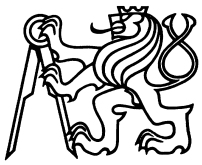
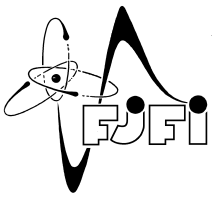


Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering



Estimating patient's life expectancy after a successful kidney transplant using machine learning methods

Odhad délky života pacienta po úspěšné transplantaci ledviny pomocí metod strojového učení

Bachelor's Degree Project

Author: **Kyrylo Stadniuk**

Supervisor: **Ing. Tomáš Kouřim**

Consultant: **Ing. Pavel Strachota, Ph.D.**

Language advisor: **PaedDr. Eliška Rafajová**

Academic year: 2022/2023

Katedra: matematiky

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Kyrylo Stadniuk
Studijní program:	Aplikovaná informatika
Název práce (česky):	Odhad délky života pacienta po úspěšné transplantaci ledviny pomocí metod strojového učení
Název práce (anglicky):	Estimating patient's life expectancy after a successful kidney transplant using machine learning methods

Pokyny pro vypracování:

1) Prozkoumejte současný přístup k transplantacím ledvin, jeho problémy a výzvy. /
Investigate the current approach to kidney transplantation, its problems and challenges.
2) Prozkoumejte příslušné metody strojového učení a metody pro hodnocení přesnosti modelu. /
Explore applicable machine learning methods and model accuracy evaluation methods.
3) Vyčistěte, předzpracujte a rozšiřte stávající datovou sadu. /
Clean, preprocess and extend the existing dataset.
4) Vytvořte prediktivní model strojového učení pro odhad délky života pacienta a ohodnoťte jeho přesnost. /
Create a predictive machine learning model estimating a patient's life expectancy and evaluate its accuracy.
5) Navrhněte úpravy skórovacího algoritmu pro transplantace ledvin na základě výsledků prediktivního modelu. /
Design an updated kidney matching compatibility scoring algorithm based on the prediction model.
6) Prozkoumejte možnost integrace dosažených výsledků do nástroje pro správu transplantací TX Matching. /
Evaluate the possibility of integrating achieved results into kidney transplantation management tool TX Matching.

Doporučená literatura:

1) P. Bruce, A. Bruce, P. Gedeck, Practical Statistics for Data Scientists, O´Reilly, 2020.
2) I. H. Witten, E. Frank, M. A. Hall, Ch. J. Pal, Data Mining : Practical Machine Learning Tools and Techniques. Morgan Kaufman, 2017.
3) A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019.
4) J. J. Kim, S. V. Fuggle, S. D. Marks, Does HLA matching matter in the modern era of renal transplantation? Pediatr Nephrol 36, 2021, 31–40.
5) R. Reindl-Schwaighofer, A. Heinzel, A. Kainz, et al., Contribution of non-HLA incompatibility between donor and recipient to kidney allograft survival: genome-wide analysis in a prospective cohort. The Lancet 393, 10174, 2019, 910-917.
6) M. Wohlfahrtová, O. Viklický, R. Lischke a kolektiv, Transplantace orgánů v klinické praxi. Grada, 2021.

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Tomáš Kouřim
Mild Blue, s.r.o., Plzeňská 27, Praha 5

Jméno a pracoviště konzultanta:

Ing. Pavel Strachota, Ph.D.
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze, Trojanova 13, 120 00 Praha 2

Datum zadání bakalářské práce: 31.10.2022

Datum odevzdání bakalářské práce: 2.8.2023

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 31.10.2022

garant oboru

vedoucí katedry

děkan

Acknowledgment:
I am grateful to Ing. Tomáš Kouřim for his expert guidance and to Dr. Pavel Strachota for his invaluable support and insightful feedback throughout this project. I would also like to extend my sincerest appreciation to PaedDr Eliška Rafajová for her language assistance.

Author's declaration:
I declare that this Bachelor's Degree Project is entirely my own work and I have listed all the used sources in the bibliography.

Prague, August 2, 2023 Kyrylo Stadniuk

Název práce:

Odhad délky života pacienta po úspěšné transplantaci ledviny pomocí metod strojového učení

Autor: Kyrylo Stadniuk

Obor: Aplikovaná Informatika

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Tomás Kourim Mild Blue, s.r.o., Plzenská 27, Praha 5

Konzultant: Ing. Pavel Strachota, Ph.D. Katedra matematiky, Fakulta jaderna a fyzikálne inzenyrska, Ceské vysoké udeni technické v Praze, Trojanova 13, 120 00 Praha 2

Abstrakt: Abstrakt max. na 10 řádků.

Klíčová slova: klíčová slova (nebo výrazy) seřazená podle abecedy a oddělená čárkou

Title:

Estimating patient's life expectancy after a successful kidney transplant using machine learning methods

Author: Kyrylo Stadniuk

Abstract: Max. 10 lines of English abstract text.

Key words: keywords in alphabetical order separated by commas

Contents

Chapter 1

Introduction

The goal of this paper is to explore fields of kidney transplantation and machine learning, create and apply machine learning model in real-world application.

Chapter 2

Medical Background

2.1 Why kidneys fail

2.2 The history of kidney transplantation.

Early Animal Experiments

Advancements in surgical methods and techniques at the beginning of the 20th century eventually led to experiments with organ transplantation. One of the first recorded transplantations was an autograft, where the donor and recipient are the same individual, performed by Emerich Ullmann on March 1, 1902, at the Vienna Medical School. Utilizing vascular suturing techniques developed by Ervin Payr, Ullmann successfully connected the dog's kidney to the vessels of its neck, resulting in the production of urine. The success of this experiment was notable enough to be presented to the Vienna Medical Society, sparking considerable interest.

That year, other experiments followed. Alfred von Decastello performed a dog-to-dog kidney allograft, a transplant between two individuals of the same species, at the Institute of Experimental Pathology in Vienna. Although initially, the transplanted kidney produced urine, it eventually ceased. Later, Ullman performed a dog-to-goat kidney xenograft, which resulted in short-term function as well.

At the same time, in Lyon, Alexis Carrel and his colleagues were working on vascular suturing methods. Carrel's technique, known as Carrel's seam, was a considerable improvement over existing methods, addressing the common issues of thrombosis, hemorrhage, stricture, and embolism [1]. His consecutive move to The Rockefeller Institute for Medical Research in the United States led to further refinements of his method and the documentation of organ rejection. For his contributions, Carrel received the Nobel Prize in Medicine in 1912 [2].

These early experiments, although varied in their outcomes, were defining in shaping the future of the organ transplantation. They not only illustrated the possibility of organ transplantation, but also highlighted the challenges, such as rejection, that would direct the course of future research. The insights gained in animal experiments laid the groundwork for the next big milestone in transplantation medicine: the onset of human organ transplantation. This transition from animals to humans marked the beginning of a new era in medical history.

Early Human Transplantation

The first recorded human renal xenograft, a transplantation between individuals of different species, is credited to Mathieu Jaboulay in 1906. It involved a pig and a goat as donor animals for two xenografts.

One kidney was transplanted to the arm and the second to the thigh. Although each kidney functioned only for an hour, these efforts marked the beginning of human transplantation attempts [1, 4].

Ernst Unger's xenografts in 1909 gained more attention. His first attempt, involving the transplantation of a kidney from a stillborn baby to a baboon, resulted in a lack of kidney function despite a successful vascular anastomosis (connection of vessels). Inspired by a successful surgery, Unger attempted a monkey-to-human xenograft, which also resulted in failure.

These early experiments demonstrated the technical feasibility of kidney transplantation but also exposed the challenge of graft rejection. Alexis Carrel, in a 1914 lecture, mentioned J. B. Murphy's work on irradiation and benzol treatment, suggesting their potential to improve graft survival [1]. Inspired by these findings, Carrel conducted his own experiments with irradiation, achieving prolonged graft survival. However, these findings were never formally published [1].

The period of the 1930s and 1940s was stagnant compared to the beginning of the century. European surgical centers that studied transplantology before were in decline. Meanwhile, the Mayo Clinic in the US was conducting some cautious experiments without considering Carrol's works and attempts at immunosuppression [1].

However, it was during that period that a significant milestone was achieved by Yurii Voronyi. On March 3, 1933, in Kherson, Ukraine, Voronyi performed the first human-to-human kidney transplant on a woman suffering from acute renal failure due to mercury chloride poisoning. Given the ethical concerns surrounding living donors and previous failures of xenografts, Voronyi considered a cadaveric transplant the only viable option. Although there was initial urine production, the transplant failed 48 hours post-surgery due to blood group incompatibility and prolonged warm ischemia, triggering an immune reaction.

Despite this setback, Voronyi continued to perform similar transplantations. He viewed these transplants as temporary measures to bridge the gap until the recipient's own kidneys could recover. Out of the six transplants he performed, two patients experienced a complete recovery, regaining normal kidney function [4].

The pioneering work of Jaboulay, Unger, and Voronyi not only demonstrated the technical feasibility of human organ transplantation but also highlighted its main challenge of graft rejection. Despite the progress, the field has yet to witness success, largely due to the lack of effective immunosuppression. In the next section, we will explore important moments of 1950s that transformed human renal transplantation from a daring experiment to feasible medical procedure.

First Successes

In 1946, at the Peter Bent Brigham Hospital in Boston, a group of surgeons: Hufnagel, Hume, and Landsteiner, performed kidney transplantation under local anesthetic on the arm vessels. The short period of kidney functioning may have helped the patient recover from acute renal failure, igniting the hospital's interest in renal transplantation.

During the same period, European surgeons were making significant advancements. Notably, Simonsen in Denmark, Dempster in London, and Küss in Paris concluded that it is preferable to place the kidney in the pelvis. Furthermore, both Simonsen and Dempster deduced that the immune response was responsible for graft failure and hypothesized that the humoral mechanism of rejection was probable.

The early 1950s marked a period of active experimentation. In Paris, Jean Hamburger reported the first live-related kidney transplant between a mother and her child, achieving the immediate function of the transplanted kidney for 22 days until it was rejected. Meanwhile, in Boston, a series of nine transplantations with the thigh position of the allograft was closely studied. Moreover, in 1953, David Hume introduced the pre-transplant use of hemodialysis. Although some success was achieved with

the administration of the adrenocorticotropic hormone (more known as cortisone), it was suggested that the immunosuppressive effect of uremia, the excess of metabolism end products in blood, were more significant than the drug regimen. Hume's further research suggested the potential benefits of prior blood transfusions, blood group compatibility, and bilateral nephrectomy (removal of both kidneys) of the host for transplant success - insights later validated with further research.

These attempts in the early 1950s taught technical aspects of kidney transplantation, and, with increased confidence, on December 23, 1954, in Boston, Joseph Murray performed kidney allograft from one identical twin to another, bypassing the rejection barrier. From that time, many similar surgeries were performed in Boston [1, 3].

To conclude, the early successes during the 1950s marked a transformative period in the history of kidney transplantation. Not only the technical possibility of kidney transplantation was shown, but also the complex immunological challenges began to unravel. Works of Hume, Murray, and others underlined the critical role of immune response in graft survival, starting the quest for effective immunosuppression. The following section tells more about that.

Attempts in Immunosuppression

The exploration of immunosuppression in transplantation began as early as the late 1940s. At the Mayo Clinic in 1948, patients with rheumatoid arthritis were administered cortisone, an adrenal cortical hormone with mild immunosuppressive properties, which provided temporary relief. Although initially praised, its effects were deemed clinically insignificant for transplantation purposes. It led researchers to revisit earlier experiments with irradiation. Experiments on mice by Joan Main and Richmond Prehn showed promising results, inspiring human trials in Boston and Paris [3].

In 1958, Murray's team in Boston employed radiation in human transplantation, achieving a significant breakthrough with a kidney transplant between non-identical twins that lasted 20 years. Similarly, in Paris, Jean Hamburger's team accomplished a 26-year functioning transplant using radiation [3].

The quest for safer immunosuppressive methods led to the anticancer drug 6-mercaptopurine (6-MP). In 1959, Schwarz and Damesehek published a paper that described how 6-MP lowered immune response to foreign proteins in rabbits. Inspired by their work, Roy Calne performed his own dog experiments, showing promising results backed by Charles Zukoski and David Hume. Despite initial setbacks, Kuss and others reported prolonged graft survival from non-related donors using total body irradiation (TBI) complemented by 6-MP. The introduction of azathioprine in 1959, a derivative of 6-MP, by Gertrude Elion and George Hitchings, further improved results. Their groundbreaking work earned them the Nobel Prize, and by 1961, azathioprine was available for human use [3].

In 1963, a conference held by the National Research Council revealed a bleak outlook for kidney transplantation, with less than 10% survival beyond three months. It changed when Tom Starzl presented a protocol combining 6-MP with prednisone, leading to over one-year graft survival in 70% of cases. His results revolutionized the field, resulting in 50 new US transplantation programs and setting a 20-year standard in post-transplant immunosuppression [3].

In summary, the late 1940s to the early 1960s was a period of discoveries and experimentation. The use of cortisone in 1948, the administration of 6-MP for immunosuppression, and the invention of azathioprine in 1959 led to better immunosuppression, culminating in Tom Starzl's protocol, which became the standard for the next two decades. As we transition from this era of discovery to the period of 1964 to 1980, we enter a phase of steady developments that would solidify the practice of kidney transplantation.

Plateau

From 1964 to 1980, kidney transplantation saw gradual progress. Dialysis, developed during WWII, finally became available for chronic renal failure as a result of the invention of Teflon arteriovenous conduits in the 1960s. Acceptance of brain death expanded donor pools. Organ preservation techniques also advanced: total body hypothermia was replaced by targeted cold solutions infusions that preserved organs better. By the mid-60s, longer preservation times gave way to organ exchanges between centers. Concerns about equitable distribution of organs led to the National Transplant Act in 1984, establishing the United Network of Organ Sharing (UNOS) for oversight [3].

As techniques were improving and donor pools expanding in the period from 1964 to 1980, so was the understanding of the nuances of immune compatibility. The following section describes the developments in tissue typing that would further expand the field of kidney transplantation.

Tissue Typing

The concept of tissue typing, suggested by Alexis Carrel in the early 20th century, remained unproven until Jean Dausset discovered the first human leukocyte antigen (HLA) in 1958 (more on HLA in . It wasn't until 1964, with Paul Terasaki's development of the microcytotoxicity assay, that testing for antibodies became reliable. The test involved mixing the donor's lymphocytes with the recipient's serum and swiftly became the standard, known as the *crossmatch test* [3].

For several years, Terasaki performed typing for most U.S. transplant centers and found a couple of observations: 1) Positive crossmatch test predicts hyperacute rejection. 2) matching can reliably identify optimal donors within a family, and it was assumed that the same principle would apply to non-related recipients.

However, in 1970, Terasaki's review of his extensive database of cadaver renal allografts revealed no correlation with the typing. It raised much agitation in the tissue typing community, and his grant was temporarily suspended until others didn't report the same. Since then, many crucial histocompatibility antigens have been discovered (Class II locus: D and DR, more on antigens in Section 2.3). Histocompatibility matching remains essential in bone marrow transplantation and in selecting family donors. Nevertheless, for unrelated organ transplants, matching considerably benefits only in cases of a perfect match [3].

While tissue typing and matching have played a crucial role in improving transplantation outcomes, the evolution of immunosuppressive drugs has been equally, if not more, important. The subsequent section tells more about new developments in immunosuppressive drugs in upcoming years.

Cyclosporine and Tacrolimus

The discovery of cyclosporine in 1976 by Jean-François Borel marked a significant milestone in the realm of transplantation. As a fungal derivative with potent immunosuppressive properties, cyclosporine drastically improved the outcomes of both renal and extra-renal transplants, surpassing the efficacy of the previously used drug, azathioprine. Similar to 6-MP, to achieve the best results, it had to be combined with prednisone. This protocol remained standard until 1989, when Tacrolimus, an even more powerful immunosuppressive agent, was introduced. Tacrolimus proved to be effective in cases where the combination of cyclosporine and prednisone was insufficient, effectively replacing cyclosporine as a usual baseline agent [3].

In conclusion, the history of renal transplantation has been marked by groundbreaking discoveries and persistent challenges. From the pioneering attempts of xenografts in the early 20th century to the

technical advancements and immunological breakthroughs of the mid-century, each milestone has been essential in shaping the current landscape of organ transplantation. As we explored the history of kidney transplantation, a deeper understanding of the immune system has become crucial. This sets the stage for the next section, which examines the fundamentals of immunology, laying the foundation for understanding the interplay between the immune system and the transplanted organ.

2.3 The Introduction to Immunology

The *immune system* is a sophisticated defense mechanism that evolved to protect multicellular organisms from pathogens such as bacteria, fungi, viruses, and parasites. It is a network of many cells and tissues that compose a complex system that detects, evaluates, and responds to the invader. It is essential to understand these mechanisms, as the immune response plays a crucial role in graft acceptance or rejection [6].

Innate and adaptive immunity are the two interconnected systems of immune response. *Innate immunity (*also *natural* or *native)* includes primitive built-in cellular and molecular mechanisms that provide rapid, albeit non-specific response to common pathogens. In contrast, *adaptive (specific* or *acquired) immunity* is slower to respond but capable of providing more targeted responses [6].

Physical barriers, including epithelia and mucous membranes constitute host's first line of defense and are a part of innate immunity. This branch of immune system not only prevents invaders from infiltrating the host but also quickly destroys many microbes that succeed in breaching these barriers. Innate immunity provides the neccessary protection until the adaptive immunity is activated. It also communicates to the adaptive immunity how to best respond to the invader. Furthermore, innate immunity plays an important role in the clearence of dead tissue and the initiation of repair after the tissue was damaged [6].

Adaptive immunity is broadly categorized into *humoral* and *cell-mediated immunity*. Central to both humoral and cell-mediated immune responses are *lymphocytes*. *B lymphocytes* (*B cells)* mediate humoral response by producing antigen-specific antibodies on encounter with the antigen. Produced antibodies then bind themselves to *antigens* — foreign molecular structures, identified by common molecular patterns known as *pathogen-associated molecular patterns* (*PAMPs*) — to mark them for destruction. The immune system uses *pathogen recognition receptors* (*PRRs*), found on the surface of T cells, and antibodies to detect and categorize *these* PAMPs, which can take the form of molecules on the surface of a pathogen or its by-products. PRRs bind to PAMPs and initiate targeted cascade of events that culminate in pathogen's elimination [6].

T cells, on the other hand, when encountering an antigen start to proliferate forming an army of T cells that will eliminate the invader and will form long-term memory about the pathogen. Activated T cells are divided into the following categories: helper T cells ($CD4^+$), that help B cells to produce antibodies and help kill ingested microbes; *cytotoxic T cells* ($CD8^+$) that target and kill infected cells; *regulatory T lymphocytes* that prevent or limit immune responses; and *memory cells*, that remain in the body long-term to provide faster and stronger immune response if the same antigen is encountered in the future [6].

Pathogen-host interaction is a continuous arms race, as pathogens usually have a short life cycle and can modify their DNA to elude the host's recognition systems. Immune system counters this with the generation of host-tolerant lymphocytes with diverse PRRs during the development in bone marrow. Cells that react to the host's own cells are eliminated, ensuring that only non-self reactive cells are allowed in circulation. The principle of recognizing self vs. non-self is called tolerance [6].

In conclusion, the immune system is a complex network of molecules, cells, tissues, and organs that cooperate in protecting the organism from pathogens. The system can be divided into two main branches:

innate and adaptive, which cooperate in protecting the host from infections while developing long-term immunity to specific pathogens. Understanding the mechanisms of the immune system is essential to understanding the domain of kidney transplantation.

2.4 Immunology of Kidney Transplant

The process of transplantation inevitably includes termination of blood flow, and, as a result, oxygenation. Therefore cell is unable to generate sufficient amount of energy to maintain homeostasis, leading to damage or death. Damage or death is associated with death (or damage) associated molecular patterns (DAMP) release that might be detected by both innate and adaptive immunity.

2.4.1 Immune system activation Peritransplant

The process of transplantation inevitably includes termination of blood flow, and, as a result, oxygenation. Therefore cell is unable to generate sufficient amount of energy to maintain homeostasis, leading to damage or death. Damage or death is associated with DAMP release that might be detected by both innate and adaptive immunity. Mostly it is the ancient innate immunity that is activated with its soluble arm - complement system.

Damage Signals Many DAMPS are recognized by the same PRRs that mediate response to PAMPs. These DAMPS include molecules that are normally hidden from the immune system and are produced during ischemia, such as extrecellular ATP, heat shock proteins (HSPs), uric acid, etc. Likewise, oxidative stress and decline in intracellular potassium may act as intracellular damage signals.

Complement Complement system is comprised of series of proteine kinases that are sequentially activated resulting in membrane attack complex (MAC) formation. MAC include complement components C5 to C9, which are inserted into pathogen cell membrane resulting in compromising cell integrity leading to cell death.

There are three pathways of complement system activation: the classical pathway, the alternative pathway, and the mannose-binding lectin (MBL) pathway. The classical pathway is activated by IgM and IgG antibodies and participates in antibody-mediated rejection, that will be discussed further. Alternative complement is always active and therefore must be controlled by a regulatory proteins, to prevent inadequate responses. The MBL pathway is activated by damaged endothelium, a cell tissue that covers organs and vessels, and carbohydrates present on pathogens. Either pathway results in C3 convertase that cleaves C3. This cleavage leads to a cascade of reactions that culminate in MAC formation.

Long ischemia time results in endothelial cell damage that is acossiated with ischemia-reperfusion injury (IRI). IRI activates MBL and alternative complement pathways.

Gene silencing using small interfering RNA (siRNA) might be a promising instrument in organ transplantation, because it can be appliend to an allograft during cold reperfusion and it has been shown to mitigate IRI in animal models. Other strategies of supressing local complement activation would also be useful.

2.4.2 Stimulation of Adaptive Alloimmunity

Immune response to a graft occurs in two main stages: afferent and efferent arms. In afferent stage, recipient lymphocytes are stimulated by donor antigens and start to proliterate and send signals to other

Table 2.1: MHC class division

MHC class I	MHC class II
HLA-A	HLA-DR
HLA-B	HLA-DP
HLA-C	HLA-DQ

cells. In efferent arm, leukocytes migrate to the transplanted organ and donor specific antibodies are produced.

For the immune system to be activated graft must express antigens that will be considered by the host's immune system as foreign. These include ABO antigens, human leukocyte antigens (HLA), and polymorphic non-HLA "auto-antigens".

ABO Blood Group Antigens

ABO system is used to group blood into groups, based on presence or absense of antigens on a blood cell surface. There are four major blood groups: A, B, O and AB.

When allocating an organ to transplant the first thing that is considered is ABO blood group antigens compatibility. ABO antigens are expressed almost by any cell in the allograft, and if the transplantation to be carried out in ABO-incompatible donor and recipient it would result in a hyperacute antibody-mediated rejection.

Donors with blood group O are so called "universal donors". Organs from them can be safely transplanted to recipients with any ABO blood group. Whereas, recipient with AB group can safely receive organ from recipient with any ABO blood group and is called a "universal recipient".

HLA

Histocompatibility antigens are genetically encoded antigens that cover cell surfaces. They differ between individuals of the same species and therefore trigger an immune response in case of allograft. In all vertebraes histocompatibility antigens are divided into single major histocompatibility complex (MHC) and numerous minor histocompatibility (miH) systems. In case of either MHC or miH incompatibility the result is an immune response to the graft, more severe in case of MHC than miH. Rejection in MHC-compatible donor-recipient pair is usually delayed, in some cases forever. Although, sometimes miH mismatch might be so severe that it would be comparable to full MHC mismatch.

MHC antigens are proteins that cover cell surfaces to help the immune system to recognize self vs. non-self. Major histocompatibility complex is divided into MHC class I and MHC class II. MHC class I cover surfaces of most cells and are liable for activation of cytotoxic CD8 cells, that help to find and destroy infected cells. MHC class II are found on certain immune cells and play crucial role in immune response coordination. In humans MHC classess are divided into three subgroups each, as can be seen on table

In clinical practice, clinicians asses and try to match donors and recipient according to the number of HLA-A, -B, and -DR mismatches, ranging from zero mismatches (0-0-0) to a maximum of 6 mismatches (2-2-2). Generally more emphasis is placed on DR loci due to capability of CD4 T cell activation, which might trigger both humoral and cellular adaptive immune responses.

Minor histocompatibility proteins can act as antigens, although weaker than MHC. However if prior sensitisation exists it could result in severe immune response that might result in graft loss.

2.4.3 Rejection

2.4.4 T Cell-mediated rejection

T cell-mediated rejection or TCMR is the most common type of allograft rejection, as it still happens in 20% of transplantations mostly within first 6 months posttrasplant. Immune system cells migrate through vessels to the graft, become activated and start to attack the organ. Complement may also play role in it.

2.4.5 B Cell-mediated rejection

B cells are immune system cells that produce antibodies. Alloantibodies are antibodies that react to donor-specific HLA antigens and might cause hyperacute rejection, *acute antibody-mediated rejection* (ABMR), and chronic ABMR. About 30% of patients have sensitivities and have certain HLA antibodies. It might cease transplantation or require antibody suppression strategy. Even low amount of antibodies below crossmatch cutoff doubles the risk of ABMR and increases the risk of graft failure by 76%. Additionally, donor specific antibodies might develop posttransplant and cause an actue ABMR.

Acute AMBR is rarely seen in patients withour prior sensitization and is highly difficult to treat. AMBR is characterized by decline in allograft function, presense of DSA and signs of acute vascular injury. A progressive reduction in graft function over time is observed almost universally.

2.4.6 Transplant Tolerance

Taking into account the detrimental effect of long-term immunosuppression one of the primary objectives in transplantation is the induction of immunologic non-responsiveness (tolerance) to an allograft. The are a couple a pathways of immune non-responsivenes generation described in literature, however it haven't gone further animal models yet.

2.4.7 Factors Influencing Rejection Beyond the Graft - Microbiome

Human body is very complex system where every subsystem influences other subsystems and the whole system in general. It is clear that gut microbiome has a profound influence on immune system. It is possible microflora on the allograft might cause rejection. Immunosupression, profilactory antibbiotics, diet changes and other restrictions asossiated with organ transplantation result in decrease in gut microbiome diversity that result in systematic inflammation, that might contribute to alloimunnity, as well as autoimunnity.

Chapter 3

Machine Learning Background

Machine learning is a subfield of computer science that consists of building algorithms capable of processing large amounts of data, finding patterns, and performing actions such as predictions or generating new data. It is an intersection of many fields of science, such as statistics, theory of probability, linear algebra, calculus, and certainly, computer science.

Machine learning excels in problems that are either overly complex or have no known algorithm.[9] It can help us generate knowledge. We can extract previously unknown correlations from the data and build knowledge. It might make fewer errors in decision-making than humans.

Based on the problem and, therefore, on our approach to building a dataset and the model, machine learning can be divided into four subfields: *supervised*, *semi-supervised*, *unsupervised*, and *reinforcement learning*. *Supervised learning* means that data is labeled, and we want to predict labels for the unlabeled data. The term labeled data is explained in the following section dedicated to supervised learning. Unsupervised learning deals with unlabeled data.

Semi-supervised learning deals with partially labeled data, and we need to label it fully either manually, or using techniques such as *clustering*.

In *reinforcement learning*, we create an environment, set up rewards for performing certain actions and punishment for others, and let the machine (actor) perform actions that produce the highest reward.

Every field is affected by human errors, and medicine is no exception. Machine learning also makes mistakes, but if we manage to get at least 1% fewer errors than humans make, this will be a substantial achievement.The human body is a complex system, where it is very difficult to comprehend all processesand and how they relate to each other. In addition, machine learning can help us gain insight into them through accumulated data and discover new relations between them.

In this chapter, we will cover all theoretical backgrounds that might prove useful for solving our problem, including classical machine learning, statistical survival analysis, basic steps that are required to **create machine learning system**s, and data preprocessing. We will begin by exploring supervised learning.

3.1 Supervised Learning

Supervised learning is the process of training a model on data where the outcome is known, to make predictions for data where the outcome is not known[12]. *Classification* and *regression* are common supervised learning tasks. In this section we will define these problems and the necessary terminology, and describe commonly used algorithms that are used to solve these types of problems.

In supervised learning the *dataset* is the collection of labeled examples $\{(\overline{x}_i, y_i)\}_{i=1}^{N}$, where each individual $\overline{x}_i$ is called a *feature vector*. A feature vector is a vector that in each its dimension $j = 1, ..., D$

contains a value that describes an example in some way. This value is called a *feature* and is denoted as $x^{(j)}$. The *label* y^i might be either a finite set of classes $\{1, 2, ..., C\}$, in case of a classification task, or a real number, a vector, a matrix or graph, in case of a regression. The goal of supervised learning algorithm is to create a model using the dataset that will take the feature vector as an input and produce a label or a more complex structure as an output.

Classification is a problem of assigning a label to an unlabeled example. This problem is solved by a classification learning algorithm that takes a labeled set of examples as input and produces a model that takes an unlabeled example as input and outputs a label. If the set of labels has only two classes we talk about *binary classification*. Consequently, if the set of labels has three or more classes, it is a *multiclass classification*. Some algorithms are binary classifiers by definition while others are multiclass classifiers. It is possible to create an *ensemble* out of binary classifiers that will be able to perform multiclass classification. An ensemble is a combination of algorithms that are connected to perform one task.

Regression is a problem of predicting a *target value* given an unlabeled example. The problem is solved by a regression learning algorithm that takes a set of labeled examples as input, and produces a model that takes an unlabeled example as input and outputs a target value.

Classification and regression tasks are similar in many ways and often for each classifier there is an equivalent regressor, and vice versa. In the following subsections we are going to explore some techniques for supervised learning.

3.1.1 Linear Regression

Linear regression is a popular regression learning algorithm. The model produced is a linear combination of all features.

The problem formulation we are trying to solve is as follows: Given a collection of labeled examples $\{(\overline{x}_i, y_i)\}_{i=1}^N$, create a model

$$f_{\bar{w},b}(\overline{x}) = \overline{wx} + b, \tag{3.1}$$

where N is the size of the collection, $\overline{x}_i$ is a *feature vector* of D dimensions of example $i = 1, \ldots, N$, every feature $x_i^{(J)} \epsilon \mathbb{R}$, $y_i \epsilon \mathbb{R}$ is the target value. $\overline{w}$ is a D-dimensional vector of parameters and $b \epsilon \mathbb{R}$. Notation $f_{\bar{w},b}(\overline{x})$ means that f is parametrized by $\overline{w}$ and b.

To train the linear regression means to find optimal values $(\overline{w}^*, b^*)$ of parameters $\overline{w}$ and b so that the model makes as accurate predictions as possible. In graphical terms, it means finding such a hyperplane that fits data points from the training set as well as possible, as shown in image3.1.

To find optimal parameters we need to minimize the following expression:

$$\frac{1}{N} \sum_{i=1...N} (\, f_{\bar{w},b}(\overline{x_i}) - y_i)^2. \tag{3.2}$$

It is called *mean squared error (MSE),* the *loss function* that comprises of *squared error loss* $(f_{\bar{w},b}(\overline{x_i}) - y_i)^2$, another loss function that evaluates individual predictions. The loss function measures the model's overall performance (MSE) or evaluates each prediction (square error loss).

There is a *closed-form solution* for finding optimal values $(\overline{w}^*, b^*)$. A closed-form solution is a simple algebraic expression that gives the result directly. In case of linear regression, it is the *normal equation*, and it looks like the following:

$$\overline{\mathbf{w}}^* = (\mathbf{x^T x})\mathbf{x^T y}. \tag{3.3}$$

Where x^T means transposed feature matrix x.

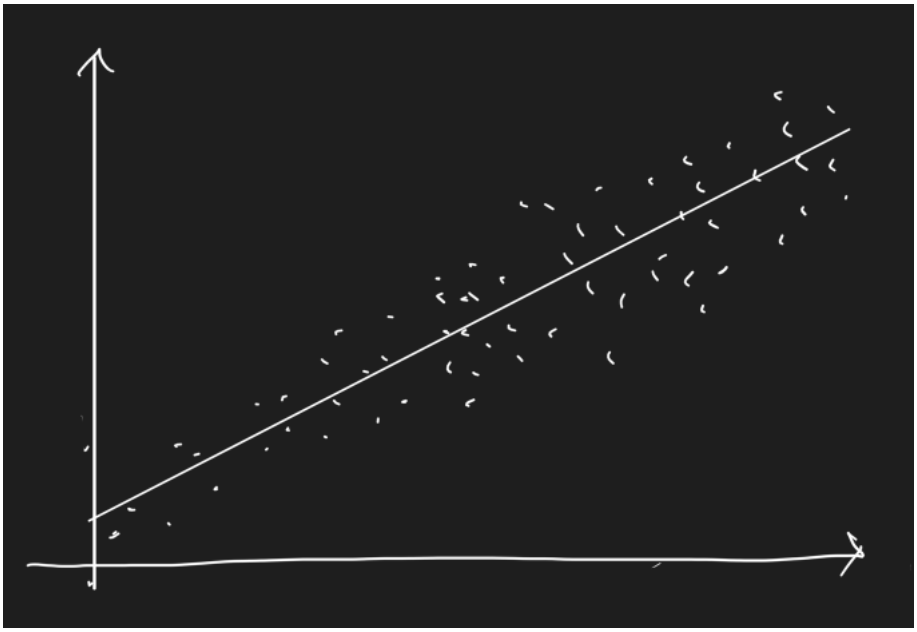


Figure 3.1: Linear regression for two-dimensional data

We could select another loss function, but according to Andriy Burkov, it would be a different algorithm. For example, we could take the absolute difference between $f(x_i)$ and y_i but that would create problems as the derivative of absolute value is not continuous. Therefore the function is not smooth, which might create unnecessary complications during the optimization process.

Linear models are usually resilient to overfitting because they are simple. The model overfits when it learns the intricacies of the training dataset so well that it remembers actual values instead of learning the underlying pattern. Such model is unable to make accurate predictions when confronted with unseen data. More on overfitting in section 3.6.

3.1.2 Logistic Regression

Logistic regression is a binary classifier that estimates the probability of an example belonging to a particular class. If the predicted probability of the instance belonging to a class is greater than 50%, then the model concludes that it belongs to the class (referred to as positive class and labeled as 1). Otherwise, it predicts that the example does not belong to that class (but belongs to the negative class, labeled 0). Logistic regression comes from statistics where its mathematical formulation is similar to a regression, hence the name. Multiclass classification is available in softmax regression, a multiclass variant of logistic regression.

As with linear regression, in logistic regression, we want to model y as a linear combination of $\overline{x}$, but in this case, it is not that straightforward.

The logistic regression model looks like the following:

$$f_{\bar{w},b}(\overline{x}) \stackrel{def}{=} \frac{1}{1+e^{-(wx+b)}}. \tag{3.4}$$

Similar to linear regression, our task is to find optimal values $(\overline{w}^*, b^*)$ for parameters $\overline{w}$ and b.

Once we found $(\overline{w}^*, b^*)$ for the 3.4, in other words, we trained the model, we can apply the model 3.4 on features x_i from an example (x_i, y_i). The output value lies in the range $0 < p < 1$. If y_i is the positive

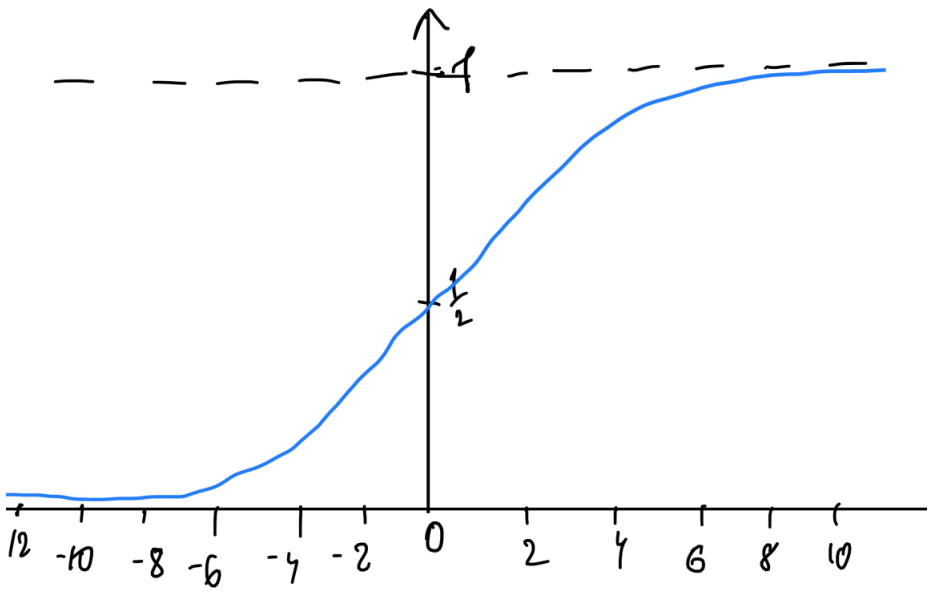


Figure 3.2: Logistic function

class, the likelihood of y_i being a positive class is given by p. Consequently, if y_i is the negative class, the likelihood for it being the negative class is given by $1 - p$.

In the figure we can see that if the y has a value lower than $\frac{1}{2}$, it has negative x values and will be marked as a negative class. If y is greater than $\frac{1}{2}$, it is positive. Although, depending on the context, the threshold may be different.

In logistic regression, instead of *minimizing* MSE we are trying to *maximize* the *likelihood function*. In statistics, the likelihood function tells how likely the example is according to our model. The objective function in logistic regression is called *maximum likelihood*. It looks like the following:

$$L_{\bar{w},b} \overset{def}{=} \prod_{i=1...N} f_{\bar{w},b}(\overline{x}_i)^{y_i}(1 - f_{\bar{w},b}(\overline{x}_i))^{(1-y_i)}. \tag{3.5}$$

On the other hand, due to the exponential function in equation 3.5, it is better to use the *log-likelihood* instead, to make calculations easier. As *Log* is a strictly increasing function, maximizing it is the same as maximizing its argument. The solution to this optimization problem is the same as the solution to the original problem. The log-likelihood function looks like the following:

$$LogL_{\bar{w},b} \overset{def}{=} ln(L_{\overline{w},b}(\overline{x})) \sum_{i=1}^{N} y_i ln f_{\bar{w},b}(\overline{x}) + (1 - y_i) ln(1 - f_{\bar{w},b}(\overline{x})). \tag{3.6}$$

Unfortunately, there is no closed-form solution for this optimization problem. Nonetheless, the function is convex, hence gradient descent (or any other optimization algorithm) pretty much guarantees the finding of the global minimum, provided that the learning rate is not too large and enough time is given.

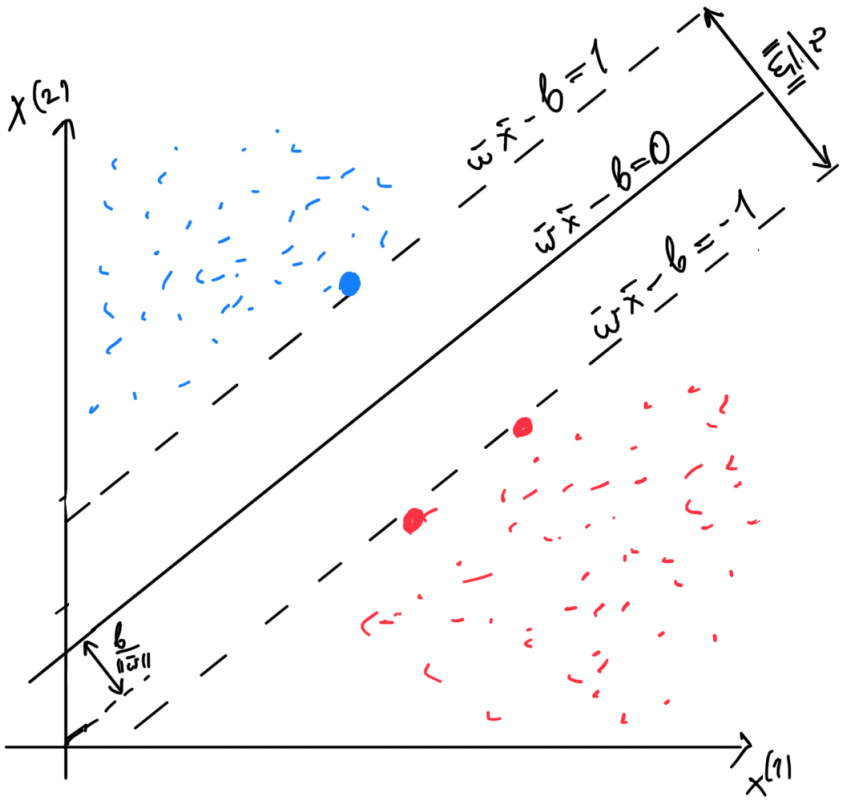


Figure 3.3: SVM demonstration for two-dimensional dataset

3.1.3 Support Vector Machines

Support vector machine (SVM) is a widely-used and powerful machine learning algorithm that can perform a wide range of tasks, including linear and nonlinear classification, regression, and outlier detection on small- to medium-sized datasets.

Linear SVM

In its classical formulation, the support vector machine is a binary classifier. Classes are called positive and negative and are labeled +1 and -1, respectively.

The model is described by the equation

$$f(x) = sign(\overline{w}x - b).$$

The function *sign* returns +1 if the input is positive, and -1 if it is negative. To train the SVM means to find optimal values $(\overline{w}^*, b^*)$ of parameters $\overline{w}$ and b so that the model makes as accurate predictions as possible. The process of finding $(\overline{w}^*, b^*)$ is called training.

The concept behind support vector machines is demonstrated in Figure 3.3. The image consists of two classes represented by red and blue dots, divided by a solid line termed the *decision boundary* $\overline{w}x - b = 0$, with two dashed lines by its sides known as *support vectors* $\overline{w}x - b = 1$ and $\overline{w}x - b = -1$. Support vectors are defined by the closest instances of a class to the decision boundary. These instances are emphasized in the figure.

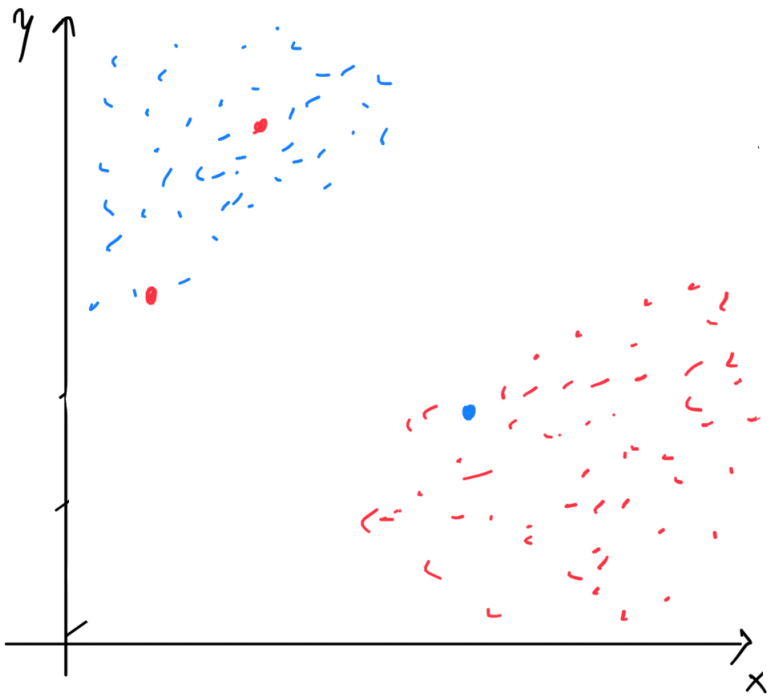


Figure 3.4: Linearly non separable dataset

The distance between the closest instances of two classes is called *margin* and is equal to $\frac{2}{||\overline{w}||}$, where $||\overline{w}||$ is the Euclidean norm and $\overline{w}$ is a parameter vector of the same dimensionality as the feature vector. Thus, the smaller the norm, the larger the margin. The larger the margin, the better the model's generalization. The primary objective of the model is to find the largest possible margin $\frac{2}{||\overline{w}||}$, so, to do that we need to *minimize* the Euclidian norm defined by the expression

$$||\overline{w}|| = \sqrt{\sum_{j=1}^{D} (w^{(j)})^2}.$$

The fundamental assumption of support vector machines is that classes are linearly separable, implying their instances can be separated by a hyperplane (decision boundary) with no examples of one class lying among the ones of the opposite class. It is illustrated in the figure 3.4. In this case, the algorithm won't be able to find an optimal solution with no instances lying between the support vectors and the decision boundary. Consequently, the model is highly sensitive to outliers.

Every optimization problem requires constraints, and for the support vector machine, they are the following:

1. $\overline{wx_i} - b \geqslant +1$ if $y_i = +1$
2. $\overline{wx_i} - b \leqslant -1$ if $y_i = -1$.

These two equations can be reduced to one $y_i(\overline{wx} - b) \geqslant 1$.

The optimization problem we want to solve is the following: Minimize $||\overline{w}||$ subject to constraint $y_i(\overline{wx_i} - b) \geqslant 1$ for $i = 1, ..., N$, where N is the number of features. This problem can be modified so that the quadratic programming techniques could be used in the optimization process. The modified

formula is $\frac{1}{2}\|\overline{w}\|^2$, and minimization of it would also mean minimization of $\|\overline{w}\|$. The updated optimization problem looks like this:

$$min\frac{1}{2}\|\overline{w}\|^2 \text{such that } y_i(\overline{wx_i} - b) \geqslant 1, i = 1, ..., N \tag{3.7}$$

Handling Noise

To introduce the ability of SVM to handle nonlinearly separable data (but not to the extreme), we define the hinge loss function: $max(0, 1 - y_i(\overline{wx_i} - b))$. It is zero if the constraints 1 and 2 are satisfied. If it is not, the data point does not lie on the right side of the decision boundary. The function value is proportional to the distance from the decision boundary. The resulting cost function looks like the following:

$$C\|\overline{w}\|^2 + \frac{1}{N}\sum_{j=1}^{N} max(0, 1 - y_i(\overline{wx_i} - b)), \tag{3.8}$$

where C is the hyperparameter that determines the trade-off between increasing the size of the decision boundary and ensuring that each x_i lies on the correct side of the decision boundary. Its value is chosen experimentally. C handles the trade-off between classifying the training data well and classifying future examples well (generalization). For higher values of C, the misclassification error will be almost negligible, so the algorithm will try to find the highest margin without considering it. For lower values of C, the algorithm will try to make fewer mistakes by sacrificing the margin size. (A larger margin is better for the generalization.) Lower values lead to wider streets and more margin violations, higher values lead to narrower streets and fewer margin violations.

SVM with the hinge loss function is called *soft-margin SVM* while the original formulation that optimizes the Euclidian norm is referred to as *hard-margin SVM*. *Soft margin classification* tries to mitigate the downsides of the *hard margin classification* by trying to find a balance between keeping the margin as large as possible and mitigating the margin outliers (instances that lie on the margin or on the opposite side).

Handling Non-linearity

We can adapt SVM to work with nonlinearly separable datasets by applying the kernel trick. The kernel trick means transforming the original space to a higher dimensional one during the cost function optimization with the hope that, in higher dimensional space, it will become linearly separable. In mathematical language: the kernel trick is mapping $\varphi : \overline{x} \rightarrow \varphi(\overline{x})$, where $\varphi(\overline{x})$ is a vector of higher dimensionality than $\overline{x}$. The kernel trick allows us to save a lot of non-necessary computations.

There are multiple kernel functions. The most widely used are linear, polynomial, radial basis function (RBF),

3.2 Unsupervised Learning

Unsupervised learning deals with a dataset that does not have labels. There are three main branches of unsupervised learning: clustering, dimensionality reduction and anomaly detection. *Clustering* is a method that identifies similar instances and groups them into sets. It has applications in data analysis, namely, *exploratory data analysis (EDA),* customer segmentation, dimensionality reduction, and anomaly detection. Clustering might be either soft, where an instance has a score of belonging to a

particular cluster, or hard, where an instance belongs to only one class. The score might be the distance from the cluster centroid or an affinity (similarity score).

Dimensionality reduction is useful for visualization and for the acceleration of learning. Datasets often have a lot of redundant data or the task requires a lot of features. Many algorithms, such as linear models, SVMs, decision trees, might have their performances compromised due to high-dimensional data. So called *curse of dimensionality* states that high dimensional data can cause slow learning and prevent us from getting an optimal model. Consequently, the reduction of the data dimensionality might be a good idea. However, it is worth noting that a dimensionality reduction algorithm might lose some useful information. A lot of modern algorithms, such as neural networks or ensemble algorithms, handle high dimensional data very well, and dimensionality reduction techniques are used less than in the past. However, they are still used for data visualization and cases when we need to build an interpretable model while we are limited in theb number of algorithms we can use.

Anomaly (outlier) detection involves the detection of instances strongly deviating from the norm. These instances are called *outliers* or anomalies while regular ones are referred to as *inliers*. Anomaly detection has many applications. For example, it can be used as a data preprocessing step to remove outliers from the dataset, which might improve the performance of the resulting model. In addition, it is used in the *fraud detection* task and the detection of faulty products in manufacturing facilities.

Novelty detection is closely related to anomaly detection. The only difference is that novelty detection assumes that the training dataset was not contaminated by outliers while anomaly detection does not make this assumption.

3.2.1 Principal Component Analysis (PCA)

Principal components are vectors that define a new coordinate system. The first vector goes in the direction of the highest variance. The second vector is orthogonal to the first one and goes in the direction of the second highest variance, and so on. If we were to reduce dimensionality to $D_{new} < D$, we would pick D_{new} largest principal components and *project* instances onto them.

It is not advised to choose the number of dimensions arbitrarily. It is recommended to choose a number of dimensions that preserves a large amount of variance (e.g. 95%), or in case of visualization to reduce the number of dimensions down to two or three. There are different versions of PCA; kernel PCA, Incremental PCA (online or batch PCA), and Randomized PCA.

3.2.2 Gaussian Mixtures

Gaussian mixtures is a common algorithm that can be used for anomaly detection. Gaussian mixtures assume that the dataset is generated by several Gaussian distributions. Any instance lying in a region of low density is an anomaly. The density threshold has to be specified. If one gets too many false positives (good products labeled as faulty) they need to decrease the threshold. Consequently, if we get too many false negatives (faulty products labeled as good) the threshold has to be increased. Gaussian mixtures belong to soft clustering. Gaussian mixtures require the number of clusters to be specified. It needs to be run a couple of times to avoid suboptimal solutions.

3.3 Data Preparation

Due to factors such as curse of dimensionality and inherent noise, we cannot load raw data to an algorithm and expect good performance. Most often, the raw data has too many features and most of them have very little predictive power. We need to build a dataset first. *Feature engineering* is responsible

for transforming raw data into a dataset. It is a labor-demanding process that requires creativity and, most importantly, domain knowledge.

The objective of this stage is to create *informative* features or features with *high predictive power*. For example, in our task of predicting survival time, donor-recipient blood group compatibility or recipient's age is likely to have much higher predictive power than the donor's or recipient's citizenship.

Moreover, it is possible to create new features with higher predictive power out of those with low predictive power. For example, the calculation of *estimated Glomerular Filtration Rate (eGFR)*, the metric of kidney function estimated on a patient's age, gender, and serum creatinine level, could potentially give more information to the learning algorithm than all features separately.

In the following subsections, we will cover some popular feature engineering techniques.

3.3.1 Handling Categorical Features

The majority of machine learning algorithms primarily operate with numerical features. To handle categorical features (the ones with only a few possible values), such as the age group or a blood group, we can use *one-hot encoding* to convert them to several binary ones. For instance, let's consider a blood group feature comprised of four primary blood groups: A, B, AB, and O. We can convert each blood group into a vector of four numerical values:

$$\begin{aligned} A &= [1, 0, 0, 0] \\ B &= [0, 1, 0, 0] \\ AB &= [0, 0, 1, 0] \\ O &= [0, 0, 0, 1] \end{aligned}$$

This technique will increase the dimensionality of the dataset but this is a trade-off we have to make because if we were to assign a number to each group (1 to A, 2 to B, etc.), that would imply gradation or ranking among these categories, while there is none.

However, if the categorical feature does suggest some gradation, for example, university marks as "fail", "average", "good", or "excellent", an enumeration of each value will be appropriate. This practice of assigning a number to categories that have ranking is called *ordinal encoding*.

Binning (or *bucketing*) is the technique used for converting numerical values into multiple binary features called *bins* or *buckets*. For example, a patient's age can be transformed into age-range bins: 0 to 18 years old, 18 to 25 y.o., 25 to 40 years old, and so on. This technique might help an a laearning algorithm learn better, particularly with smaller datasets.

3.3.2 Feature Scaling

Different ranges of feature values might pose a problem to some machine learning algorithms as they do not handle them very well. It might result in a slower training time or a poorer performance. This problem is solved by *normalization* and *standardization* scaling techniques.

Normalization (also known as *min-max scaling*) is a technique of converting an actual range of numerical feature values into a standard range of values: $[-1, 1]$ or $[0, 1]$ without losing any information. The normalization formula for value $x^{(j)}$ for feature j, looks like the following:

$$\bar{x}^{(j)} = \frac{x^{(j)} - min(j)}{max(j) - min(j)},$$

where $min(j)$ and $max(j)$ are minimal and maximal values of feature j.

Standardization is a scaling technique that scales numerical data in such a way that after scaling, it has properties of the *standard normal distribution* with the mean μ=0 (average value) and the standard deviation from the mean $\sigma = 1$. The standardization formula for value $x^{(j)}$ for feature j, looks like the following:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}.$$

Typically, standardization is used for supervised learning, in case feature values are formed by standard distribution (bell curve) or a feature has outliers. In other cases, the normalization is preferred.

3.3.3 Handling Missing Feature Values

Datasets frequently have missing values and to handle them, we have one of the following options:

1. **Removal of rows with missing values**. The most direct and straightforward approach to managing missing data. If missing values are sparse or the dataset is large enough, the usage of this technique would be appropriate.

2. **Feature removal**. If the dataset has a feature with an excessive amount of missing values relative to its size, it is better to remove the feature.

3. **Regression imputation**. This technique implies the filling in a missing feature value with predictions of a machine learning regression algorithm.

4. **Mean/median imputation**. This method involves the filling of missing feature values with their mean or median value.

5. **Constant value imputation**. This technique entails the filling the missing values with clearly too high or too low values. The motivation is for the algorithm to discern the value as an outlier while considering other features. This method is not recommended as it can introduce bias.

It is often impossible to tell which data imputation method would work the best and therefore, it should be checked experimentally.

3.4 Model Training and Hyperparameter Tuning

It is a common practice to divide a dataset into three parts

- Training set (70% of the dataset)
- Validation set (15% of the dataset)
- Test set (15% of the dataset)

The training set, being the largest of them, is employed to train the machine learning model. Validation and test sets, which are of identical sizes and often called hold-out sets, are used in subsequent stages of model evaluation.

The rationale behind the use of separate training and validation sets is to prevent overfitting - a situation when the model performs well on the training data but poorly on the unseen data. Overfitting can occur if the model is tested and evaluated on the same dataset. As a result, the model may memorize the training examples and fail to make accurate predictions on the unseen data. To alleviate this, we use

the validation set to fine-tune the model, and the test set to assess its performance before deploying it to production.

A typical workflow involves training the model on the training set, validation on the validation set using the selected metric, then adjusting the model's parameters to improve its performance. This process is repeated until no substantial improvement is observed. Finally, the model's performance is assessed on the test set. This iterative process is referred to as hyperparameter tuning.

An alternative to the three-set technique is *k-fold cross-validation*. This technique involves splitting the dataset into k subsets, or folds, of equal size. One fold is used as a validation set, while the other k-1 folds constitute a training set. The model is trained exactly k times, with each fold serving as a validation set only once. The only drawback is that it is highly computationally demanding, particularly with a high k value and larger datasets, as the model will be trained k times.

A *hyperparameter* is a parameter specified before model training, in contrast to regular parameters that are calculated during training. Each model possesses a different set of hyperparameters and they profoundly influence the model's performance. The number of trees in Random Forest and the C hyperparameter in Support Vector Machines are examples of hyperparameters. The task of finding the optimal combination of hyperparameters is called hyperparameter tuning. One strategy might be to select hyperparameters manually and observe their impact on the performance. However, utilizing the grid search is a better way.

Grid search is a standard way of performing hyperparameter fine-tuning. It includes defining hyperparameters to experiment with, providing values for each hyperparameter to be tested, and training a model for each possible combination of hyperparameters. The performance of each individual model is assessed using k-fold cross-validation and the best combination of hyperparameters is selected. This approach is used in sci-kit-learn's implementation - GridSearchCV.

Grid search proves to be effective when dealing with relatively few hyperparameter combinations. However, with larger number of hyperparameter combinations, it is advisable to use RandomizedSearch (RandomizedSearchCV in sci-kit-learn). This method is very similar to grid search but instead of trying every possible combination of provided values, it tests only a specified number of randomly selected hyperparameter combinations. The primary advantage of this method over grid search lies in more control over computational power and the time dedicated to hyperparameter tuning.

3.5 Survival Analysis

Survival analysis, often referred to as *time-to-event analysis,* is a statistical technique employed to analyze and predict the time until an event of interest occurs. Its name originates from clinical and biological research, where these methods are used to analyze survival time, hence the name. These methods, however, found their uses in areas far beyond clinical settings: in business to predict the time until the customer "churns" from a subscription, in engineering, to estimate the product longevity or the longevity of their parts, in social sciences, estimate the longevity of a marriage or a student dropout rate in an academic setting.

In the context of survival analysis, *time* (also *survival time*) refers to the duration from the start of an individual's follow-up to the occurrence of an event, measured in days, weeks, months, or years [13]. The term *event* (also referred to as *deat*h or *failure*) encompasses any occurrence that permanently changes the state of the subject. It can be death, the onset of a disease, a relapse from remission, recovery, or any other specified experience of interest that an individual might encounter [13, 30]. Usually, only one event of interest is considered. When evaluating multiple events, the problem is categorized as *competing risks* or *recurrent events* problem [13].

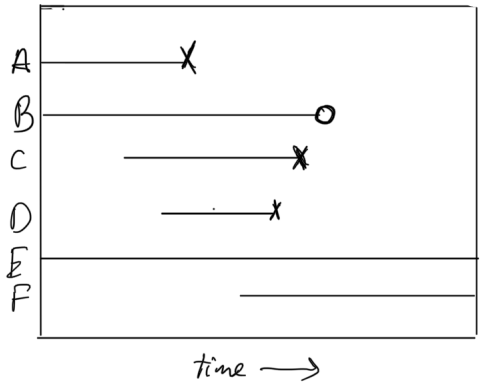


Figure 3.5: Censoring illustration

In this section, we will delve into fundamental survival analysis terminology, such as censoring and survival function. We will explore the classification of the survival analysis methods, highlighting the most popular statistical methods, as well as one machine learning method tailored to the needs of survival analysis. Additionally, we will discuss ways to evaluate survival models effectively.

3.5.1 Basic Terminology

In this subsection, we are going to cover basic terminology required for survival analysis such as censoring, censoring assumptions, survival, and hazard functions.

Censoring

The most distinct feature of survival analysis methods is the ability to handle censored data. *Censoring* refers to a circumstance when the information about survival time is only partially known. For example, the dataset utilized in our research has 370,000 censored instances out of 500,000 performed transplantations. These patients were either still alive at the last date of observation or were lost to follow-up. This lack of complete information indicates *censoring* in survival analysis. Censoring makes the application of standard statistical and machine learning approaches to survival data impractical [27].

Look at the Figure 3.5. On the y-axis, we can see individual patients, while the x-axis corresponds to the study timeline (the right side is the end of the study). Cross (X) denotes an occurrence of the event, and circle (O) corresponds to the patient's exit from the study.

There are three types of censoring: *left*, *right*, and *interval* censoring. *Right censoring*, which is more common, occurs when we are sure that the event did not happen by a specific time and we don't know when it will happen. The situation arises when the patient drops out of a study, or the study ends when they are still alive, as illustrated with patients B, E, and F in Figure 3.5.

Left censoring is less common and happens when the event occurs before the study begins or before the initial observation. We know the event happened before a specific time, but the exact time is unknown. This type is typical in cases where a patient has already experienced the event (e.g., developed a disease) before enrolling in the study.

Interval censoring happens when the event occurs within a particular timeframe, but the exact time is unknown. It can be the case in studies involving periodic patient follow-up, where the event can happen at any point between two visits.

Understanding right, left, and interval censoring is essential in survival analysis. We will next turn our attention to the assumptions associated with these censoring types. These assumptions are inherent to many survival analysis methods and are critical in selecting the appropriate technique.

Censoring Assumptions

There are three types of censoring assumptions: *random*, *independent*, and *non-informative*. Each shares certain similarities but also possesses unique distinctions, which we'll explore in detail. Censoring assumptions is the way censoring is managed.

1. **Random Censoring**: Subjects censored at time t are assumed to have the same failure rate as remaining subjects provided the same survival experience. Subjects that were censored are selected randomly, meaning the study does not influence or bias which participants are censored.

2. **Independent Censoring**: Independent censoring occurs when the censoring is random within certain subgroups, defined by certain covariates. If no covariates are present, it defaults to random censoring. This distinction might not be apparent when examining a single subgroup.

3. **Non-Informative Censoring**: Non-informative censoring occurs when censored instances do not provide any information on their survival prospects. In other words, whether or not the patient is censored has no influence on experiencing the event of interest.

Generally, it is safe to assume *non-informative* censoring when censoring is *independent* and/or *random*. However, these assumptions are not equivalent. To better understand these concepts, let's look into examples.

Consider a three-year disease occurrence study with 100 subjects at risk. By the end of the study, 20 of them contracted the disease, giving a 20% three-year disease risk. Suppose we want to extend the study for another two years on the remaining 80 individuals. However, 40 refuse to continue in the study and, are, therefore, lost to follow-up (censored). Of the remaining 40, 5 contracted the disease. Assuming those who left were representative of the remaining subjects (random and independent censoring), another 5 among the censored would have contracted it. Consequently, the five-year risk is 30% and the five-year survival is 70% under random and independent censoring assumptions. In this case, random and independent censoring is the same, as no predictor variables are considered.

To illustrate the difference between random and independent censoring, let us introduce another group to the study: group B (the group before is A) with 100 individuals. In the first three years, 40 contracted the disease, and 10 left the study. So, the calculated three-year risk for group B is 40%. In the next two years, 10 out of 50 get the disease, yielding 20% risk for years between 3 and 5. Under the independent censoring assumption, we assume that out of 10 censored, 2 contracted the disease. The five-year risk for group B is 52% with 48% survival under independent censoring assumptions.

As we can see, the five-year risk in the two groups differs significantly (30% against 52%), and the censoring proportion is also very different (50% against 17%). Hence, the overall censoring is not random. However, it is random within each group, so the censoring is independent. If, instead, in group B, 30 subjects out of 60 were censored at the three-year mark, the censoring proportion would be the same in both groups, and the overall censoring would be random, as those censored would be the representatives of those who remained at risk.

To best illustrate what non-informative censoring is, let us demonstrate informative censoring. Let us take a group of subjects under random and independent censoring assumptions. Every time subject A gets an event, subject B leaves the study (e.g., B is A's relative). If the censored subjects are representative of subjects at risk it would be random and independent censoring. Here, the censoring mechanism is directly related to event occurrence, so the censoring is informative.

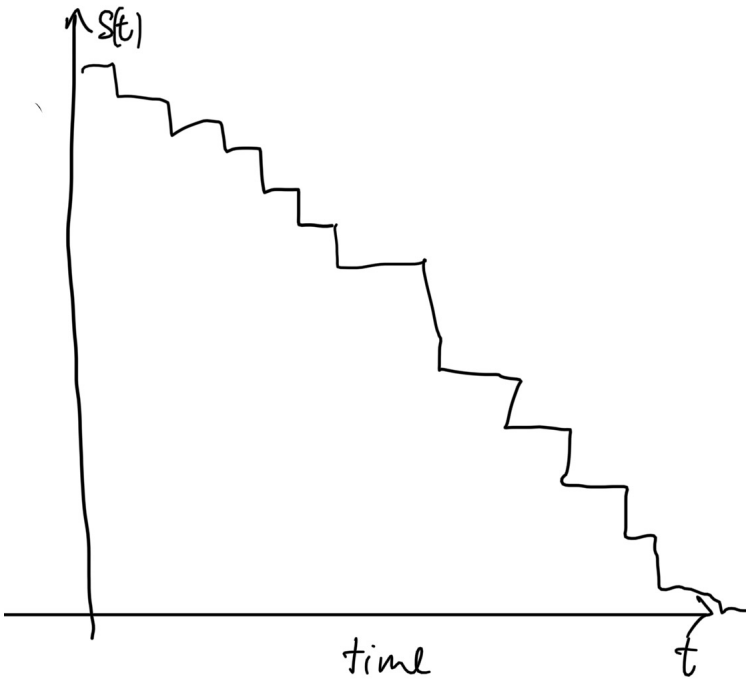


Figure 3.6: Survival function

Survival Function

The *survival function*, also known as the *survivor function*, denoted by $S(t)$, represents the probability that the patient survives, in other words, does not experience the event of interest beyond a given time t. Mathematically, it is denoted by:

$$S(t) = P(T > t). \tag{3.9}$$

Where P represents the probability, t is any specific time of interest, and T is the random variable for the subject's survival time. For instance, if we want to know the likelihood that a patient is going to live for more than five years after a kidney transplant, we set t equal to 5, and we evaluate $S(t)$ to determine the probability that T, actual survival time, is greater than 5 years.

The survival function has several key characteristics:

1. It continually decreases or maintains its value over time, theoretically extending from 0 to infinity. That is, if the study lasted indefinitely, the survival function would eventually fall to 0. In practical research scenarios, studies do not last forever, and not every patient experiences an event by the end of the study.

2. As it represents a probability, the function value ranges from 0 to 1. It starts at 1, at the beginning of the observation period, indicating 100% survival probability, and declines over time, potentially reaching 0, as the probability of survival decreases.

3. Although, by definition, the graph of the survival function is smooth, in reality, it is a step function. This stepwise representation is due to the nature of real-world data, where events are recorded at specific, discrete time points rather than continuously [13].

Survival function $S(t)$ gives a comprehensive understanding of the probability of an individual surviving beyond a specific time point. It presents a declining or constant trend starting at 1 and potentially declining to 0. While understanding the probability of not failing at every time point is important, knowing the instantaneous risk of failing is useful in some cases, leading us to the next tool: the hazard function.

Hazard Function

The survival function provides the probability of an individual surviving at each given point in time. This function is often preferred over the *hazard function* due to its more intuitive appeal: it directly communicates the chance of survival. However, there are scenarios where knowing the risk at each point in time is necessary, entailing the use of the hazard function.

The *hazard function*, denoted as $h(t)$, represents the instantaneous potential per unit of time for the event to occur, provided the subject's survival up to time t. It is expressed by the following equation:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}. \tag{3.10}$$

Here, Δt represents an infinitesimally small increment of time. The function $h(t)$ is equal to the limit, as Δt approaches zero, of a conditional probability, divided by Δt. The conditional probability statement gives the probability that a person's survival time, T, will lie in the time interval between t and $t + \Delta t$, given that the survival time is greater than or equal to t.

Occasionally, due to its structure, the hazard function is referred to as a *conditional failure rate*. It is a rate because it represents a conditional probability per unit of time Δt, and it is conditional on the subject surviving until time t. Unlike probability, this rate has a scale from 0 to infinity — depending on the measure of time in days, weeks, or years. Essentially, by considering the limit as the Δt approaches zero we basically get the instantaneous potential of failing at time t, given survival until that moment.

To best illustrate the concept of instantaneous potential, let us refer to the concept of velocity. Velocity gives us the speed at a specific point in time. For instance, the velocity of 80 km/h, means that maintaining the same speed for an hour would result in traveling 80 kilometers. However, it doesn't predict how much the car will travel in reality. The same works with the instantaneous potential: it might be high at one point, but low at another, reflecting that both are measurements at an instant in time rather than an interval [13].

The cumulative hazard function further extends our understanding by quantifying the accumulated risk over time. It is an area under the hazard function that allows us to say which group has a greater risk. It is defined by the following function:

$$H(t) \stackrel{def}{=} \int_0^t h(u)du, \text{ where } t > 0. \tag{3.11}$$

Where $h(u)$ represents a hazard function. This integral measure enables a comprehensive comparison of risk between groups, showing which has a greater risk from the perspective of accumulated potential for the event to occur over time.

The relationsip between the survival and hazard functions

Some models, such as Cox Proportional Hazards, are written in terms of the hazard function, and it is necessary to be able to estimate the survival function out of the hazard function to make the model more flexible and accessible to a larger audience. Fortunately, there are ways to convert one into the other and vice versa. In this subsection, we are going to cover ways to do that.

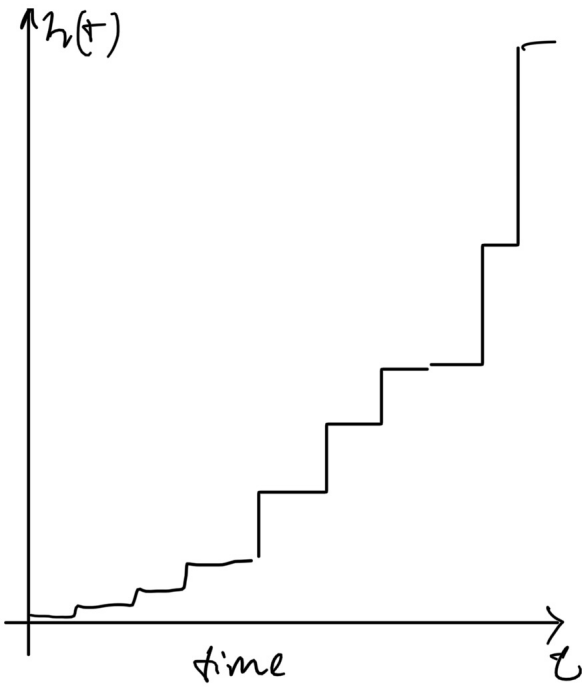


Figure 3.7: The example of a hazard function. Note that it may signifficantly differ.

Let us start by considering how to get the survival function $S(t)$ from the hazard function $h(t)$. The relationship is captured in the following equation:

$$S(t) = exp\left[-\int_0^t h(u)du\right]. \tag{3.12}$$

This equation 3.12 illustrates that the survival function $S(t)$ is equal to the exponential of the negative cumulative hazard function from zero to t. As we can see, the integral in the equation is the $H(t)$ from the 3.11. And we can simplify our equation to:

$$S(t) = e^{-H(t)}$$

On the other hand, to obtain the hazard function from the survival function, we can utilize the following relationship:

$$h(t) = -\left[\frac{dS(t)/dt}{S(t)}\right]. \tag{3.13}$$

Here, the Equation 3.13 denotes that the hazard function $h(t)$ is the negative derivative of the survival function $S(t)$ with respect to time t divided by $S(t)$ itself.

Considering the fact that the survival function describes the probability of a patient surviving up to time t and the hazard function shows us the instantaneous risk of a person dying at a specific instant t, we can say that they provide complementary information about survival and risk over time [13]. Of the two discussed functions, the survival function is used more often as it is more intuitive, and in the practical part, we will estimate the survival function as well. Fortunately, we will not convert them manually. The *scikit-survival* library (more on it in the section 3.7) will do that for us.

3.5.2 Taxonomy of Survival Analysis Methods

Survival analysis methods can be broadly categorized into *statistical* and *machine learning based* methods. Both aim to estimate the survival time and the survival probability at the estimated survival time [23]. Statistical methods primarily characterize distributions of the event times and the statistical properties of the parameter estimation by estimating the survival curves. Typically, statistical methods are employed for low-dimensional data [23].

On the other hand, machine learning methods focus on the prediction of the event occurrence at a given time. They harness the strengths of traditional survival analysis while integrating different machine learning techniques, leading to more potent algorithms. Machine learning methods are mostly used with high-dimensional data [23].

Statistical methods can be further subdivided based on their assumptions and parameter usage into parametric, non-parametric, and semi-parametric methods. Machine learning methods, encompassing methods like survival trees, ensembles (random survival forests), neural networks, and support vector machines form a distinct category. Advanced machine learning techniques, such as active learning, transfer learning, and multitask learning are included as well [23].

In this section, we will cover selected statistical methods and one machine learning technique. Machine learning techniques related to neural networks will be covered in Section 3.6, dedicated to these advanced survival analysis techniques.

3.5.3 Statistical Methods

This section provides a concise overview of statistical techniques, given our primary emphasis on machine learning methods. Here, we introduce three different types of statistical methods that are commonly used to estimate the survival and hazard functions: *non-parametric*, *semi-parametric*, and *parametric methods*.

Non-parametric methods are preferred in situations where the event time does not adhere to any known distribution or when the proportional hazards assumption is not met [23]. There are three main non-parametric methods: the Kaplan-Meier (KM) method, the Nelson-Aalen (NA) estimator, and the Life-Table (LT) method. In the next section, we will cover Kaplan-Meier in more detail. The Life-Table method is more convenient than Kaplan-Meier for the estimation of survival curves when data subjects are segmented into distinct time intervals when dealing with an extensive number of subjects or a broad population scope. On the other hand, the Nelson-Aalen method is used for the estimation of hazard functions [23].

Semi-parametric models offer a middle ground between fully parametric models, which make specific distributional assumptions, and non-parametric models, which make very few assumptions. Among semi-parametric methods, the Cox model is the most frequently employed model for survival regression analysis. It is semi-parametric as the distribution of the outcome is unknown. Unlike other approaches, this method is based on the proportional hazards assumption and uses partial likelihood for parameter estimation (more on that in 3.5.3.2). There are a couple of variants of the basic Cox model: the penalized Cox model, which will be used in the practical part of this work, the CoxBoost algorithm, and the Time-Dependent Cox model [23].

Parametric methods shine in their accuracy and efficiency when the time to the event conforms to a known distribution that can be specified in terms of certain parameters. With parametric models, estimating the time to the event is straightforward, whereas the Cox model can make this task somewhat cumbersome or unfeasible. In the domain of parametric models, linear regression is central. However, the Tobit model, Buckley-James regression, and penalized regression are the most favored. Beyond this,

other parametric models like the Accelerated Failure Time (AFT) have gained traction. The AFT model represents survival time as a function of covariates [23].

To conclude, statistical methods in survival analysis can be broadly categorized into non-parametric, semi-parametric, and parametric techniques, each with its unique strengths and applications. Among non-parametric methods, Kaplan-Meier stands out as particularly significant. It provides a robust and intuitive way to estimate survival functions. The following section delves deeper into the Kaplan-Meier method.

3.5.3.1 Kaplan-Meier Survival Curves

Kaplan-Meier is a non-parametric method of survival function creation. It is *non-parametric* because it does not take into account any covariates, or parameters, and requires only the survival time and the censoring indicator. It works under an independent censoring assumption. The general Kaplan-Meier formula for plotting the survival function is the following:

$$\hat{S}(t_{(f)}) = \hat{S}(t_{(f-1)}) \times \hat{P}(T > t_{(f)} | T \geq f_{(f)}). \tag{3.14}$$

That can be read as the survival function $\hat{S}$ of time $t_{(f)}$ is equal to the probability of surviving past the previous time point $t_{(f-1)}$ times the conditional probability of surviving past the time $t_{(f)}$ [13]. This function can also be expressed as a product limit:

$$\hat{S}(t_{(f)}) = \prod_{i=1}^{f-1} \hat{P}(T > t_{(i)} | T \geq f_{(i)}).$$

In the Equation 3.14, we replaced $\hat{S}(t_{(f-1)})$ with the product of all fractions estimating the conditional probabilities for failure times t_{f-1} and those preceding it [13]. Because of that the Kaplan-Meier estimator is sometimes referred to as the *product-limit method [23].* Figure 3.8 shows the survival curve created with the KM method.

These survival curves are often compared using the log-rank test. The *log-rank test* is a way to compare two survival functions, that is often used in studies, where there is a target group and a placebo (control) group to assess the efficacy of the treatment or intervention in the study by comparing the survival curves of the two groups [13].

To summarize, the Kaplan-Meier method provides a robust non-parametric approach for estimating survival functions, emphasizing its independence from covariates. While the log-rank test offers a mechanism to compare the survival curves and assess treatment efficacy, there are still situations that require taking covariates into consideration. That leads us to the Cox proportional hazards model, which inherently handles covariates.

3.5.3.2 Cox Proportional Hazards Method

The *Cox proportional hazards* (also Cox PH) model is widely used in survival analysis semi-parametric model. This section explores its formulation, key properties, and reasons for its widespread use in research.

The Cox proportional hazards model is defined in terms of a hazard at time t for a subject with a given vector of explanatory variables $\mathbf{X}$:

$$h(t, \mathbf{X}) = h_0(t) \times exp\left[\sum_{i=1}^{p} \beta_i X_i\right], \tag{3.15}$$

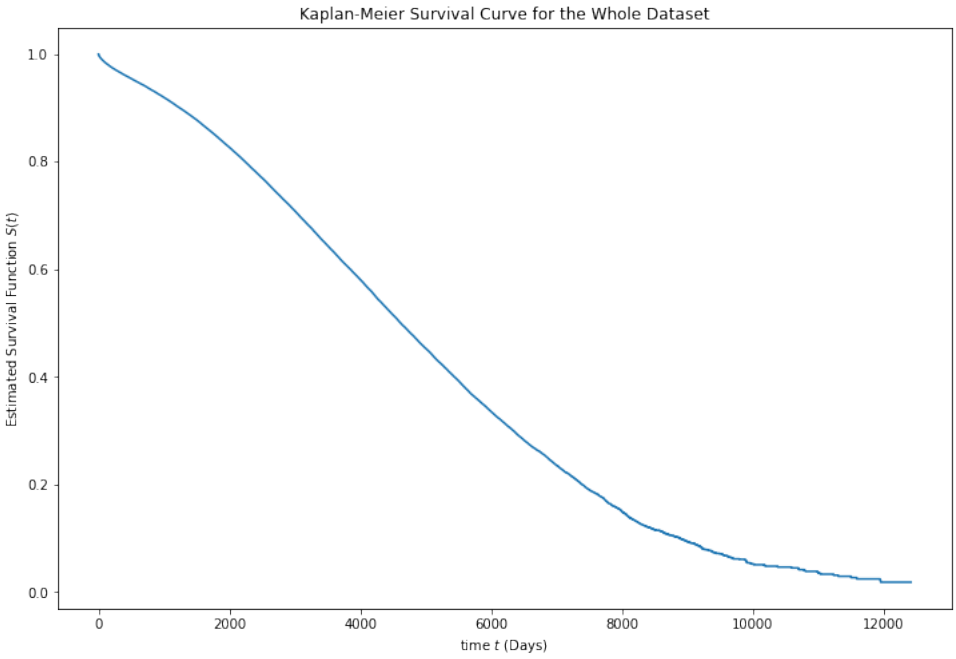


Figure 3.8: Survival curve estimated with KM

where $h_o(t)$ stands for the *baseline hazard function*. Coefficients β are the parameters of interest in the model. Since the exponential function has no t, $\mathbf{X}$ is called *time-independent.* The exponential function ensures that the function is non-negative, satisfying the definition of the hazard function[13].

Even though Equation 3.15 contains the baseline hazard function, the function is not specified. Fortunately, we can calculate the hazard ratio, a measure of effect, without having to estimate the baseline hazard function. Similarly, the hazard function $h(t, \mathbf{X})$ and the survival function $S(t, \mathbf{X})$ can be estimated without the baseline function. So, with minimal assumptions, we can estimate everything we need (h, S, and HR).

The *hazard ratio* (HR) is a measure of the influence of an intervention on the outcome. A hazard ratio is defined as the hazard for one individual divided by the hazard for the other, as illustrated with the following formula:

$$\hat{HR} = \frac{h(t, X^*)}{h(t, X)} = exp\left[\sum_{i=1}^{p} \hat{\beta}_i (X^* - X)\right] \tag{3.16}$$

As can be seen, the equation does not contain t and the basic hazard function, as they are canceled out, making it a *proportional hazard assumption*.

Like logistic regression, the CoxPH uses the *maximum likelihood* function 3.5 to calculate its parameters ($\hat{\beta}_i$). However, since the maximal likelihood considers only a part of patients, namely those who experienced an event, the formula is called ***partial*** *likelihood* [13].

Let us define the partial likelihood. The subject's survival can be defined by $h(t, X)dt$ with $dt \to 0$. Consider J (where $J \leq N$) as the total number of events of interest observed for N instances. $T_1 < T_2 < \cdots < T_J$ represents the unique and sequentially ordered times to the event of interest. Let X_j be the vector of covariates for a subject who experiences the event at time T_j. R_j is the set of risk subjects at T_j. Given that the event happens at time T_j, the individual probability associated with X_j can be described as follows:

$$\frac{h(T_j, X_j)dt}{\sum_{i \epsilon R_j} h(T_j, X_i)dt}.$$

By taking the product across all subject's probabilities we get the partial likelihood. Based on Cox assumption and the presence of censoring, partial likelihood is defined as follows:

$$PL(\beta) = \prod_{j=1}^{N} \left[\frac{exp(X_j\beta)}{\sum_{i \epsilon R_j} exp(X_i\beta)} \right]^{\delta_J}. \tag{3.17}$$

δ_j is the indicator of censoring of a given instance (1 for event occurrence, 0 for censoring). Therefore, if the subject is censored, $\delta = 0$, the individual probability is equal to 1 and the subject does not affect the result. The vector of coefficients $\hat{\beta}$ is estimated by either maximizing partial likelihood, defined above, or maximizing the negative *log-partial likelihood* to improve the efficiency:

$$LL(\beta) = -\sum_{j=1}^{N} \delta_j \{X_j\beta - log[\sum_{i \epsilon R_j} exp(X_i\beta)]\}.$$

[23]

Being a semi-parametric, this model is a safe choice because it consistently delivers a sufficiently reliable result. The risk of choosing the wrong model as often happens with parametric models, is practically non-existent. However, if one is sure that a parametric model suits the problem, one should use the parametric model [13].

3.5.3.3 Penalized Cox Models

The Cox proportional hazards model is often chosen, since its coefficients can be interpreted in terms of hazard ratios, offering meaningful insights. Yet, when estimating coefficients for many features, the standard Cox model collapses due to matrix inversion getting disrupted by correlations between features. Feature (column) correlations make a matrix singular, i.e. impossible to revert. In this section, we aim to explore the *Ridge regression*, *LASSO*, and *Elastic Net* methods as extensions or modifications to the Cox model. These techniques address the inherent challenges in the standard Cox model, offering solutions for regularization and feature selection.

Ridge By incorporating an l_2 penalty term on the coefficients, shrinking them to zero, we can avoid the problem of the inability to revert singular matrix. Consequently, our objective has the following form:

$$arg \max_{\beta} \log PL(\beta) - \frac{\alpha}{2} \sum_{j=1}^{p} \beta_j^2.$$

In the equation, $PL(\beta)$ denotes the partial likelihood of the Cox model (3.17), terms $\beta_1, \ldots, \beta_p$represent the coefficients corresponding to p features, $\alpha \geq 0$ is a hyper-parameter that controls the amount shrinkage. The resulting objective is referred to as *ridge regression.* If α is set to zero, we get the regular Cox model [25].

LASSO While the l_2 ridge penalty solves the mathematical problem of fitting the Cox model, we would still need to take into account all features, no matter how many there are. Preferably, we would like to select a small subset of the most predictive features and ignore the rest, as too many features might result in overfitting. *LASSO* (Least Absolute Shrinkage and Selection Operator) does exactly that. Rather

than merely shrinking the coefficients to zero, it performs feature selection as a part of the optimization process, where a subset of coefficients is set to zero and is, therefore, excluded, reducing the number of features we would need for prediction. Mathematically, we would replace the l_2 penalty with the l_1 penalty, leading to the following optimization problem:

$$arg\max_{\beta} \log PL(\beta) - \alpha \sum_{j=1}^{p} |\beta_j|.$$

The main drawback is that we cannot directly control the number of features selected. However, the value of α essentially determines the number of features selected. To achieve a refined model that requires fewer features, we need a data-driven way to determine the appropriate α. This can be accomplished by first determining the α that would ignore all features (coefficients are set to zero) and then gradually decreasing its value, possibly down to 1% of its starting value. Fortunately, it is implemented in scikit-survival's *sksurv.linear_model.CoxnetSurvivalAnalysis.* We would need to set *l1_ratio=1.0* and *alpha_min_ratio=0.01* to search for 100 α values up to 1% of the estimated maximum [25].

Elastic Net The LASSO method is effective for selecting a subset of discriminative features. However, it is not without its shortcomings. The first is that LASSO is unable to select more features than there are instances in the training data set. The second is its tendency to randomly select only one feature out of a set of highly correlated ones. The *Elastic Net* alleviates these issues by incorporating the $l_1$and l_2 penalties in a weighted manner, as is shown in the following optimization problem:

$$arg\max_{\beta} \log PL(\beta) - \alpha \left(r \sum_{j=1}^{p} |\beta_j| + \frac{1-r}{2} \sum_{j=1}^{p} \beta_j^2 \right),$$

where $r\epsilon[0;1]$ is the relative weight of the $l_1$and l_2 penalty ($r = 1$ is a LASSO penalty, $r = 0$ is a ridge penalty). The Elastic Net penalty combines the LASSO's feature selection capability and Ridge's regularization power. As a result, it is more stable than LASSO, and in a situation of highly correlated features, it would select them all, while LASSO would randomly select only one. Usually, it is sufficient to give the l_2 penalty only a small weight to improve the stability of the LASSO, for example, $r = 0.9$ might suffice.

Similar to LASSO, the weight α inherently dictates the size of the chosen subset. Its optimal value is typically estimated in a data-driven way [25]. More on that in the practical part, where we will choose the best α for our data set.

To conclude, Ridge, LASSO, and Elastic Net offer powerful extensions of the Cox PH model, especially when dealing with high-dimensional datasets and highly correlated features. They help to regularize the Cox model and select only the most significant features, avoiding overfitting and assuring a more stable and interpretable model. Nevertheless, it is important to acknowledge their limitations. As linear models, they inherently capture only linear relationships, and non-linear relationships are enigma to them. That leads us to the next section dedicated to the Random Survival Forests, a machine learning approach. Machine learning techniques are known for their ability to catch complex non-linear relationships, often outperforming traditional statistical methods in predictive accuracy.

3.5.4 Random Survival Forest

Random Survival Forest (RSF) is an ensemble machine learning method tailored for survival analysis. It is derived from the original Random Forest method developed by [28]. Random Forests have gained popularity in the machine learning community due to their effectiveness in classification and regression

machine learning tasks. Random Forest is built from multiple decision trees, averaging their predictions allows for more accurate predictions than any single tree can provide. Similar to Random Forest, Random Survival Forest leverages the power of multiple survival trees making its predictions more robust.

Random Survival Forest is comprised of *Survival Trees*. A survival tree is essentially a binary tree. It is developed by the iterative splitting of children nodes into two nodes until a certain criterion is met. An optimal node split is the one maximizing the survival difference between the children nodes. It is done by iterating through all features and finding such a value for a given feature that maximizes the survival difference [26].

Random Survival Forest training is comprised of the following steps:

1. Randomly draw L bootstrap samples of size n with replacement from the training dataset. n is about two-thirds of the original data. The remaining instances, about one-third, are termed out-of-bag (OOB) observations and are not included in the bootstrap sample.

2. For each sample, develop a full-grown survival tree based on the selected splitting criterion. At each node, randomly select $\sqrt{p}$ (or other number) covariates, where p is the total number of covariates. Stop developing when a certain condition is met (for example, when the terminal node has fewer observations than a predetermined threshold or the node reaches purity).

3. For each tree, calculate the cumulative hazard function with the Nelson-Aalen estimator. Find a mean of all trees to find ensemble CHF.

4. Use OOB data to determine the prediction error of the ensemble [27].

In summary, the Random Survival Forest is a robust and powerful approach to survival analysis that harnesses the collective power of multiple survival trees. Having explored several survival analysis methods, it is crucial to be able to evaluate them properly, since the standard machine learning performance metrics are not always applicable to survival analysis due to the presence of censoring. The next section will introduce essential criteria for the evaluation of survival analysis models and the primary performance metrics created for this purpose.

3.5.5 Performance Metrics

Survival prediction models play a vital role in healthcare. They are often used to estimate the risk of developing a particular disease and are crucial in guiding the clinical management of patients. It is, therefore, essential to assess their performance accurately. Similar to machine learning, this process of model evaluation is referred to as model validation. There are three aspects we can asses our model on:

1. **Overall performance,** which is the distance between the predicted and observed survival time.

2. **Discrimination,** or the model's ability to distinguish between high- and low-risk patients.

3. **Calibration** is the agreement between the observed and predicted survival times [21].

The absence of bias in a situation when the validation set contains censored instances is a sign of a good performance measure. Otherwise, in the presence of high levels of censoring, the evaluation would be unreasonably optimistic [21].

In this section, we will cover three measures of discrimination and one measure that assesses both discrimination and calibration (overall performance).

3.5.5.1 Harrel's and Uno's Concordance Indices

One way to measure discrimination is through *concordance*. A pair of patients is *concordant* if the subject who experiences an event earlier has a greater estimated risk [30]. *Measures of concordance* quantify the rank correlation between the predicted risk and the observed survival times. Typically, their values range between 0.5 and 1. A value of 0.5 indicates no discrimination, while 1 corresponds to the ideal discrimination [21].

The *concordance index*, or *C-index*, is the most widely used performance metric in survival analysis. It is defined through the concordance probability. The *concordance probability* is the probability that from the arbitrarily selected pair of patients (i, j), the one with a shorter survival time T, has the higher predicted risk M [21]. Mathematically, this is expressed as:

$$C = P(M_i > M_j | T_i < T_j).$$

Harrell's concordance index is the most widely used implementation of concordance index. To compute Harrel's concordance index C_H, we consider every comparable pair of patients where the one with the shorter time failed. Pair is "comparable" if we can determine which of them experienced the event first [30]. C_H is estimated as the proportion of these pairs in which the subject with the shorter survival time has a higher estimated risk. A modified version of this estimator, $C_H(\tau)$, only considers patients with $T_i < \tau$ and may provide more stable estimates [21].

Mathematically, Harrell's C-index is defined as a ratio between the number of concordant and comparable pairs:

$$\hat{C} = \frac{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} \left[I(T_i^{obs} < T_j^{obs}) + (1 - \Delta_j) I(T_i^{obs} = T_j^{obs}) \right] \left[I(M_i > M_j) + \frac{1}{2} I(M_i = M_j) \right]}{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} \left[I(T_i^{obs} < T_j^{obs}) + (1 - \Delta_j) I(T_i^{obs} = T_j^{obs}) \right]}. \quad (3.18)$$

where $I(\cdot)$ is the indicator function. It is equal to 1 if its argument is true, and 0 if it is not. T^{obs} is the observation time. Δ_i is a binary variable and $\Delta_i = 1$ if the subject i experienced an event during the time of observation and $\Delta_i = 0$ if they did not.

According to the Scikit-survival's documentation [22] and Rahman et. al. [21], Harrell's concordance index becomes biased in the presence of censoring. The bias increases with the level of censoring. Uno et. al. [31] introduced a modified concordance index, $C_U(\tau)$, which incorporates weights based on the probability of being censored. While their estimator proved robust to the choice of τ, they noted that the error of the estimate might be quite large if there are too few instances beyond this time point [21, 22].

Having explored the concordance index and its modifications, it is evident that it provides valuable insight into the model's discriminatory abilities. Yet, it offers a singular value that characterizes a model's performance across different time points without considering fluctuations in concordance that tend to happen over time. To enhance our evaluation of model performance in terms of discrimination across varying time points, we turn our attention to another popular metric in survival analysis: the ROC AUC. The subsequent section explores it in detail.

3.5.5.2 Time-dependent Area under the ROC

The *area under the receiver operating characteristic curve* (ROC AUC) is a popular performance measure for binary classification tasks. In survival analysis, it is used to determine how well estimated risk scores can distinguish diseased patients from healthy ones [22].

In binary classification, the *receiver operating characteristic (ROC)* is a curve that plots the *true positive rate (TPR or sensitivity)* against the *false positive rate (FPR)*. The FPR is the ratio of negative

instances that are falsely classified as positive. It is equal to 1− the *true negative rate* (the ratio of negative instances that are correctly classified, often referred to as *specificity*). TPR, or sensitivity, represents the ratio of positive instances classified as positive [9]

In survival analysis, we extend the ROC to continuous outcomes, where a patient is alive at the start of the observation, but might experience an event at some point later. Specificity and sensitivity, therefore, become time-dependent measures. It is specifically important, as model accuracy tends to be different at different points in time. Here we consider *cumulative cases* and *dynamic controls* at any given point in time t. *Cumulative cases* are all subjects who experienced an event prior to or at time t, while *dynamic controls* are those who are yet to experience the event after time t. By calculating the ROC AUC for any given time point t, we assess the model's ability to differentiate between patients. Specifically, how well the model can distinguish patients who fail by a given time $t_i < t$ from subjects who fail after this time $t_i > t$. The time-dependent ROC AUC is especially useful when we want to predict an event happening in a period up to time t, rather than at a specific time-point t [22].

While the time-dependent ROC AUC provides a valuable measure of a model's discrimination ability at various time points, it falls short in providing insight into the accuracy of individual predictions – calibration. To address this limitation and provide a more comprehensive model assessment, we turn our attention to the time-dependent Brier score.

3.5.5.3 Time-dependent Brier Score

Time-dependent ROC AUC and concordance index are great for assessing the overall discrimination among all time points (mean AUC and c-index) and the discrimination at any individual time point (the ROC graph), but they tell us nothing about the accuracy of individual predictions [22]. A metric analogous to regression performance measures used in machine learning would be ideal. Fortunately, such a metric exists. *Time-dependent Brier score* is a modification of mean squared error (MSE) that handles right censored data.

While the concordance index and time-dependent ROC AUC measure only discrimination, the time-dependent Brier score measures both discrimination and calibration, making it a metric of "overall performance". It is defined by the following equation:

$$BS^c(t) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \leq t \wedge \delta_i = 1)\frac{(0 - \hat{\pi}(t|\mathbf{x}_i))^2}{\hat{G}(y_i)} + I(y_i > t)\frac{(1 - \hat{\pi}(t|\mathbf{x}_i))^2}{\hat{G}(t)}$$

where $\pi(\hat{t}|\mathbf{x})$is a model's predicted probability of remaining event-free up to the time point t for feature vector $\mathbf{x}$, and $\frac{1}{\hat{G}(t)}$ is the inverse probability of censoring weight [22]. $I(\cdot)$ is the indicator function. y_i is the subject's survival time, δ_i is the event/censoring indicator.

The limitation of the time-dependent Brier Score is its applicability exclusively to the models that are capable of estimating the survival function. *Integrated Brier Score* provides a scalar value for general model evaluation. It is beneficial for model comparison, as time-dependent measures are a bit harder to compare than scalar values, and for the model fine-tuning process.

Survival analysis provides a suite of techniques for the prediction of a time to event. As we explored its methods and evaluation metrics, it is evident that the traditional SA provides robust tools for handling censored data, evaluating risk factors, and making predictions over time. However, as data grows in both complexity and volume, there is a growing need for more advanced modeling techniques, leading us to advanced machine learning: deep learning. Deep Learning, with its ability to handle large and complex datasets, promises to expand traditional survival analysis methods. In the following section, we will explore how deep learning can be adapted for survival analysis to harness the power of neural networks for more precise predictions.

3.6 Deep Learning

Feed Forward Neural Networks
Recurrent Neural Networks
Generative Models for Survival Analysis

3.7 Overview of Machine Learning Libraries and Tools

for the survival analysis the python library scikit-survival was used

3.7.1 Comparison

3.8 Conclusion

Chapter 4

Data Preparation and Analysis

In this chapter we age going to look into the UNOS dataset. Make sense of the dataset. Explore important features and their relationship with each other. Look into survival time for

The dataset provided by the IKEM (Institute of Clinical and Experimental Medicine in Prague) that I had from the beginning was not suitable for any meaningful analysis. That is why it was decided to look for the dataset elsewhere.

The data reported here have been supplied by the United Network for Organ Sharing as the contractor for the Organ Procurement and Transplantation Network. The interpretation and reporting of these data are the responsibility of the author(s) and in no way should be seen as an official policy of or interpretation by the OPTN or the U.S. Government.

The dataset is not for the open use. If you are interested in testing the results achieved in this paper, you need to acquire the data first. The requirements for the data acquirement are written here

The dataset consists of 993 806 of records for both transplanted patients and ones from teh waiting list, and 450 features comprised of waiting list data and already transplanted patients for kidney and pancreas transplant from October 1, 1987 to the present. Kidney transplants have 490 172 records.

The following features will be considered.

4.1 Data Loading

The data were provided in a form of a MongoDB database dump. It is impossible to perform data analysis with the database dump. So it was necessary to run the database first and import the database

Feature description	Type	Abbreviation
Donor Age		
Recipient Age		
Donor Type		
Donor gender		
Recipient gender		
Donor blood group		
Recipient blood group		
Recipient on dyalisis		
Recipient creatinine at the time of tx		

Table 4.1: Features

dump there. We set up the database in a Docker container (Docker is container management system + **explain what is container**) locally on my Mac, as the university cluster unfortunatelly does not have Docker. The data from the database and table kidpan were then exported to CSV, compressed into zip and uploaded to the cluster.

The pandas DataFrame method read_csv() loaded data for too long to work comfortably (5 minutes), as the CSV file had the size of 80GB, so it was decided to use parquet file instead. It was done by dumping the pandas DataFrame into Parquet database file using DataFrame.to_parquet() method. Parquet is used for efficient cloud computing. It provides more efficient way of loading data, as it works on the principles of databases, so the loading time of the whole dataset was decreased to 38 seconds. Additionally, it allows for specifying what columns to load, reducing the data loading time to 21 seconds. Thus using this technology has signifficantly improved the workflow.

4.2 Data preprocessing pipeline

In this section I will describe the data pipeline that I use to create the dataset out of the raw data. The pipeline can be found in github repository of this paper: survival_pipeline.py.

The work with the pipeline is pretty straightforward: we initialize the class and call the *load()* method. As is shown in the following block of python code:

```
1 from surv_data_pipeline.survival_pipline import
     ScikitSurvivalDataLoader
2
3 loader = ScikitSurvivalDataLoader()
4 X, y = loader.load()
```

Two main constants of the class are *categorical_values* and *numerical_values.* Categorical and numerical features must be specified there. It is important for following preprocessing steps.

The main method of the class *ScikitSurvivalDataLoader* is *load().* This method loads the data into the pandas DataFrame, applies exclusion criteria (more on that later), handles NaN values and returns X and y, X being numerical (categorical values were handled with OneHot encoding and numerical values were scaled) and y having format of (PSTATUS, PTIME), first one is the boolean censoring indicator (True - event happened, False - otherwise), PTIME is the number of days survived. This format is required by the Scikit-survival Library to build survival estimators.

The first step is to load the data into pandas DataFrame from the parquet file. Fortunately, pandas has support for this kind of files. It is performed by the Pandas method *read_parquet(path, engine, columns).* In *path* we need to specify the path to the parquet file, *engine* specifies what parquet library should be used, I use 'auto', it tries *pyarrow* if it doesn't work it uses *fastparquet.* In *columns* we need to specify the columns we want to load. (explain more what is pyarrow and parquet)

Description of feature engineering step:

The next step is to divide the dataset into training, validation and test sets, the reasons behind that, were explained in the datapreprocessing section of the previous chapter. These sets are then assigned as class variables to the class and are sent to preprocessing method *_handle_nan(),* where the NaN values are filled with median, specific value, or examples with such values are deleted with the pandas DataFrame method *drop_na(),* depending on the *fill_na_with_median* boolean parameter.

After the NaN handling step, the training set is send to the method *_get_X_y()* where the numerical values are standartized and categorical are encoded with the OneHot encoding with the Scikit-Survival methods *standardize()* and *encode_categorical().* Numerical and categorical values then comprise the *X* set, directly used in the training. The target value set is constructed with *Surv.from_arrays()* utility that

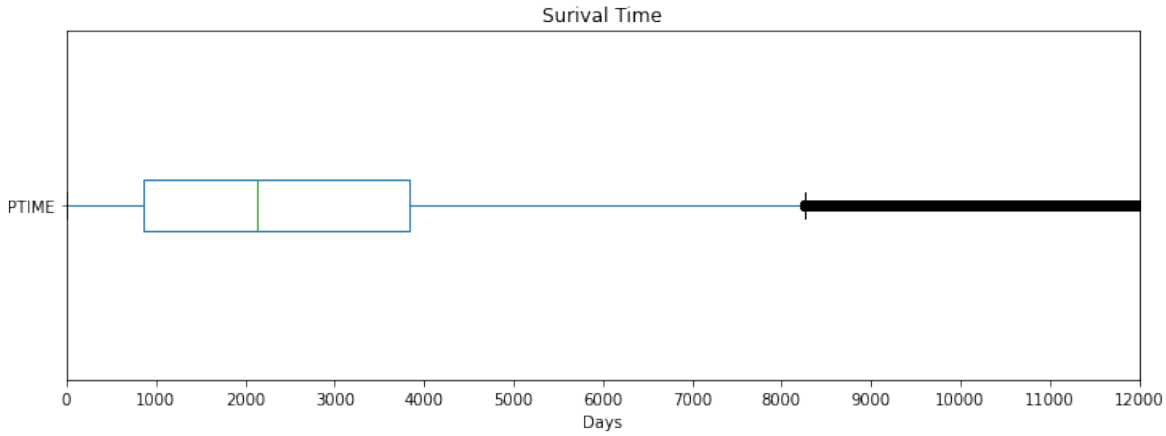


Figure 4.1: Box plot for the survival time

accepts event and survival time and builds the y value acceptable to scikit-survival algorithms. The class variable *df* is then set to None with the goal of memory optimisation. X and y are then returned.

When we need the validation and the test sets, we just call methods *get_validate_X_y()* and *get_test_X_y()* which will provide us with the sets for the hyperparameter tuning step with the validation set and the final evaluation step with the test set.

4.3 Exploratory Data Analysis

In this section we are going to cover all the most signifficant features. The data were not adjusted to limit the influence of other factors, so the correlations I am trying to make here might not be fully correct, however some are confirmed in literature.

4.3.1 Survival Data

In this subsection we are going to explore the y axis that is going to be used for the training of the survival estimators. The y value consists of censoring status, which is a boolean value, and time to event, which is a numerical value representing the survival time or the time at which it was censored. The y value is called the *survival data*. The column PSTATUS is a censoring status, while the PTIME column represents the time-to-event variable.

To best visualize the time-to-event variable we are going to use a box plot. A box plot is a simple, yet powerful statistical graph based on quartiles, that allows to quickly make sense of the data distribution. (odkaz na statistics for data scientists) It is based on three quartiles: the first(Q_1), the second(Q_2) and the third (Q_3). First quartile corresponds to 25 percentile and means that 25% of the datapoints are below it. The second quartile corresponds to 50% percentile, or median, and it means that below and above that point lies an equal amount of data points. The third quartile corresponds to 75 percentile and it means that below it lies 75% of data points. The quartiles form the box: the first quartile forms the left edge(or bottom edge, in case of horizontal box plot), the third quartile forms the right edge (or top edge) of the box and the median is drawed inside of the box. The box itself represents interquartile range (IQR), that is calculated as $IQR = Q_3 - Q_1$. The lines that lie beyond the box are called *whiskers* and indicate a range for "a bulk of the data". The whiskers extend to the furthest points outside of the box, except they cannot be longer than 1,5 times the IQR. The values lying outside of the whiskers are considered outliers.

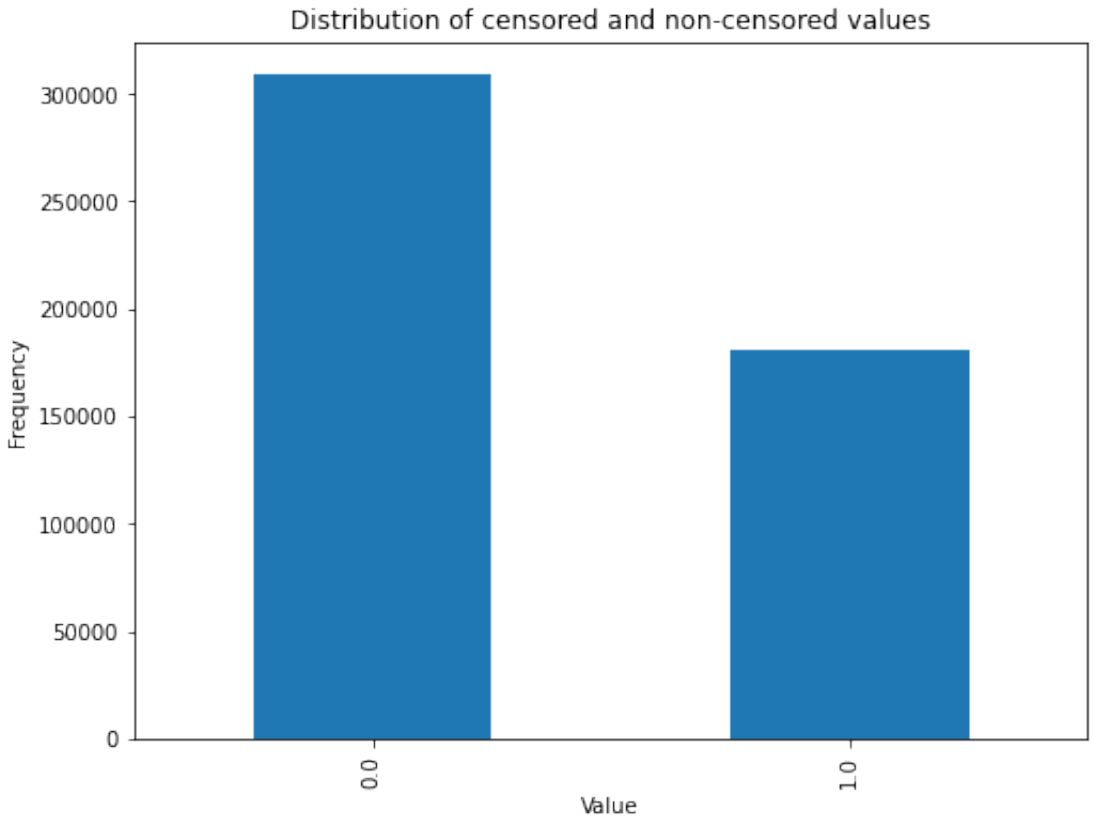


Figure 4.2: Bar chart of PSTATUS colum that provides information on censoring. 0 corresponds to censored instance, 1 corresponds to event happening. There are 308823 censored instances, and 181349 times event happened.

Look at the Figure 4.1, there you can see the box plot of patient survival time (PTIME column). The $Q_1$is equal to 867 days (2.4 years), the median is 2136 days (5.85 years) and the third quantile $Q_3$is equal to 3828 days (10.5 years). The interquartile range (IQR) is equal to 2961 days. This makes up the box. The left whisker extends from 0 day up to the first quartile of 867 days. The right whisker is much longer, and extends from the third quartile up to the $Q_3 + 1.5 * IQR$, which in our case is equal to 8269,5 days. The values above 8269,5 can be considered outliers. As can be seen from the figure (almost straight black line, consisting of individual dots) there are a lot of them.There are less than **10 000** (**get the actual value**) outliers, which is not that much compared to the 490 000 of total kidney transplantations. It is not the wisest choice to simply remove the outliers, as they still might have useful information to the model. However it has to be estimated experimentally. The handling of outliers will be described in the section dedicated to the dataset building.

Look at the figure 4.2, where there is plotted the distribution of the PSTATUS column values. 0 corresponds to censored instances, 1 corresponds to event happening. As can be seen, the distribution is quite uneven, and there is much larger amount of censored instances than ones with the event happening, as there are 308 823 censored instances and only 181 349 non-censored ones. The percentage of censoring is 63%, which is quite high.

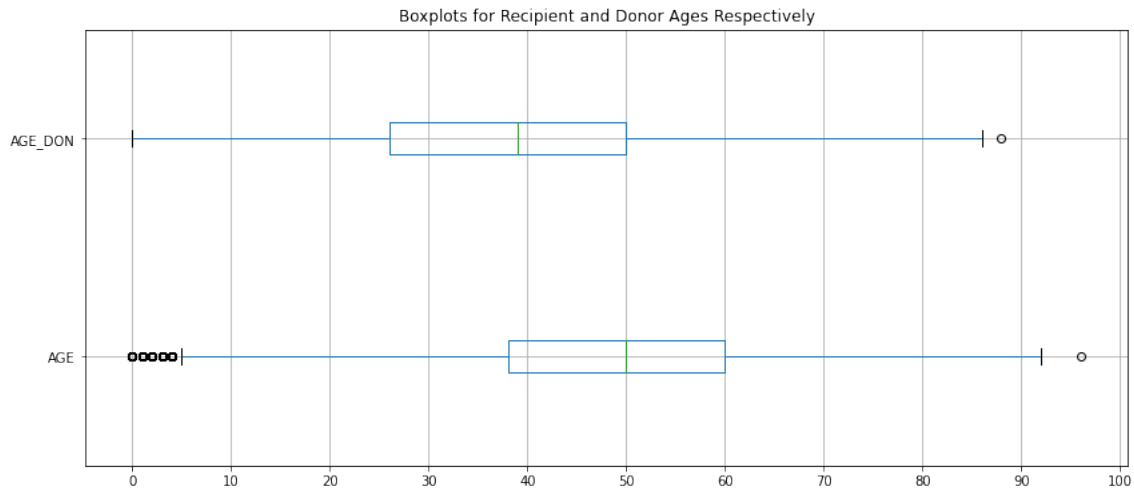


Figure 4.3: Box plot for the recipient age versus donor age.

4.3.2 Age

In this subsection we are going to explore both the donor and recipint ages in the dataset, and will see what the literature tells about their importance to long-term survival.

As can be seen on the box plot in the Figure 4.3, in this dataset donors are usually younger than the recipients by median 11 years. Median recipient age in the dataset is 50, median donor age is 39. The vast majority, 75% of the recipients aged lie between 38 and 60 years old, making IQR of 22, while 75% of the donor age lie between 26 and 50 years old, making IQR of 24. Interestingly, there are not many outliers, as the whiskers cover ages from about 5 to 93, covering the most of human life range.

In the Figure 4.4 you can see the hexagonal binning for the recipient age vs. the survival time. The hexagonal binning is a substitute for a scatter plot for large datasets, as scatter plots do not handle large data sets very well. The hexagonal binning plot consists of colored hexagons, and the darker the color is, the more instances lie in it.

The figure 4.4 more or less corresponds to box plots 4.1 and 4.3, as the majority of colored hexons lie in box ranges for age and the survival time.

4.3.3 Donor Type

In this subsection we are going to explore the influence of donor type (living or diseased) on the survival. Recipients with kidneys from living donors live longer, this is a well established fact[29] [30].

Let's look at the Figure 4.5, where are plotted two Kaplan-Meier survival curves for all patients from the dataset, without taking into account other covatiates. On the graph we can see that the survival probability of living donor transplant is indeed significantly higher than the survival probability of deceased.

This is the case because often there is no time to make full HLA screening, that may allow for HLA mismatches. Additionally, deceased transplants may suffer from mild kidney damage due to the delay in transplantation. While living donor transplants are often performed between siblings that have similar HLA, that creates better compativility.

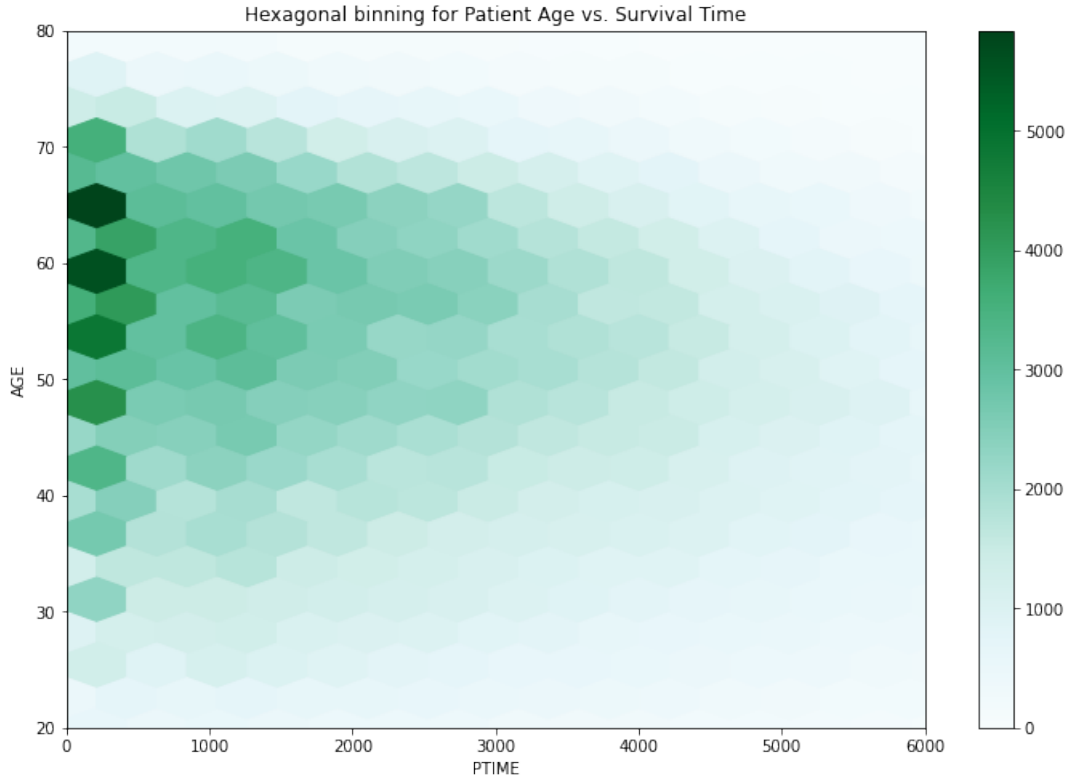


Figure 4.4: Hexagonal binning for the recipient age versus survival time. The brighter the color of the hexagon is, the more instances lie in it.

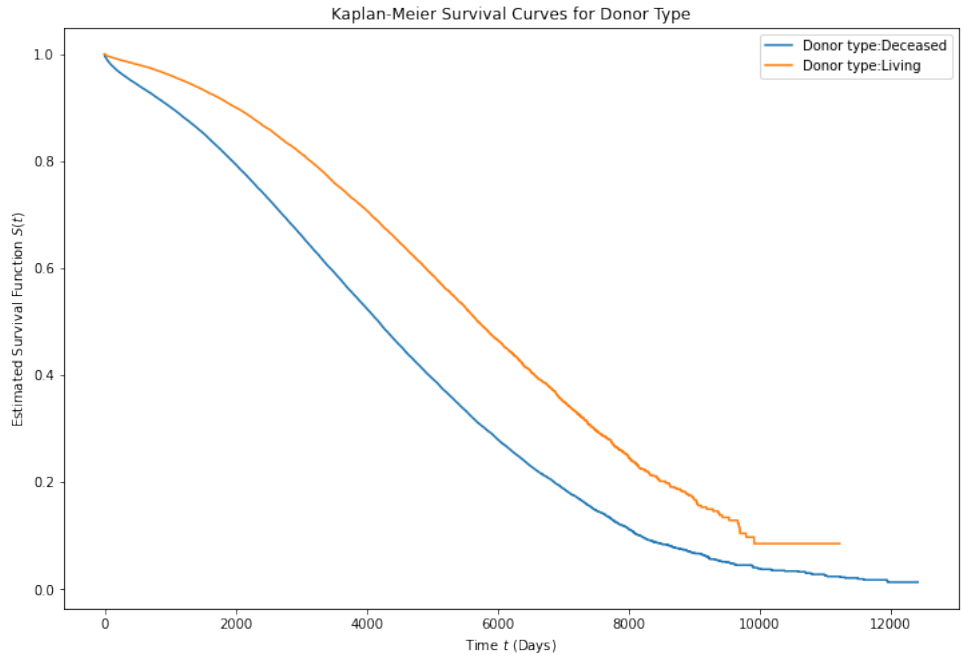


Figure 4.5: Kaplan-Meir survival curve donor types

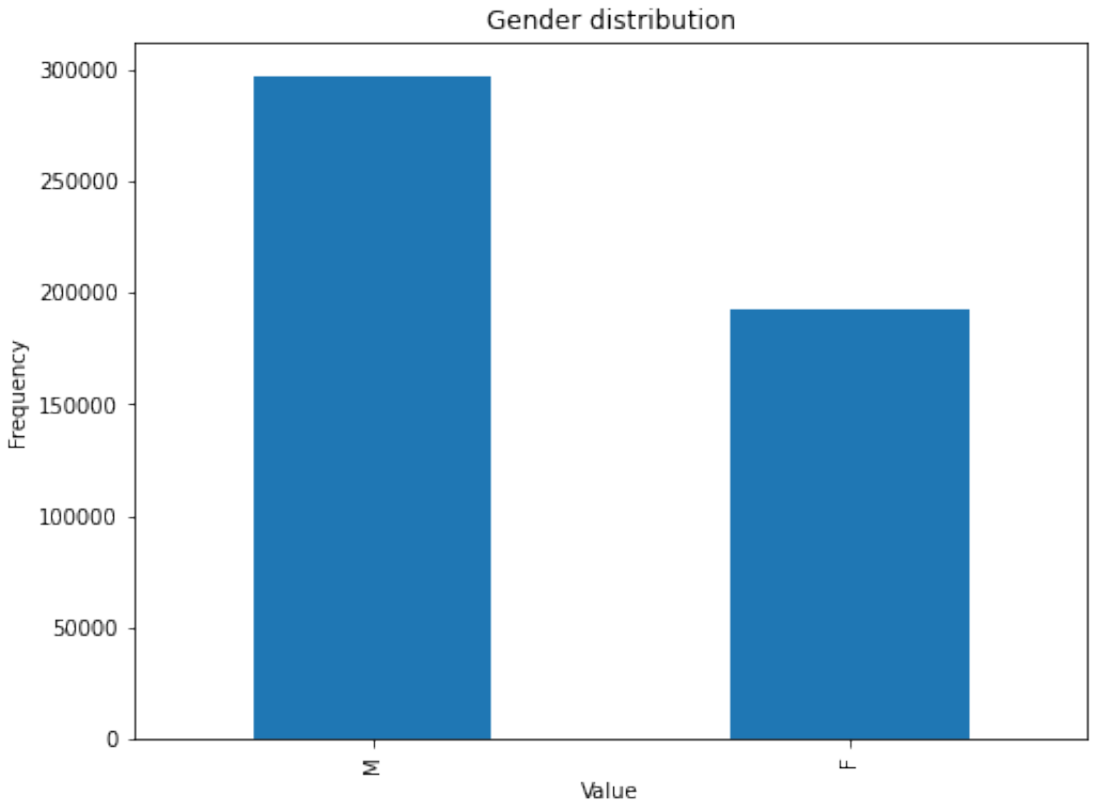


Figure 4.6: Bar chart for gender

4.3.4 Gender

In this subsection we are going to explore the gender distribution in our dataset, and its influence on survival.

Let's explore the distribution of gender in our dataset. Take a look at the figure 4.6. There are 297279 men and 192882 women - the 35% difference. Despite the fact that the chronic kidney disease is more common in women, the end stage kidney failure and therefore kidney transplantation is more common in men[19].

Let's take a look at gender's influence on survival. In the Figure 4.7 we can see the Kaplan-Meier survival curves for men and women on the whole dataset. As can be seen from the graph, females generally have less risk than their male counterparts. Women usually live longer [14]. Quite signifficant factor is the difference between male and female immune responses - males ususally have greater risk to get an infection, than females, and the intesity of the infection is higher[15]. Furthermore, the influence of immunosuppresants make the problem of infection even worse.

4.3.5 The Use of Dialysis

In this subsection we are going to explore the influence of dialysis on survival. In the figure 4.8 we can see two survival curves for the patients who were on dialysis, and for those who were not. The whole dataset was used. As can be seen in the Figure, the patients who were on dialysis before the transplantation have a greater risk, while those who were not. This agrees with ..... [20].

Why exactly this is the case, unfortunately, I was not able to find. The literatures just states it as fact.

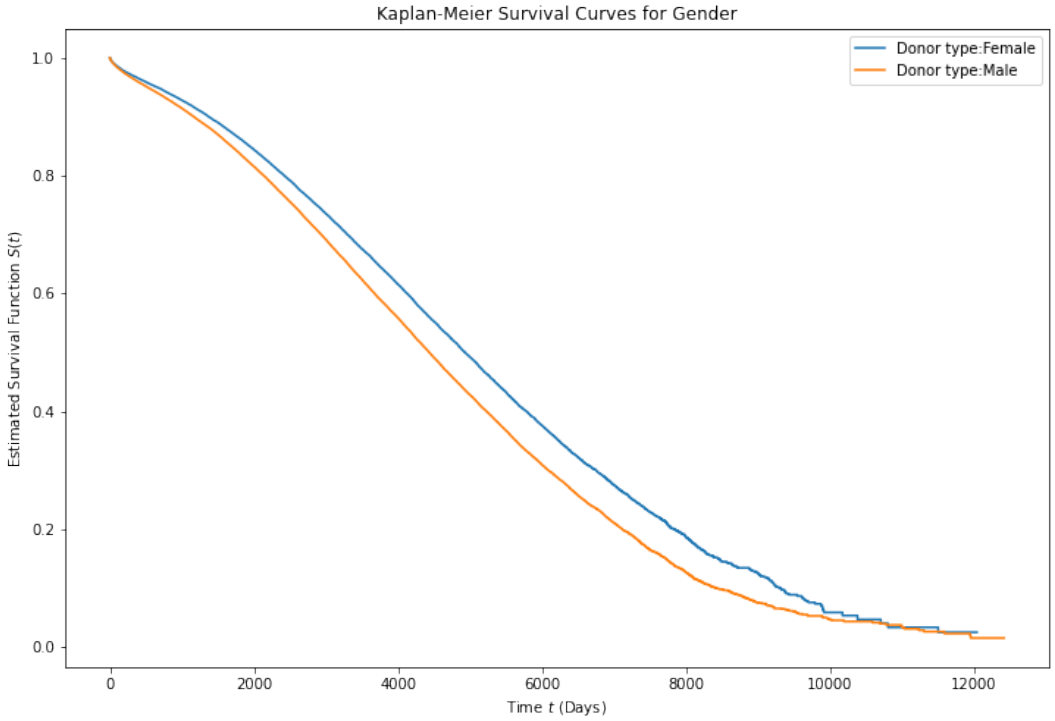


Figure 4.7: Kaplan-Meir survival curve for genders

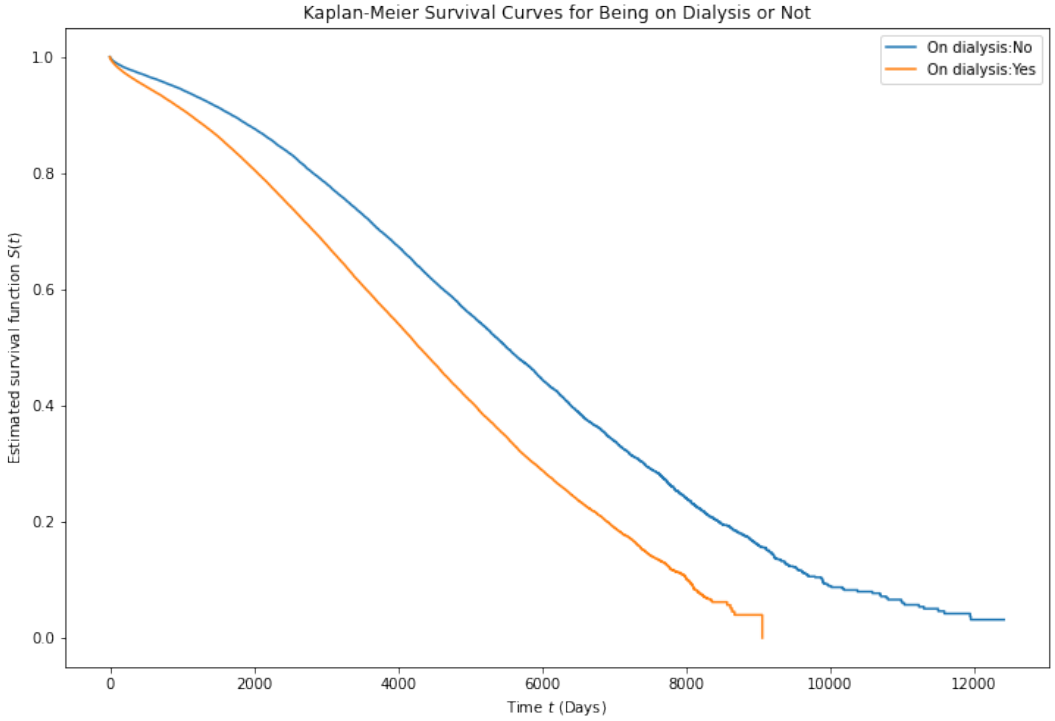


Figure 4.8: Kaplan-Meir survival curve for using dialysis or not

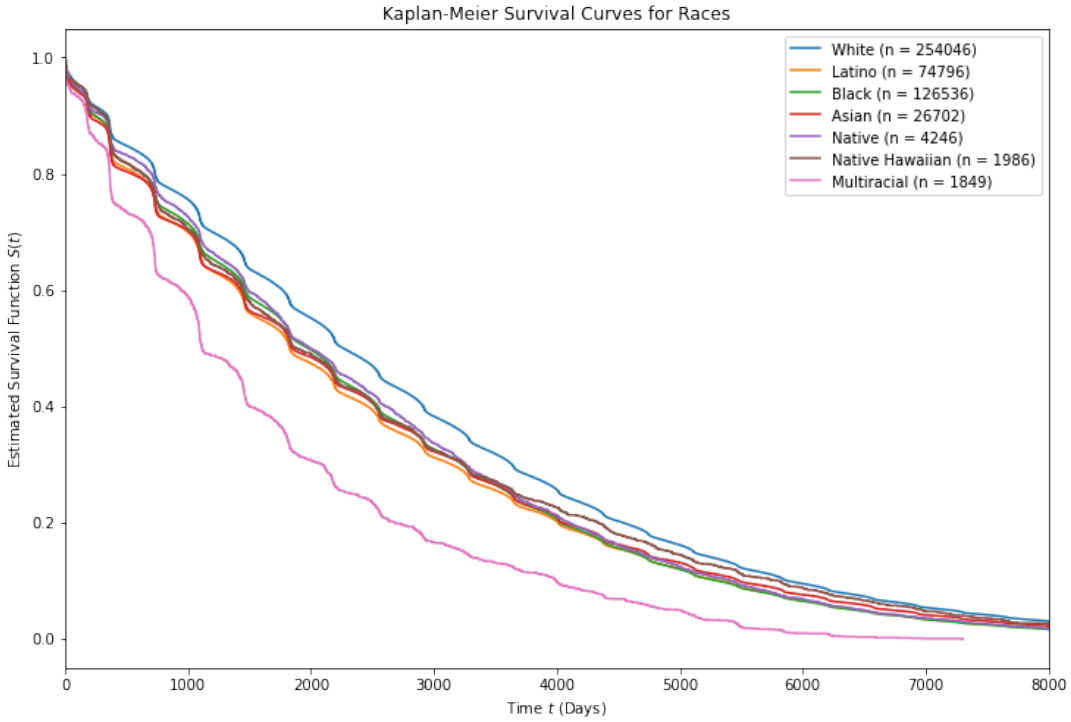


Figure 4.9: Kaplan-Meir survival curve for ethnicities

4.3.6 Race

In this subsection we are going to explore the survival curves for different ethnic groups. In the Figure [31] are plotted 7 survival curves for different ethnicities. As can be seen on the image, 5 ethnicities share about the same survival probability during the cource of 8000 days, while two of them differ substantially from the other. White americans had higher survival probability than other ethnical groups, later converging to the others. This corresponds to ([31]) examining graft survival.

The least survival probability had multiracial group. Their number, however, was the lowest - only 1849 instances, so it is not enough to make any conclusions.They might be later removed, as the definition is too vague and there are not many instances.

4.4 Dataset building, Exclusion criteria and noice reduction

Chapter 5

Machine Learning Model

5.1 Problem Formulation

Predicting the survival time after a successful kidney transplant can be approached in three ways: as a regression problem, classification problem, or through survival analysis.

A *regression* model may seem an intuitive choice, as we want to predict a numerical value – the survival time. But it is not the best option for the following reasons:

1. **The censored dataset**. The dataset has a high level of censoring – 76%. The dataset contains the number of days survived, along with the survival status. Including both living and deceased patients would introduce too much noise to the model, making it highly inaccurate. It is impossible to predict the number of days survived with regression methods based on a dataset comprised of both living and deceased patients.

2. **Censoring removal would produce bias and signifficantly reduce the dataset**. We could remove all censored instances, but that would reduce the dataset from 500 000 to roughly 120 000 examples. It would also introduce significant bias, as the dataset would contain only deceased patients, and most of them passed away before the introduction of modern techniques for treating the rejection. As a result, the model created from such a dataset would be highly inaccurate.

3. **Regression predicts only one single number.** It poses a problem, especially over extended time frames, as there are too many factors that we can't account for, leading to incorrect predictions.

Another way of formulating the problem is *classification*. We can theoretically divide the dataset into groups: "less than one year", "one to five years", "five and more", or even more groups and train a classifier based on them, as it was done by .... et al.. And again, we would face problems of censoring and bias mentioned above. So the classification is also not the best option.

A more appropriate way of problem formulation is in terms of *survival analysis*. Survival analysis methods handle censoring and provide a better form of prediction: survival function or hazard function, which represents survival probability or the failure rate at each moment in time, respectively.

5.2 Model selection

The algorithms provided by the scikit-survival do not handle large datasets very well (never ending training process and worse results probably due to the noise) that is why I chose to train different models for different demographics, as one specific model for one specific demographic will perform better that one model trained for all demographics. In addition, the living donor transplantation differs a bit from the diseased transplantation, that might introduce some noice into the model.

Dataset	Uno c-index	IBS	Mean AUC
Living	0.705	0.135	0.704
Deceased	0.681	0.165	0.714
Kaplan-Meier (for IBS reference)	-	0.247	-

Table 5.1: Coxnet performance on the test set

Model	Uno c-index	IBS	Mean AUC
Coxnet (standard hyperparams, living)	0.668		
Coxnet (the best hyperparams, living)	0.705	0.135	0.704
Kaplan-Meier (for IBS reference)	-	0.247	-

Table 5.2: Cox before and after fine tuning.

The way I approach the model selection model automation with the class SurvivalEstimators defined in estimator_automation.py.

Run the following class and short list the most promising models. In this case it is survival gradient boosting and random survival forests.

5.3 Results

In this section we are going to discuss two models, Coxnet and Random Survival Forest. These models were chosen because they both are able to generate survival functions in the scikit-survival library, unlike others that only estimate the risk score. Unfortunately, the older version of scikit-survival was used - 0.14.0, due to the limitation in python version on cluster where these models were trained. The survival function will later be used in the application KidneyLife to visually illustrate the probability of survival in each moment in time.

5.3.1 Coxnet

Coxnet, or an elastic net, is a linear model, so it is fast even with large datasets, a bit worse results, compared to the Random survival forest. It makes prediction both in a form of the risk score or a survival function.

As can be seen in the table 5.1, the Coxnet performed the best for the living group, the worst for the deceased, and somewhere in between for the both living and deceased.

Discussion of the AUC figure:

As was covered in 3.4, the hyperparameter tuning is usually performed with either GridSearch or RandomizedSearch. Unfortunately, the older Python version on the cluster did not allow to install the newest version of scikit-survival, where the GridSearch was implemented. So, the hyperparamer tuning was performed with a custom script that is designed to imitate the GridSearchCV but without the k-fold cross-validation. The script optimizes for the Integrated Brier Score (IBS) that directly tells the accuracy of predictions. The script can be found: here (**add link**).

Discussion of the Brier Figure:

The coxnet has only two hyperparameters: L1 ratio and alpha, the latter is calculated by fitting the model and we then need to choose the best of them. L1 ratio defines the relative weight of the $l_1$and l_2 penalty.

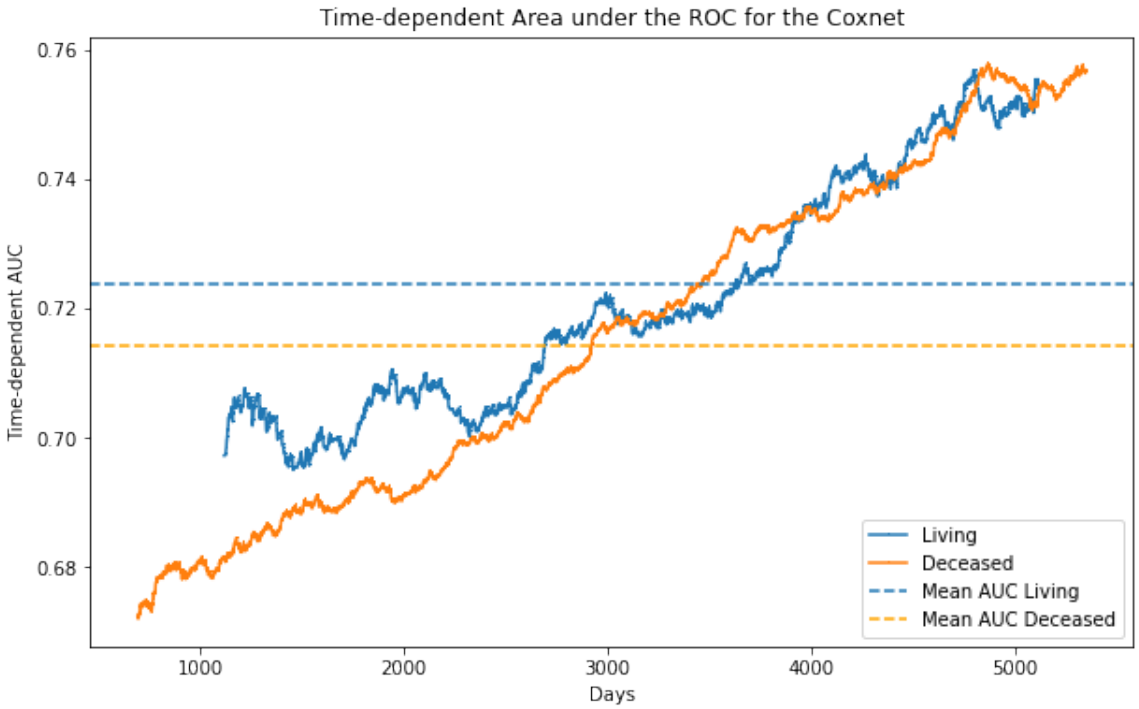


Figure 5.1: AUC curve for coxnet

Figure 5.2: Time-dependent Brier score for the Coxnet

Feature description	Importance	Abbreviation
Donor Age		
Recipient Age		
Donor Type		
Donor gender		
Recipient gender		
Donor blood group		
Recipient blood group		
Recipient on dyalisis		
Recipient creatinine at the time of tx		

Table 5.3: Coxnet Feature Importance

Dataset	Uno c-index	IBS	Mean AUC
Living	0.727	0.129	0.743
Deceased	0.678		
Kaplan-Meier (for IBS reference)	-	0.247	-

Table 5.4: Random Survival Forest performance on the test set

On the table 5.3 we can see the importances of features for the prediction. Features with zero influence were omitted.

5.3.2 Random Survival Forest

Random survival forest is a powerful ensemble machine learning algorithm, that is comprised of multiple submodels, and therefore it takes a lot of time to train, a lot of time to make a prediction, depending on the selection of hyperparameters, making it a bit difficult to work with, especially during the hyperparameter tuning. Extremely memory hungry. The prediction is a survival function or a hazard score. It was covered in detail in (**RSF subsection**). The living and deceased subsets had 34951 and **70 000** instances respectively.

As can be seen from the table 5.4, the survival forest performed the best for the living group, the worst for the deceased, and somewhere in between for the both living and deceased, just as it was in case with the Coxnet.

discussion of AUC image:

discussion of the Brier image: On the Figure 5.4 is plotted the time dependent Brier score for the Random Survival Forest. As we can see, it increases over time, meaning that predictions get worse over time. This totally makes sense, as the more time passes after the transplant, the less pretransplant information, that was used for training, has influence on the survival.

The fine-tuning was performed with a custom script, described in the previous subsection dedicated to the coxnet. The hyperparameters that were fine-tuned are the following: n_estimators, max_depth, min_sample_split and max_features.

On the table 5.5 we can see the performance of regular RSF with the standard hyperparameters compared to the performance of RSF with fine-tuned hyperparams.

Feature importance can be seen on the table 5.6.

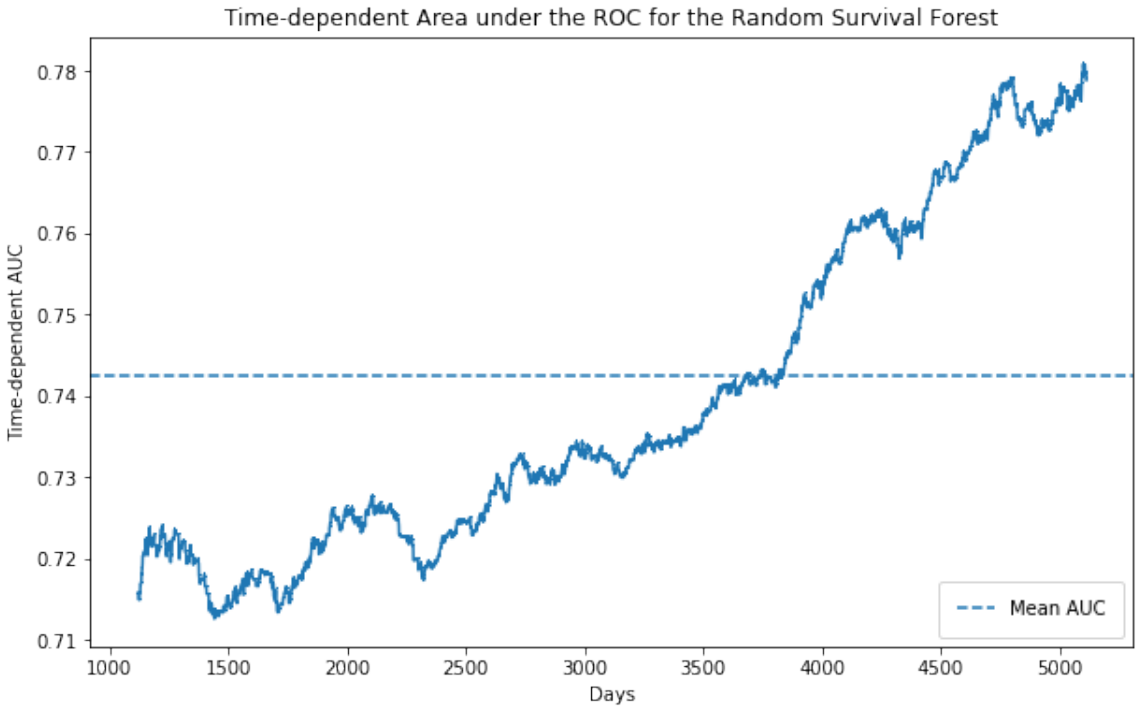


Figure 5.3: AUC for RSF

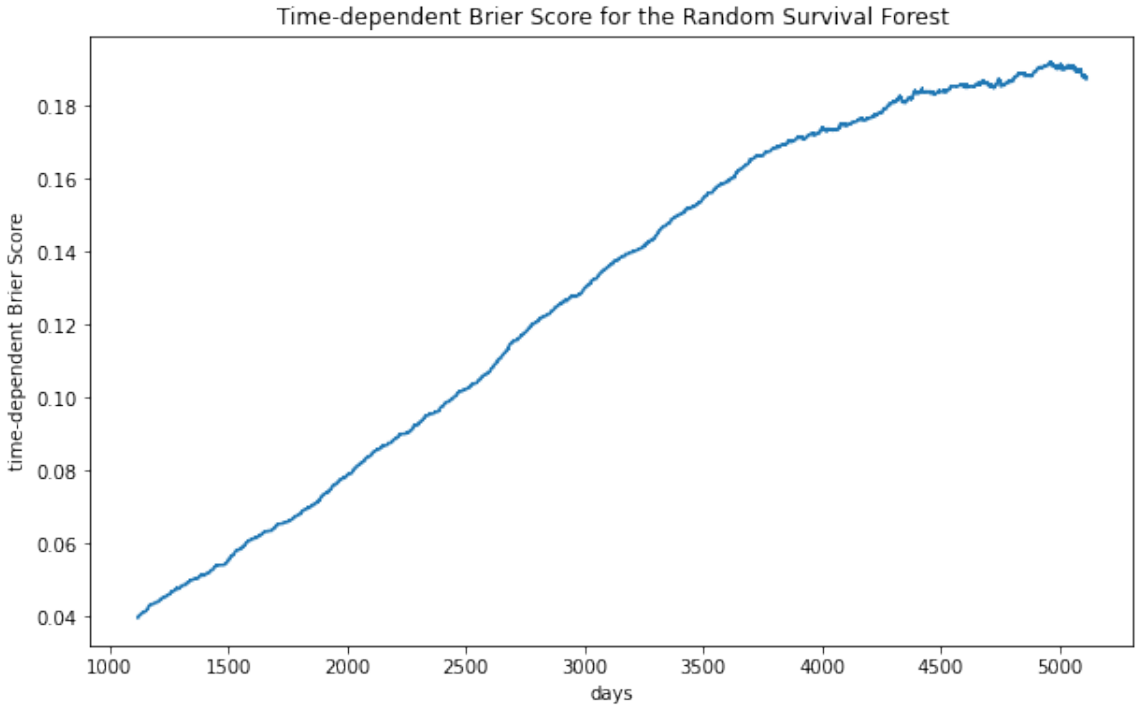


Figure 5.4: Time-dependent Brier score for the Random Survival Forest

Model	Uno c-index	IBS	Mean AUC
Random Survival Forest (standard hyperparams, living)	0.708	0.160	
Random Survival Forest (the best hyperparams, living)	0.723	0.129	0.743
Kaplan-Meier (for IBS reference)	-	0.247	-

Table 5.5: RSF before and after fine tuning.

Feature description	Importance	Abbreviation
Donor Age		
Recipient Age		
Donor Type		
Donor gender		
Recipient gender		
Donor blood group		
Recipient blood group		
Recipient on dyalisis		
Recipient creatinine at the time of tx		

Table 5.6: Feature Importance for RSF

5.3.3 Comparison

image with the two BS and AUC graphs for each model

As can be seen on the table 5.7 and on the image ..., the Random survival forest performed better than the elastic net. However, there are some differences in timeframes, where at certain points in time the coxnet performed better than the RSF.

AUC RSF vs. Coxnet:

Brier Coxnet vs. RSF:

5.4 Scoring algorithm

the cumulative hazard suits the place of transplantation score very well

5.5 Limitations

these models probably aren't suitable for KEP (check results), bc they're slow, but are good for prediction estimated survival and when it is best to intervene.

the RSF deceased could be better fine-tuned

Model	Uno c-index	IBS	Mean AUC
Random Survival Forest (living)	0.723	0.129	0.742
Coxnet (living)	0.705	0.135	0.704
Random Survival Forest (deceased)			
Coxnet (deceased)	0.681	0.165	0.714

Table 5.7: Model comparison on the test set

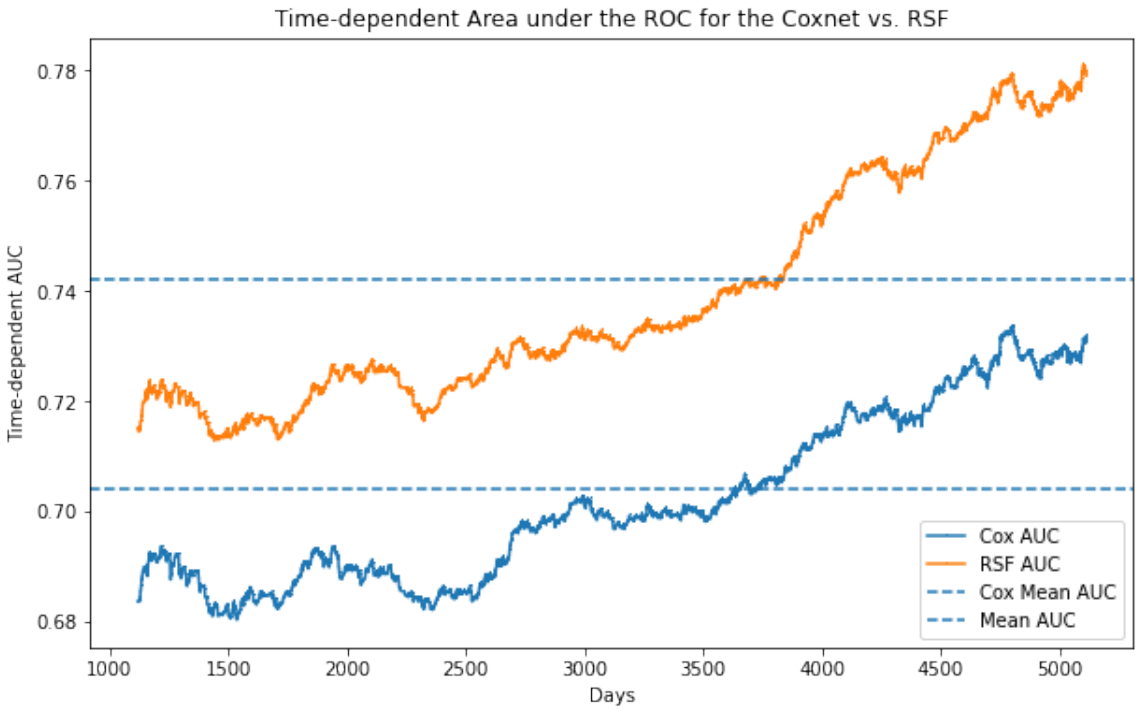


Figure 5.5: AUC: Coxnet vs. RSF, Living

5.6 Further work

more thorough hyperparameter tuning, especially with the RSF on the deceased dataset.
another model for follow up data
deep survival neural network

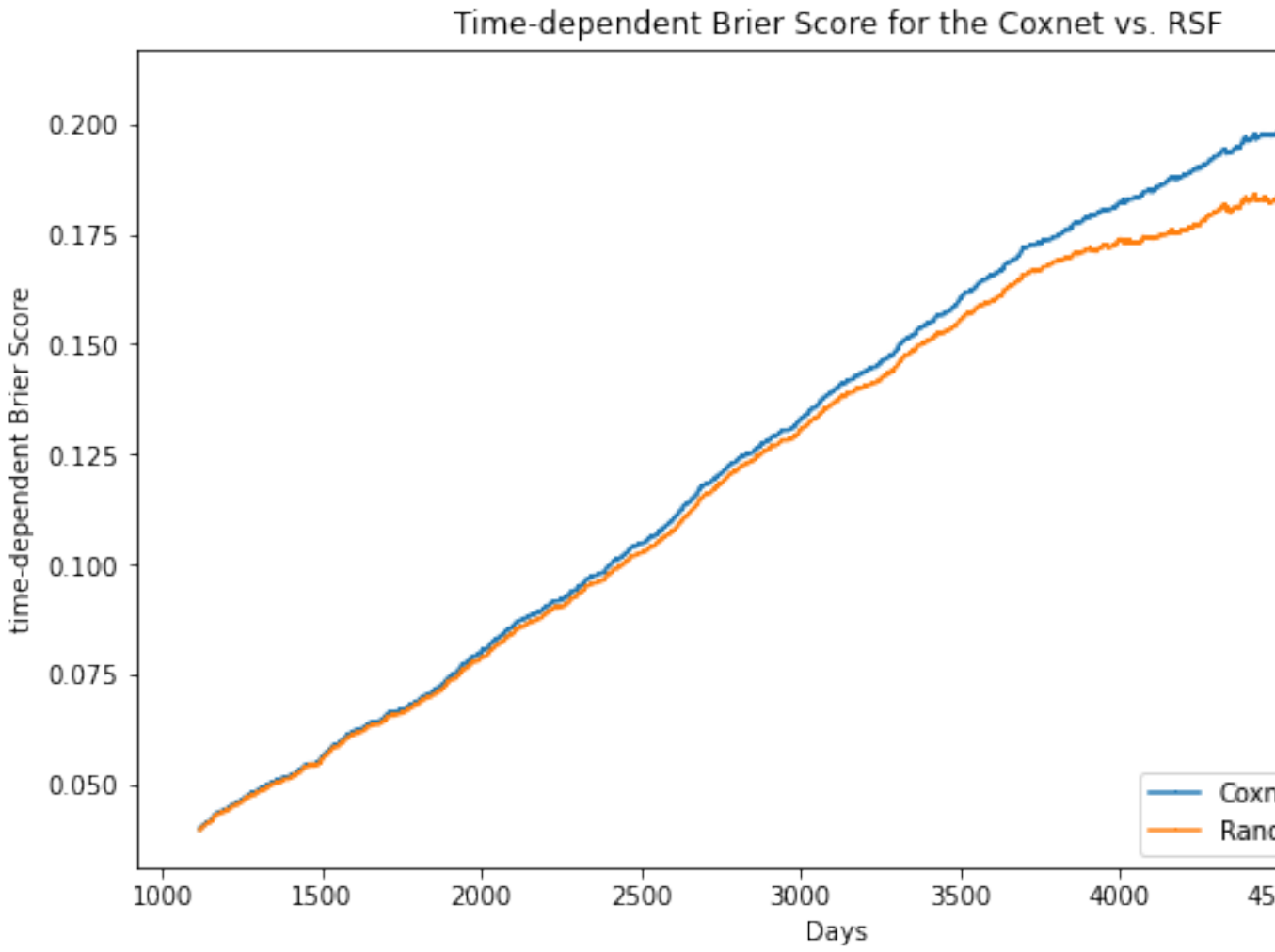


Figure 5.6: Brier: Coxnet vs. RSF, Living

Chapter 6

Applications

txmatching is something totally different, so it was decided to create separate application for accesing the model.

6.1 Existing Solutions

6.1.1 Txmatching

Txmatching is

6.2 KidneyLife

6.2.1 Frontend

6.2.2 Backend

For backend was used flask. Flask is popular python web framework. The model was saved as a binary in pickle format and the back end provides an api to access the model from the web page.

Conclusion

Text of the conclusion...

Bibliography

[1] Knechtle, S. J., Marson, L. P., & Morris, P. (2019). Kidney transplantation - principles and practice: Expert consult - online and print (8th ed.). Elsevier - Health Sciences Division

[2] Nobel prize in physiology or medicine (2022) Our Scientists. Available at: https://www.rockefeller.edu/our-scientists/alexis-carrel/2565-nobel-prize/ (Accessed: February 6, 2023).

[3] Barker, C. F., & Markmann, J. F. (2013). Historical Overview of Transplantation. Cold Spring Harbor Perspectives in Medicine, 3(4). https://doi.org/10.1101/cshperspect.a014977

[4] Matevossian, Edouard, et al. "Surgeon Yurii Voronoy (1895-1961)-a pioneer in the history of clinical transplantation: in memoriam at the 75th anniversary of the first human kidney transplantation." Transplant International 22.12 (2009): 1132.

[5] PUNT, Jenni et al. Kuby immunology. Eight. vyd. New York: Macmillan Education, 2019. ISBN 9781319114701;1319114709;

[6] ABBAS, Abul K., Andrew H. LICHTMAN a Shiv PILLAI. Basic immunology: functions and disorders of the immune system. Sixth. vyd. Philadelphia: Elsevier, 2020. ISBN 9780323549431;0323549438;

[7] NCI Dictionary of Cancer terms (no date) National Cancer Institute. Available at: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/abo-blood-group-system (Accessed: March 6, 2023).

[8] Dean L. Blood Groups and Red Cell Antigens [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2005. Chapter 2, Blood group antigens are surface markers on the red blood cell membrane. Available from: https://www.ncbi.nlm.nih.gov/books/NBK2264/

[9] Aurélien Géron. Hands-on Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., Sept. 2019.

[10] Andriy Burkov. THE HUNDRED-PAGE MACHINE LEARNING BOOK. Andriy Burkov, 2019.

[11] Makary M A, Daniel M. Medical error—the third leading cause of death in the US BMJ 2016; 353 :i2139 doi:10.1136/bmj.i2139

[12] Bruce, P., Bruce, A., & Gedeck, P. (2020). Practical statistics for data scientists: 50+ Essential concepts using R and python (2nd ed.). O'Reilly Media. p. 141

[13] Kleinbaum, D. G., & Klein, M. (2011). Survival analysis: A self-learning text, third edition (3rd ed.). Springer.

[14] Ostan R, Monti D, Gueresi P, Bussolotto M, Franceschi C, Baggio G. Gender, aging and longevity in humans: an update of an intriguing/neglected scenario paving the way to a gender-specific medicine. Clin Sci (Lond). 2016 Oct 1;130(19):1711-25. doi: 10.1042/CS20160004. PMID: 27555614; PMCID: PMC4994139.

[15] vom Steeg LG, Klein SL. SeXX Matters in Infectious Disease Pathogenesis. PLoS Pathog. 2016 Feb 18;12(2):e1005374. doi: 10.1371/journal.ppat.1005374. PMID: 26891052; PMCID: PMC4759457.

[16] Rodrigues S, Escoli R, Eusébio C, Dias L, Almeida M, Martins LS, Pedroso S, Henriques AC, Cabrita A. A Survival Analysis of Living Donor Kidney Transplant. Transplant Proc. 2019 Jun;51(5):1575-1578. doi: 10.1016/j.transproceed.2019.01.047. Epub 2019 Jan 21. PMID: 31155195.

[17] Nemati E, Einollahi B, Lesan Pezeshki M, Porfarziani V, Fattahi MR. Does kidney transplantation with deceased or living donor affect graft survival? Nephrourol Mon. 2014 Jul 5;6(4):e12182. doi: 10.5812/numonthly.12182. PMID: 25695017; PMCID: PMC4317718.

[18] Pisavadia B, Arshad A, Chappelow I, Nightingale P, Anderson B, Nath J, Sharif A. Ethnicity matching and outcomes after kidney transplantation in the United Kingdom. PLoS One. 2018 Apr 13;13(4):e0195038. doi: 10.1371/journal.pone.0195038. PMID: 29652887; PMCID: PMC5898720.

[19] Guillermo García García, Arpana Iyengar, François Kaze, Ciara Kierans, Cesar Padilla-Altamira, Valerie A. Luyckx, Sex and gender differences in chronic kidney disease and access to care around the globe, Seminars in Nephrology, Volume 42, Issue 2, 2022, Pages 101-113, ISSN 0270-9295, https://doi.org/10.1016/j.semnephrol.2022.04.001. (https://www.sciencedirect.com/science/article/pii/S0270929522000092)

[20] Mange KC, Joffe MM, Feldman HI. Effect of the use or nonuse of long-term dialysis on the subsequent survival of renal transplants from living donors. N Engl J Med. 2001 Mar 8;344(10):726-31. doi: 10.1056/NEJM200103083441004. PMID: 11236776.

[21] Rahman MS, Ambler G, Choodari-Oskooei B, Omar RZ. Review and evaluation of performance measures for survival prediction models in external validation settings. BMC Med Res Methodol. 2017 Apr 18;17(1):60. doi: 10.1186/s12874-017-0336-2. PMID: 28420338; PMCID: PMC5395888.

[22] Evaluating survival models — scikit-survival 0.21.0. (n.d.). Readthedocs.Io. Retrieved July 18, 2023, from https://scikit-survival.readthedocs.io/en/stable/user_guide/evaluating-survival-models.html

[23] Ping Wang, Yan Li, and Chandan k. Reddy. 2019. Machine Learning for Survival Analysis: A Survey. ACM Comput. Surv. 51, 6, Article 110 (February 2019), 36 pages.https://doi.org/10.1145/3214306

[24] Definitions:, 1. 1. (n.d.). Survival distributions, hazard functions, cumulative hazards. Stanford.edu. Retrieved October 23, 2023, from https://web.stanford.edu/~lutian/coursepdf/unit1.pdf

[25] Penalized Cox Models — scikit-survival 0.22.1. (2015). Readthedocs.io. https://scikit-survival.readthedocs.io/en/stable/user_guide/coxnet.html

[26] Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests.

[27] Wang, H., & Li, G. (2017). A selective review on random survival forests for high dimensional data. Quantitative bio-science, 36(2), 85.

[28] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[29] Gönen M, Heller G. Concordance probability and discriminatory power in proportional hazards regression. Biometrika 2005;92(4):1799–09.

[30] Enrico Longato, Vettoretti, M., & Barbara Di Camillo. (2020). A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. Journal of Biomedical Informatics, 108, 103496–103496. https://doi.org/10.1016/j.jbi.2020.103496

[31] Uno H, Cai T, Pencina MJ, D'Agostino RB, Wei LJ. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Stat Med. 2011 May 10;30(10):1105-17. doi: 10.1002/sim.4154. Epub 2011 Jan 13. PMID: 21484848; PMCID: PMC3079915.