individually. For example in the majority of the cases where the function $m(l_i) = i$ is used as the mapping function and rounding to the nearest integer is used when decoding delivers poorer results than choosing the function based on intervals in $\mathbb{R}$. One example would be: Decode values in range $[1, 1.3)$ as l_1 and values in range $[1.3, 2]$ as l_2.

One approach aims to calculate pairwise label distances and apply them when translating the responses from $\mathbb{R}$ into D_y [104], however, the improvement is visible only to a certain extent.

2.9.2 Standard nominal classification

This strategy discards the ordering information of the labels completely and treats the response variable as a nominal categorical variable. Subsequently, different models for regular classification can be applied (e.g. NNs or RFs).

It is only natural to infer that ignoring information when training the model will lead to a deterioration of its performance when compared to the methods which include the discarded information. This method is no exception. It typically compensates for this deficiency by employing more training data [69].

An extension of this approach can be seen in the *Cost Sensitive Classification* [105], where the matrix with the costs of misclassification is included in the training procedure to increase the performance of the models [69].

CHAPTER 3

State of the art

This chapter offers a concise review of the current state-of-the-art research on the topics of predicting various COVID-19 metrics and advanced ML approaches to ordinal regression. The current literature has been screened and the most important facts about each research paper are briefly summarized in each paragraph. Firstly, the existing prediction models for COVID-19 statistics will be discussed. Several examples of the previously mentioned COVID-19 metrics are the number of infections, the number of deaths, the need for an ICU, and Length of hospital stay (LoS). Furthermore, the main research papers on the Random Forest approaches to ordinal regression will be presented along with their results. Afterward, the discussion will be focused on the use of Neural Networks as the primary architecture for the ordinal regression models. The chapter is then finalized with the statement about the novelty and the scientific merit that this thesis brings to the research community.

3.1 Current research on predicting the COVID-19 metrics

In this section, an overview of the results of four influential papers on the topic of predicting various COVID-19 measures is provided. The first paper discussed is a meta-analysis/review of the COVID-19 prediction models (multiple metrics included). The second paper focuses on Long COVID and finding the influential predictors of the disease. In the third paper, the authors compare four ML Models for the predictions of LoS and extract relevant features. Finally, the researchers of the fourth paper develop a comprehensive framework with the goal of improving ICU resource allocation.

Owing to the urgency and the necessity related to COVID-19 research numerous studies have emerged which predict various aspects of the disease. One systematic review of the models for diagnosing COVID-19 patients, predicting the number of infections, and mortality rates, and detecting people with increased risk of infection or hospitalization is provided by [106]. According to this review, the majority of the models, unfortunately,

contain a great risk of bias owing to the non-representative selection of control patients, risks of model overfitting, and unclear reporting methods.

Research has also been done on predicting the long-lasting form of the disease, also called long COVID, which became a very prominent topic lately. The study by Sudre et al. [107] attempts to find attributes and predictors of long COVID, for example. However, as the authors state themselves, the data used might not be very representative of the whole population and other factors which might influence the structure of the data have not been taken into account. Nevertheless, this paper could find its use as a foundation for further research.

In their paper, [108] aim to compare the performance of four ICU LoS prediction algorithms for COVID-19 patients. In their work, the algorithms discussed are Random Forests (RFs), Gradient Boosting (GB), Extreme Gradient Boosting (XGB), and Ensemble models. After applying the four algorithms to the local sample of 895 COVID-19 patients from eastern Saudi Arabia, they found out that RFs were able to attain the highest accuracy rate (of 94.16%), which compared to the results of other studies is relatively large. The second goal of the paper was to find the factors which stand in correlation with the LoS. The three factors with the highest correlation found were age, C-Reactive Protein (CRP), and Nasal Oxygen Support Days. The models developed should, however, be applied outside of the local scope in order to get a sounder estimate of their performance.

One more complex approach to predicting the COVID-19 metrics can be seen in [109]. The goal of this paper was to create an ML Framework that would answer the following three questions:

1. Which patients will need ICU care and what are the associated risk factors? (Response variable 1)
2. Which patients are at high risk of passing away at the ICU? (Response variable 2)
3. How long will the patient stay in the ICU? (Response variable 3)

The purpose of the ML Framework is clear: it offers support to the ICU management staff in the process of decision-making w.r.t. allocation of their resources. The data of 733 patients were used and 909 variables were given at the beginning. Furthermore, the top 10 variables which were the most useful for predicting Responses 1 and 2 were selected (based on ROC and AUC metrics). The binary classification for the first two response variables was performed with a very high-quality standard, where 22 SVM models were fitted for 22 balanced training subgroups and their results were finally aggregated with majority voting. For the LoS metric, a LASSO regression model was used in order to gain a clearer insight into the strongest predictors. The accuracies for the two binary target variables were 83% and 92% respectively. Finally, the Mean Absolute Error (MAE) of the LoS predictions was around 0.72, meaning: on average an error of less than one day was made when predicting the duration of hospital stay among the survivors.

3.2 Random Forest approaches to Ordinal Regression

This section contains highlights of two influential papers on the topic of employing Random Forests for ordinal regression modeling.

In their work, [4] use classical regression forests (random forests for continuous responses as introduced by [77]) with optimized score values. The approach assumes that there is an underlying continuous variable for the ordinal response. Instead of mapping the class values of the response into $\mathbb{R}$ and dividing $\mathbb{R}$ into optimized intervals in order to classify the new observations, this approach represents these intervals, also called *class widths*, as optimized scores. As proven by [110] the scale of the class scores does not influence the final result, i.e. if the classes y_i are mapped to $\mathbb{R}$ with the function $m(y_i) = i$ or $m(y_i) = i^2$ it does not have significant effect on the performance of the predictors. Therefore the authors of [4] have decided to limit the score interval to $[0, 1]$. The algorithm presented aims to build the trees with optimized boundaries of the Q adjacent sub-intervals of $[0, 1]$, where Q denotes the number of response classes. The boundaries of the intervals are firstly repeatedly sampled as heterogeneously as possible to obtain heterogeneous divisions of the $[0, 1]$ interval. Then the regression forests are grown for each of the divisions and, subsequently, the k divisions that achieve the lowest out-of-the-bag error metric upon evaluation are kept and a performance score is assigned with a dedicated performance function. The averages are then taken for each of the boundaries individually of the k best results and used to train the final regression forest. The authors of the paper performed an empirical comparison of the performance of their approach, the classical nominal RF and the naive RF approach where the class values are mapped to $\mathbb{R}$ with $m(y_i) = i$ function. The result of this analysis shows that the OF algorithm achieves only a slight improvement over other approaches. The authors also provide an implementation of their algorithm in the form of an R package `ordinalForest`, which will be used in this thesis as well.

Another approach to the problem of ordinal regression, which uses random forests is described in the work of [110]. In contrast to [4], this approach uses conditional inference trees [111] as base learners instead of regression trees to avoid bias. However, it should be noted that these trees take longer to train and are not recommended for use on high-dimensional datasets. Furthermore, this work uses two novel variable importance measures, specially designed for ordinal categorical response, in order to improve the performance of random forests. It is noted by the authors that they incorporate ordering information into the stages of tree construction and when computing the variable importance. One can go even further and employ ordering information in the process of the result aggregation before the final class prediction. According to an empirical analysis performed by the authors on the extent of the improvement this methodology yields, it has been established that only a slight improvement can be seen w.r.t. the naive nominal classification with random forests. The authors themselves state that it is recommended to use ordinal regression trees along with a permutation VIM with ordinal information incorporated instead of conditional inference trees with the ordinal response.

3.3 Neural Network approach to Ordinal Regression

On the other hand, [112] describe a refined neural network approach to the problem based on the research of [113]. Contrary to naive implementations of neural network classification where *only one* output node has value 1, the authors also consider the order of the categories by assigning each of the output nodes 1..k the value of 1 if the observation's response belongs to the k-th category, and the model assigns the value 0 to the remaining output nodes (k+1)..m (where m is the total number of output nodes/response classes). For example, *1110* is the encoding of the third category and there are four categories in total. Essentially, each output node O_i predicts the probability of the observation belonging to a category greater or equal to i. Furthermore, the training process is very similar to the classical neural networks, where the backpropagation algorithm is used. However, the authors also decided to use independent *sigmoid* activation functions for the nodes of the output layer, in order to avoid the constraint imposed by classical softmax, which asserts that the sum of the nodes must be equal to 1. This is important because the model is based on a novel encoding of the outputs. Furthermore, by employing this activation function it is *not ensured* that the monotonic relationship will exist between the output node results (probabilities that a node is belonging to a class lower than i). In particular $o_1 \geq o_2 \geq ... \geq o_Q$ is not asserted, where o_1 denotes the probability result of the first output node and Q the number of categories in the response (see Figure 3.1). This problem was addressed by [3] and will be discussed in the following paragraph. The additional merit of this approach is its capability to learn both in batch and online mode, making it an excellent instrument for adaptive real-time learning. The methodology exhibited significant improvement in the performance compared to naive nominal classification methodology with neural networks and it is comparable to other state-of-the-art approaches such as SVMs or Gaussian processes.

The approach described by [75] utilizes a modified Convolutional Neural Network (CNN) to estimate a person's age. The key difference to the classical nominal classification with NNs is that this approach transforms the regression problem into multiple binary classification problems, each classifier [1] i deciding whether the given observation has the rank larger than i or not. One can immediately observe that this methodology considerably resembles the approach from [112]. This is because they are both based on the work by [114], who first introduced the reformulation of the ordinal regression as a series of binary classifiers. The Neural Network also has $Q - 1$ outputs, for each task. The loss function for the final layer is softmax normalized cross-entropy loss. Since the methodology is applied for age estimation from images, the first part of the network contains convolutional layers. This approach also does not ensure that the monotonicity principle of the output classifiers remains preserved (see Figure 3.1), however, an extension of this methodology has been proposed [3], which addresses the issue.

[1] referred to as the *task* in the original paper

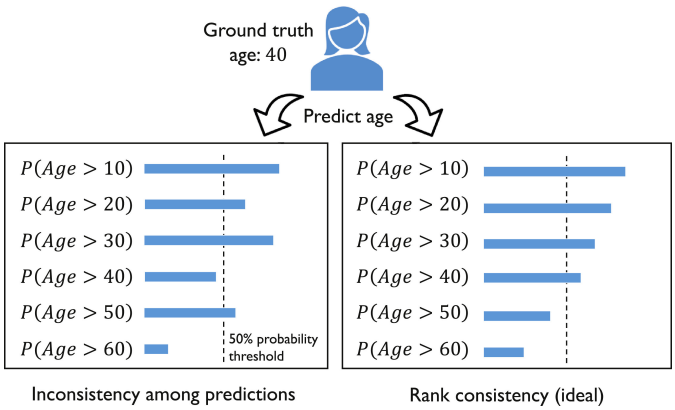

Figure 3.1: Visualization of the predicted probabilities for the same observation without monotonicity principle ensured (left) and with the principle ensured (right) [3]

As previously noted, [3] offer a solution to the problem of output prediction inconsistency mentioned in the previous two paragraphs. This is achieved by a slight modification of the output layer in the model proposed in [75], namely, the authors propose using the same weight parameter for all classifiers in the final layer and leaving the bias terms independent. The proof that this ensures monotonicity of the individual outputs can be found in the original paper [3]. Another name for the procedure is Consistent Rank Logits (CORAL) as proposed by the authors. This adjustment brought statistically significant improvement over the algorithm described by [75] and will be used later in this thesis.

3.4 Novelty and the merit of the research provided in this thesis

The scientific merit of this thesis is twofold: Firstly, by using a COVID-19-related dataset for predicting the hospital length of stay, the process of resource distribution in the ICUs is facilitated and, secondly, a direct comparison of the state-of-the-art Random Forest and Neural Network models that are oriented towards ordinal regression is conducted. Both of these aspects are going to be described more thoroughly in the following.

As mentioned in Section 3.1 the majority of the models developed are at risk of containing bias. Another weakness of these models would be the size of the datasets used for verifying the models. These are typically on a scale from 100 to 10.000 observations, which is relatively small in comparison to the dataset used in this thesis, which contains around 314.000 observations. Due to its extent, we believe that an improvement in the performance of the models could be expected and a more reliable evaluation of the models achieved. The utilization of this dataset for the models presented in this thesis should provide further insights into the capabilities of ML algorithms in assisting the medical staff.

Moreover, it can be observed that the Random Forests and Neural Networks are, indeed, individually quite well documented, even when used for Ordinal Regression. However, to the author's knowledge, a direct comparison of the two methodologies with the application to Ordinal Regression is absent in the current state of the literature. This thesis delivers this missing comparison and offers a further understanding of the suitability of each algorithm for the problem at hand. In addition, the research conducted on the employment of Random Forests and Neural Networks for Ordinal Regression typically uses publicly available toy datasets or simulated data for the model evaluation. On the contrary, this thesis will consider a "real-world" dataset and assess both methodologies in a more practical environment.

CHAPTER 4

Results

The details about how the models were developed and the evaluation of their performance are provided in this chapter. Firstly, the used frameworks, environments, and tools are named. Then, the description of the dataset is given, including its origin, dimensions, variable descriptions, and variable types. Furthermore, the exploratory data analysis is conducted and reported. In particular, the following tasks are included in exploratory data analysis: investigating various statistical properties of the dataset, studying the correlation/association between the features, and visualizing the individual variables. After examining the dataset and obtaining the information needed for further steps, the implementation of the models is described. Here, the focus lies on the specifics of the data preprocessing, fitting, and evaluation phases. Lastly, the results of the model evaluation are presented and visualized.

4.1 Tools Used

The two programming languages which were used for the analyses are *Python* and *R*. Additionally, the R package `ordinalForest` was used as an implementation of the random forest ordinal regression algorithm, proposed by [4] and the python library CORAL was used as an implementation of the neural network approach proposed by [3]. The main R packages for plotting were `ggplot2`, `ggcorrplot` and `graphics`. The naive solutions were implemented with the help of the R package `ranger` [115] (RFs) and the python library `PyTorch-Lightning` [116] (NNs).

4.2 Dataset Description

The dataset [1] used for training and evaluating the models is a public dataset found on the *Kaggle* website (`https://www.kaggle.com/`). Kaggle could be seen as a data science and machine learning-oriented platform that allows its users to participate in

#	Variable	Description	(Data) Type
1	case id	ID number of the case	integer
2	Hospital	ID number of the hospital	nominal categorical
3	Hospital type	ID number of the hospital type	nominal categorical
4	Hospital city	ID number of the hospital city	nominal categorical
5	Hospital region	ID number of the hospital region	nominal categorical
6	Available Extra Rooms	Number of available extra rooms in the hospital	integer
7	Department	name of the department that is overlooking the case	nominal categorical
8	Ward Type	ID Letter of the ward type	nominal categorical
9	Ward Facility	ID Letter of the ward facility	nominal categorical
10	Bed Grade	Condition of the bed in the ward [1]	ordinal categorical
11	patient id	ID number of the patient	integer
12	City Code Patient	ID Number of the city	nominal categorical
13	Type of admission	Type of admission (*Emergency/Trauma/Urgent*)	nominal categorical
14	Illness Severity	Illness severity **at admission**	ordinal categorical
15	Patient Visitors	**Not defined in [1] Presumably the number of visits (not visitors) the patient had**	integer
16	Age	Age category of the patient	ordinal categorical
17	Admission Deposit	Deposit made at admission	integer
18	*Stay Days*	Number of days spent at the hospital	ordinal categorical

Table 4.1: Dataset Variables Summary

case id	Hospital	H. type	H. city	H. region	Available Extra Rooms	Department	Ward Type	W. Facility
1	8	2	3	2	3	radiotherapy	R	F
2	2	2	5	2	2	radiotherapy	S	F
3	10	4	1	0	2	anesthesia	S	E
4	26	1	2	1	2	radiotherapy	R	D
5	26	1	2	1	2	radiotherapy	S	D
6	23	0	6	0	2	anesthesia	S	F

Table 4.2: First nine columns of the dataset

(and create) various data science competitions, publish datasets, create and fit models and discuss various topics related to ML and Data Science in general.

It should be noted that since the dataset is public, it does not contain any clinical or symptom-related data, which is typically found in medical datasets, especially the ones that were already used to predict the LoS of COVID-19 patients. This is one of the challenges this dataset presents w.r.t. fitting the models, since typical factors that predict the LoS are indeed found in clinical hospital records, which is not the case with this dataset. Some common clinical features, that are missing, are for example: *fatigue*, *fever*, *headache* or *measurements from blood samples (e.g. antibody counts)*.

The original dataset [1] contains 318 438 observations of 18 variables, including the response (Length of hospital stay). The summary of variable types and their semantics are summarized in Table 4.1.

The first six observations of the dataset can be seen in the Tables 4.2 (first nine variables) and 4.3 (last nine variables).

Bed Grade	patientid	City Code Patient	Type of Admis-sion	Illness Severity	Patient Visitors	Age	Admission Deposit	Stay Days
2.00	31397	7.00	Emergency	Extreme	2	51-60	4911.00	0-10
2.00	31397	7.00	Trauma	Extreme	2	51-60	5954.00	41-50
2.00	31397	7.00	Trauma	Extreme	2	51-60	4745.00	31-40
2.00	31397	7.00	Trauma	Extreme	2	51-60	7272.00	41-50
2.00	31397	7.00	Trauma	Extreme	2	51-60	5558.00	41-50
2.00	31397	7.00	Trauma	Extreme	2	51-60	4449.00	11-20

Table 4.3: Last nine columns of the dataset

4.3 Exploratory Data Analysis

This section contains the exploratory analysis of the dataset at hand. An overview of the statistical properties of each variable is firstly presented. The visualizations of the response variable as well as of certain covariates of interest are then provided. Furthermore, the correlation matrix is plotted in order to gain insight into the dependence between the variables, especially between the response and the predictors. The variables with the highest correlation with the response will be plotted against the response variable.

4.3.1 Statistical Properties

A selection of basic statistical characteristics for the numerical variables of the dataset is provided in Table 4.4. Respectively, the Table 4.5 contains the value counts of each categorical variable in the dataset. These two tables along with variable visualization in the next subsection offer a rough overview of the value distributions in the respective columns (variables).

Statistic	Available_Extra_Rooms_in_Hospital	Patient_Visitors	Admission_Deposit
Min.	0.0	0.0	1800
1st Qu.	2.0	2.0	4186
Median	3.0	3.0	4741
Mean	3.2	3.3	4881
3rd Qu.	4.0	4.0	5409
Max.	24.0	32.0	11008

Table 4.4: Statistical characteristics of numerical columns

Hospital_type	Hospital_region	Department	Ward_Type	Ward_Facility	Bed_Grade	Type.of.Admission	Illness_Severity	Age	Stay_Days
0:143312	0:133224	anesthesia : 29647	P: 5046	A: 27906	1: 26505	Emergency:117624	Minor : 56682	0-10 : 6254	0-10 :23602
1: 68946	1:122427	gynecology :249387	Q:106125	B: 35156	2:123671	Trauma :152200	Moderate: 85850	11-20 :16763	11-20 :78120
2: 45928	2: 62674	radiotherapy : 28506	R:127875	C: 35462	3:110583	Urgent : 48501	Extreme :175793	21-30 :40828	21-30 :87454
3: 20389		surgery : 1201	S: 77793	D: 51809	4: 57566			31-40 :63613	31-40 :55137
4: 24770		TB & Chest disease: 9584	T: 1477	E: 55351				41-50 :63716	41-50 :11735
5: 10703			U: 9	F:112641				51-60 :48497	51-60 :35005
6: 4277								61-70 :33681	61-70 : 2740
								71-80 :35784	71-80 :10250
								81-90 : 7887	81-90 : 4837
								91-100: 1302	91-100: 2764
									>100 : 6681

Table 4.5: Value counts of categorical columns

4.3.2 Visualization of the variables

In the following Figures (4.1, 4.2, 4.3 and 4.4) the histograms which depict the distribution of each variable are plotted in order to gain a better insight into the distribution of the respective variables on the one hand, and on the other hand to establish if the classes of the response variable should be balanced out.

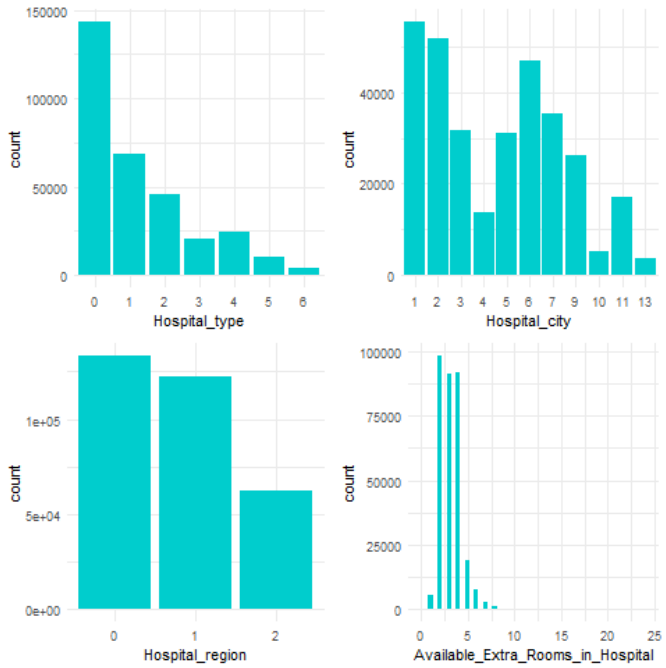

Figure 4.1: Histograms (distributions) of the Hospital Type, Hospital City, Hospital Region and Available Extra Rooms variables

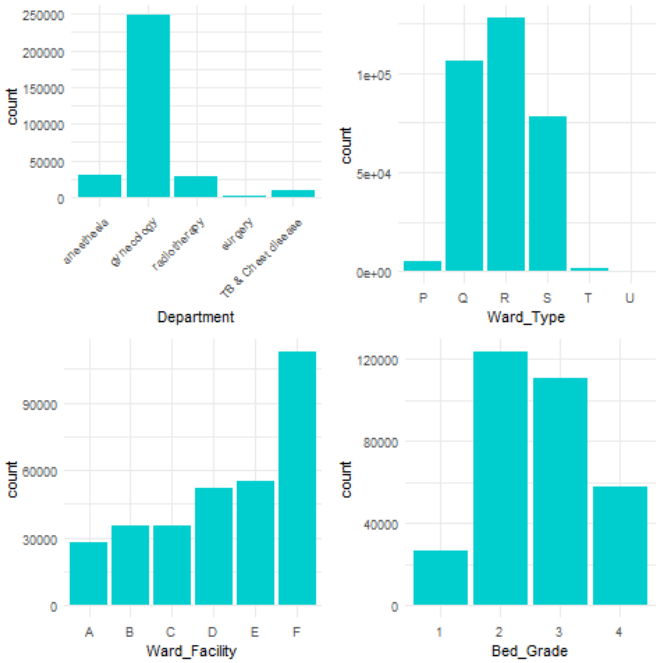

Figure 4.2: Histograms (distributions) of the Department, Ward Type, Ward Facility and Bed Grade variables

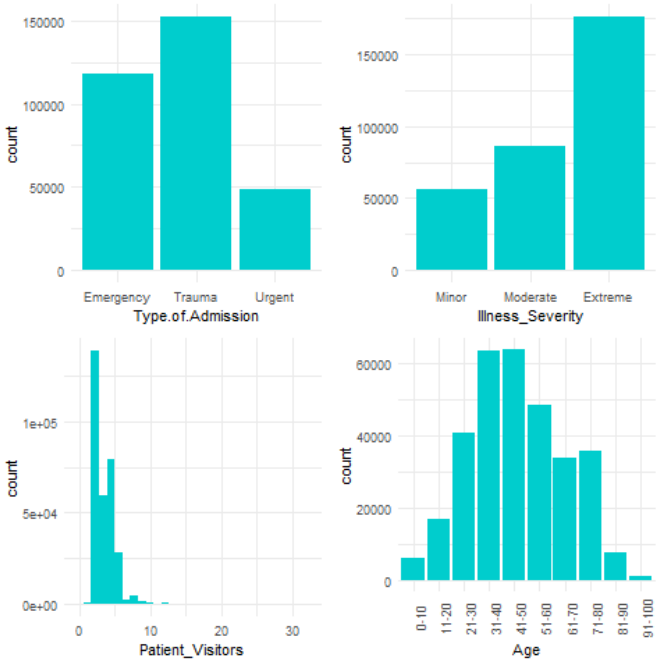

Figure 4.3: Histograms (distributions) of the Type of Admission, Illness Severity, Patient Visitors and Age variables

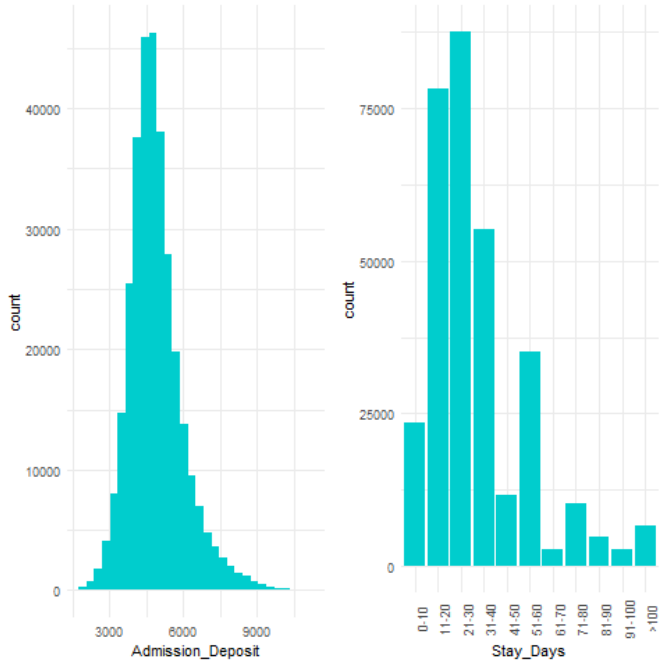

Figure 4.4: Histograms (distributions) of the Admission Deposit and Stay Days variables

4.3.3 Correlation Matrices

Correlation is one of the standard tools in the exploratory data analysis because it allows the analyst to determine the extent of the (typically linear) relationship between two variables. If the two variables are correlated it means that they both carry similar information. On the one hand, this can be useful if one of the variables is the response and the goal is to predict it as accurately as possible, on the other hand, if both variables belong to the feature variables and carry approximately the same information, one of them is usually redundant and can be dropped, which is generally done in Dimensionality-Reduction-Step.

In the following, four correlation matrices will be plotted. Firstly, the correlation matrix of the "raw" data will be plotted, where the relationship between all variables will be investigated. Further, the variables are separated by their types: numeric (continuous) and categorical (nominal and ordinal). The correlation matrix of the variables of the same type is then plotted.

Correlation matrix of the raw data transformed to numericals

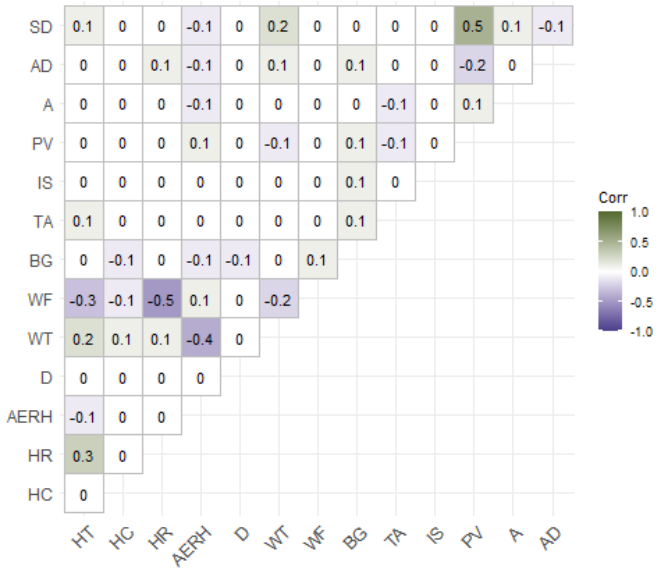

Figure 4.5: Correlation matrix of the raw data, where all variables were cast to numerical type

All variables of the dataset were transformed into numericals and the correlation matrix was calculated. Since the nominal categorical variables do not carry any ordinal information, their transformation to numericals is pointless. Hence this process is quite suboptimal in terms of determining the relationships between the variables. Therefore in the following, the variables of different types are considered separately. Large absolute correlations can be seen for the variable pairs: `Ward_Facility / Hospital_Region`, which is not of interest for this thesis, and `Patient_Visitors / Stay Days`. The latter pair will be plotted separately in the next section.

Correlation matrix of the numerical variables

Since there are only three numerical variables in the dataset, the correlation matrix shown in Figure 4.8 is relatively small. Furthermore, it can be observed that there are no significant correlations between the three variables.

Correlation matrix of the categorical variables

The categorical variables will in the following be further divided into *nominal* and *ordinal* categoricals.

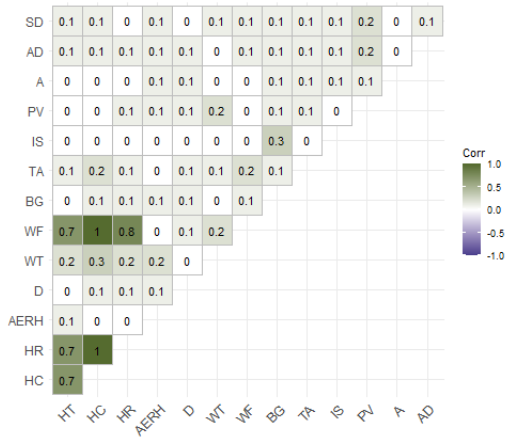

Figure 4.6: Correlation matrix of the categorical variables calculated with Cramer's V

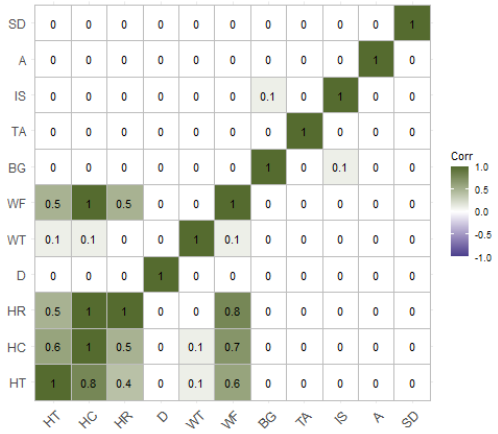

Figure 4.7: Correlation matrix of the categorical variables calculated with Uncertainty Coefficient

The Figure 4.6 depicts the correlation coefficients between the categorical variables calculated with the Cramer's V [117]. As stated by [118] this might introduce the problem with losing prediction asymmetry. This is because for each pair of variables (x, y), the predictive performance is not equal when one predicts x from y and y from x. In order to address this problem *Uncertainty Coefficient* [119] was used in the second figure (Figure 4.7).

Both figures show that there is a considerable degree of correlation between the pairs of `Hospital_*` variables, as well as between the `Ward_Facility` and `Hospital_*` variables, especially `Ward_Facility/Hospital_City`. Since they are both covariates in the models developed, these high correlations can only indicate that certain variables can be removed and the model performance will not be affected by a significant amount.

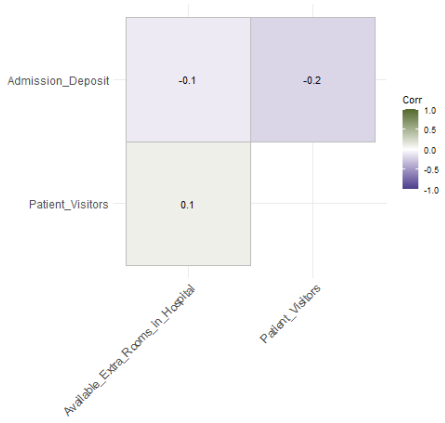

Figure 4.8: Correlation matrix of the numerical variables

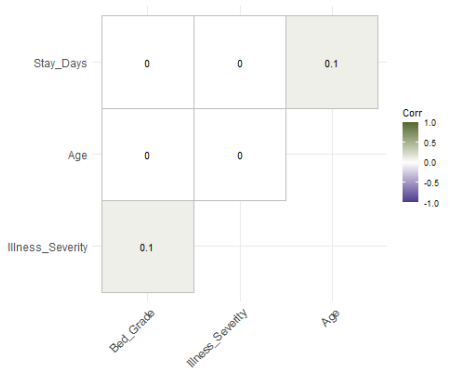

Figure 4.9: Correlation matrix of ordinal categorical variables calculated by mapping them to $\mathbb{R}$

Finally, the Figure 4.9 illustrates the correlation matrix of ordinal categoricals. The variables were firstly mapped to $\mathbb{R}$ with the "identity" function: $m(l_i) = i$ where l_i is the i-th category and the ordering rule is $l_i < l_j$ iff $i < j$. Then the correlation coefficients were calculated with the Pearson method [120].

From the correlation matrix shown in Figure 4.9 it can be concluded that the ordinal categorical variables are not linearly correlated. An additional examination has been performed using the Spearman Rank method [121] to test the monotonic dependency between these variables, which yielded very similar results.

4.3.4 Associations between the response and other variables of interest

In this subsection, the relationships between the response and the variables which could be efficient predictors will be examined and visualized. After the respective figure, a brief explanation and discussion will be provided.

Patient Visitors vs Stay Days

Since it has been established that a slight correlation exists between the variables `Patient_Visitors` and `Stay_Days`, the observations of the two variables have been plotted in the Figure 4.10.

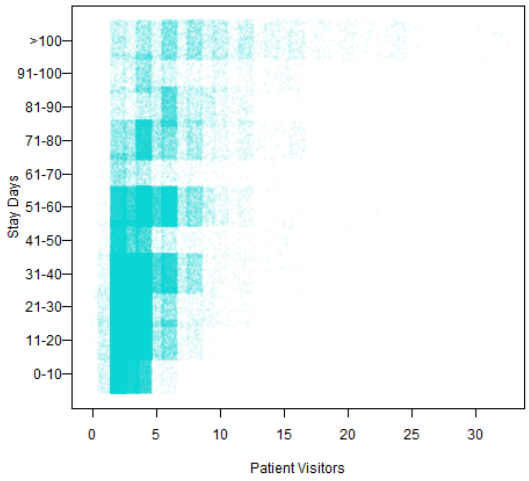

Figure 4.10: Patient Visitors vs Stay Days (jittered)

The high concentration of observations in the lower left quadrant is a direct consequence of the (right) skewness that both variabless exhibit. As a potential explanation for the slight correlation between the response and the Patient Visitors variable one could assume that patients who are hospitalized longer typically have a higher number of visits. The extent to which this variable could be used for predicting (Variable Importance) will be discussed in the Subsection 5.1.2.

Bed Grade, Age and Illness Severity, Ward Type vs Stay Days

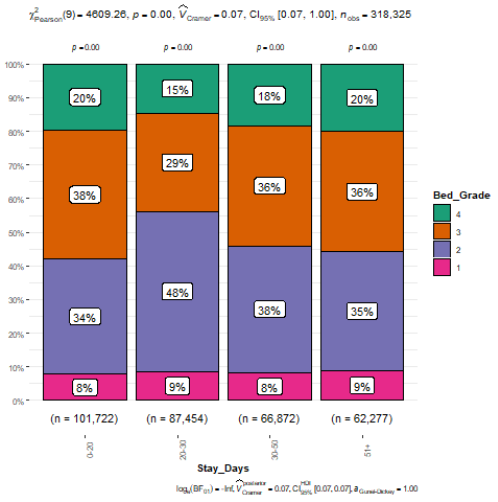

Figure 4.11: Relationship between Stay Days and Bed Grade

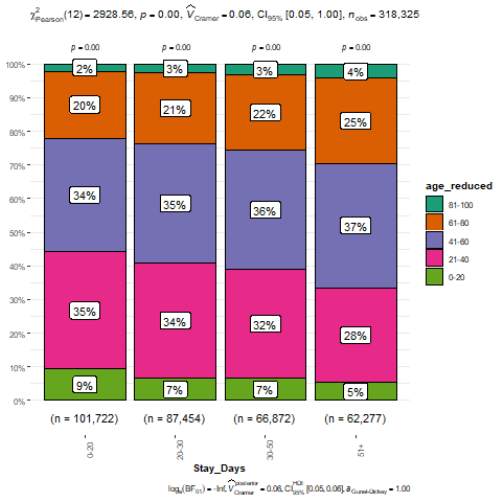

Figure 4.12: Relationship between Stay Days and Age

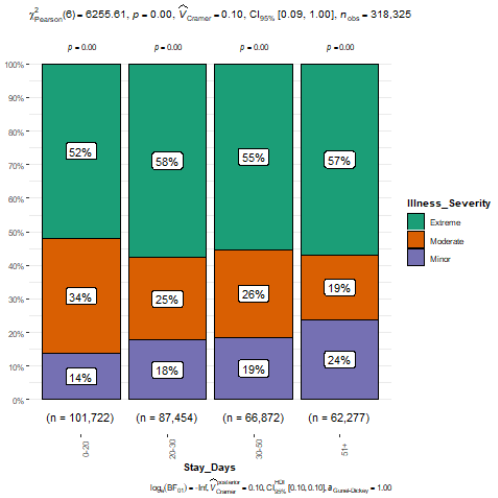

Figure 4.13: Relationship between Stay Days and Illness Severity

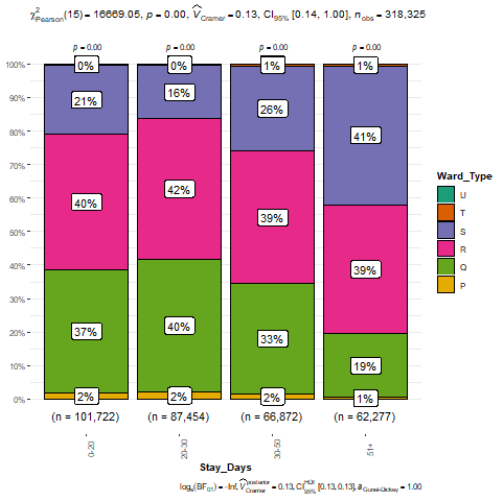

Figure 4.14: Relationship between Stay Days and Ward type

Figures 4.11, 4.12 4.13, and 4.14 have the four classes of stay days variable (see the Balancing of the response variable in the next Section) on the x-axis. Further for each of the three variables `Bed_Grade`, `Age`, `Illness_Severity` and `Ward_Type` the plots are made as follows: Observations of one response class are represented by one bar. Each of the bars is then divided with respect to the proportions of classes of the second variable. These plots allow the association of the class distributions of two categorical variables

to be more visible, which in turn enables the researcher/practitioner to obtain a clearer insight into the data at hand. Since the variables `Age` and `Stay_Days` have a larger amount of classes, they have been reduced before plotting in order to obtain a clearer chart.

The plot shown in the Figure 4.11 illustrates that the class distribution of the `Bed_Grade` variable is similar for almost all classes of `Stay_Days`. In the case of the second response class, a difference in the distribution is visible. Namely, almost half of the beds assigned to patients who were staying in the hospital for 20-30 are of the bed grade 2.

In the case of Figure 4.12, a decline in the proportion of people that are 0-40 years old is visible as the length of stay increases. The reverse can be said for the rest of the age classes (40-100 years). In other words, a tendency can be observed that out of the patients who stay the longest, the majority are over 40 years of age. This proportion drops as one moves to the shorter LoS categories and the proportion of people that are under the age of 40 increases.

Furthermore, with regard to Figure 4.13, it can be observed that the majority of cases demonstrate Extreme illness severity. However, the increase in the length of stay does not indicate an increase in the illness severity, but quite the opposite: the proportion of patients with minor illness severity rises along with the LoS, while the proportion of patients with the Extreme illness severity remains constant. One possible explanation for this could be that the illness severity was recorded at admission time and not in the course of hospital stay or at discharge or at the time of death. Therefore those patients who have transitioned from minor to moderate and in some cases to extreme illness severity during their stay have been assigned only the "minor illness severity" class.

Lastly, the comparison of the Ward Type and the response variable is provided in Figure 4.14. A trend is visible in the distribution of the ward type classes among the observations of individual response classes as one moves from the first to the final Stay days class. This characteristic could be very useful for fitting the classification models as it will be observed later on.

4.4 Implementation

This section contains a detailed description of every step included in the *data preprocessing stage*. All of these steps were applied to the dataset before feeding the data to the models. Then, the details regarding the model fitting will be discussed and the manner in which the models were trained and evaluated will be presented.

4.4.1 Data Preprocessing

Each step of the data preprocessing phase will now be separately described and the justification of each will be provided:

Removing Unnecessary Variables

`case_id` and `patient_id` contain the identification number of the case and the patient and they contain no relevant information, hence these columns were dropped.

As mentioned by Pargent et al. [122], a large number of classes in the predictor variables typically adds additional, unnecessary complexity to the models and poses additional difficulties for the models. Therefore, the variables `Hospital` and `City_Code_patient`, which have a larger number of categories (32 and 37 respectively), have been removed. Instead of removing them, it was also possible to encode them. A comprehensive comparison of different encoding techniques for high-cardinality features is also provided in [122].

Lastly, the variable `Hospital_city` was removed due to its high correlation with other Hospital variables. Since it is highly correlated with certain variables it conveys almost identical information, hence it is redundant.

Removing observations containing NA (null) values

It has been noticed that there are undefined (NA) values in the dataset. The majority of these values are contained in the *City Code Patient* variable (4532 observations). The remaining NA values are located in the *Bed Grade* variable (113 observations). These values are generally hard to incorporate into the modeling process and there are two common ways to deal with missing values: 1) Imputation and 2) Elimination. Since the dataset already contains a considerably large amount of observations, the latter approach will be employed and the observations containing missing values removed.

Balancing the response variable

The response variable (Stay Days) is relatively unbalanced, as it can be observed from the variable distribution depicted in Figure 4.15. This imbalance can introduce certain biases into the models because some classes are more common than others. In order to address this, certain response classes will be merged. The histogram of the response variable after the merging of the classes can be seen in Figure 4.16.

Since the response class '>50' is more important for the hospital managers than the other classes, the response variable will further be reduced to only two classes '<50' and '>50' days. The models will be trained both with this final reduction and without it. This will considerably relieve the two-class models[1] and allow easier classification because the number of response classes is reduced.

Due to the imbalance between the classes '<50' and '>50' the binary classification models did not perform as well. The negative effect of an imbalanced response was especially visible with the neural network model, where high accuracy was achieved merely by

[1]It is worth noting that two-class models will be used synonymously with two-class response models, binary classification models and binary response models. The same holds for four-class models.

classifying all observations as '<50'. As a consequence of this classification strategy, the sensitivity and precision of the '>50' class were zero, which indicates the worst-possible performance w.r.t. the main use-case this thesis describes. To address this, an additional step was added for the purpose of introducing the balance between the two classes. The datasets used for fitting (only) the binary classifiers were under-sampled. The *Near Miss* under-sampling strategy (implemented in the Python library `imblearn` by [123]) was used for the neural network model and the *Cluster-based* under-sampling approach (implemented in the R package `IRIC` by [124]) for the RF model.

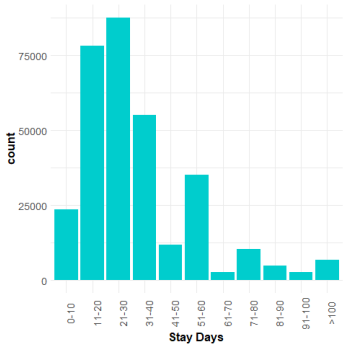

Figure 4.15: Original distribution of the stay days variable

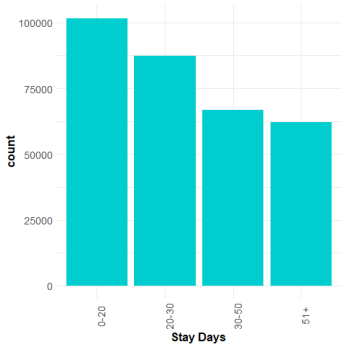

Figure 4.16: Distribution of the stay days variable after merging

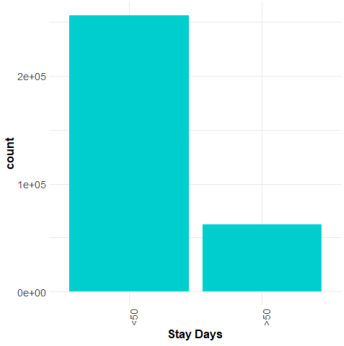

Figure 4.17: Distribution of the stay days variable after merging it into only two categories (*<50* and *>50* days)

Transforming the strings and integers into categoricals

Since certain variables are stored as *integers* or *strings* but semantically represent categorical variables (ordinal/nominal), these are transformed into the correct datatype. In the language R, this datatype is named *factor* and it can be initialized as an ordered or an unordered (nominal) factor. In Python, on the other hand, the respective datatype is called categorical. However, this only denotes the nominal categorical variables. In order to assign an ordinal categorical data type to a variable in a pandas data frame, one needs to create an object of the class `CategoricalDtype` from the pandas API and initialize it with the classes and the `ordered` flag set to `True`.

Encoding the categorical predictors and the response

The categorical predictor variables are encoded by the procedure described in the following. It should be noted, however, that these encodings are exclusively used for the data that is fed to the Neural Network models. Random forest models can easily handle the categorical variables and transforming them in this way could lead to prolonged training time (especially the OneHot Encoding).

1. Nominal Predictors - OneHot Encoding : If a variable has k different classes, k

new variables are created with possible values 0 or 1 s.t. the value the observation has for the m-th of these k variables is equal to one iff the observation had the m-th class in the original variable and is equal to zero otherwise. For example if the variable A has classes x, y and z, three new variables are created: Ax, Ay and Az. If an observation had class y as the value of the original variable A, the new encoding is $Ax = 0$, $Ay = 1$, and $Az = 0$

2. Ordinal Predictors - Label (ordinal) Encoding : The ordinal categories are transformed into integers from 1 to k where k is the number of classes (mapped to $\mathbb{Z}$)

The response variable is only transformed for the NN approach and the transformation is already described in Section 3.3.

After the encodings, the initial variables are, of course, removed from the dataset.

Splitting the data into train and test sets

The train-test ratio used was 80-20, meaning that 80% of the observations were used for training and 20% for the evaluation of the model performance.

Scaling the data

Before feeding the data into the NN model it has been centered and scaled column-wise. In particular, the mean of each column is equal to zero and the variance of each column is equal to one.

Randomization and Reproducibility

Since random forests rely on randomization quite heavily, the random numbers used for their construction need to be the same in order to achieve the reproducibility of the research. In each of the notebook files, where the implementation was written, the random seed was always manually set at the beginning to ensure reproducibility. This is achieved through the utilization of the "seed-setting" functions: `torch.seed()` [Python] and `set.seed()` [R] which guarantee that the algorithms in question will draw the same random numbers in the same order. One model (ordFor four-class) was retrained with another seed, which resulted in a slight decrease in the model performance ($\sim 10\%$ Accuracy, $\sim 1\%$ Sensitivity, and $\sim 5\%$ Precision).

4.4.2 Application of each algorithm

The specifics of method calls and training of the models will now be described separately for the RFs and NNs. After that, the chosen metrics are named and the reasoning behind this choice is provided.

Random Forests

The random forest approach has been implemented in R. The package used is called `ordinalForest` and was developed by the authors of the original paper [4].

The function call, for training the RF is given in the following listing:

```
ordfor(depvar = "Stay_Days_reduced", data=training_set, nsets=100, nbest=10,
                          ntreefinal = 500, ntreeperdiv = 100,
                          importance="rps")
```

Listing 4.1: Ordinal Forests function call

The arguments of the function call depicted in the Listing 4.1 are interpreted as follows [125]:

1. `depvar` : is the dependent (response) variable, which is being predicted
2. `data` : dataset dedicated for training (training set)
3. `nsets` : number of score sets used for determining the optimal score set
4. `nbest` : number of *best* score sets used to determine the optimal score set
5. `ntreefinal` : number of trees constructed in the final step (the final forest)
6. `ntreeperdiv` : number of trees constructed in the smaller RFs (one forest per score set)
7. `importance` : VIM used, ('rps' denotes ranked probability score and was used in the final model).

The naive RF solution was calculated with the function `ranger` belonging to the R package of the same name [115]. The number of trees grown is by default 500 and their depth is unlimited. In order to make the results comparable, the same number of trees is contained in the ordinal forest model as well.

Neural Networks

On the other hand, the neural network methodology analysis has been implemented in Python with the use of `PyTorch`, `Pytorch-Lightning` and `CORAL`[3]. packages. A multi-layer perceptron architecture was used, as described in [3] and [112] (see Section 3.3 for more details). Multiple models were fitted, each containing 36, 36, 36, 36, and 18 hidden layers respectively. The final layer is a so-called CORAL layer with the sigmoid activation function as described in [3].

Naive solution for NNs is implemented with the `PyTorch Lightning` library in Python as well. The number of hidden layers remains the same, but the final layer is not a CORAL layer but a simple linear layer with cross-entropy (+argmax) activation function.

Both CORAL and naive models (four class and two class) have been trained with the *Batch size* of **1024** observations, *learning rate* equal to **0.025** and **15** *epochs*. All four models also contain the same hidden layer structure: **[36, 36, 36, 36, 18]**. Meaning they have 36 neurons in the first four hidden layers and 18 in the final *hidden* layer. The final layer is then added depending on the methodology (naive or CORAL).

Evaluating the models with different performance metrics

After the model training, each of the models will be evaluated with the following metrics: Accuracy, mean AUC, unweighted, and linear weighted Kappa. Since the most important response class for the medical staff and ICU management is the '>50 days' class, the additional metrics examined *specially for this class* will be sensitivity and *precision*. These metrics have been briefly described in the Subsection 2.3.6.

4.5 Findings

The performance of the models presented in this section is expressed in both tabular and graphical forms. Since the models will be compared later on (in the Chapter 5) the models were grouped by certain criteria. The results of the models of the same "group" are summarized together to make them easier to read and interpret. The model groups are:

- Random Forest models (Both naive and OrdFor, fitted with four-class and two-class response)
- Neural Network models (Both naive and CORAL, fitted with four-class and two-class response)
- Optimized models only (Ordinal Forest and CORAL, fitted with four-class and two-class response)

The models were assessed through the following performance metrics: *Accuracy, Sensitivity of the '>50 days' class, Precision of the '>50 days' class, AUC, Unweighted* and *Linear Kappa*.

Since the RF models provide insights about *variable importance*, this information is also visualized in the final part of this section.

4.5.1 Random Forests

Table 4.6 contains the results of the four RF models, which were fitted, two naive (classic) RFs and two Ordinal Forests, each with four and two response classes respectively. The most preferable value of each metric (column-wise) is highlighted in bold. The visual representation of these results is presented in Figure 4.18.

Model	Accuracy	Unweighted Kappa	Linear Weighted Kappa	Sensitivity of '>50' class	Precision of '>50' class	AUC
Classic RF	0.72	0.61	0.68	0.81	**0.85**	0.83
Classic RF 2C	**0.86**	**0.73**	**0.73**	**0.89**	**0.85**	**0.86**
OrdFor	0.61	0.46	0.60	0.85	0.67	0.81
OrdFor 2C	0.85	0.70	0.70	0.86	**0.85**	0.85

Table 4.6: Performance of the Random Forest models, evaluated with different metrics

Model	Accuracy	Unweighted Kappa	Linear weighted Kappa	Sensitivity of '>50' class	Precision of '>50' class	AUC
CORAL NN	0.48	0.29	0.45	0.57	0.69	0.64
Classic NN	0.51	0.33	0.48	0.70	0.64	0.66
CORAL NN 2C	**0.84**	**0.68**	**0.68**	**0.85**	0.83	**0.84**
Classic NN 2C	**0.84**	0.67	0.67	0.80	**0.86**	**0.84**

Table 4.7: Performance of Neural Network models, evaluated with different metrics

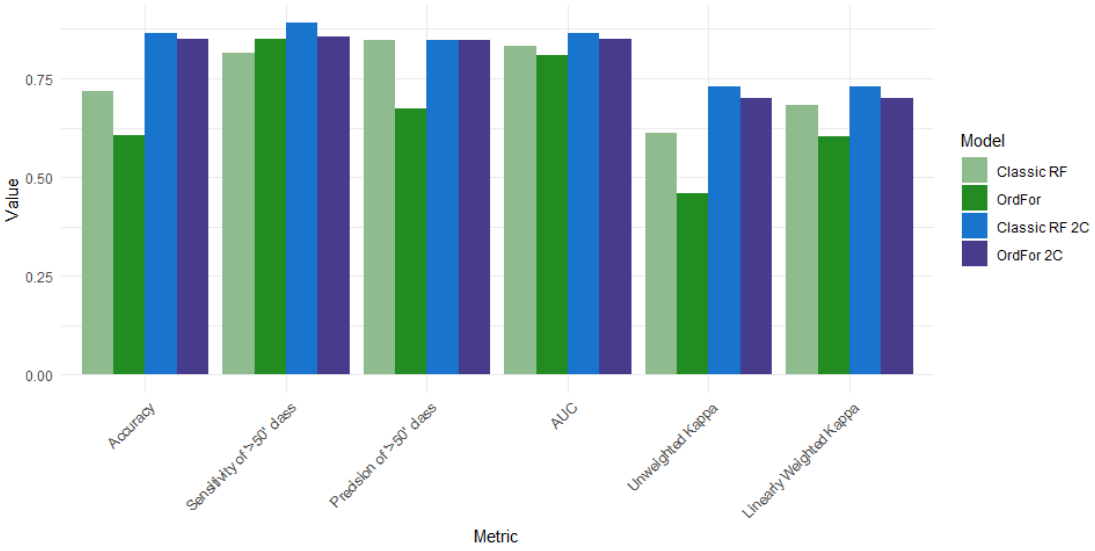

Figure 4.18: Comparison of different metrics for **Random Forest** Models

4.5.2 Neural networks

The performance of Neural Network models is presented in Table 4.7. The most favored values are, again, in bold font. A visualization of this table is provided in the Figure 4.19, where bar groups represent the metric and different bar colors represent different models (see legend of the plot).

Model	Accuracy	Kappa	Linear weighted Kappa	Sensitivity of '>50' class	Precision of '>50' class	AUC
OrdFor	0.61	0.46	0.60	0.85	0.67	0.81
CORAL NN	0.48	0.29	0.45	0.57	0.69	0.64
OrdFor 2C	**0.85**	**0.70**	**0.70**	**0.86**	**0.85**	**0.85**
CORAL NN 2C	0.84	0.68	0.68	0.85	0.83	0.84

Table 4.8: Performance of Ordinal forest and CORAL models (both four-class and two-class classifiers), evaluated with different metrics

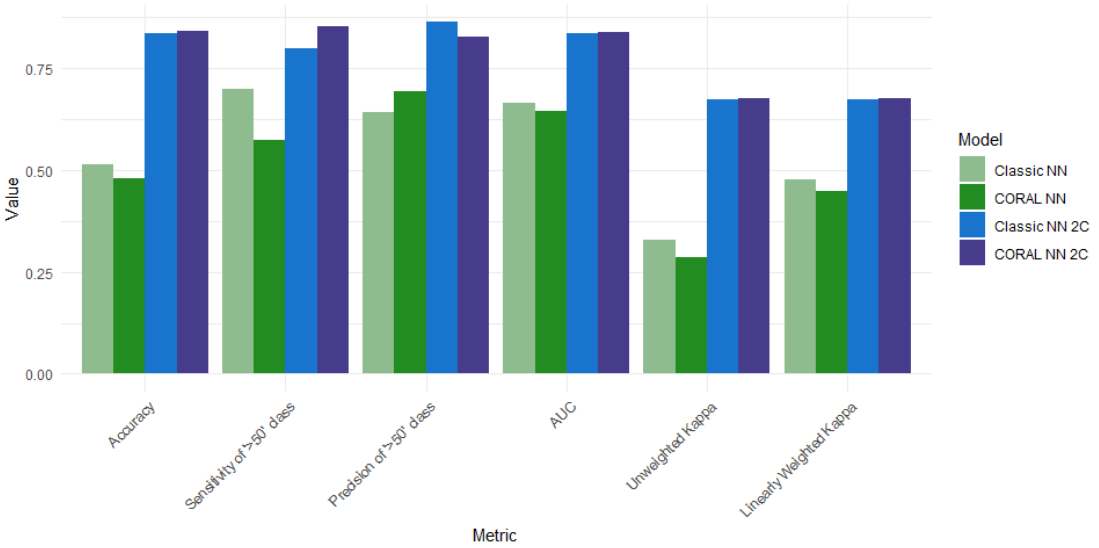

Figure 4.19: Visual comparison of different performance metrics for **Neural Network** Models

4.5.3 Ordinal Forests vs CORAL

The performance evaluations of OrdFor and CORAL models (fitted with both four-class and two-class responses) are depicted in the Table 4.8. The most preferable value for each metric is represented in bold font.

The visual results are split by the number of classes in the response for better readability. Figure 4.20 illustrates the performance of the naive and optimized (non-naive) neural network and random forest models, that were fitted with the response that contains four classes, across the above-defined metrics. The corresponding results for the different models that predicted the *two-class* response are visible in the Figure 4.21.

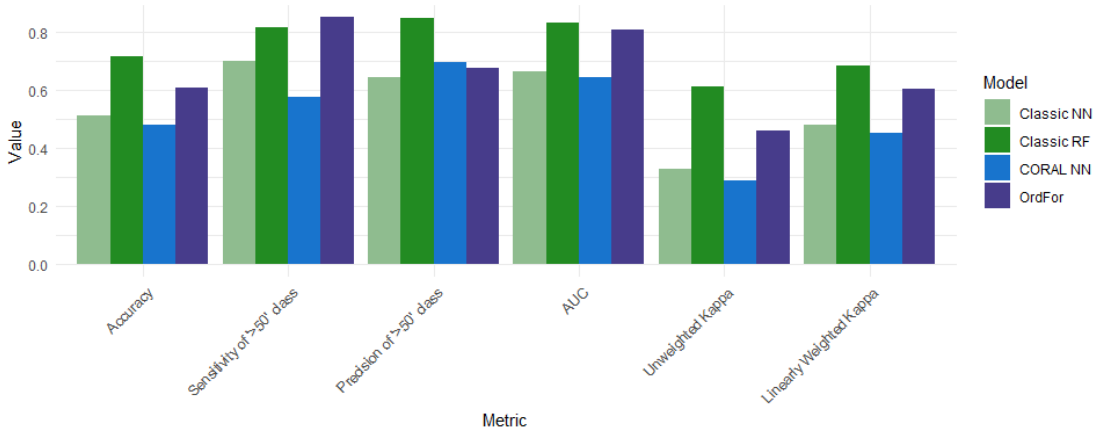

Figure 4.20: Comparison of different performance metrics for both Neural Networks and Random Forests (**naive and optimized models**) where the response contains **four** classes

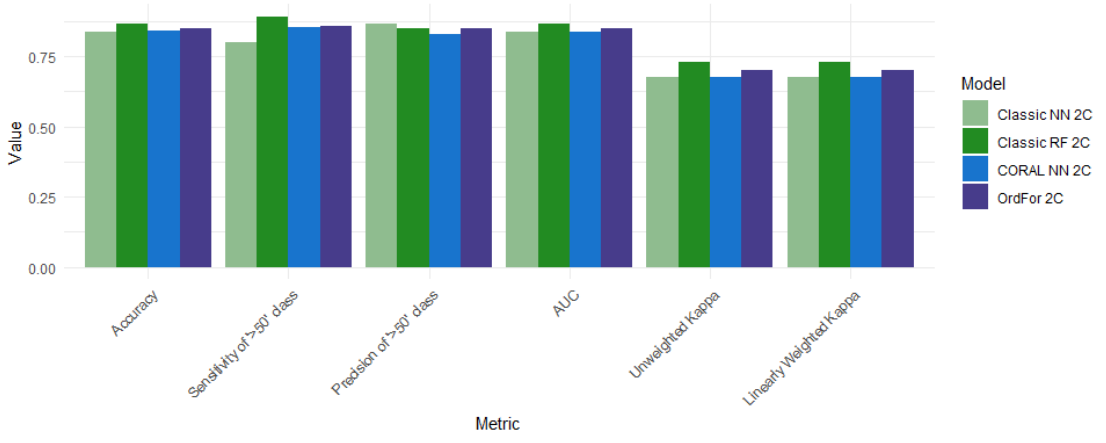

Figure 4.21: Comparison of different performance metrics for both Neural Networks and Random Forests (**naive and optimized models**) where the response contains **two** classes

4.5.4 Variable Importance found with Random Forest models

Since the default variable importance measure for the ordinal forest algorithm is the *Ranked Probability Score (RPS)*, another model was fitted by using the *accuracy* as a variable importance measure, for the sake of further exploration. This model exhibited slightly worse performance w.r.t. classification accuracy and other metrics, which was expected, since this accuracy, as a variable importance measure, disregards the ordering of the response variable [125]. Therefore, the VIM chosen for ordinal forests was *RPS*.

Variable importance measure used with the naive RFs is the *Gini Impurity* and in the case of the ordinal forest, the *Ranked Probability Score (RPS)* was used. Given that the RPS is used for ordinal forests and not accuracy, the two-class case produces NaN values for the variable importance, therefore it is not included in the plot. The results have been accumulated and are visualized in the Figure 4.22

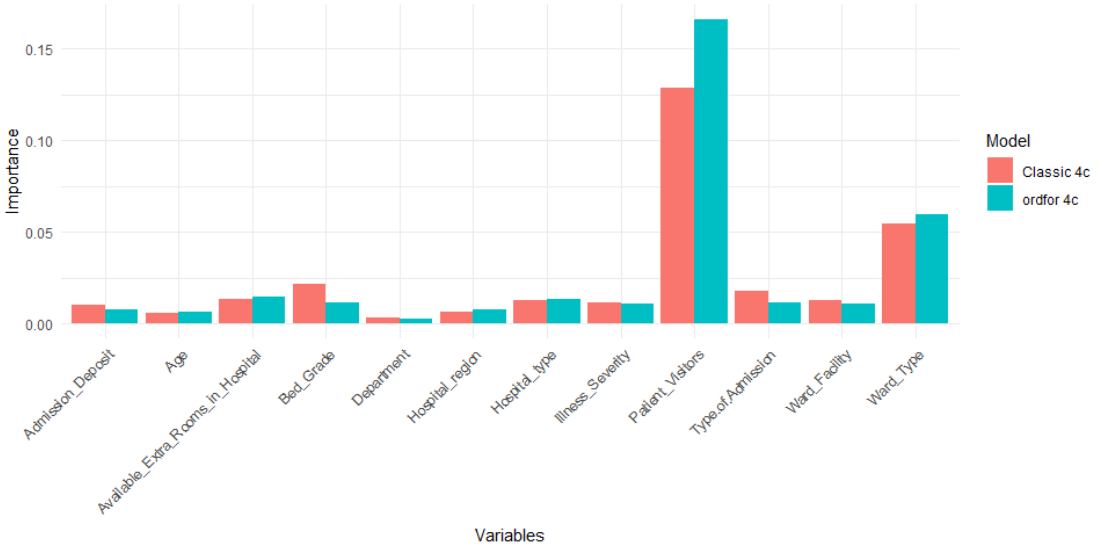

Figure 4.22: Variable importances obtained with the three RF models: Naive with four classes, Naive with two classes and Ordinal Forest with four classes

CHAPTER 5

Discussion and future work

The first part of this chapter contains a discussion of the results presented in the previous chapter. The results will, firstly, be interpreted from the ML perspective and an explanation will be provided to elucidate the causes of these findings. The models will be compared to one another based on the predefined metrics. After comparing the best-performing model developed in this thesis to other state-of-the-art models for predicting LoS of COVID-19 patients, the implications it has for the hospital management staff will be pointed out. The chapter will be concluded with a summary of the work carried out in this thesis along with a few remarks regarding the possible extensions of this work in form of suggestions for future research.

5.1 Discussion

This section encompasses a pairwise comparison of different model groups that were created based on multiple criteria. Firstly, the two-class response models will be compared to the four-class ones. Then the naive solutions will be compared to their "optimized" counterparts for both RFs and NNs. After that, the results of ordinal forest and CORAL models will be compared and discussed. For the final comparison, all models will be included and their similarities and differences will be examined. After comparing the algorithms with each other, the importance of variables produced by the ordinal forest and naive RF model will be discussed. Furthermore, the models of this thesis are compared to several models from the state-of-the-art literature on the matter of predicting hospital LoS. Finally, the statements about the capabilities of the best-performing model will be summarized based on the employed metrics with the focus on the viewpoint of the ICU management staff.

5.1.1 Comparison of the models

The above-described model groups will now be compared based on their performance, characteristics, training time, and benefit for the medical staff:

Two-class vs Four-class models

The most notable difference among all models is visible through the separation of the four-class case and the two-class case. The models which are fitted to predict a binary response variable achieve significantly better results w.r.t. *accuracy* in both RF and NN models. The same cannot be said for the majority of the remaining performance metrics in the case of RFs. On the other hand, the binary classification NN models outperform their "four-class" counterparts in all metrics. This is because the four-class model needs to differentiate between a larger number of classes, introducing more possibilities of wrong assignments and, consequently, lowering the probability of correct classification. Moreover, both categories of the two-class response have *identical* number of observations, since they have been undersampled to meet this criterion. This, however, does not hold for the four-class response, where one can observe the differences as large as almost 40k observations (between the '0-20' and '>50' class).

Naive RFs vs Ordinal Forests (Four class response)

The results indicate that the performance of the Naive RF and the Ordinal forest models differ substantially. In all metrics except Mean AUC and sensitivity of the '>50' class, the naive solution outperforms the ordinal forest model considerably. It can be observed that the Ordinal Forest model has reached only a slightly higher value of the '>50' class sensitivity and a slightly lower value of the AUC than its naive counterpart. The high values of AUC indicate that the model performs quite well at *distinguishing* between the classes and high values of sensitivity metric indicate that the ordinal forest model can identify around 85% of the patients who will stay in the hospital longer than 50 days, which could be considered beneficial for the medical practitioners.

Given that only the AUC and '>50' class sensitivity are similar between the ordinal forest and naive RF, and that in all other metrics the naive solution produces superior results, we can infer that the ordinal forest methodology does not provide any improvement over the naive solution in the context of the dataset and problem at hand. The authors of the paper where the Ordinal Forest algorithm was introduced [4] state that Ordinal forests achieve comparable and in some cases even worse performance than the naive nominal classification RFs, with the only exception being the weighted Kappa metrics, where the Ordinal Forests perform better. They also conclude that the Ordinal forests are better at predicting values *close* to the actual class, rather than exact values, which is opposite from the naive RFs. The authors also claim that Ordinal Forest models perform better if there is a bell-shaped distribution of the classes, which is not the case here since the classes have been merged to form uniform-like distribution.

It should, finally, be mentioned that training ordinal forests, comes with significantly increased training time, therefore, caution by model selection is advised.

Naive NNs vs CORAL (Four class response)

Similar can be said for the CORAL classifier. The naive solution produced slightly better results in almost all metrics except precision. The largest difference in the performance of the two models can be observed in the sensitivity metric, where the naive classifier reached 70% of correctly classified true positives, whereas the CORAL only succeeded to classify 57% of the true positives correctly. It is worthy of point out, however, that the CORAL classifier exhibited slightly higher sensitivity in the binary classification case than the naive approach. For further explanation of such model behavior please refer to the final paragraph of this subsection (*Comparison of all models*).

Having the CORAL classifier perform almost identically, and in some cases even moderately worse, it can be concluded that no significant improvement can be seen over the naive implementation when using the provided dataset. Moreover, the training times for both naive and the CORAL classifier are quite similar, which makes the choice of either model adequate.

Ordinal Forests vs CORAL

The Figure 4.20 indicates that the Ordinal Forest approach performs significantly better than the CORAL approach for the majority of the tracked metrics in the four-class case. The most significant superiority of the ordinal forest approach is reported with the metrics *sensitivity of '>50' class* and *AUC*, where the values attained by the ordinal forest model and the CORAL model differ by 28% and 17% respectively.

On the other hand, in the binary classification setting, both CORAL and ordinal forest models exhibit similar performance across all examined metrics. This was expected since the ordering information is not that relevant for binary classification/regression, which is also why these models perform comparably well to their naive counterparts in the binary response context.

Comparison of all models

The first noticeable difference, present across all results, is that the four-class models tend to have lower values of the observed metrics than the two-class models. Possible causes of this occurrence have already been explained in one of the previous paragraphs. Along with that, both naive and optimized two-class response models achieve almost identical performance across all metrics. As previously noted, this is due to the insignificance of the ordering information in the binary classification setting.

It can further be observed, that both Kappa values remain the same for the two-class case and that larger weighted kappa is observed in the four-class-response models. The equality of the metrics in the former can be explained by the fact that there are only two

response classes, therefore there is only one way to misclassify an observation and with that only one distance penalty termm which remains the same for both unweighted and weighted versions of the metric.

It is evident that both random forests do indeed perform better than neural network approaches. This could be a consequence of the dataset structure and how the categorical features are handled in each of the models. In particular, since the majority of features are categorical, the *splitting method* for deciding the class assignment (employed in decision trees) is a more "natural" way to handle and classify these observations. On the other hand, in order for a neural network to process these variables, they have to be encoded in integers or real numbers first. This shows that neural networks are more appropriate for datasets where the features are continuous (integer or real) variables.

Gutierrez et al. have established that it is challenging for the ordinal regression-based models to achieve significantly better performance than their naive equivalents for certain datasets [69]. This could be one further argument that the performance of the ordinal regression models heavily depends on the dataset, which is evident in this thesis as well.

One further point that should be highlighted is that the ordinal forest algorithm took significantly more time to train. This can be explained by the fact that this algorithm has a multitude of steps before fitting the final RF: it firstly computes the class width samples, fits multiple smaller RFs, and evaluates them. Having this many steps results in high time costs.

Finally, the findings indicate that the naive RF solution outperforms all other four-class response models w.r.t. almost every metric, with the only exception being the ordinal forest model that performed slightly better in terms of '>50' class sensitivity. Among all fitted models, with both binary and four-class responses, the naive two-class random forest prevails (although in some cases only by a small proportion). The only model that performs by $\sim 1\%$ better than the naive binary response RF in terms of '>50' class precision is the naive two-class NN. With that it can be concluded that the naive two-class RF is the best performing model developed in the course of this thesis.

5.1.2 Importance of variables

Findings in the Figure 4.22 indicate that both naive and the ordinal forest model identify variables `Patient_Visitors` and `Ward_Type` as very influential, with the emphasis on the `Patient_Visitors` variable.

The correlation coefficient (see Subsection 4.3.3) between the `Patient_Visitors` variable and the response could be viewed as the justification for the latter statement. As previously discussed, the longer the patients stay in the hospital, the more visits they receive. In spite of the high variable importance of this variable, it must be mentioned that the total number of visits is only observable after the patient has been dismissed, meaning that values of this variable cannot be a priori known and employed for predicting the LoS in a real-world scenario.

Furthermore, in the Subsection 4.3.4 it has been mentioned as well, that the value distributions of the `Ward_Type` variable across the response classes could play a role in the classification. In the Figure 4.14 one can observe that patients who were hospitalized for longer than 50 days are predominantly assigned to ward types 'S' and 'R', while patients who remained in the hospital shorter (e.g. 0-30 days) were mostly situated in wards of type 'R' and 'Q'.

Interestingly, the variables `Age` and `Illness_Severity` have relatively low importance ranking, even though one would expect them to be quite relevant for predicting the LoS.

5.1.3 Comparison to other studies

As described in the Section 3.1, Alabbad et al. develop four ML models with the same goal as the models developed for this thesis. The four model types are: RFs, Extreme Gradient Boosting (XGB), Gradient Boosting (GB), and Ensemble Classifier. The researchers aim to predict the hospital LoS as well, however, the data which is used to train and evaluate the models stems from a *single* Hospital in eastern Saudi Arabia and only has 895 entries. The authors have a similar problem description, where the response variable is an ordered categorical variable. Contrary to the research conducted in this thesis, the authors of that paper ignore the ordering between the classes and fit the models as typical classifiers. Since their response variable was also imbalanced, they used SMOTE oversampling strategy [126] in order to create uniform distribution between the classes (after this transformation there were 1296 observations). This number is still significantly smaller than the number of observations used for training the models presented in this thesis. On the other hand, the features contained in the dataset are more symptom-oriented and they contain clinical data (e.g. measurements from blood samples of each individual). Some of these features are highly correlated with the hospital length of stay [108], which makes it easier to achieve better model performance. Having access to a dataset of that kind could have been a crucial step in achieving high accuracy ($\sim$94%), precision ($\sim$93%), and recall ($\sim$93%).

The performance of the models developed by Alabbad et al. in their work [108] is comparable and even greater than the majority of other state-of-the-art results in the field of predicting the hospital LoS of COVID-19 patients.

Seven studies were mentioned in [108], which also predict the LoS. The majority of them are shaped as regression models. Two studies have reported their results with AUC: the model developed in [127] reaches the AUC of *0.89*, while the one presented in [128] reports slightly lower AUC of *0.84*. Given these two studies, all random forest models developed in this thesis achieve comparable performance w.r.t. the AUC. On the other hand, neural network models that achieve comparable performance are only the ones where the response contains two categories.

Furthermore, Alabbad et al. mention one study, apart from their own, where the performance of the models has been evaluated via Accuracy. The authors of [129] report attaining the accuracy of 80% with their models. In comparison to the four-class models

fitted in this thesis, that accuracy is relatively larger (by $\sim$ 19% [RF] and by $\sim$ 32% [NN]). On the other hand, the two-class response models have achieved similar and even slightly greater performance to that which was reported in [129]. An explanation for this could be identical as to why the binary classification models performed better than the four class models in general (please refer to the Subsection 5.1.1).

Finally, it should be mentioned that the dataset used for this thesis, unfortunately, does not include some of the relevant predictor variables identified by other researchers [107]. In the following, the argumentation is provided that the obtained results, can, nevertheless, be considered beneficial for the main use case: assisting the ICU staff.

5.1.4 Implications for the ICU staff

The following statements can be made regarding the capabilities of the best-performing model:

- The model was able to differentiate between the patients who are staying in the hospital for longer than 50 days and those who are staying for less than 50 days with the accuracy of **86%**.
- Among all patients who were hospitalized for longer than 50 days, the model was able to correctly identify **89%** of them. (metric: sensitivity)
- Out of all patients who were identified by the model as those who will stay for longer than 50 days, **85%** were correct predictions. (metric: precision)
- Based on the AUC, there is a **86%** chance of the model being able to distinguish between the classes. The quantity of discrimination between the classes recorded falls into the range between 0.8-0.9, which is considered *excellent* in the literature [130].
- According to the value of Cohen's Kappa (0.73), the agreement between the model and the actual classes is deemed **"substantial"** by [131].

5.2 Conclusion

The COVID-19 pandemic has placed a great burden on the medical staff in all medical fields. Due to the nature of the disease, the greatest pressure was observed by practitioners of the Intensive Care Unit. This department was the most susceptible to reaching its maximum capacity and it has proven to be the most important department for treating the COVID-19 patients across numerous hospitals. Due to the urgency of the matter at hand, the prompt delivery of the assistance and of the appropriate results was needed.

To provide the contribution to the current state of the research, this thesis considers a machine learning approach to predicting the length of hospital stay for COVID-19 patients. The response variable (LoS) of the chosen dataset is an ordinal categorical variable, which

required special algorithm variations in order to make performant predictions. The two approaches chosen were: 1) a variation of the random forest called *ordinal forest*, and 2) a variation of the neural network approach called *CORAL*.

The motivation and goals behind this thesis are firstly presented in the Chapter 1 along with the problem statement and the methodological approach. The theoretical foundation for the central topics of the work is then provided. Here COVID-19 and Long COVID are described as the medical element of the thesis and, on the other hand, Machine Learning Models, their development, and two common representatives, RFs and NNs are introduced as the technical aspect of the thesis. Having the basic concepts introduced, the foundations from the current state of the literature on the topics of *Predicting COVID-19 metrics* and *Ordinal Regression with RFs & NNs* are explained. These explanations offer a theoretical introduction to the methods which are used in the practical part to obtain the results. Afterward, the results of applying each of the two models in two different settings (four class and two class case) are presented in the Chapter 4. The models were evaluated using the predefined metrics and their performance was visualized in order to display the results in a more intuitive manner. Furthermore, the presented findings were discussed and it has been established that certain models have achieved comparable performance to those reported in the present literature and others were outperformed by their naive counterparts and that the performance of such models is heavily dependent on the dataset. The former was based on two metrics: Accuracy and AUC. Furthermore, the key metrics of the best-performing algorithm have been highlighted and briefly explained for the purpose of enabling the ICU management staff to quickly assess the performance of the best-performing model. Lastly, suggestions regarding future research on this topic are brought forward.

5.3 Recommendations for future work

Through this thesis, it has been shown that a great potential for supporting the ICU management staff lies in the machine learning field. As with any problem which is being solved with the Machine Learning paradigm, more quality data results in more reliable outcomes.

Therefore, as the first possibility of extending the research presented in this thesis and providing improved assistance to the medical staff, the choice of a more sophisticated dataset is recommended. The dataset should, for example, contain the relevant predictors of Long COVID, identified in [107], typically in form of symptoms, and possibly the number of stay days as an integer variable, which would allow the researcher to create pure regression models. Unfortunately, access to those kinds of datasets is typically restricted. As a consequence, the dataset used in this thesis is a public dataset, without many (public) high-quality alternatives.

Secondly, additional data-preprocessing steps could be considered. On the one hand, a simple transformation, which could produce minor improvements in the performance would be balancing the imbalanced categorical features (like `Age` for example). On the

other hand, more advanced data-preprocessing steps would, for instance, be encoding the variables with a high number of categories (`Hospital` or `City_Code_Patient`) instead of removing them. It has been proven by [122] that encoding presented by [132] performs quite well with high-cardinality features. It would, further, be possible to completely balance the four-class response variable, i.e. all categories would then have the same number of observations. This could be done either by over- or under-sampling (as it was performed for the two-class response).

Along with supplementary data modifications and utilization of other datasets, it would also be advised to employ other ML models for predicting the response variable. SVMs and Boosting (AdaBoost or XGBoost) algorithms could serve as additional model types to make the predictions about COVID-19 metrics, including LoS, more reliable. Employing varying strategies in this manner has the potential of, firstly, providing clearer insights into the structure of the prediction problem from the ML perspective, and, secondly, making a greater contribution to the central use-case of this thesis (Medical viewpoint).

Another potential improvement of the random forest and neural network models, in particular, could be achieved through comprehensive and fine-grained hyperparameter tuning. In the course of this thesis, only a basic hyperparameter search was conducted for both models due to time and computation limitations. Optimizing the hyperparameters further, would consume considerably more time and probably bring some advancements, especially in the case of the NN models. Nevertheless, it is speculated that these improvements would be insufficient to justify the additional temporal overhead needed to achieve them.

Lastly, medical practitioners and ICU management staff could be contacted and interviewed for the sake of a domain-based evaluation of the models. Through this procedure, potential flaws and improvement points could be identified for each of the models, which would bring these models closer to deployment and their application in practice.

List of Figures

List of Tables

Acronyms

ICU Intensive Care Unit

NN Neural Network

OR Ordinal Regression

RF Random Forest

SVMs Support vector machines

Bibliography

[1] Möbius, "Covid-19 hospitals treatment plan," accessed on 11.05.2022. [Online]. Available: https://www.kaggle.com/datasets/arashnic/covid19-hospital-treatment

[2] E. M. Rees, E. S. Nightingale, Y. Jafari, N. R. Waterlow, S. Clifford, C. A. B Pearson, T. Jombart, S. R. Procter, G. M. Knight *et al.*, "Covid-19 length of hospital stay: a systematic review and data synthesis," *BMC medicine*, vol. 18, no. 1, pp. 1–22, 2020.

[3] W. Cao, V. Mirjalili, and S. Raschka, "Rank-consistent ordinal regression for neural networks," *arXiv preprint arXiv:1901.07884*, vol. 1, no. 6, p. 13, 2019.

[4] R. Hornung, "Ordinal forests," *Journal of Classification*, vol. 37, no. 1, pp. 4–17, 2020.

[5] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.

[6] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[7] WHO, "Naming the coronavirus disease (covid-19) and the virus that causes it," https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it, February 2020, accessed on 23.04.2022.

[8] J. Feehan and V. Apostolopoulos, "Is covid-19 the worst pandemic?" *Maturitas*, vol. 149, pp. 56–58, 2021.

[9] Y. Shi, G. Wang, X.-p. Cai, J.-w. Deng, L. Zheng, H.-h. Zhu, M. Zheng, B. Yang, and Z. Chen, "An overview of covid-19," *Journal of Zhejiang University-SCIENCE B*, vol. 21, no. 5, pp. 343–360, 2020.

[10] Y.-C. Wu, C.-S. Chen, and Y.-J. Chan, "The outbreak of covid-19: An overview," *Journal of the Chinese medical association*, vol. 83, no. 3, p. 217, 2020.

[11] WHO, "Who coronavirus (covid-19) dashboard," https://covid19.who.int/table, accessed on 23.04.2022.

[12] ——, "Who director-general's opening remarks at the media briefing on covid-19 - 11 march 2020," https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020, March 2020, accessed on 23.04.2022.

[13] C. Wu, Y. Liu, Y. Yang, P. Zhang, W. Zhong, Y. Wang, Q. Wang, Y. Xu, M. Li, X. Li *et al.*, "Analysis of therapeutic targets for sars-cov-2 and discovery of potential drugs by computational methods," *Acta Pharmaceutica Sinica B*, vol. 10, no. 5, pp. 766–788, 2020.

[14] B. Hu, H. Guo, P. Zhou, and Z.-L. Shi, "Characteristics of sars-cov-2 and covid-19," *Nature Reviews Microbiology*, vol. 19, no. 3, pp. 141–154, 2021.

[15] WHO, "Zoonoses," https://www.who.int/news-room/fact-sheets/detail/zoonoses#:~:text=A%20zoonosis%20is%20an%20infectious,food%2C%20water%20or%20the%20environment., (Accessed on 04/24/2022).

[16] E. Team, "The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (covid-19)—china, 2020," *China CDC Weekly*, vol. 2, no. 8, p. 113, 2020.

[17] G. Gao, W. Liu, P. Liu, W. Lei, Z. Jia, X. He, L.-L. Liu, W. Shi, Y. Tan, S. Zou *et al.*, "Surveillance of sars-cov-2 in the environment and animal samples of the huanan seafood market," 2022.

[18] J. E. Pekar, A. Magee, E. Parker, N. Moshiri, K. Izhikevich, J. L. Havens, K. Gangavarapu, L. M. Malpica Serrano, A. Crits-Christoph, N. L. Matteson, M. Zeller, J. I. Levy, J. C. Wang, S. Hughes, J. Lee, H. Park, M.-S. Park, K. Ching Zi Yan, R. Tzer Pin Lin, M. N. Mat Isa, Y. Muhammad Noor, T. I. Vasylyeva, R. F. Garry, E. C. Holmes, A. Rambaut, M. A. Suchard, K. G. Andersen, M. Worobey, and J. O. Wertheim, "SARS-CoV-2 emergence very likely resulted from at least two zoonotic events," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6291628

[19] M. Worobey, J. I. Levy, L. M. M. Serrano, A. Crits-Christoph, J. E. Pekar, S. A. Goldstein, A. L. Rasmussen, M. U. G. Kraemer, C. Newman, M. P. G. Koopmans, M. A. Suchard, J. O. Wertheim, P. Lemey, D. L. Robertson, R. F. Garry, E. C. Holmes, A. Rambaut, and K. G. Andersen, "The Huanan market was the epicenter of SARS-CoV-2 emergence," Feb. 2022, corrected date in filename. [Online]. Available: https://doi.org/10.5281/zenodo.6299600

[20] J. S. Peiris, Y. Guan, and K. Yuen, "Severe acute respiratory syndrome," *Nature medicine*, vol. 10, pp. S88–97, 01 2005.

[21] C. S. Goldsmith, K. M. Tatti, T. G. Ksiazek, P. E. Rollin, J. A. Comer, W. W. Lee, P. A. Rota, B. Bankamp, W. J. Bellini, and S. R. Zaki, "Ultrastructural characterization of sars coronavirus," *Emerging infectious diseases*, vol. 10, no. 2, p. 320, 2004.

[22] R. Rana, A. Tripathi, N. Kumar, and N. K. Ganguly, "A comprehensive overview on covid-19: Future perspectives," *Frontiers in Cellular and Infection Microbiology*, vol. 11, 2021.

[23] G. Lippi, C. Mattiuzzi, and B. M. Henry, "Updated picture of sars-cov-2 variants and mutations," *Diagnosis*, vol. 9, no. 1, pp. 11–17, 2022.

[24] M. Meselson, "Droplets and aerosols in the transmission of sars-cov-2," *New England Journal of Medicine*, vol. 382, no. 21, pp. 2063–2063, 2020.

[25] N. Van Doremalen, T. Bushmaker, D. H. Morris, M. G. Holbrook, A. Gamble, B. N. Williamson, A. Tamin, J. L. Harcourt, N. J. Thornburg, S. I. Gerber *et al.*, "Aerosol and surface stability of sars-cov-2 as compared with sars-cov-1," *New England journal of medicine*, vol. 382, no. 16, pp. 1564–1567, 2020.

[26] Y. Wu, C. Guo, L. Tang, Z. Hong, J. Zhou, X. Dong, H. Yin, Q. Xiao, Y. Tang, X. Qu, L. Kuang, X. Fang, N. Mishra, J. Lu, H. Shan, G. Jiang, and X. Huang, "Prolonged presence of SARS-CoV-2 viral RNA in faecal samples," *Lancet Gastroenterol Hepatol*, vol. 5, no. 5, pp. 434–435, 05 2020.

[27] M. Ciotti, M. Ciccozzi, A. Terrinoni, W.-C. Jiang, C.-B. Wang, and S. Bernardini, "The covid-19 pandemic," *Critical reviews in clinical laboratory sciences*, vol. 57, no. 6, pp. 365–388, 2020.

[28] G. Kampf, D. Todt, S. Pfaender, and E. Steinmann, "Persistence of coronaviruses on inanimate surfaces and their inactivation with biocidal agents," *Journal of hospital infection*, vol. 104, no. 3, pp. 246–251, 2020.

[29] L. Dong, J. Tian, S. He, C. Zhu, J. Wang, C. Liu, and J. Yang, "Possible vertical transmission of sars-cov-2 from an infected mother to her newborn," *Jama*, vol. 323, no. 18, pp. 1846–1848, 2020.

[30] A. Alshahrani, "Leadership preparedness to combat the economic disaster caused by covid-19 crisis," 05 2020.

[31] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong *et al.*, "Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in wuhan, china," *Jama*, vol. 323, no. 11, pp. 1061–1069, 2020.

[32] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The lancet*, vol. 395, no. 10223, pp. 497–506, 2020.

[33] G. Nepal, J. H. Rehrig, G. S. Shrestha, Y. K. Shing, J. K. Yadav, R. Ojha, G. Pokhrel, Z. L. Tu, and D. Y. Huang, "Neurological manifestations of covid-19: a systematic review," *Critical Care*, vol. 24, no. 1, pp. 1–11, 2020.

[34] P. Zhai, Y. Ding, and Y. Li, "The impact of covid-19 on ischemic stroke," *Diagnostic Pathology*, vol. 15, no. 1, pp. 1–5, 2020.

[35] R. Wölfel, V. M. Corman, W. Guggemos, M. Seilmaier, S. Zange, M. A. Müller, D. Niemeyer, T. C. Jones, P. Vollmar, C. Rothe *et al.*, "Virological assessment of hospitalized patients with covid-2019," *Nature*, vol. 581, no. 7809, pp. 465–469, 2020.

[36] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei *et al.*, "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study," *The lancet*, vol. 395, no. 10223, pp. 507–513, 2020.

[37] Y.-J. Han, Z.-G. Ren, X.-X. Li, J.-L. Yan, C.-Y. Ma, D.-D. Wu, and X.-Y. Ji, "Advances and challenges in the prevention and treatment of covid-19," *International journal of medical sciences*, vol. 17, no. 12, p. 1803, 2020.

[38] D. van Riel and E. de Wit, "Next-generation vaccine platforms for covid-19," *Nature materials*, vol. 19, no. 8, pp. 810–812, 2020.

[39] WHO, "Covid-19 vaccine tracker and landscape," https://www.who.int/publications/m/item/draft-landscape-of-covid-19-candidate-vaccines, April 2022, accessed on 28.04.2022.

[40] N. I. for Health and C. E. (NICE), "Covid-19 rapid guideline: managing the long-term effects of covid-19," December 2020, accessed on: 04.05.2022. [Online]. Available: https://www.nice.org.uk/guidance/ng188

[41] A. Raveendran, R. Jayadevan, and S. Sashidharan, "Long covid: an overview," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 15, no. 3, pp. 869–875, 2021.

[42] A. Nalbandian, K. Sehgal, A. Gupta, M. V. Madhavan, C. McGroder, J. S. Stevens, J. R. Cook, A. S. Nordvig, D. Shalev, T. S. Sehrawat *et al.*, "Post-acute covid-19 syndrome," *Nature medicine*, vol. 27, no. 4, pp. 601–615, 2021.

[43] H. Crook, S. Raza, J. Nowell, M. Young, and P. Edison, "Long covid—mechanisms, risk factors, and management," *bmj*, vol. 374, 2021.

[44] A. Azevedo and M. F. Santos, "Kdd, semma and crisp-dm: a parallel overview," *IADS-DM*, 2008.

[45] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Emerging technology in modelling and graphics*. Springer, 2020, pp. 99–111.

[46] C. Shearer, "The crisp-dm model: the new blueprint for data mining," *Journal of data warehousing*, vol. 5, no. 4, pp. 13–22, 2000.

[47] D. S. P. A. DSPM, "What is crisp-dm?"

[48] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2020.

[49] J. Cohen, "Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. psychology," *Bulletin, 70, 213â*, vol. 220, 1968.

[50] H. E. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 2007.

[51] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

[52] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes; par AM Legendre...* chez Firmin Didot, libraire pour lew mathematiques, la marine, l . . . , 1806.

[53] C. F. Gauss, *Theoria combinationis observationum erroribus minimis obnoxiae*. H. Dieterich, 1823, vol. 2.

[54] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[55] ——, *Robust statistics*. John Wiley & Sons, 2004, vol. 523.

[56] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, 2020.

[57] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[58] Y. Dodge, *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[59] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.

[60] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[61] J. A. Hartigan, "Statistical theory in clustering," *Journal of classification*, vol. 2, no. 1, pp. 63–76, 1985.

[62] R. A. Fisher, "The statistical utilization of multiple measurements," *Annals of eugenics*, vol. 8, no. 4, pp. 376–386, 1938.

[63] P. Filzmoser, *Multivariate Statistics - Course Notes*. TU Wien, 2021.

[64] E. Anderson, "The species problem in iris," *Annals of the Missouri Botanical Garden*, vol. 23, no. 3, pp. 457–509, 1936.

[65] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[66] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data mining and knowledge discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[67] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[68] O. Z. Maimon and L. Rokach, *Data mining with decision trees: theory and applications*. World scientific, 2014, vol. 81.

[69] P. A. Gutiérrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2015.

[70] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 42, no. 2, pp. 109–127, 1980.

[71] P. McCullagh and J. A. Nelder, *Generalized linear models*. Routledge, 2019.

[72] J. A. Anderson, "Regression and ordered categorical variables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 46, no. 1, pp. 1–22, 1984.

[73] C. Winship and R. D. Mare, "Regression models with ordinal variables," *American sociological review*, pp. 512–525, 1984.

[74] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 283–287.

[75] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Ordinal regression with multiple output cnn for age estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4920–4928.

[76] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 493–507, 2012.

[77] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[78] ——, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[79] ——, "Out-of-bag estimation," 1996.

[80] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, 2013.

[81] I. Mollas, G. Tsoumakas, and N. Bassiliades, "Lionforests: Local interpretation of random forests through path selection," *arXiv preprint arXiv:1911.08780*, 2019.

[82] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[83] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[84] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[85] M.-H. Roy and D. Larocque, "Robustness of random forests for regression," *Journal of Nonparametric Statistics*, vol. 24, no. 4, pp. 993–1006, 2012.

[86] B. Macukow, "Neural networks–state of art, brief history, basic models and architecture," in *IFIP international conference on computer information systems and industrial management*. Springer, 2016, pp. 3–14.

[87] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[88] L. Guesmi, H. Fathallah, and M. Menif, *Modulation Format Recognition Using Artificial Neural Networks for the Next Generation Optical Networks*, 02 2018.

[89] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[90] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[91] H. Hassan, A. Negm, M. Zahran, and O. Saavedra, "Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: Case study el burullus lake." *International Water Technology Journal*, vol. 5, 12 2015.

[92] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[93] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[94] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[95] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[96] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[97] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of clinical epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[98] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[99] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," *arXiv preprint arXiv:1705.10694*, 2017.

[100] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.

[101] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[102] B. Komer, J. Bergstra, and C. Eliasmith, "Hyperopt-sklearn," in *Automated Machine Learning*. Springer, Cham, 2019, pp. 97–111.

[103] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[104] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tiňo, and C. Hervás-Martínez, "Exploitation of pairwise class distances for ordinal classification," *Neural computation*, vol. 25, no. 9, pp. 2450–2485, 2013.

[105] S. B. Kotsiantis and P. E. Pintelas, "A cost sensitive technique for ordinal classification problems," in *Hellenic Conference on Artificial Intelligence*. Springer, 2004, pp. 220–229.

[106] L. Wynants, B. Van Calster, G. S. Collins, R. D. Riley, G. Heinze, E. Schuit, M. M. Bonten, D. L. Dahly, J. A. Damen, T. P. Debray *et al.*, "Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal," *bmj*, vol. 369, 2020.

[107] C. H. Sudre, B. Murray, T. Varsavsky, M. S. Graham, R. S. Penfold, R. C. Bowyer, J. C. Pujol, K. Klaser, M. Antonelli, L. S. Canas *et al.*, "Attributes and predictors of long covid," *Nature medicine*, vol. 27, no. 4, pp. 626–631, 2021.

[108] D. A. Alabbad, A. M. Almuhaideb, S. J. Alsunaidi, K. S. Alqudaihi, F. A. Alamoudi, M. K. Alhobaishi, N. A. Alaqeel, and M. S. Alshahrani, "Machine learning model for predicting the length of stay in the intensive care unit for covid-19 patients in the eastern province of saudi arabia," *Informatics in Medicine Unlocked*, p. 100937, 2022.

[109] T. Dan, Y. Li, Z. Zhu, X. Chen, W. Quan, Y. Hu, G. Tao, L. Zhu, J. Zhu, Y. Jin *et al.*, "Machine learning to predict icu admission, icu mortality and survivors' length of stay among covid-19 patients: Toward optimal allocation of icu resources," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 555–561.

[110] S. Janitza, G. Tutz, and A.-L. Boulesteix, "Random forest for ordinal responses: prediction and variable selection," *Computational Statistics & Data Analysis*, vol. 96, pp. 57–73, 2016.

[111] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased recursive partitioning: A conditional inference framework," *Journal of Computational and Graphical statistics*, vol. 15, no. 3, pp. 651–674, 2006.

[112] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1279–1284.

[113] K. Crammer and Y. Singer, "Pranking with ranking," *Advances in neural information processing systems*, vol. 14, 2001.

[114] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," *Advances in neural information processing systems*, vol. 19, 2006.

[115] M. N. Wright and A. Ziegler, "ranger: A fast implementation of random forests for high dimensional data in C++ and R," *Journal of Statistical Software*, vol. 77, no. 1, pp. 1–17, 2017.

[116] W. Falcon, "Pytorch lightning," 2019, version 1.4. [Online]. Available: https://www.pytorchlightning.ai

[117] H. Cramér, "Mathematical methods of statistics (pms-9), volume 9," in *Mathematical Methods of Statistics (PMS-9), Volume 9*. Princeton university press, 2016.

[118] H. A. Ngo, "Correlation between discrete (categorical) variables," December 2019, https://rstudio-pubs-static.s3.amazonaws.com/558925_38b86f0530c9480fad4d029a4e4aea68.html Accessed on: 25.05.2022.

[119] H. Theil, "Economic forecasts and policy," 1961.

[120] F. Galton, "Regression towards mediocrity in hereditary stature." *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886.

[121] C. Spearman, "The proof and measurement of association between two things." *The American Journal of Psychology*, vol. 15, pp. 72–101, 1904. [Online]. Available: https://www.jstor.org/stable/1412159

[122] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl, "Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features," *Computational Statistics*, pp. 1–22, 2022.

[123] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[124] J. Z. Bing Zhu, Zihan Gao, "Integrated r library for imbalanced classification," 2019, accessed on 10.06.2022. [Online]. Available: https://github.com/shuzhiquan/IRIC

[125] R. Hornung, *ordinalForest: Ordinal Forests: Prediction and Variable Ranking with Ordinal Target Variables*, 2021, r package version 2.4-2. [Online]. Available: https://CRAN.R-project.org/package=ordinalForest

[126] M. M. Rahman and D. N. Davis, "Addressing the class imbalance problem in medical datasets," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 224, 2013.

[127] R. Gunduz, B. S. Yildiz, I. H. Ozdemir, N. Cetin, M. B. Ozen, E. O. Bakir, S. Ozgur, and O. Bayturan, "Cha2ds2-vasc score and modified cha2ds2-vasc score can predict mortality and intensive care unit hospitalization in covid-19 patients," *Journal of Thrombosis and Thrombolysis*, vol. 52, no. 3, pp. 914–924, 2021.

[128] Y. Hong, X. Wu, J. Qu, Y. Gao, H. Chen, and Z. Zhang, "Clinical characteristics of coronavirus disease 2019 and development of a prediction model for prolonged hospital length of stay," *Annals of translational medicine*, vol. 8, no. 7, 2020.

[129] A. Henzi, G.-R. Kleger, M. P. Hilty, P. D. Wendel Garcia, J. F. Ziegel, and R.-.-I. I. for Switzerland, "Probabilistic analysis of covid-19 patients' individual length of stay in swiss intensive care units," *PloS one*, vol. 16, no. 2, p. e0247265, 2021.

[130] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315–1316, 2010.

[131] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[132] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *ACM SIGKDD Explorations Newsletter*, vol. 3, no. 1, pp. 27–32, 2001.