Survival Model Analysis applied to Kidney Transplant Data

Karyn Silva de Azevedo
Master's degree in Computer Science
Department of Computer Science
2021

Advisor
Prof. João Pedro Pedroso, Faculty of Science, University of Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

Abstract

Kidney transplant is a therapy for irreversible graft failure that can provide longer life to the recipients. However, to achieve a successful transplant to prolong a patient's life, it takes a complex matching system to combine donors and recipients based on several matching criteria and policies. To understand and interpret the results of the matching system, this work was developed. To accomplish this task, all the following survival analysis tools were used: Kaplan-Meier, Cox models, Survival Support Vector Machine, Random Survival Forest and Gradient Boosting Survival. These models were used to predict the survival time and their performance were evaluated using two different implementations of the concordance index. The database used in this project was the Scientific Registry of Transplant Recipients, which contains the records of all kidney transplant patients in the USA starting in 1987, with more than one million entries. Unfortunately, given the restrictions in the computational power available, the dataset was not entirely used, so a subset of 20.000 entries was randomly selected to represent the population. When using this subset, the best performance was achieved by the Ensemble models: Random Survival Forest (Harrell's c-index: 0.6602, Uno's c-index:0.6916) and Gradient Boosting (Harrell's c-index: 0.6624, Uno's c-index:0.6941). Also, according to most of the models, a set of features was identified as the ones that have the greatest impact in the survival time: type of donor (living or dead), Diabetes type II (Yes or No), Age of the donor and age of the recipient (years) and the ethnicity, specifically Hispanic.

Resumo

O transplante renal é uma terapia para falência irreversível do órgão que pode proporcionar vida mais longa aos receptores. No entanto, para obter um transplante bem-sucedido para prolongar a vida de um paciente, é necessário um sistema de atribuição complexo para combinar doadores e receptores com base em vários critérios e políticas de correspondência. Para entender e interpretar os resultados do sistema de emparelhamento, este trabalho foi desenvolvido. Nesse projeto, todas as seguintes ferramentas de análise de sobrevivência foram utilizadas: Kaplan-Meier, modelos de Cox, *Survival Support Vector Machine, Random Survival Forest e Gradient Boosting Survival*. Esses modelos foram utilizados para prever o tempo de sobrevida e para avaliar sua performance foram utilizadas duas implementações diferentes do índice de concordância. A base de dados utilizada neste projeto foi o *Scientific Registry of Transplant Recipients*, que contém os registros de todos os pacientes de transplante renal nos EUA a partir de 1987, com mais de um milhão de entradas. Infelizmente, dadas as restrições no poder computacional disponível, o conjunto de dados não foi totalmente utilizado, então um subconjunto de 20.000 entradas foi selecionado aleatoriamente para representar a população. Ao usar este subconjunto, o melhor desempenho foi alcançado pelos modelos *Ensemble: Random Survival Forest* (índice de concordância de Harrell: 0,6602, índice de concordância de Uno: 0,6916) e *Gradient Boosting* (índice de concordância de Harrell: 0,6624, índice de concordância de Uno: 0,6941). Ainda, de acordo com a maioria dos modelos, identificou-se um conjunto de variáveis como as que têm maior impacto no tempo de sobrevida: tipo de doador (vivo ou morto), Diabetes tipo II (Sim ou Não), Idade do dador e idade do receptor (anos) e a etnia, especificamente hispânica.

Acknowledgements

First, I would like to thank my mom and sister for the support throughout this process. Even though we are currently spread in three different continents, I can feel your love and caring every day, regardless of the distance. I would also like to thank my partner in life, Paulo, for the support and for believing in me even when I couldn't. And, last but not least, a special thanks to Professor João Pedro Pedroso, who had the patience and kindness to guide me through this crazy journey.

Contents

List of Tables

List of Figures

Acronyms

AUROC area under the receiver operating characteristic

CMS Centers for Medicare Medicaid Services

CPRA Calculated Panel-Reactive Antibody

ESRD end-stage renal disease

HLA Human Leukocyte Antigen

KAS Kidney Allocation System

KDPI Kidney Donor Profile Index

MHC Major Histocompatibility Complex

NTIS National Technical Information Service's

OPTN Organ Procurement and Transplantation Network

PRA Panel-Reactive Antibody

PDF Probability Density Function

RSF Random Survival Forest

RVM Relevance Vector Machine

SRTR Scientific Registry of Transplant Recipients

SVM Support Vector Machine

SVR Support Vector Regression

UNOS United Network for Organ Sharing

Chapter 1

Introduction

Kidney transplant is a therapy for irreversible graft failure end-stage renal disease (ESRD) that surpasses dialysis treatments both for the quality and quantity of life that it provides and for its cost-effectiveness [8]. However, to achieve a successful kidney transplant, many factors are taken into consideration. One of the most important aspects of graft transplant is the compatibility between donor and recipient, which is not only defined by blood type compatibility, for example, but also for criteria as the HLA antigens of the individuals [24]. With that in mind, the main responsibility of the Kidney Allocation System (KAS) is to find the best combination between available donors and recipients, given all the restrictions in place.

However, it is important to keep in mind that the goal is not only to provide the best matches, but also to be fair [3]. So, this is not an optimization problem that only focuses on the survival time, it should also take into consideration the fact that there are patients that, depending on the criteria in place, would always be sent to the end of the line, and consequently, would never receive a transplant. This is the most interesting aspect of the KAS, it has a set of rules that tries to add fairness to the selection and matching process, while prolonging the recipients lives.

So, when evaluating a combination of donor and recipient, many variables can impact the allocation algorithm [25]. Besides the compatibility issue and fairness, there is also the matter of availability. So, first, the number of kidneys available for transplant is low when compared to the size of the waiting list, and whenever an organ becomes available due to the death of an individual, for example, that kidney will be allocated according to a set of criteria, taking into consideration the quality of the kidney expressed in the Kidney Donor Profile Index (KDPI), the age and survival expectancy of the kidney are also considered [25]. At the end, it is not only about the perfect individual match, but the matching taking into consideration the entire set of donors and recipients.

The KAS impacts many lives by extending the lifetime of the patients with quality. So, it is important to understand this system, identifying which characteristics impact the most the survival time of the graft. The goal of this project is to explore the complexity of the allocation system through the modeling of the survival time. To achieve that, machine learning methods

were used, in the context of survival analysis, to model the graft survival time based on a meaningful selection of information from the recipient and the organ donor.

The data used in this project was obtained through a request performed to the Scientific Registry of Transplant Recipients (SRTR) which is responsible for centralizing data collected from different organizations and combining them, providing summary reports and analysis to the transplant community.

The Organ Procurement and Transplantation Network (OPTN) is the largest contributor of the SRTR database, along with the Centers for Medicare Medicaid Services (CMS) and the National Technical Information Service's (NTIS) Death Master File [1]. The OPTN was created by the United Network for Organ Sharing (UNOS), which had its first data collection system launched in 1986. The currently internet-based system provides registration of candidates and matching capabilities to the organ transplant programs, besides the followup information submission.

In order to have access to the data, a Data Use Agreement must be signed, which restrains the use of the data. Any attempt to identify the individuals represented there or the sharing of the data without UNOS' permission is forbidden.

The data set provided corresponds to 53 tables containing information from many graft transplants, e.g. Intestine, Kidney, Pancreas, Liver, performed in the United States, from both citizens and foreigners. The data dates from October 1, 1987, to December 2020, the last being the month the files were shared. The complete list of tables and a short description can be found in its documentation.

1.1 Organization

Chapter 2: Background In this chapter, the theoretical aspects are introduced. The main topics are the statistical theory behind survival analysis and the proposed methods, both statistical and machine learning, to model survival data. Also, it provides further details on the selected database and the important medical concepts of the kidney transplant context.

Chapter 3: Related Work This chapter discusses two papers and their approach to the survival analysis of kidney transplant. Some authors have previously proposed improvements to the machine learning algorithms, while others proposes to modify the data representation. Both works used the same database as this dissertation.

Chapter 4: Development & Results this chapter describes the manipulation of the data and how the chosen models were trained to reach the obtained result. It also describes the results obtained from all the models used, with a comparative analysis at the end.

Chapter 5: Conclusion This chapter reviews the work done and proposes the next steps for future developments.

Chapter 2

Background

This chapter's goal is to present the statistical concept of Survival Analysis, focusing on the models used in this dissertation. A brief description of the database along with the definition of its most important concepts is also provided here.

2.1 Survival Analysis

The study of kidney transplant survival time consists of evaluating the time to graft failure. However, this requires a specific set of mathematical tools capable of dealing with the particularities of the survival data. And for that, there is the survival analysis.

The survival analysis is a subfield of statistics which models time-to-event data, and its goal is to model the time until an event of interest happens [16]. In the context of this project, the kidney failure is the event of interest.

When considering the time-to-event data, the standard statistical approaches that assume a normal distribution cannot be used with this type of data, because they are often asymmetric, differently from the normal distribution assumed by most of those tools [16]. For example, in a study evaluating the time to graft failure in a population with high risk of rejection, the rejection episodes (event of interest) will most likely happen in the early followups, decreasing as time goes by. The opposite is also observable, in cases where events are expected to happen later than sooner, depending on the data observed.

Not only the distribution of the observations has its peculiarities, but also the data itself. For example, the data is considered incomplete because the event of interest is not always observed for every subject, those observations are called censored data. That happens because, first, studies are limited in time, with specific start and end dates, that restricts the time frame for the event to be observed. Second, any participant can miss the followups, drop from the study, or even die from unrelated causes, which results in incomplete information for those subjects.

The picture 2.1 illustrates a survival data example. The subject A and D both had the event

of interest observed during the study time frame (2010 - 2020), regardless of their difference in time of enrollment in the study, these two represent a complete data also known as uncensored data. The element B, on the other hand, at a given time, was lost during the study, so the information is incomplete or censored. Last, subject C was observed during the entire study, but the event of interest did not happen in that time frame. So, both B and C have right-censored times, because the time of the event is equal or greater than the observed time.

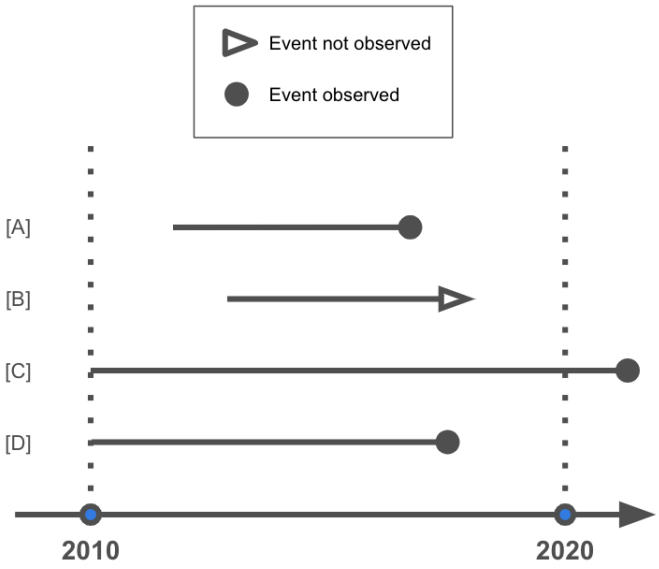

Figure 2.1: Right Censored Data Example: four subjects

Even though subjects B and C have not experienced the event of interest until the end of the study represented in 2.1, their cases contribute to the analysis and cannot be simply discarded. The fact that they did not experience the event of interest is valuable. So, to estimate the likelihood of the event, there are survival analysis techniques that make use of the censored data as well as the uncensored, which makes survival data unique.

2.1.1 Survival Function, Probability Density Function and Hazard Function

This section introduces the statistical basis for the survival analysis. There are three main functions that illustrate different aspects of the survival distribution, all intrinsically related, they are: survival function, probability density function and hazard function [16], which are defined as follows.

Given a random variable T that denotes the survival time of an individual, the survival probability $S(t)$ is defined as the probability of the survival time being greater than a certain time t, where $t \geq 0$ and $S(t) = P(T > t)$. Based on the definition of cumulative distribution function $F(t)$ of T, the survival function, $S(t)$, can be rewritten as 2.1.

$$\begin{aligned} S(t) &= 1 - P(T \leq t) \\ S(t) &= 1 - F(t) \end{aligned} \tag{2.1}$$

$S(t)$, also known as the cumulative survival rate, can be better observed in the survival curve in graph 2.2. This graph is frequently used to identify the median, which can be more interesting than the mean whenever outliers are present. $S(t)$ is a non-increasing function with $S(t) = 1$ when $t = 0$, and for $t = \infty$, $S(t) = 0$, which means that the probability of surviving goes to 0 as time goes to infinity [16].

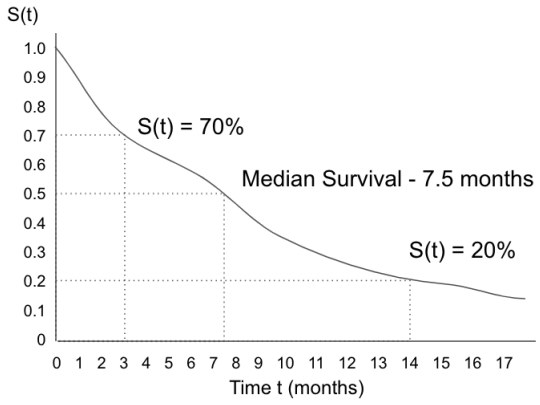

Figure 2.2: Survival Curve Graph Example

The estimator of the function $S(t)$ is described in the equation 2.2, which can be easily used to estimate the survival time for complete data, however, if some subjects are censored, $\hat{S}(t)$ cannot be obtained [16].

$$\hat{S}(t) = \frac{\#\text{ subjects with survival time T} > \text{t}}{\#\text{ subjects}} \tag{2.2}$$

By using the survival equation estimator 2.2 to estimate the censored data example in table 2.1 which corresponds to the example in image 2.1, for $t \leq 4$, we have the following results. The $\hat{S}(t \leq 4) = 100\%$, since all the subjects have the survival time greater than four. However, $\hat{S}(5)$ cannot be estimated, since the number of survivors is unknown, and that is because there is information on subject B until $t = 4$, but beyond that is undetermined, for example.

Table 2.1: Censored Data Example

Subject	Survival Time
A	5
B	4+
C	10+
D	7

The other function related to survival analysis is the Probability Density Function (PDF) 2.3, which is defined by the probability of time to event of interest happening in the small interval $(t, t + \Delta t)$ per unit time, it is a non-negative function, meaning that $f(t) \geq 0$ for $t \geq 0$ and $f(t) = 0$ for $t < 0$ [16].

$$f(t) = \frac{\lim_{\Delta t \to 0} P[T \text{ in the interval } (t, t + \Delta t)]}{\Delta t} \tag{2.3}$$

For uncensored observations, the PDF $f(t)$ 2.3 estimator is defined as the proportion of individuals that experienced the event of interest in the interval per unit width, the PDF estimator is seen in equation 2.4 [16].

$$\hat{f}(t) = \frac{\#\text{ subjects experienced the event of interest in the interval beginning at time t}}{(\#\text{ total subjects})(\text{interval Width})} \tag{2.4}$$

Finally, the Hazard function 2.5 gives the conditional failure rate. It is calculated similarly to the PDF 2.3, however, it provides the probability of time to event of interest falling in the small interval $(t, t + \Delta t)$ given the time to event of interest is greater than or equal to the lower bound $T \geq t$ [16].

$$h(t) = \frac{\lim_{\Delta t \to 0} P[T \text{ in the interval } (t, t + \Delta t) \mid T > t]}{\Delta t} \tag{2.5}$$

The Hazard function can be rewritten using the cumulative function, as seen in equation 2.6 [16]

$$h(t) = \frac{f(t)}{1 - F(t)} \tag{2.6}$$

An estimate to the Hazard function can be seen in equation 2.7, which, for uncensored information, represents the instantaneous potential of a given subject to experience an event at time t, given that the event did not happen until that time [16].

$$\hat{h}(t) = \frac{\#\text{ subjects experienced the event of interest per unit time in the interval}}{\#\text{ subjects not experienced the event of interest at time t}} \tag{2.7}$$

To serve the purpose of predicting the survival function using censored data, it is important to make use of the appropriate methods that are prepared for that. The next section will present methods that are compatible with censored data, they will be introduced divided into two main groups [28]:

Statistical Models Non-parametric and Semi-parametric

Machine Learning Models Support Vector Machines, Gradient Boosting, Random Survival Forest

There are more models in the context of survival analysis than the ones presented in the following sections, both statistical and machine learning ones. But, the models introduced here are limited to the ones used in this dissertation.

2.2 Statistical Models for Survival Analysis

Statistical models are used to propose an estimator to the survival function or the hazard function [28]. They can be divided in three major groups: Non-parametric, Semi-parametric and Parametric estimators. But in this section, only the first two will be further detailed.

2.2.1 Non-Parametric Estimators

Non-parametric estimators do not rely on assumptions about the distribution of the underlying population. They describe the data by defining an estimator to the survival function S(t) seen in the previous section 2.2. As it was stated previously, this function cannot be calculated straight from the censored data, otherwise resulting in an underestimated survival time for the censored subjects [2].

These estimators are easy to understand, to use and more efficient than Parametric estimators when the survival time is not described by a known distribution [16, 28]. They are often used as a first step in the survival analysis to evaluate the data or are integrated to other survival analysis approach as part of the prediction process [2].

2.2.1.1 Kaplan-Meier estimator

Kaplan-Meier's is one of the statistical approaches to perform survival analysis that works for censored data. To accomplish that, some assumptions are made:

- Censoring does not impact the probability of developing the event of interest, i.e. censored and non-censored individuals have the same survival probability;
- Subjects with different enrollment times (i.e. later or earlier enrollment) in the study have the same survival probabilities;
- The time of the event happens at the specified time [28].

The Kaplan-Meier estimator for the survival function is in the equation 2.8. Given a time t with $t > t_j$, d_j is the number of subjects that had already experienced the event of interest up to time t_j, n_j stands for the number of subjects that have not experienced the event up to time t_j.

$$\hat{S}(t) = \prod_{j|t_j<t} \left(1 - \frac{d_j}{n_j}\right) \tag{2.8}$$

For time t = 0, we have $d_0 = 0$ and n_0 = total number of subjects enrolled in the study, resulting in $\hat{S}(0) = 1$, as previously stated. In the case of a censored subject, the t_i will correspond to the last time available, the censorship time [16].

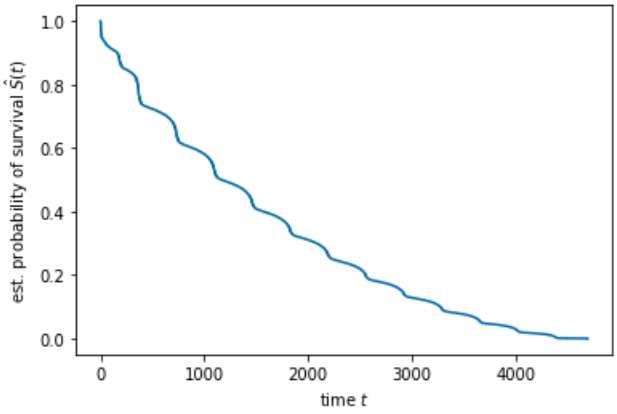

Figure 2.3: Kaplan-Meier Survival Curve Graph - Kidney Transplant

Using the equation 2.8, the Survival function in graph 2.3, which represents the survival time for Kidney transplant, can be described as a step function, each step corresponding to a time to event t_i of the uncensored data. The 'steps' are not regularly distributed, it depends on the number of subjects that experienced the event and when they happened [16].

To define the estimator using Kaplan-Meier survival function, it is clear that only two variables are needed: time to event and censored subjects. However, in many studies, it is also interesting to evaluate the influence of other variables in the survival time. For example, when analyzing the survival time after a graft transplant, other variables like age, blood type or even diabetes can contribute with information to the model. To use this estimator in those cases, the data should be grouped according to each covariate values and a specific survival curve drawn for each grouped covariate as seen in figure 2.4, which presents different curves for each cell type in the context of cancer survival time. However, as the number of covariates grow, this approach becomes unfeasible, so other models should be considered [28].

2.2.2 Semi-Parametric Estimators

Semi-Parametric estimators do not require the assumption of a distribution to the survival times but they include a parametric component to estimate the survival time [28]. Differently from the Non-Parametric, they are capable of incorporating covariates more easily, enriching the analysis.

There is a set of different semi-parametric estimators that can be explored. In this section, the Cox model and its variations will be described.

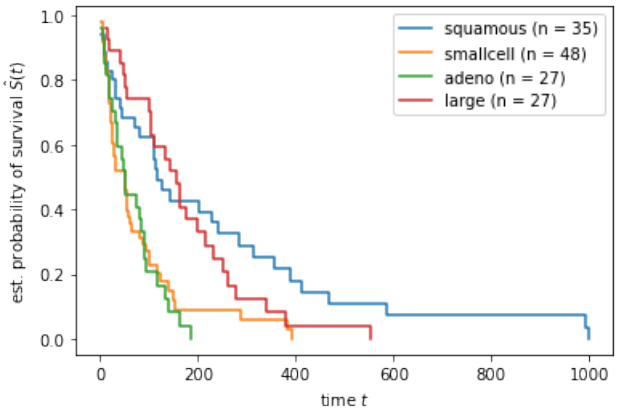

Figure 2.4: Veterans' Administration Lung Cancer Trial grouped by Cell type - Scikit-survival (Source: Scikit-survival datasets)

2.2.2.1 Cox Models

Cox Models are considered semi-parametric methods that can estimate the relationship between the hazard rate and the covariates without the need to make assumptions about the distribution of the baseline hazard function [16]. In the hazard function 2.9 the term $h_0(t)$ is the baseline hazard function, which is the non-parametric part of the formula described as a non specified function [2], X_i stands for the vector of covariates of the individual i while β is the vector of coefficients. The $exp(X_i\beta)$ is the parametric portion, which contains the covariate vector and the coefficient vector [28].

$$h(t, X_i) = h_0(t)exp(X_i\beta) \tag{2.9}$$

An important assumption for the cox models is that the hazard function is proportional, or the hazard ratio is constant. In other words, given two individuals with hazard function in time t = 1, $h(t = 1, X_1)$ and $h(t = 1, X_2)$ with the hazard function of subject one being two times the hazard function of subject two $h(t = 1, X_1) = 2h(t = 1, X_2)$, for $t_j > 1$ that proportion will be maintained throughout time. There are approaches to evaluate whether the proportional hazards' assumption holds, like plotting the Kaplan-Meier curve to verify any crossings between subjects or, a more robust approach, the complementary log-log plot can add more precision to the analysis [2].

For the Cox's Proportional Hazard's model, all the covariates are considered and added to the final model, regardless of their relevance. For a smaller set of features, this approach may work properly, however it is not recommended for a larger set of covariates (p » n), p = number of covariates and n = sample size, given the higher risk of correlated features being present, resulting in a possibly non-singular matrix [9].

To deal with that problem, Ridge's Penalized Cox Model uses a penalty variable that can shrink certain coefficients to zero, however it maintains the need to evaluate the expression levels of all features to make predictions. The penalty coefficient is described in equation 2.10. The first part, the λ is the parameter that controls the shrinkage of the variables, while β is the

coefficients vector with dimension p, which is the total number of features [9].

$$\frac{\lambda}{2}\sum_{j=1}^{p}\beta_j^2 \tag{2.10}$$

Ridge's approach assigns a shrinkage penalty to the less relevant variables, but does not effectively exclude the redundant features. Consequently, this approach has a high computational cost. So, instead of using the entire set of features, LASSO approach proposes to continuously select the most predictive subset of features. Its penalty coefficient is similar to the one seen in Ridge's equation 2.10, except that instead of the β^2 it uses $|\beta|$ as seen in equation 2.11 [28]

$$\lambda\sum_{j=1}^{p}|\beta_j| \tag{2.11}$$

The two main drawbacks of LASSO's approach is that, first, it is unable to select a number of features greater than the sample size, which affects its results in high-dimensional dataset. Secondly, when the model encounters a set of features closely related, it randomly selects any feature. To tackle those issues, the Elastic Net approach proposes to use a combination of LASSO and Ridge described in the equation 2.12 [9], this way, the Elastic Net penalty has the subset selection property of the LASSO with the regularization capability of the Ridge penalty.

$$\lambda\left[\alpha\sum_{j=1}^{p}|\beta_j| + \frac{1}{2}(1-\alpha)\sum_{j=1}^{p}\beta_j^2\right] \tag{2.12}$$

2.3 Machine Learning Models for Survival Analysis

Machine Learning is a successful approach to model non-linear and linear output data. However, the same challenges faced by statistical modeling when it comes to censored data are also present in traditional Machine Learning. Because of that, adapted versions of the traditional machine learning models were proposed. Some of those models are presented in the following section.

2.3.1 Survival Support Vector Machine Model

Due to the flexibility of Support Vector Machine (SVM) models, researchers proposed different approaches to adapt it to address the survival analysis problem: (i) ranking-based, (ii)regression-based and (iii) a combination of the (i) and (ii) [23]. The first approach, ranking-based, uses the order of the predicted data and the original data, however it does not consider the actual time from neither of them, only the ordering. The second approach, proposed the use of Support Vector Regression (SVR) for censored Data (SVRc), using an asymmetric loss function [28].

Finally, a SVR-based approach was proposed, which combines (i) and (ii) in the context of survival analysis. Other different approaches were proposed using Relevance Vector Machine

(RVM), which uses Bayesian inference, adding a probabilistic classification on the formulation of SVM, for example [28].

In the context of this project, only one SVM implementation was explored, the linear Support Vector Machine model. It is a linear implementation of the survival SVM presented above in (iii).To better understand how it uses censored data, the optimization function described in 2.13 is discussed.

$$argmin_{\mathrm{w},b}\frac{1}{2}\mathrm{w}^T\mathrm{w}+\frac{\alpha}{2}\left[r\sum_{i,j\in P} max(0,1-(\mathrm{w}^T\mathrm{x}_i-\mathrm{w}^T\mathrm{x}_j))^2+(1-r)\sum_{i=0}^{n}(\zeta_{\mathrm{w},b}(y_i,\mathrm{x}_i,\delta_i))^2\right] \quad (2.13)$$

The equation's 2.13 input is the triplet $(\mathrm{x}_i, y_i, \delta_i)$ for the i-th observation, being x_i the feature vector, the y_i the positive survival time and δ_i the censoring flag, which can be zero or one, only. It also has the r variable, which serves as the trade-off parameter which balances between the ranking objective and the regression one, when $r = 1$ the equation is equivalent to the ranking-base approach and when $r = 0$ to the regression approach [7]. Finally, the α parameter that is a positive parameter that controls the regularization level which means that, for smaller values of α, the regularization is greater and for bigger values of α, the regularization is smaller.

Like other SVM models, the goal of this equation is to find the parameters w and b that satisfies the optimization described above with the best values for α and r. However, when it comes to censored data, the SVM has to take into consideration the censored elements as well.

To do so, the last part of the equation uses the censoring flag δ_i of the i-th observation, it is further described in equation 2.14.

$$\zeta_{w,b}(y_i,\mathrm{x}_i,\delta_i)=\begin{cases} max(0,y_i-w^T\mathrm{x}_i-b), \text{ if } \delta_i=0 \\ \\ y_i-\mathrm{w}^T\mathrm{x}_i-b, \text{ if } \delta_i=1 \end{cases} \quad (2.14)$$

For the function 2.14 the input is the triplet of the i-th element: y_i survival time, x_i the features vector and the censoring flag δ_i. Depending on the value of δ_i which indicates a censored data with $\delta_i = 1$ and $\delta_i = 0$ otherwise, it provides different calculation.

There are other approaches to the Survival SVM, the one described here is better described in the publication "Fast Training of Support Vector Machines for Survival Analysis, Machine Learning and Knowledge Discovery in Databases" [23].

2.3.2 Random Survival Forest Model

Random Survival Forest (RSF) is a machine learning model that relies on smaller predictive structures called survival trees. Survival tree method is a fully non-parametric approach to survival analysis, able to successfully model high-dimensional datasets [20].

The idea behind RSF is that having a wide variate of not too complex survival trees is better than one complex survival tree. So, to build that, the first step is to randomly select samples from the original dataset, each sample will originate a survival tree. Then, while building the three, at each step a subset of the variables is selected [15].

On the first step, the bootstrapping selects different sets of elements, repeated or not, from the original dataset. However, around 37% of the original data will not be selected and that unselected part is called out-of-bag (OOB) data [15]. On the second step, the variables in each node are used to maximize survival different between daughter nodes. And finally, each leaf should have at least one unique element.

Once the Random Forest is built, the Cumulative Hazard Function is calculated by the average of all trees' cumulative hazard function. And to evaluate the quality of the model, the OOB data is used to calculate the prediction error [15].

2.3.3 Gradient Boosting Survival Model

Gradient Boosting models are ensemble learning methods that builds the model iteratively using weak predictive structures. So, given data in the format x_i, y_i, $0 < i < n$, with n representing the sample size, x_i the covariates and y_i the target variable of the i-th element, the general goal is to learn a functional mapping $y = F(x, \beta)$, where β is the set of parameters of F, such that a generic cost function, represented by equation 2.17, is minimized [11].

$$\sum_{i=1}^{n} \Phi(y_i, F(x_i, \beta)) \tag{2.15}$$

F(x) is defined as in equation 2.16, which defines the steps taken by the gradient boosting model when combining the weak learners. Looking at each component, first, the M is the total number of iterations, and τ is part of β.

$$F(x) = \sum_{i=1}^{M} \rho_m f(x, \tau_m)) \tag{2.16}$$

The Gradient Boosting process works like this: (i) the initial $f_0(x)$ is set; (ii) at each iteration m, calculate the pseudo residuals, then fit a weak learner to the chosen values and finally (iii) calculate the prediction the new learner provides. On the last step, the learning rate $v, 0 < v \leq 1$ is used to reduce the impact each new weak predictor has on the prediction, with the goal to improve accuracy in the long run.

When applying Gradient Boosting to survival analysis, the Cox model is used and the cost function is defined as the negative log partial likelihood [11]. This is one possible approach to address censored data.

$$\Phi(y, F) = -\sum_{i=1}^{n} \delta_i \left\{ F(x_i) - \log(\sum_{j|t_j \geq t_i} e^{F(x_j)}) \right\} \tag{2.17}$$

2.4 Metrics

The same way censored data requires specific models, there are specific metrics to evaluate the performance of those models. Depending on the primary interest of the analysis or the proportion of censored data, some metrics can be better to evaluate the performance of the model than others. Three metrics are listed below with a short description.

Concordance index (C-index) this evaluation method is based on the ordering of the prediction compared to the original data. So, given (i, j) a comparable pair with survival times t_i, t_j and predicted survival times as $\hat{S}(t_i), \hat{S}(t_j)$. The pair (i, j) is concordant if $t_i > t_j$ and $\hat{S}(t_i) > \hat{S}(t_j)$ and discordant otherwise. In the case of a pair containing a censored element k, it is only comparable with any other uncensored data m if $t_k > t_m$, being t_k the observed time of k. The c score is generically calculated in $c = P(\hat{y}_1 > \hat{y}_2 | y_1 \geq y_2)$, there are multiple ways of calculating it, depending on the model used [28]. The score varies from [-1,1], being 1 the perfect ordering.

2.5 Scikit-survival

The Scikit-survival [22] is a Python library developed to address survival analysis specifically. It was build on top of Scikit-learn [21], a machine learning library in Python. All the methods and metrics previously covered in this chapter are available in this library and were used in the context of this dissertation.

This library provides the entire set of tools to perform the analysis, training and testing of the survival models. Before using the models, just like any other machine learning method, they have to be trained. So, the first step is to run the `.fit()` command using the training data. Once the model is trained, it can be used to predict the survival time, using the `.predict()` function, of any similar data, e.g. the test data or real-life data. The input data of the training step can vary depending on the model, it can require only the survival time and the event array, or require more information like the features.

In the library's documentation, further the implementation details are available for each model [7].

2.6 SRTR Database and Data Processing

Medical data is abundant, which does not mean they are accessible. Both hospitals and medical studies collect various data from patients, however, due to the sensitive nature of this type of data, it can be difficult for researchers to have access to that valuable information [12]. Also, the entire data does not lay in one place only, it is scattered throughout different healthcare

providers, universities, and organizations, stored in different formats [12]. And that is why it is important to have Organs, for instance, responsible for the ownership of the data, anonymizing it and making it available.

2.6.1 The Scientific Registry of Transplant Recipients Database

The data used in this project was obtained through a request performed to the Scientific Registry of Transplant Recipients (SRTR). The data set provided is in .DAT format, with the header presented in table format in HTML files. The entire database corresponds to 53 tables containing information from many graft transplants, e.g. Intestine, Kidney, Pancreas, Liver, performed in the United States, from both citizens and foreigners.

The dataset dates from October 1, 1987, to December 2020, the last being the month the files were shared by the Organization. The complete list of tables and the corresponding description for each variable can be found in the documentation. The features present in SRTR database can be complex and, in some cases, specific to the context of the American transplant system. This section further details some of those variables that are relevant to this dissertation.

The main table used in this project is named KIDPAN DATA, which means Kidney and Pancreas Data, it is composed of 480 columns and more than one million entries, which corresponds to 3.7 GB of memory once loaded in Python as a dataframe. This table merges information from patients, donors, medical followup and waiting lists. The types of transplants represented in this table are: Kidney, Kidney and Pancreas and Pancreas only.

2.6.2 Medical Concepts

There is a set of features worth exploring with more details, given the importance they have in the Kidney transplant context. Unlike weight and Body Mass Index (BMI), which are more common concepts and familiar to most, the Human Leukocyte Antigen (HLA) or Calculated Panel-Reactive Antibody (CPRA) are less common to the public. The information presented in this topic can be encountered in the Organ Procurement and Transplantation Network (OPTN) website and documentation.

2.6.2.1 Human Leukocyte Antigen (HLA)

The immune system's main job is to distinguish self from nonself elements. When a nonself element is identified, an immune response can be triggered with the goal to eliminate it from the body. Not only external elements can trigger an immune response, but also abnormal cells deriving from the host tissues. So, if the system is able to recognize a molecule, regardless of where it comes from, it is considered an antigen [4].

The responsible for recognizing those molecules is the HLA, which is defined by the genes on

a region known as Major Histocompatibility Complex (MHC). The genes of that region are of extreme importance to the immune response of the body. Some HLA antigens are encountered on every tissue of an individual body, not only on blood cells [24].

For example, in every nucleated cells of the body, the class I HLA can be found, which is divided into three subgroups according to the loci on the MHC region of the DNA: -A, -B and -C. There is also the HLA of class II, which is usually only present in professional antigen-presenting cells and is represented by the loci -DP, -DQ and -DR [4, 24].

The MHC class I and II molecules play a big part in the transplant compatibility. According to the author of "The Human Leukocyte Antigen (HLA) System", in the context of renal transplants, there are two main requirements to prevent complications with the transplant: HLA compatibility and ABO compatibility [24]. However, obtaining a good HLA matching is a difficult task.

The challenge of matching the HLA of donors and recipients is a known problem, not only because of the diversity of HLA antigens, but also because of the anti-HLA every individual produces under certain circumstances [24]. When in contact with different antigens from their own, through a pregnancy, blood transfusion or organ transplant, an individual can produce anti-HLA antibodies against the foreigner HLA.

Besides that, the diversity of the HLA antigens makes the search for a compatible donor even more difficult. To solve such problem, the matching criteria used by the UNO's algorithm allows each antigen to match itself and, depending on the case, other related antigens [5]. The rules to determine a related antigen are complex and will not be covered here.

Table 2.2: HLA variables available in the SRTR Database - Recipient and Donor

	Recipient	Donor
A	A1	DA1
	A2	DA2
B	B1	DB1
	B2	DB2
DR	DR1	DDR1
	DR2	DDR2

The SRTR Database not only provides the information of the main HLA to the renal transplant context, i.e. A, B and DR, but also provide the mismatch levels [24]. The HLA system is represented in the database using six columns for the A, B and DR HLA loci of both the donor and the recipient as seen in table 2.2. While the mismatch levels between donor and recipient for A, B and DR are stored in three different columns AMIS, BMIS and DRMIS, with a forth column summarizing the mismatch level of all HLA antigens, the HLAMIS as seen in table 2.3

The mismatch levels are calculated based on the columns from table 2.2. For example, when the pair of columns A1, DA1 and A2, DA2 have the same values or values that represent related antigens, the A mismatch level is zero.

Table 2.3: Values for the variables AMIS, BMIS, DRMIS and HLAMIS in the SRTR Database

Variables	Values
AMIS	[0, 1, 2]
BMIS	[0, 1, 2]
DRMIS	[0, 1, 2]
HLAMIS	[0, 1, 2, 3, 4, 5, 6]

2.6.2.2 PRA and CPRA

The Panel-Reactive Antibody (PRA) is a concept closely related to HLA discussed previously. Each individual has a set of HLA antigens and when in contact with different HLA antigens from their own, the body triggers an immune response by producing HLA antibody. There are three main ways for a person to get in contact with different HLA antigens: blood transfusions, prior transplants or pregnancies [24].

To calculate the PRA percentage of a patient, their blood is test against white blood cells from 100 blood donors, and the number of reactions within that panel corresponds to the value of PRA. The blood donors represent the potential HLA makeup for a donor from that area, so if a patient reacts to 40 of the samples, theoretically, if a donor of that region becomes available, that patient's chances of experiencing acute rejection are 40% [6]. For higher PRA values, the patient have to wait longer to find a compatible donor available.

PRA does not provide the information of which HLA antigens a patient is reacting to, only the percentage of reactions. To complement that information, the transplant center input the unacceptable antigens for the candidates, based on the antibody produced by them, and the potential donors with those antigens will not even be considered as transplant candidates for them. Since the antigen information in the national donor pool is also available, the calculation of the likelihood that the recipient and donor would be incompatible is possible. And this is known as the CPRA [6]. In summary, the cPRA is calculated based on the unacceptable HLA antigen frequency in the pool of available organs.

Depending on the PRA or CPRA values, recipients can receive points in the kidney allocation formula, otherwise sensitized patients would wait much longer than others.

2.6.2.3 Blood type system

The compatibility of blood type between donor and recipient is also important to a successful transplant. The blood type phenotypes considered in the SRTR database are: O, A, A_1, A_2, B, AB, A_1B, A_2B with the subtypes A_1 and A_2.

The blood type is defined by the antigen(s) present in an individual's red blood cells. Besides those antigens, humans also have the antibodies to the antigen(s) they are lacking from, so, for example, the blood type O has neither antigen A nor B, so the antibodies present in the

serum are both Anti-A and Anti-B. Moreover, there is a variation in A antigen expression, which corresponds to the types A_1 and A_2 [26]. Other variations of the A antigen exist, but only those two are represented in the SRTR database.

The blood types are not equally likely in the population, there are certain blood types, like type B, that are less common than other and, consequently, the average waiting time for a transplant is higher. Because of that, United Network for Organ Sharing (UNOS) has included a policy that allows a recipient with type B blood, who meet certain criteria, to receive a kidney from type A, non-A_1, AB and non-A_1B donors [25]. Based on that, the compatibility information used by the SRTR database (the ABO_MAT variable), can be observed in table 2.4, which also includes those special cases.

Table 2.4: Compatibility level of ABO blood types

Recipient \Donor	O	A	A_1	A_2	B	AB	A_1B	A_2B
O	1	3	3	3	3	3	3	3
A	2	1	1	1	3	3	3	3
A_1	2	1	1	1	3	3	3	3
A_2	2	1	1	1	-	-	-	-
B	2	3	3	3	1	3	3	3
AB	2	2	2	2	2	1	1	1
A_1B	2	2	2	2	2	1	1	1
A_2B	2	2	2	2	2	1	1	1

Chapter 3

Related work

In this chapter, studies related to the prediction of graft survival time after a kidney transplant are presented and discussed, focusing on the differences of their approaches and the corresponding results. The approaches used by those studies are both proposing new models or by evaluating the matching process and the variables used. This is a rich subject that had shown to have different approaches to it, with the ultimate goal to improve the patients lives.

In the paper "Deep Learning for Patient-Specific Kidney Graft Survival Analysis" [18], the authors propose a deep learning method based on the principle of multitask learning to predict the survival times for graft patients. Their motivation is the limitations present in the Cox model, which only models the hazard function and assumes that the effect of the analyzed features does not vary over time. The paper uses both C-index and a modified area under the receiver operating characteristic (AUROC) as performance metrics to evaluate the results. The best deep learning method presented, with a c-index of 0.655, was MLP Efron's + rank.

This paper suggests a better prediction model than the traditional Cox models approaches, however, when considering its applicability on the context of medical use, the black box model does not provide an easy visibility on the meaningful features and their individual impact on the survival time or on the performance metric. Basically, to achieve this visibility, the same strategy used with Kaplan-Meier is recommended, plotting the survival curve for each feature of interest and their unique values. That is not an easy task when hundreds of variables can be used as the input of the prediction model. In comparison to the proposed model, There are many other machine learning models that have an interesting performance while allowing an easier analysis of the features' impact on the survival time. Besides that, this paper does not list the features used and the overall impact each variable has on the outcome.

The paper "Machine Learning Approaches for Prediction of Kidney Transplant Survival" [29] instead of proposing improvements to the model itself, suggests that the data can be better represented to achieve better survival time predictions. It compares the results from five different representations of the data, mainly focusing on the HLA representation. It uses a set of models, from Cox Models to machine learning ones, to evaluate the impact of the different representations.

The performance metric used was the c-index as well.

The previous study suggested a set of different representations of the data, varying from 44 features to more than 3,000 features. So, instead of pursuing the simplification of the dataset, it adds complexity to it.

Both studies used the Scientific Registry of Transplant Recipients (SRTR) database, which is also used in this dissertation, with different selection of variables and filtering criteria. On the first paper, the dataset had 131,079 deceased donor transplants, from 2000 to 2014 and most features were used except for the ones that had more than 20% of missing values, but no proper information is disclosed on the specific features selected. While in the second paper, the author used a different table from the dataset, filtered out according to specific criteria, like PRA $<$ 80% and age, which resulted in 56.299 transplants. However, when selecting the features, they used literature review method, which resulted initially in 35 features, later on being expanded.

The results obtained by this author can be seen in the table below. As he referred in his work, the best performance was presented by the *GradientBoostingSurvivalAnalysis* with the best c-index of 64.85. The table shows the results obtained by each model for each different data representation proposed by the author.

Algorithms	1	2	3	4	5
IPCRidge	50.18	49.49	51.60	51.13	50.04
CoxPHSurvivalAnalysis	62.94	63.17	51.19	62.57	62.82
CoxnetSurvivalAnalysis	62.98	63.48	56.47	62.74	62.99
FastSurvivalSVM	61.10	61.63	53.32	61.08	61.82
FastKernelSurvivalSVM	52.62	53.21	49.78	53.39	52.54
GradientBoostingSurvivalAnalysis	64.23	64.85	62.26	64.38	64.28
ComponentwiseGradientBoostingSurvivalAnalysis	59.25	59.98	57.06	60.02	59.59

Figure 3.1: C-index comparisons of all algorithms and datasets of paper "Machine Learning Approaches for Prediction of Kidney Transplant Survival"

In summary, previous works have explored improvements to the models and data representation. One proposes a neural network applied to a highly modified dataset with more than 400 features. The other proposes changes to the representation of the data, making it more complex.

However, when we consider the application of the model and the replicability in different contexts and different Kidney Allocation System (KAS), those proposals fall short. One by relying on the amount of data available and not performing a proper feature selection, and the other by adding to the data dimensionality, making it heavier to run and less interpretable.

Because of that, this dissertation proposes a different approach. For this study, the features were carefully selected based on the analysis of the database and on a literature review, by understanding the meaning of each variable individually. The goal is to demonstrate that, by

selecting the right set of features, we can reach as good results as complex modeling of the data while allowing this approach to be replicated in different contexts, given the features used.

Chapter 4

Development & Results

This chapter's objective is to present the work developed using the concepts introduced in the background chapter. It is divided into two main parts, the first being related with the dataset, the preprocessing it went through and the feature engineering, the second presents the methods used and their training approach.

4.1 Dataset

The Scientific Registry of Transplant Recipients (SRTR) database information is distributed in many tables, so the first challenge was to separate the useful information from the rest. According to the documentation provided, with all the files, the entire dataset has a total of 53 tables, but only 16 of those are somehow related to Kidney transplant and the others refer to other organ type. After an evaluation of the information presented in each table, both "KIDNEY_FOLLOWUP" and "KIDPAN_DATA" were considered relevant. Another option was to merge a couple of tables, however, due to memory restrictions this option was discarded. So, "KIDPAN_DATA" was chosen as the foundation of the study with the output variables defined as ['GTIME_KI', 'GSTATUS_KI'], representing the graft survival time and the censoring flag, respectively.

The KIDPAN_DATA table contains data of transplants performed from 1987 to 2020. It has information on both the donor and the recipient before and after the transplant happened. This table provides a full panorama on the transplants because it merges data from the followups, waiting list, deceased and living donor list, etc. Each row on the table corresponds to a unique transplant, with a total count of 1.026.968 transplants, and each feature is represented by a column, being around 480 columns in total.

When importing the data, there were mixed typing in the features and the missing values were represented by single dots. So, the first step was to set the missing values to null and then identify the type of every feature correctly. An automated process was implemented to perform the second step, so the dataset could be finally explored.

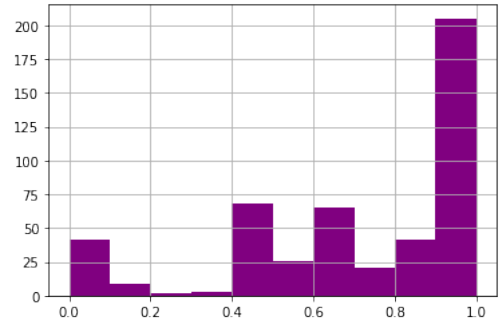

Figure 4.1: Database % null values per feature analysis

The figure 4.1 is a histogram that displays the distribution of features according to the percentage of missing values on the original dataset, represented by the x-axis. There are around 200 features with more than 90% of missing values. That happens because, according to SRTR documentation, features have changed since 1987, new information was added to the dataset and certain features have been deprecated. The same visualization but on the rows' context can be seen in image 4.2, almost half of the transplants in the dataset have more than 80% of the features as null.

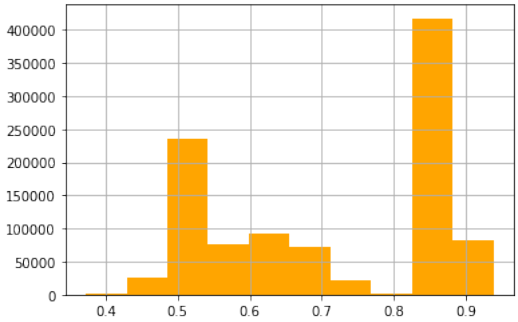

Figure 4.2: Database % null values per row analysis

Since this is a high-dimensional database with many records, this first analysis helped to identify which actions should be taken next. In this case, by analyzing the features available, we opted by filtering the data and see if this would already reduce the missing values.

4.1.1 Data Filtering

As seen above, there are many features that are not interesting, given they have many missing values, and because of that, those features may not add much to the analysis. But before focusing on them, based on the goal of this dissertation, five important features were used on the filtering process, and they are listed below:

ORGAN this variable indicates which transplanted organ each row corresponds to, with three option: KI, KP and PA. The value ORGAN = KI refers to Kidney only, which is the only

case useful for this study.

AGE_GROUP this feature has two possible values: A for Adult and P for Pediatric, with a proportion of 95% and 5%, respectively, only adult recipients were selected because, according to the database documentation, pediatric transplants have a different set of features when comparing with adult transplants, which could add complexity to the analysis.

NUM_PREV_TX refers to the number of previous transplants. It is relevant information because transplant is one of the triggers to HLA antibody production that can increase the chances of acute rejection. So, only patients with no previous transplant were chosen.

DON_TY there are three types of transplant in the SRTR database, L - living donor, D - deceased donor and F - foreigner. The Foreigner flag does not indicate whether the donor was alive or not, so only L and D were considered.

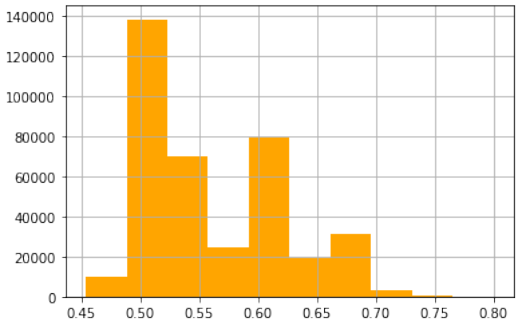

Figure 4.3: Database % null values per row after data filtering

After the filtering, the resulting dataset has 375.498 entries. Even thought that represents less than half of the initial dataset, when comparing with other studies, it is a significant amount of transplants to work with [14]. The figure 4.3 shows that after the filtering, the rows with less information have been removed when comparing with the previous graph 4.1. The next step was to select the relevant features.

4.1.2 Feature selection

Two different strategies were considered when selecting the relevant features. First, a manual removal of irrelevant columns was performed, among those were columns related only to Pancreas transplant, identifiers, and payment methods. Then, an automated process based on the null proportion on each row and the correlation between columns was used. This was the first attempt to navigate through all the features, looking for the relevant information that had the least amount of missing values. This strategy, however, was not pursued, because more than 200 features were selected, making it harder to process the database and differ the meaningful features from those specific to the U.S. transplant system, for example. So, instead of exploring the dataset features blindly, a literature review was performed on that topic.

After a literature review [14, 17], the set of features listed in the table 4.1 were selected in addition to the output variables GTIME_KI and GSTATUS_KI, that corresponds to graft survival time and the censoring flag, respectively. Both categorical and numeric features were used, and the corresponding preprocessing methods were applied.

The days in dialysis (DIAL_DAYS) is an information that is not present in the original dataset, so it was created based on three other variables: DIAL_TRR, TX_DATE (Transplant date) and DIAL_DATE (First day of dialysis). This feature was considered relevant to the survival time according to the paper "election of donor-recipient pairs in renal transplantation: comparative simulation results" [17].

Table 4.1: KIDPAN_DATA - Selected variables

Variable	Variable Type	Description
AGE AGE_DON	Numeric	Age (years) of both recipient and donor
GENDER GENDER_DON	Categorical	Gender of both recipient and donor
BMI_CALC BMI_DON_CALC	Numeric	Body Mass Index for both the recipient and donor
ETHCAT ETHCAT_DON *	Categorical	Ethnicity group of both recipient and donor
ABO_MAT	Categorical	Blood type match level between recipient and donor
AMIS, BMIS, DRMIS, HLAMIS	Categorical	The mismatch levels for A, B and DR antigens
DIAB	Categorical	Diabetes information on the recipient
DAYSWAIT_CHRON_KI	Numeric	Total days on Kidney waiting list
DON_TY	Categorical	Donor type - deceased or living
DIAL_TRR DIAL_DAYS	Categorical, Numeric	recipient pre-transplant dialysis (Y, N), days of dialysis before transplant
END_CPRA	Numeric	candidate most recent calculated PRA

4.1.3 Data preprocessing

The preprocessing of the data consisted in, first, dealing with the null values. In this case, the entries with any missing values were completely removed, to avoid the imputation of data. With the removal of rows, the final data set has 195.085 entries and a total of 19 features.

Next, the categorical features were transformed using `get_dummies` Pandas built-in function, which increased the dimensionality according to the number of unique values of each categorical feature as seen in table 4.2. Every categorical feature was processed except for AMIS, BMIS, DRMIS and HLAMIS, following the goal of this dissertation, to avoid increasing the dimensionality,

which, consequently could impact the interpretability of the results and the models used.

Table 4.2: KIDPAN_DATA Categorical features - results after transforming categorical features

Original feature	Categories	Corresponding new features
GENDER	M, F	GENDER_M
GENDER_DON	M, F	GENDER_DON_M
ETHCAT	1, 2, 4, 5, 6, 7, 9	ETHCAT_2, ETHCAT_3, ETHCAT_4, ETHCAT_5, ETHCAT_6, ETHCAT_7 ETHCAT_9
ETCHCAT_DON	1, 2, 4, 5, 6, 7, 9	ETHCAT_DON_2, ETHCAT_DON_3, ETHCAT_DON_4, ETHCAT_DON_5, ETHCAT_DON_6, ETHCAT_DON_7, ETHCAT_DON_9
DIAB	1, 2, 3, 4, 5, 998	DIAB_2, DIAB_3, DIAB_4, DIAB_5, DIAB_998
DON_TY	L, D	DON_TY_L
DIAL_TRR	U, Y, N	DIAL_TRR_Y, DIAL_TRR_U

The numeric features were transformed just for the Cox models using the Scikit-survival library built-in functions. According to the Scikit-survival documentation, those functions normalize the data by subtracting the mean and dividing by the l2-norm [7].

4.1.4 Data Analysis

After the data preprocessing, the final dataset contains 195.293 entries, with a total of 36 features, after the categorical transformation. The graft survival times in the final dataset can be seen in figure 4.4, with the orange part representing the censored times, which corresponds to 78% of all records. The x-axis corresponds to the number of transplants and the y-axis to the survival time. The censored data is concentrated in the first half of the graph, which represents the shortest survival times.

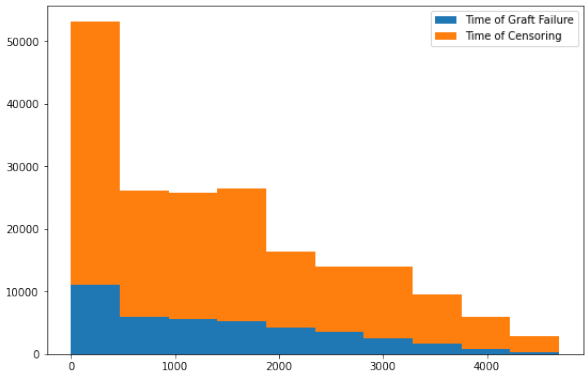

Figure 4.4: Histogram with survival time - censored and non-censored data

The transplants in the dataset were performed from 2007 to 2020, as shown in the graph 4.5. Given that the kidney half life is around 12 years [10], this is an interesting time window to observe the data. As can be seen in the graph, the blue part that represents the observation of the event of interest (i.e. graft failure) proportionally decreases over time, which means older transplants are more likely to have already experienced the event of interest compared to the more recent ones.

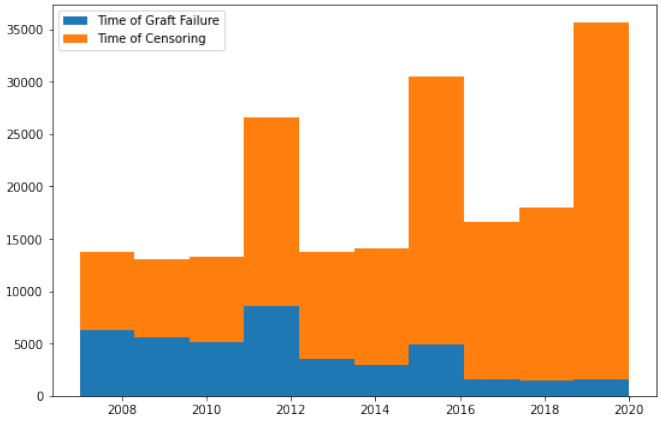

Figure 4.5: Histogram with transplant year - censored and non-censored data

In figure 4.6, the correlation between features is shown. It is important to notice that the only features with a high correlation are AMIS, BMIS, DRMIS and HLAMIS and that is because HLAMIS takes all the other three to be calculated. Besides that, all other features are not correlated enough to be removed from the selected features.

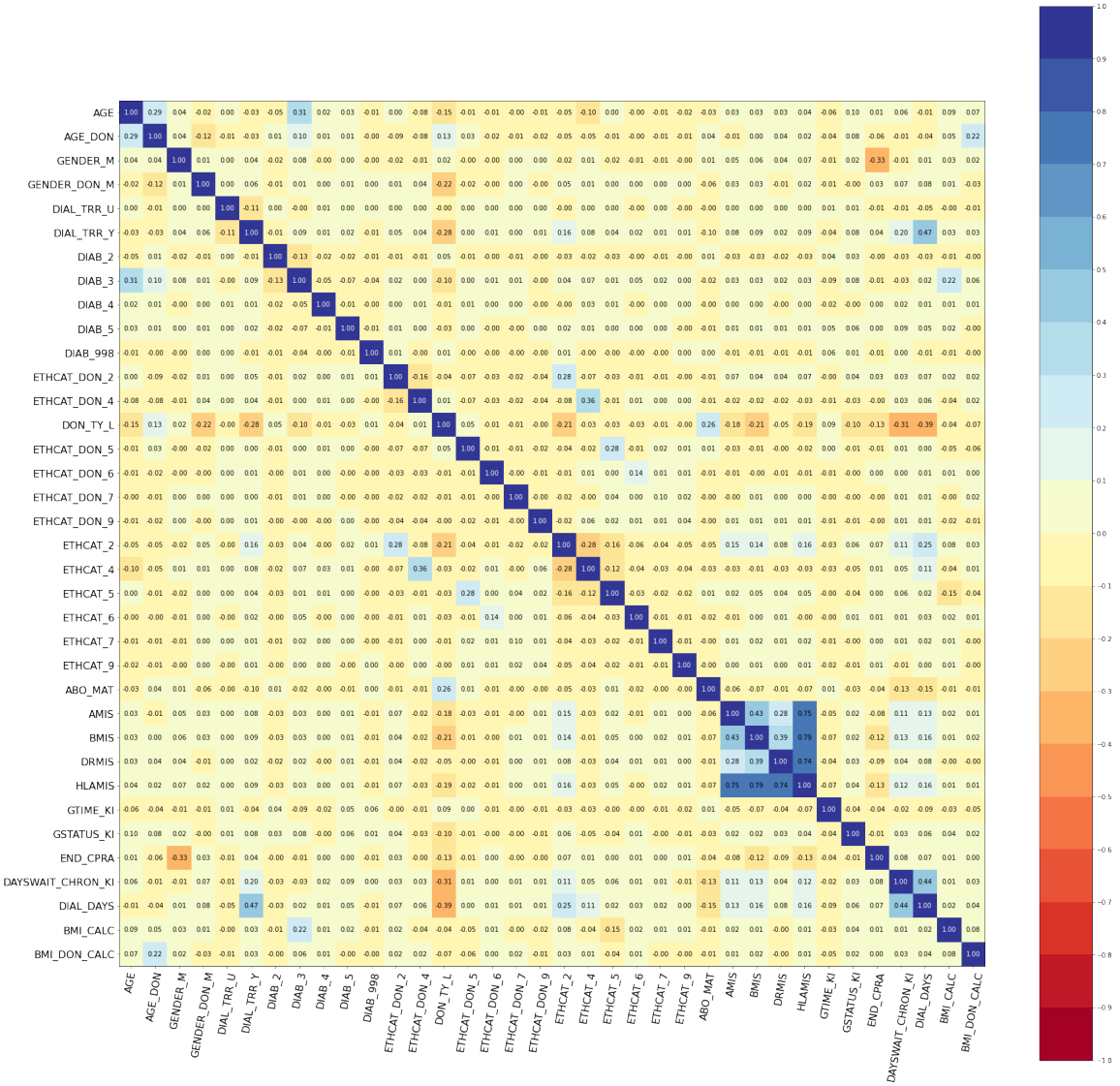

Figure 4.6: Correlation matrix for selected features

4.2 Methods and Results

The goal of the study is to assess the quality of the transplant given by the graft survival time, while identifying the most significant features. To achieve that, the SRTR database was selected, filtered and transformed so both statistical and machine learning methods could be applied to the data and the results interpreted.

However, at the first attempt to train the models using the resulting dataset, the memory available was not enough to complete the processing of the data, forcing the program to restart. Because of that, a sample of the data was used instead of the entire dataset due to those memory restrictions. While some models could be trained using the entire dataset, others required more resources to perform the training, and based on the restrictions imposed by those heavier models, the sample size was defined as 20.000 entries, which was randomly selected from the original

dataset, after the filtering, using the `.random()` function from the Pandas library with the random_state set as 31 for the reproducibility purpose.

Next, the sample set was split into training (80%) and test (20%) sets, maintaining the proportion of censored patients. All models used the same training and test sets.

To evaluate and compared the results from the different models, two different implementations of the c-index were used. The c-index is an important metric, since it is widely used, allowing us to compare the results more easily with different approaches and papers, but depending on the implementation, it can be too optimistic for data with high proportions of censored data, and that is why two different versions were used here [27].

The models and metrics used in this dissertation are implemented in the Scikit-survival library [22], which is a Python library built on top of Scikit-learn. The corresponding functions are listed below:

- `kaplan_meier_estimator`
- `CoxPHSurvivalAnalysis`
- `CoxnetSurvivalAnalysis`
- `RandomSurvivalForest`
- `GradientBoostingSurvivalAnalysis`
- `ComponentwiseGradientBoostingSurvivalAnalysis`
- `FastSurvivalSVM`

Further details on those functions and how they were configured and used are covered for each specific model in the following sections.

The c-index performance metrics are also implemented in Scikit-survival and the ones used correspond to the list below:

- `concordance_index_censored`
- `concordance_index_ipcw`

The first metric is the implementation of Harrell's approach and the second was proposed by Uno, as stated in Scikit-survival documentation [7].

The following sections will cover the training process of each model and how was the process to define the corresponding parameters. Also, they will provide further details on the specific details and differences between the models.

4.2.1 Kaplan-Meier implementation and results

The Kaplan-Meier approach provides as the output an estimator of the survival function. Given the survival time and the censoring information, as explained in the background chapter in section 2.2.1.1, the function `kaplan_meier_estimator()` returns an array with the survival probability for each time t.

The method's output can be plotted as the survival curve seen in 4.7. The graph shows the survival curve for the entire dataset after the preprocessing, this was the only method that used the original dataset after the preprocessing and not the 20.000 individuals sample. The graph does not have evident steps, and that is due to the elevated number of transplants in the sample, making it look smoother. We can observe, based on the 50% percentile, that half the kidneys survived at least 3900 days. The Kaplan-Meier method is an interesting approach to provide a first visualization of the data, it helps understand the survival behavior of the population as a whole.

This survival curve does not provide deeper insights than that. If a new patient starts his treatment today, this curve only provides a generic survival expectancy, regardless of the health status of this person. So, to include further details to the analysis, like blood type of Human Leukocyte Antigen (HLA) mismatch levels, a specific survival curve should be drawn for the specific details.

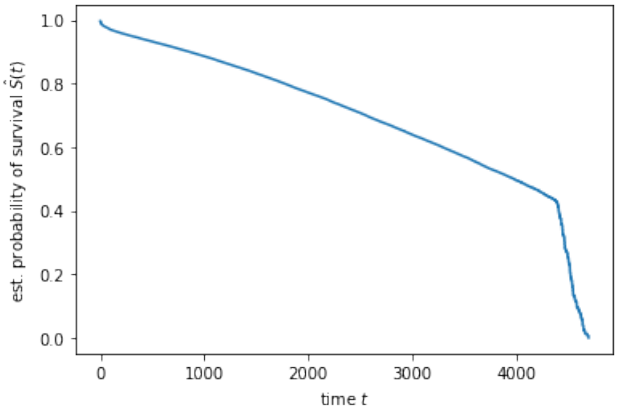

Figure 4.7: Kaplan-Meier Survival Curve Graph - Kidney Transplant

For example, as we can see in figure 4.8, for the HLA levels, there are different survival curves, each representing the survival probability through time for the different groups. This is the approach to evaluate the impact of specific features on the survival curve, stratifying for each unique value of variables available. By observing the curves, we can suppose that the lowest levels of HLA mismatch, i.e. 0,1 and 2, corresponds to a better prognosis than other levels like 3,4,5,and 6.

To actually verify whether the previously stated hypothesis stands, we should validate it using the Log-rank statistic test. To expand the analysis of the dataset using the Kaplan-Meier method,

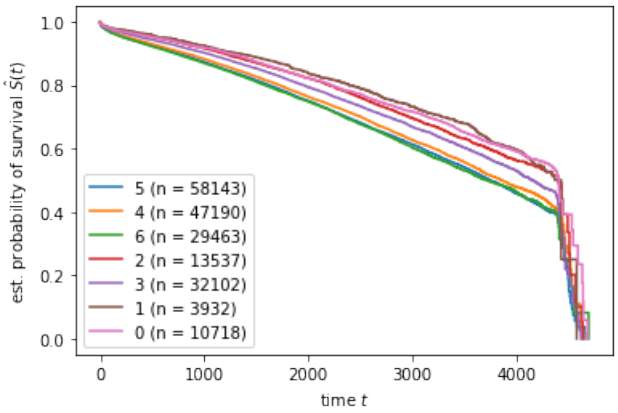

Figure 4.8: Kaplan-Meier Survival Curve Graph - Kidney Transplant divided by HLAMIS values

the survival curve should be calculated for all the features individually and their combinations as well. However, this approach requires an exhaustive exploration, and that is why this method was not further explored in the context of this dissertation.

Nonetheless, it is important to mention that other methods use Kaplan-Meier as part of their predictive process, which will be further explained in the following topics.

4.2.2 Cox Models implementation and results

As discussed in section 2.2.2.1, the Cox models are more flexible than the Kaplan-Meier approach because they can, more easily, use the data variables. For the Cox's Proportional Hazard's model, for instance, all the covariates are considered and added to the final model while the other approaches use penalty parameters to control which variables are considered in the final equation.

The first model used was the Cox's Proportional Hazard's model applied to the raw data. To do so, we used the `CoxPHSurvivalAnalysis()` with no parameters defined. However, as explained in the section 2.2.2.1, this approach can be unsuccessful in the case of too many variables in the data, which increases the chances to have a singular matrix, and that is exactly what happened in this scenario as seen in the output 4.9.

This issue was one of the motivations to further process the data set with a better feature selection, which resulted in the final data sample used. Besides that, even though the Cox's Proportional Hazard's model was explored during the development of this dissertation, it was decided to only use the penalized Cox models instead, starting with the Ridge's Cox model.

Ridge's model is one of the penalized models, so the first step to use this method is to define the penalty variable λ as seen in equation 2.10. In the context of the scikit-survival function for the Ridge's implementation, the λ is represented by the alpha parameter in `CoxPHSurvivalAnalysis(alpha)`. Besides the α parameter, there are more parameters in

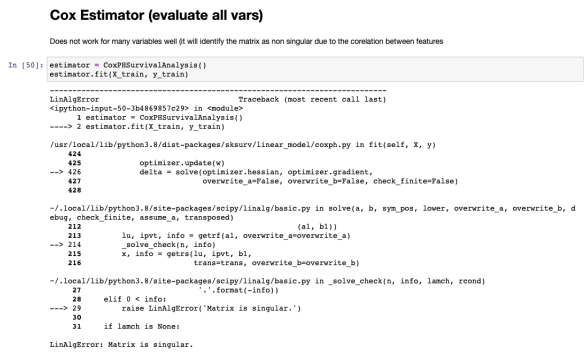

Figure 4.9: Cox's Proportional Hazard's model - Code Error

this function, however, in the context of this dissertation, the default values were used [7].

To identify a good alpha value for the training set, a range of alpha values were manually tested to identify the best performance based on the c-index. The manual process consisted of, first, training the model with a different α value and then verifying the corresponding c-index for that value. At the end, the α with the highest c-index was selected. For Ridge's Cox model, the $\alpha = 0.001$ gave the best Harrell's c-index 0.6569 as seen in table 4.3.

The manual approach was used to explore the LASSO's penalty parameter. Using the scikit-survival function `CoxnetSurvivalAnalysis(l1_ratio)`, with the parameter `l1_ratio` which corresponds to the λ in equation 2.11. With the `l1_ratio` manual selection described above, the chosen value was the `l1_ratio`= 0.5 which presented the best Harrell's c-index 0.6556 as seen in table 4.3.

Finally, the Elastic Net model was the last penalized model used. Through the manual approach, the selected set of penalty variables were alpha_min_ratio = 0.001, l1_ratio = 0.9, with these parameters the corresponding Harrell's c-index was 0.6567 as seen in table 4.3.

The performance metrics for each of the Cox models are in the table 4.3. The three models had a similar performance, but the best model according to Harrell's c-index was the Ridge model. The parameters used to train each model is also in table 4.3 right next to the model's name.

Table 4.3: The performance of the Cox Models

	Harrell's c-index	Uno's c-index
Ridge ($\alpha = 0.001$)	0.6569	0.6829
LASSO (`l1_ratio` = 0.5)	0.6556	0.6812
Elastic Net ($\alpha = 0.01$,`l1_ratio` = 0.9)	0.6567	0.6825

4.2.3 Random Survival Forest Model implementation and results

The Random Survival Forest is an ensemble model, which uses survival trees as base learners. To train this model, it is necessary to identify the size of the forest while avoiding the overfitting to the data. The scikit-survival function for this model is `RandomSurvivalForest`, which has a set of parameters that can be configured to fine-tune the model. In the context of this dissertation, the only parameter configured was the number of estimators, that represents the number of trees in the forest, all the other parameters were not changed, their default values were used, more details can be found in scikit-survival documentation [7].

To identify the best value for the number of estimators variable, the performance of the model was evaluated based on the c-index for a range of estimator numbers. As seen in graph 4.10, we have forests with size 10 to 100 and the best performance was observed around the number 50.

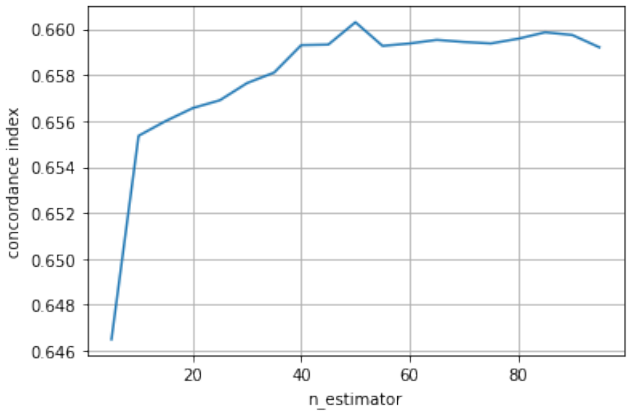

Figure 4.10: Random Survival Forest: c-index for different numbers of estimators n_estimator

This evaluation was performed during the training phase. Once the best number of estimators was identified, the trained model was applied to the test set, and the corresponding c-indexes are 0.6602 for Harrell's and 0.6916 for Uno's c-index.

Once the model is trained, and we apply it to the data, the initial dataset gets divided at every node, and when they reach the leaves, each leaf represents a small subsets of the initial dataset. For each of those groups, the survival curve is calculated as seen in graph 4.11 using the Kaplan-Meier method.

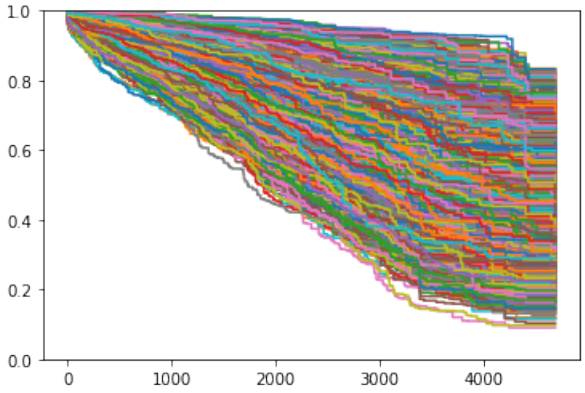

Figure 4.11: Random Survival Forest: final survival curves

4.2.4 Gradient Boosting Models

Two different boosting models were used with the respective functions `GradientBoostingSurvival` and `ComponentwiseGradientBoosting` from the Scikit-survival library. Both use the Cox's partial likelihood as the loss function, but one has stump regression trees as base learners and the other uses component-wise least squares base learners. In summary, the first is a non-linear model while the second is linear.

These were one of the most time and resource consuming models in this dissertation, because they require a heavy processing to avoid the overfitting to the data. To evaluate the best approach to avoid the overfitting, some regularization strategies were tested, as seen in image 4.12.

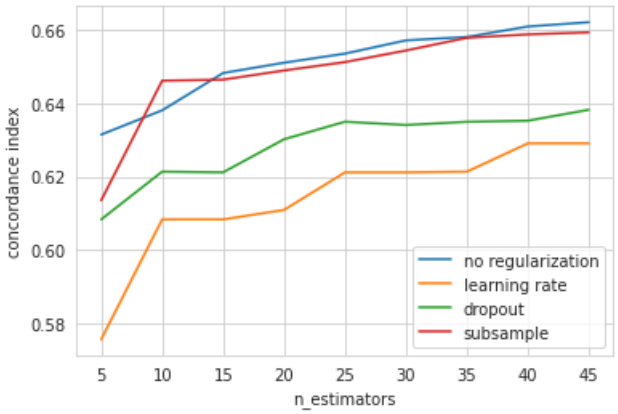

Figure 4.12: Gradient Boosting regularization strategies compared

The image shows that using dropout or setting a learning rate are the most effective strategies in avoiding overfitting, while the sub-sampling strategy shows a similar result to the no regularization one. Nonetheless, we are not limited to using only one type of regularization, actually they can be combined. The graph presented is just one example of the many evaluations performed with the models.

The final parameters chosen were, for the non-linear model, 50 stumps, while the linear model

used 90 base learners, both used the learning rate as 0.5. The corresponding c-indexes for the two methods are in table 4.4.

Table 4.4: The performance of the Gradient Boosting Models

	Harrell's c-index	Uno's c-index
Non-linear model	0.6624	0.6941
Linear model	0.6538	0.6804

4.2.5 Support Vector Machine (SVM)

The SVM model used corresponds to the function `FastSurvivalSVM` in Scikit-survival, which corresponds to the modeling of a linear Survival SVM.

To identify the best set of parameters, a grid search was used. This was the most time-consuming model, given that during the hyperparameter search phase it tested 1300 different fits.

The parameter α, as seen in equation 2.13, with the best performance was 0.0018585950321648853 and the model trained with that parameter had the c-indexes of 0.6602 and 0.6916, for Harrell's and Uno's implementations, respectively.

To further evaluate the result obtained, we can see how the test score varies for different α values in the graph 4.13. With the increase of the α value, the model's result remain stable, which means that the alpha is not an important parameter here and the model is relativelly robust.

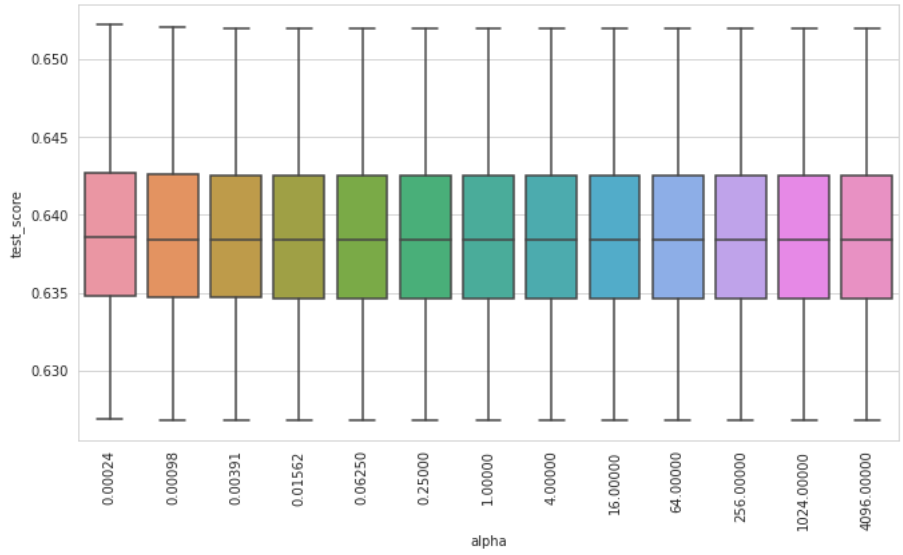

Figure 4.13: Survival SVM - c-index for different α values

4.3 Comparing the Models

In this section, the results obtained with all the models are compared. Also, the most predictive parameters are presented and discussed according to their meaning. Finally, the computational cost of training each model is compared.

The results obtained with the Random Survival Forest and the Non-linear Gradient Boosting models were slightly better than the others and also better than the previous works [18, 29]. However, when compared with the SVM result, it presents the biggest difference of around 0.02.

The two metrics analyzed have a big difference between them, being the Uno's c-index more optimistic than Harrell's. That difference is the result of the proportion of censored data in the dataset, to which Uno's metric is more sensitive [27].

Besides the performance metric, the most important features were also evaluated. Even though some models had different approaches that could provide the information on the most important feature, the function `PermutationImportance` provided by Scikit-learn was used. This approach was chosen to provide a standardized evaluation throughout the models with comparable values. The function output can be seen in table 4.5, the weight corresponds to the impact that feature has on the performance metric, with higher weights, the higher the impact.

Table 4.5: The five most important all models

	Harrell's c-index	Uno's c-index
Cox Penalized Model - Ridge	0.6569	0.6829
Cox Penalized Model - LASSO	0.6556	0.6812
Cox Penalized Model - Elastic Net	0.6567	0.6825
Random Survival Forest	0.6602	0.6916
GradientBoosting	0.6624	0.6941
ComponentwiseGradientBoosting	0.6538	0.6804
SVM	0.6382	0.6388

Besides the metrics, the five most important feature for all models seen in table 4.7 are provided. All the models have selected the type of donor (DON_TY_L) as the most important feature, and that is because it truly has an impact in the patient's life. For example, when comparing the functionality of the kidney after five years of transplant, those patients that have received a living donor kidney have a better renal clearance than the deceased donor recipient [13].

Another feature that was identified as important for most of the models is the age of the donor and the recipient's age. This is more intuitive, because we are evaluating the life expectancy of the organ, and when it comes to an older donor, chances are that the organ is more likely to fail sooner than a kidney from a younger donor.

The ethnicity features, on the other hand, are not so intuitive. However, there are studies

that relate the ethnicity with groups of HLA antigens, so if the recipient and the donor are ethnically compatible, they are more likely to have matching HLA antigens [24]. The feature ETHCAT_4 refers to the Hispanic ethnicity.

Now, the diabetes related variable was also selected by all the models. The feature DIAB_3 contains the information of which recipients have type II diabetes, which also has an impact on the normal functioning of the organ [19].

Lastly, the models can also be compared and evaluated based on their performance. In table 4.6 two different times are presented for each model, the parameter exploring and the training time. The first refers to the phase of testing the model with different parameters to identify the best configuration, and the second corresponds to the training of the same model but with a defined set of parameters. When comparing all the models, the worst performance is from SVM, followed by the linear Gradient boosting. The cox models are not represented in this table because their times are too low.

Besides their performance times, the models also have different requirements when it comes to memory. The comparative data is not displayed here, but the Random Survival Forest (RSF) was not capable of running the training phase with more than 20,000 entries due to memory consumption, while most of the other models could.

Table 4.6: Computational performance compared all models

Survival Models (20,000 entries)	Parameter exploring	Training Time
Random Survival Forest	23min	1 min
GradientBoosting	32 min	8 min
ComponentwiseGradientBoosting	88 min	15 min
SVM	156 min	5 min

Table 4.7: The five most important features for all Models

Cox Models			Random Survival	Gradient Boosting		Survival
Ridge	LASSO	Elastic Net	Forest	Non-linear	Linear	SVM
DON_TY_L	DON_TY_L	DON_TY_L	DON_TY_L	DON_TY_L	DON_TY_L	DON_TY_L
0.0351±0.0044	0.0267±0.0040	0.0361±0.0046	0.0361 ± 0.0046	0.0326±0.0058	0.0500±0.0073	0.0431±0.0046
DIAB_3	DIAB_3	DIAB_3	AGE_DON	AGE	DIAB_3	AGE_DON
0.0200±0.0067	0.0187±0.0067	0.0179±0.0066	0.0205 ± 0.0062	0.0191±0.0033	0.0335±0.0092	0.0178±0.0025
AGE_DON	AGE_DON	AGE_DON	AGE	DIAB_3	AGE_DON	DIAB_3
0.0162±0.0040	0.0136±0.0036	0.0165±0.0042	0.0188 ± 0.0065	0.0184±0.0061	0.0174±0.0037	0.0116±0.0018
ETHCAT_4	AGE	ETHCAT_4	DIAB_3	AGE_DON	ETHCAT_4	AGE
0.0087±0.0030	0.0073±0.0040	0.0076±0.0027	0.0166 ± 0.0047	0.0161±0.0030	0.0087±0.0024	0.0072±0.0018
DIAL_DAYS	ETHCAT_4	DIAL_DAYS	ETHCAT_2	ETHCAT_2	DIAL_DAYS	ETHCAT_4
0.0070±0.0013	0.0058±0.0018	0.0067±0.0010	0.0082 ± 0.0038	0.0090±0.0032	0.0079±0.0008	0.0054±0.0011

Chapter 5

Conclusion

This work was developed to explore survival analysis using kidney transplant data. To achieve that, a careful exploration and feature selection of the Scientific Registry of Transplant Recipients (SRTR) database was performed to, then, apply the survival models described in chapter 2. The models were used to understand the relationship between the features and the survival expectancy of the transplanted kidney, given the complexity of the Kidney Allocation System (KAS).

The methods used were first prepared to optimize the result obtained from them. After that, their performance was compared, to identify the best model and understand what information they provide on the allocation system and the data used. When comparing the result obtained through this work with previous works, it presents an improvement in the performance metric, probably due to the careful selection of the features and the avoidance of using records with missing values.

The feature selection was performed given the computational restrictions presented by the development environment used, so the goal was to maintain the most important features that are easy to interpret and are meaningful to the survival time prediction. Another motivation for the feature selection was to be able to replicate the study in different KAS around the world. With that in mind, the models used here could be trained using a different data source and the results compared, which could result in updates to KAS allocation policy.

Finally, a natural next step would be the application of the methods to the context of the Portuguese KAS providing more insights and validation to the policymakers. Besides the policies, a future work could include the development of an integrated data system for data collection and model improvement.

Bibliography

[1] Scientific Registry of Transplant Recipients. Data That Drives Development. https://www.srtr.org/about-the-data/the-srtr-database/. Accessed: 18-07-2021.

[2] Time-To-Event Data Analysis. https://www.publichealth.columbia.edu/research/population-health-methods/time-event-data-analysis. Accessed: 07-05-2021.

[3] Ethical Principles in the Allocation of Human Organs. https://optn.transplant.hrsa.gov/professionals/by-topic/ethical-considerations/ethical-principles-in-the-allocation-of-human-organs/. Accessed: 13-03-2021.

[4] Human Leukocyte Antigen (HLA) System. https://www.msdmanuals.com/professional/immunology-allergic-disorders/biology-of-the-immune-system/human-leukocyte-antigen-hla-system, . Accessed: 07-03-2021.

[5] Interpretation of HLA Typing Results for Entry into UNet - Implications for the HLA Matching Algorithm. https://optn.transplant.hrsa.gov/resources/guidance/interpretation-of-hla-typing-results-for-entry-into-unet/, . Accessed: 17-08-2021.

[6] Organ Procurement and Transplantation Network - About CPRA. https://optn.transplant.hrsa.gov/resources/allocation-calculators/about-cpra/#pros. Accessed: 17-07-2021.

[7] Scikit-survival API reference. https://scikit-survival.readthedocs.io/en/stable/api/index.html. Accessed: 10-02-2021.

[8] Michael Abecassis, Stephen T Bartlett, Allan J Collins, Connie L Davis, Francis L Delmonico, John J Friedewald, Rebecca Hays, Andrew Howard, Edward Jones, Alan B Leichtman, et al. Kidney transplantation as primary therapy for end-stage renal disease: a national kidney foundation/kidney disease outcomes quality initiative (nkf/kdoqi™) conference. *Clinical Journal of the American Society of Nephrology*, 3(2):471–480, 2008.

[9] Axel Benner, Manuela Zucknick, Thomas Hielscher, Carina Ittrich, and Ulrich Mansmann. High-dimensional cox models: the choice of penalty as part of the model building process. *Biometrical Journal*, 52(1):50–69, 2010.

[10] J Michael Cecka and Paul I Terasaki. The unos scientific renal transplant registry. *Clinical transplants*, pages 1–16, 1992.

[11] Yifei Chen, Zhenyu Jia, Dan Mercola, and Xiaohui Xie. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. *Computational and mathematical methods in medicine*, 2013, 2013.

[12] Krzysztof J Cios and G William Moore. Uniqueness of medical data mining. *Artificial intelligence in medicine*, 26(1-2):1–24, 2002.

[13] Ingrid B de Groot, J Iraida E Veen, Paul JM van der Boog, Sandra van Dijk, Anne M Stiggelbout, Perla J Marang-van de Mheen, and PARTNER study group. Difference in quality of life, fatigue and societal participation between living and deceased donor kidney transplant recipients. *Clinical transplantation*, 27(4):E415–E423, 2013.

[14] Covadonga Díez-Sanmartín and Antonio Sarasa Cabezuelo. Application of artificial intelligence techniques to predict survival in kidney transplantation: a review. *Journal of clinical medicine*, 9(2):572, 2020.

[15] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841 – 860, 2008. doi:10.1214/08-AOAS169.

[16] Elisa T Lee and John Wang. *Statistical methods for survival data analysis*, volume 476. John Wiley & Sons, 2003.

[17] Bruno Alves Lima and Helena Alves. Selection of donor-recipient pairs in renal transplantation: comparative simulation results. *Acta medica portuguesa*, 30(12):854–860, 2017.

[18] Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245*, 2017.

[19] S Michael Mauer, Michael W Steffes, and David M Brown. The kidney in diabetes. *The American journal of medicine*, 70(3):603–612, 1981.

[20] Justine B Nasejje, Henry Mwambi, Keertan Dheda, and Maia Lesosky. A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data. *BMC medical research methodology*, 17(1):1–17, 2017.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

[23] Sebastian Pölsterl, Nassir Navab, and Amin Katouzian. Fast training of support vector machines for survival analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 243–259. Springer, 2015.

[24] Umapathy Shankarkumar. The human leukocyte antigen (hla) system. *International Journal of Human Genetics*, 4(2):91–103, 2004.

[25] DE Stewart, AY Kucheryavaya, DK Klassen, NA Turgeon, RN Formica, and MI Aeder. Changes in deceased donor kidney transplantation one year after kas implementation. *American Journal of Transplantation*, 16(6):1834–1847, 2016.

[26] JR Storry and Martin L Olsson. The abo blood group system revisited: a review and update. *Immunohematology*, 25(2):48, 2009.

[27] Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

[28] Ping Wang, Yan Li, and Chandan K Reddy. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[29] Haonan Zhang. *Machine Learning Approaches for Prediction of Kidney Transplant Survival*. The University of Toledo, 2019.