



CZECH TECHNICAL UNIVERSITY IN PRAGUE  
Faculty of Nuclear Sciences and Physical Engineering



# **Predicting Kidney Transplant Survival: A Machine Learning Approach**

## **Predikce přežití po transplantaci ledvin pomocí technik strojového učení**

Bachelor's Degree Project

Author: **Peter Nutter**  
Supervisor: **Ing. Tomáš Kouřim**  
Consultant: **Ing. Pavel Strachota, Ph.D.**  
Academic year: 2022/2023

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nutter** Jméno: **Peter** Osobní číslo: **502467**  
Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**  
Zadávací katedra/ústav: **Katedra matematiky**  
Studijní program: **Matematické inženýrství**  
Specializace: **Matematická informatika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Predikce přežití po transplantaci ledvin pomocí technik strojového učení**

Název bakalářské práce anglicky:

**Predicting Kidney Transplant Survival: A Machine Learning Approach**

Pokyny pro vypracování:

1. Proveďte rešerši literatury týkající se metod strojového učení a analýzy přežití. Seznamte se s různými modely, algoritmy a metodami pro hodnocení jejich úspěšnosti.
2. Prozkoumejte problematiku transplantace ledvin. Analyzujte aktuální přístupy k hodnocení kompatibility dárců a příjemců, zejména v ČR, EU a USA, a identifikujte jejich případné nedostatky.
3. Seznamte se s dostupnými softwarovými nástroji pro vývoj modelů strojového učení, například TensorFlow, SciKit Learn, PyTorch.
4. Zpracujte a analyzujte reálná data z provedených transplantací ledvin. Porovnejte výsledky transplantací na základě různých faktorů, jako je geografická oblast, věk, pohlaví a etnicita.
5. Porovnejte různé modely pro predikci přežití štěpu u příjemců transplantovaných ledvin a vyhodnoťte jejich výkonnost na základě reálných dat.

Seznam doporučené literatury:

- [1] A. Géron, Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly, Sebastopol, CA, 2019.
- [2] I. H. Witten, E. Frank, M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2016.
- [3] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, 2016.
- [4] K. P. Murphy, Probabilistic Machine Learning. MIT Press, 2022.
- [5] W. Mariana, V. Ondřej, L. Robert, Transplantace orgánů v klinické praxi. Grada, 2021.
- [6] J. J. Kim, S. V. Fuggle, S. D. Marks, Does HLA matching matter in the modern era of renal transplantation? Pediatric Nephrology 36, 2021, 31-40.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Tomáš Kouřim Mild Blue, s.r.o.**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:


**Ing. Pavel Strachota, Ph.D. katedra matematiky FJFI**

Datum zadání bakalářské práce: **09.06.2023**

Termín odevzdání bakalářské práce: **02.08.2023**

Platnost zadání bakalářské práce: **30.09.2024**

  
Ing. Tomáš Kouřim  
podpis vedoucí(ho) práce

  
prof. Ing. Zuzana Masáková, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

  
doc. Ing. Václav Čuba, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

21.6.2023

Datum převzetí zadání



Podpis studenta

*Acknowledgment:*

I would like to thank my supervisor, Tomáš Kouřim, for his expert guidance and express gratitude to my thesis consultant, Pavel Strachota, for his exceptional editorial assistance.

*Author's declaration:*

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

Prague, August 2, 2023

Peter Nutter

*Název práce:*

**Predikce přežití po transplantaci ledvin pomocí technik strojového učení**

*Autor:* Peter Nutter

*Obor:* Matematické inženýrství

*Zaměření:* Matematická informatika

*Druh práce:* Bakalářská práce

*Vedoucí práce:* Ing. Tomáš Kouřim, Mild Blue, s.r.o.

*Konzultant:* Ing. Pavel Strachota, Ph.D., Katedra matematiky FJFI ČVUT v Praze

*Abstrakt:* Chronické onemocnění ledvin, které postihuje více než 10% světové populace, představuje vážný zdravotní problém v globálním měřítku. Transplantace ledvin je jednou ze stěžejních léčebných možností. Tato bakalářská práce se zaměřuje na využití strojového učení pro predikci délky přežití transplantované ledviny. Reálná data z databáze americké organizace, United Network for Organ Sharing (UNOS), a Institutu klinické a experimentální medicíny (IKEM) byla použita k vytvoření několika modelů, včetně Coxovy regrese, Random Survival Forests, neuronové sítě DeepSurv a dalších parametrických modelů. Námi vyvinuté modely nabízejí možnosti pro zdokonalení skórovacího systému aktuálně používaného v USA, nebo dokonce pro vytvoření a zavedení komplexnějšího systému skórování pro Českou republiku, který by zohlednil dříve nepoužité faktory. Nejpřesnější z testovaných modelů dosáhly výsledků srovnatelných s aktuální literaturou. Tato práce nejen potvrzuje potenciál strojového učení v oblasti transplantační medicíny, ale také otevírá možnosti pro zlepšení úspěšnosti léčby. Zároveň nastiňuje cestu pro budoucí výzkum zaměřený na optimalizaci těchto modelů a zlepšení jejich praktického využití.

*Klíčová slova:* Coxova regrese, Institut klinické a experimentální medicíny (IKEM), DeepSurv, Random Survival Forests, neuronové sítě, predikce mortality pacientů, predikce přežití štěpu, prediktivní modelování, strojové učení, analýza přežití, transplantace ledvin, United Network for Organ Sharing (UNOS)

*Title:*

**Predicting Kidney Transplant Survival: A Machine Learning Approach**

*Author:* Peter Nutter

*Abstract:* Chronic kidney disease, a global health issue, impacts over 10% of the world's population, making kidney transplantation a critical treatment option. This thesis delves into the application of machine learning for predicting the longevity of kidney grafts post-transplantation. Real data from the United Network for Organ Sharing (UNOS) and the Institute for Clinical and Experimental Medicine (IKEM) have been utilized to develop various models, such as Cox regression, Random Survival Forests, and the DeepSurv neural network, among others. The development of these models opens potential avenues for improving the existing scoring system used in the USA, or even establishing a more comprehensive scoring system for the Czech Republic that considers previously unused variables. Our top models have shown performance levels that are comparable with those currently used in the literature. This study not only reaffirms the potential of machine learning in transplantation medicine but also creates opportunities for improving patient outcomes. It additionally illuminates a path for future research, optimizing these models, and better understanding their practical implications.

*Keywords:* Cox proportional hazards model, DeepSurv, graft survival prediction, Institute for Clinical and Experimental Medicine (IKEM), kidney transplantation, machine learning, neural networks, patient mortality prediction, predictive modeling, Random Survival Forests, survival analysis, United Network for Organ Sharing (UNOS)

# Contents

<b>Introduction</b>	<b>8</b>
<b>1 Overview of Machine Learning</b>	<b>9</b>
1.1 Definition of Machine Learning	9
1.2 Types of Machine Learning Algorithms	9
1.3 Essential Concepts and Techniques in Machine Learning	10
1.3.1 Preparing the Data	10
1.3.2 Model Development	11
1.3.3 Model Evaluation	13
1.4 Popular Machine Learning Algorithms	13
1.4.1 Supervised Learning Algorithms	13
1.4.2 Unsupervised Learning Algorithms	15
1.4.3 Neural Networks	15
1.4.4 Challenges and Opportunities	16
<b>2 Kidney Transplantation: Current Practices and Challenges</b>	<b>17</b>
2.1 Overview of Kidney Transplantation	17
2.2 Factors Affecting Kidney Transplant Outcomes	17
2.2.1 Human Leukocyte Antigen (HLA) System	17
2.2.2 Other Medical Factors Affecting Transplant Outcomes	18
2.2.3 Waiting Time	18
2.2.4 Types of Kidney Transplants	18
2.3 Donor-Recipient Pairing Strategies Across Regions	19
2.3.1 United States	19
2.3.2 European Union and Related Organ Allocation Systems	20
2.3.3 Czech Republic	21
2.4 Current Limitations, Challenges, and Prospects for Improvement in Allocation Systems	22
2.4.1 Opportunities for Enhancement	22
2.4.2 Persistent Challenges	22
<b>3 Survival Analysis and Machine Learning in Kidney Transplantation</b>	<b>24</b>
3.1 Survival Analysis	24
3.1.1 Censoring	24
3.1.2 Terminology and Notation	25
3.1.3 Likelihood and Censoring	27
3.1.4 Non-Parametric Models	27
3.1.5 Parametric Models	28

3.1.6	Semi-parametric Models . . . . .	30
3.1.7	Machine Learning Techniques for Predictive Analysis . . . . .	32
3.1.8	Evaluation Metrics in Survival Analysis . . . . .	33
<b>4</b>	<b>Data Processing and Analysis</b>	<b>36</b>
4.1	Hardware and Software Configuration, Libraries, and Packages Used . . . . .	36
4.1.1	Python Libraries and Packages: . . . . .	36
4.2	Data Collection . . . . .	37
4.3	Inclusion Criteria and Data Cleaning . . . . .	38
4.4	Feature Selection and Engineering . . . . .	38
4.5	Data Partitioning . . . . .	40
4.6	Data Imputation . . . . .	40
4.7	Data Transformation . . . . .	40
<b>5</b>	<b>Descriptive and Comparative Analysis</b>	<b>42</b>
5.1	Descriptive Statistics of the Dataset . . . . .	42
5.1.1	Numerical Variables . . . . .	42
5.1.2	Categorical Variables . . . . .	44
5.2	Survival Analysis Using Kaplan-Meier and Nelson-Aalen Models . . . . .	45
5.2.1	Kaplan-Meier Survival Curve . . . . .	46
5.2.2	Nelson-Aalen Cumulative Hazard Curve . . . . .	47
5.2.3	Histogram of Transplant Year . . . . .	47
5.3	Comparative Analysis of Key Variables Across Groups . . . . .	47
5.3.1	Kaplan-Meier Survival Curves by Ethnicity . . . . .	47
5.3.2	Kaplan-Meier Survival Curves by Age . . . . .	47
5.3.3	Survival Analysis by Donor Type and Gender . . . . .	49
<b>6</b>	<b>Model Development and Evaluation</b>	<b>51</b>
6.1	Model Selection and Evaluation . . . . .	51
6.1.1	Parametric Models . . . . .	51
6.1.2	Semi-Parametric Models . . . . .	51
6.1.3	Other Machine Learning Models: . . . . .	52
6.2	Model Training . . . . .	53
6.3	Hyperparameter Tuning . . . . .	53
6.4	Model Evaluation . . . . .	53
<b>7</b>	<b>Results and Previous Research</b>	<b>54</b>
7.1	Performance Evaluation of the Machine Learning Models . . . . .	54
7.2	Feature Importance Analysis . . . . .	56
7.3	Previous Research on Kidney Transplant Survival Prediction . . . . .	57
7.4	Comparison to Previous Research . . . . .	58
<b>8</b>	<b>Conclusion</b>	<b>59</b>
8.1	Limitations of the Study and Future Work . . . . .	59
8.2	Practical Implications and Applications . . . . .	59
8.3	Summary of the Main Findings . . . . .	60
	<b>Bibliography</b>	<b>61</b>

# Introduction

Kidney transplantation stands as a life-saving intervention for individuals with end-stage kidney disease. The efficacy of this procedure, however, depends heavily on a complex matching process that uses numerous criteria to pair donors with recipients [33]. Machine learning algorithms have shown significant promise in enhancing the effectiveness of this process by predicting patient survival post-kidney transplantation. This thesis dives into this promising field, aiming to explore and compare various predictive models and machine learning techniques specific to kidney graft survival [37].

To accomplish this, a detailed review of existing literature on machine learning methods and predictive analysis techniques will be conducted. A broad spectrum of models, algorithms, and evaluation methods will be examined, setting the stage for a comprehensive comparison of their predictive capabilities. We will also delve into the existing kidney transplant allocation policies in the United States, the European Union, and the Czech Republic, identifying potential areas of improvement.

Our research will make use of two primary resources: the United Network for Organ Sharing (UNOS) database, which contains records of over a million kidney transplant patients in the USA since 1987, and a smaller dataset from IKEM, the Institute for Clinical and Experimental Medicine in Prague.

While developing the predictive models, we will evaluate machine learning tools and libraries, such as PyTorch, SciKit Learn, SciKit Survival, and PySurvival. The ultimate goal is to evaluate these models' capability in predicting kidney graft survival after transplantation using real-world data. Through this research, we aim to provide meaningful contributions that could improve the transplantation allocation system and enhance kidney transplant patients' survival rates.

## Contextual Background

This thesis explores the use of machine learning techniques to predict kidney transplant survival, at the intersection of medical informatics and transplantation medicine. The topic was chosen for its potential to advance the field of kidney transplantation and its relevance to Mild Blue, a company specializing in medical software development.

In collaboration with the Institute for Clinical and Experimental Medicine (IKEM), Mild Blue developed a software tool, TXM (Transplant Exchange Matcher), which optimizes paired kidney exchanges to increase the number of possible transplants. The optimization algorithm primarily considers Human Leukocyte Antigen (HLA) matching.

This thesis aims to expand the variables considered in kidney transplantation by exploring additional factors that could influence graft survival. We will develop and compare multiple machine learning models for predicting graft survival, potentially offering valuable insights for refining tools like TXM.

As a member of the Mild Blue team, I have a unique perspective on the challenges and opportunities in this field. This experience informs the methodology of this thesis, ensuring it is grounded in practical realities while advancing the field of computer science in the context of medical software development.



# Chapter 1

## Overview of Machine Learning

### 1.1 Definition of Machine Learning

Machine learning (ML) is a subfield of artificial intelligence that aims to develop algorithms and models capable of learning from data and making predictions. Its wide range of applications include healthcare, finance, marketing, and many other fields [4].

ML algorithms are designed to automatically identify patterns in data and use them to predict outcomes for new, previously unobserved data. There are three main types of ML algorithms: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning involves training algorithms on a labeled dataset where the desired outcomes are known. It is typically used for classification and regression tasks. Unsupervised learning, on the other hand, involves training algorithms on an unlabeled dataset where the desired outputs are unknown. It is commonly used for clustering and dimensionality reduction. Reinforcement learning involves algorithms that learn through feedback in the form of rewards or punishments. This type of learning is often used in the context of games and decision making.

There are many ML algorithms, each with its own strengths and weaknesses. The most commonly used algorithms include linear regression, logistic regression, decision trees, random forests, k-nearest neighbors, support vector machines, neural networks, and gradient boosting [4].

Choosing the right ML algorithm depends on the nature of the data and the specific problem to be solved. In this paper, we focus on the use of ML algorithms to predict graft survival after kidney transplantation.

### 1.2 Types of Machine Learning Algorithms

#### Supervised Learning

Supervised learning is the most widely used type of machine learning algorithm, and it is typically used for classification and regression tasks. In this type of learning, the algorithms are trained on a labeled dataset, where the desired outputs are known. The algorithm then uses the patterns it identifies in the data to make predictions about new, unseen data. This type of learning is suitable for problems where the desired outputs are well-defined and the relationship between the inputs and outputs can be modeled [4].

Classification is a type of supervised learning where the algorithm is trained to predict the class label of a given input. For example, given a dataset of patient information, a classification algorithm can be

trained to predict whether a patient will survive after a kidney transplant or not. Logistic regression and decision trees are two commonly used classification algorithms.

Regression is another type of supervised learning where the algorithm is trained to predict a continuous output. For instance, given a dataset of patient information, a regression algorithm can be trained to predict the number of days a patient is likely to survive after a kidney transplant. Linear regression and support vector regression are two common regression algorithms.

## **Unsupervised Learning**

Unsupervised learning is used for tasks such as clustering and dimensionality reduction. In this type of learning, the algorithms are trained on an unlabeled dataset, where the desired outputs are not known. The algorithm then identifies patterns in the data and groups similar data points together. This type of learning is used when the desired outputs are not well-defined or when the relationship between the inputs and outputs cannot be modeled [4].

Clustering is a common unsupervised learning technique that aims to group similar data points together. For instance, a clustering algorithm can be used on a dataset of patient information to group patients with similar medical histories. K-means and hierarchical clustering are two frequently used clustering algorithms.

Dimensionality reduction is another type of unsupervised learning where the algorithm aims to reduce the number of variables in a dataset while preserving the relevant information. This can be especially useful for datasets with a high number of variables, where identifying the most important variables can help simplify the data analysis process. For instance, a dimensionality reduction algorithm can be used on a dataset of patient information to identify the most important variables that contribute to patient survival after a kidney transplant. Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are two commonly used dimensionality reduction algorithms.

## **Reinforcement Learning**

Reinforcement learning is a category of machine learning that employs rewards or penalties as feedback to train the algorithm. This approach is geared towards maximizing the reward signal, as the algorithm learns to make decisions. Reinforcement learning has proven particularly useful in game-playing and decision-making tasks. Two commonly used algorithms for reinforcement learning are Q-learning and Monte Carlo Tree Search [4].

# **1.3 Essential Concepts and Techniques in Machine Learning**

In this section, we will discuss essential concepts and techniques that underpin the development of effective machine learning models. These concepts and techniques apply across various algorithms, ensuring that the models are robust, accurate, and well-suited for the problem at hand.

## **1.3.1 Preparing the Data**

### **Data Preprocessing**

Data preprocessing is a crucial step in the machine learning pipeline, as it prepares the dataset for training and evaluation. This process involves several techniques, such as data cleaning, which deals with inconsistencies, duplicate entries, and inaccuracies in the data. Handling missing values is another essential aspect of data preprocessing, and it can be addressed using various methods, including deletion,

imputation, or interpolation. Outlier detection and removal help identify and eliminate data points that significantly deviate from the norm and may adversely impact model performance. Data transformation techniques, such as normalization and standardization, are used to scale and adjust the distribution of the data to facilitate better learning by the algorithms.

## **Feature Engineering**

Feature engineering plays a vital role in improving the performance of machine learning models by selecting, creating, and transforming features. This process begins with feature extraction, which involves extracting relevant information from raw data to create new, informative features. Feature scaling ensures that all features have the same range and magnitude, preventing those with larger values from dominating the learning process. Dimensionality reduction techniques, such as Principal Component Analysis 1.4.2, help reduce the number of features while retaining important information, which can alleviate overfitting and reduce computational complexity. Feature selection methods, such as filter, wrapper, and embedded methods are employed to identify the most relevant and informative features for the problem at hand, which can enhance model performance and interpretability.

## **Handling Imbalanced Data**

Imbalanced datasets, where one class is significantly underrepresented, pose challenges for machine learning algorithms, as they tend to be biased towards the majority class. Techniques for handling imbalanced data include resampling methods (oversampling the minority class or undersampling the majority class), using cost-sensitive learning, employing ensemble methods like SMOTE (Synthetic Minority Over-sampling Technique), or using evaluation metrics that are less sensitive to class imbalance, such as precision-recall curves or F1 score.

### **1.3.2 Model Development**

#### **Model Selection**

Model selection is the process of choosing the most appropriate machine learning algorithm for a given problem. It involves considering factors such as dataset size, complexity, and expected model performance. Different algorithms have varying strengths and weaknesses, and the choice of algorithm can significantly impact the quality of predictions. To make an informed decision, it is essential to understand the problem domain, data distribution, and the underlying assumptions of the algorithms. Comparing the performance of multiple algorithms using cross-validation and performance metrics can help identify the most suitable model for the task.

#### **Bias-Variance Tradeoff and Overfitting/Underfitting**

Understanding the bias-variance tradeoff is crucial in developing effective machine learning models. Bias refers to the error introduced by approximating a real-world problem using a simplified model, while variance refers to the error arising from a model's sensitivity to small fluctuations in the training data. A model with high bias is prone to underfitting, meaning it does not capture the underlying structure of the data, while a model with high variance is prone to overfitting, meaning it learns noise in the training data, leading to poor generalization on unseen data. Striking the right balance between bias and variance is essential for optimal model performance. Techniques such as regularization, cross-validation, and ensemble learning can help manage the tradeoff and reduce the risk of overfitting or underfitting.

## Hyperparameter Tuning

Hyperparameter tuning is the process of optimizing the parameters of a machine learning algorithm to achieve the best possible performance. Hyperparameters are external factors that govern the learning process and are not learned during training. Examples of hyperparameters include the learning rate in gradient descent, the depth of a decision tree or the number of hidden layers in a neural network. Techniques for hyperparameter tuning include grid search, which exhaustively tries all possible combinations of hyperparameter values within a predefined range; random search, which randomly samples hyperparameter values from a specified distribution; and Bayesian optimization, which uses a probabilistic model to guide the search for optimal hyperparameters. By fine-tuning hyperparameters, a model's performance can be significantly improved, making it better suited for the problem at hand.

## Optimization and Training Techniques

Effective training and optimization techniques are essential in developing accurate and efficient machine learning models. Methods such as gradient descent, batch gradient descent, mini-batch gradient descent and stochastic gradient descent help minimize the loss function, and find the best parameters for the model. Advanced optimization algorithms, like Adam, RMSprop, and AdaGrad, adapt learning rates during training to enhance convergence speed and model performance [4].

## Regularization

Regularization is a technique used to prevent overfitting in machine learning models by adding a penalty term to the loss function. Regularization methods constrain the model's complexity, making it less prone to overfitting. Common regularization techniques include Lasso (Least Absolute Shrinkage and Selection Operator) and Ridge regression. Lasso regularization adds an L1 penalty term, which is the sum of the absolute values of the model coefficients, while Ridge regression adds an L2 penalty term, which is the sum of the squared values of the model coefficients. By incorporating regularization, researchers can develop models that are more robust and generalize better to new, unseen data.

## Kernel Methods and the Kernel Trick

Kernel methods are powerful techniques that allow linear algorithms to solve non-linear problems by mapping the input data into a higher-dimensional space where they become linearly separable. The kernel trick involves using a kernel function to compute the inner product of the transformed data points in this higher-dimensional space without explicitly performing the transformation. This approach reduces computational complexity and enables the use of algorithms like Support Vector Machines and Kernel Principal Component Analysis on non-linear data [4].

## Ensemble Methods

Ensemble methods are learning techniques that combine multiple models to improve the overall performance of machine learning algorithms. These techniques leverage the strengths of individual models to create a more accurate and robust final prediction. Bagging, or bootstrap aggregating, is an ensemble method that trains multiple models independently on random subsets of the dataset, with replacement, and combines their predictions through majority voting or averaging. Boosting is another ensemble technique that trains models sequentially, with each model focusing on the instances that were difficult for the previous model to predict correctly. Stacking, or stacked generalization, trains multiple models on the same data and then uses their predictions as input for a meta-model that makes the final prediction.

By employing ensemble methods, the performance of machine learning algorithms can be enhanced, resulting in better predictions and more accurate models.

### 1.3.3 Model Evaluation

#### Cross-Validation

Cross-validation is a crucial technique for assessing the performance of machine learning models and preventing overfitting. It involves dividing the dataset into multiple subsets, using one subset for validation and the rest for training, and repeating this process for each subset. This way, the model is tested on different portions of the data, providing a more reliable estimate of its performance. Techniques such as k-fold cross-validation, where the dataset is split into k equal-sized folds, and leave-one-out cross-validation, where a single observation is used as the validation set, are commonly used. Stratified k-fold cross-validation ensures that the proportion of classes is maintained across all folds, which is particularly useful when dealing with imbalanced datasets. Cross-validation helps to fine-tune model parameters and select the best-performing model for the task at hand.

#### Model Evaluation Metrics

Selecting appropriate evaluation metrics is vital for measuring the performance and efficacy of machine learning models. Each metric is designed for specific tasks, and understanding their characteristics is key to making sense of model performance. For classification tasks, typical metrics are accuracy, precision, recall, F1 score, and area under the ROC curve. In regression tasks, mean squared error, mean absolute error, and R-squared are commonly used. For clustering tasks, silhouette score, adjusted Rand index, and mutual information are frequently employed. By comprehending the properties of these metrics, data scientists can accurately assess and interpret the performance of machine learning models [28].

## 1.4 Popular Machine Learning Algorithms.

In addition to the three broad categories of machine learning, there are several commonly used algorithms and techniques that are applied in various applications. Some of the most popular algorithms include:

### 1.4.1 Supervised Learning Algorithms

#### Linear Regression

Linear regression is a supervised learning algorithm for predicting continuous numerical outcomes. It models the relationship between a dependent variable  $y$  and one or more independent variables  $X$  by fitting a linear equation  $y = X\beta + \epsilon$  to the observed data, where  $\beta$  represents the coefficients, and  $\epsilon$  represents the error term. Although designed for linear relationships, it can handle non-linear relationships with appropriate data transformations. It has a closed-form solution ( $\beta = (X^T X)^{-1} X^T y$ ) for efficiency, but iterative methods like gradient descent can also be used. Linear regression is simple, interpretable, and serves as a useful baseline for comparing other algorithms [4].

## Naive Bayes

Naive Bayes is a family of probabilistic machine learning algorithms that are based on Bayes' theorem, with the "naive" assumption that all features are independent of each other. Despite this simplifying assumption, Naive Bayes has been proven effective in many applications, particularly in text classification tasks such as spam detection and sentiment analysis. Naive Bayes classifiers are computationally efficient and easy to implement, making them a popular choice for tasks where the independence assumption is reasonable [28].

## K Nearest Neighbors

K Nearest Neighbors (KNN) is a simple and popular machine learning algorithm used for both regression and classification problems. In KNN, the algorithm is trained on a set of labeled data, and then, for a given test sample, it identifies the  $k$  nearest labeled data points and classifies the test sample based on the majority class of the  $k$  nearest data points.

KNN works by calculating the distance between the test sample and all the training samples, then selecting the  $k$  training samples with the shortest distance to the test sample. The majority class of these  $k$  nearest neighbors is then used to classify the test sample.

This algorithm is particularly useful for small and moderate-sized datasets, and it is easy to implement. However, KNN can be computationally expensive, as the time complexity increases with the number of samples and features in the dataset. Additionally, KNN may not perform well in high-dimensional feature spaces, as it can be sensitive to irrelevant features.

KNN is widely used in many applications, such as image classification, anomaly detection, and recommendation systems. In the medical field, KNN has been used for classification tasks, such as diagnosing diseases, and it has also been used in gene expression analysis [11, 31].

## Support Vector Machines

Support Vector Machines (SVM) is a powerful machine learning algorithm used for classification and regression tasks. It efficiently identifies complex relationships in high-dimensional spaces by utilizing kernel functions. The algorithm works by finding the optimal hyperplane that maximizes the margin between different classes, thus providing the best separation. SVM's versatility makes it suitable for various applications, including modeling kidney transplant outcomes based on features like patient demographics, donor characteristics, and clinical factors [29].

## Decision Trees

Decision Trees, renowned for their simplicity and interpretability, serve as the foundation for more advanced techniques. The construction of a tree involves a recursive partitioning of data. This partitioning is based on selecting the feature that maximizes information gain, a decision guided by metrics like Gini impurity or entropy. However, single decision trees tend to overfit when faced with complex datasets [4].

## Random Forests

Random Forest enhances decision tree performance by constructing multiple trees during training. Notable for its accuracy, generalization capability, and interpretability, this ensemble algorithm selects random subsets of features and samples at each node when constructing trees. This diversification reduces noise susceptibility. Predictions are made by aggregating individual tree results through majority

voting (classification) or averaging (regression), increasing model stability and reducing variance. Random Forests also provide feature importance measures, beneficial for identifying significant variables in various applications [4].

### **Gradient Boosting Machines**

Gradient Boosting Machines (GBM) is a powerful ensemble machine learning algorithm used for classification and regression tasks. It iteratively builds an ensemble of weak learners, typically decision trees, to minimize a loss function by focusing on improving the model's performance on the most challenging cases. GBM's ability to adapt to complex patterns makes it effective in handling large datasets and providing accurate predictions [29].

## **1.4.2 Unsupervised Learning Algorithms**

### **Principal Component Analysis**

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique that aims to transform the original dataset into a new coordinate system such that the first few principal components capture most of the variance in the data. By reducing the number of variables in a dataset, while preserving the relevant information, PCA can help to minimize the computational complexity and mitigate the "curse of dimensionality" in high-dimensional datasets.

### **1.4.3 Neural Networks**

Neural networks are a subset of machine learning algorithms that are inspired by the structure and function of the human brain. They are designed to learn from input data and make predictions based on that data.

In a neural network, multiple layers of interconnected nodes, or artificial neurons, process and analyze the input data. Each neuron receives input from other neurons and then processes the information and produces an output. The output of one layer serves as input for the next layer, until a final output is produced.

The structure and interconnections of a neural network can be adjusted to optimize the processing of input data. This is done through a process called training, where the network is exposed to a large amount of input data and the weights of the connections between neurons are adjusted in order to produce more accurate predictions.

Several different types of neural networks exist, each with a unique structure and purpose. Some common types include feedforward neural networks, recurrent neural networks, convolutional neural networks, and autoencoders.

Feedforward neural networks are the simplest type of neural network and are used for a wide range of tasks, including image recognition, natural language processing, and recommendation systems. Recurrent neural networks are designed to handle sequential data and are often used in speech recognition and language translation. Convolutional neural networks are used specifically for image recognition tasks and are optimized for processing data with spatial relationships. Autoencoders are neural networks that are used for dimensionality reduction and data compression.

Neural networks have proven to be highly effective for a wide range of machine learning tasks and continue to be a subject of active research. The ability to adjust a network's structure and interconnections allows for a high degree of customization and specialization, making neural networks a powerful tool in the machine learning toolbox.

### 1.4.3.1 Multi-Layer Perceptrons (MLPs)

Multi-Layer Perceptrons (MLPs) represent a specific subclass of feedforward neural networks. Composed of an input layer, one or more hidden layers, and an output layer, they provide an algorithmic architecture wherein each layer is fully interconnected to the next. This comprehensive interconnection ensures that each node in a given layer receives input from all nodes of the previous layer and forwards output to all nodes in the subsequent layer.

The input layer accepts raw data with each node corresponding to an individual feature in the data. The hidden layers, which house the majority of computational processing, apply weights to their inputs and pass the results through a non-linear activation function, such as ReLU, sigmoid, or tanh. This transformation allows the MLP to capture complex, non-linear patterns within the data [4].

The output layer, receiving data processed from the hidden layers, reformats it according to the nature of the problem – a softmax function for multi-class classification, a sigmoid for binary classification, or no activation function for regression tasks.

The training process, typically based on backpropagation with optimization techniques such as stochastic gradient descent (SGD), Adam, or RMSProp, involves iterative adjustment of the weights connecting nodes. The optimization aims to minimize a loss function that measures the difference between the network's predictions and the true values. This process enables the MLP to learn the underlying function that maps inputs to outputs.

MLPs are instrumental in learning and approximating a broad range of functions, allowing their application in various tasks, from computer vision to natural language processing. The inherent flexibility of MLPs to learn intricate patterns makes them a foundational tool in deep learning research and applications.

## 1.4.4 Challenges and Opportunities

Machine learning has the potential to significantly improve healthcare outcomes. It can enhance the work of healthcare professionals in various areas, such as diagnosis, prognosis, treatment selection, and planning interventions. By using data-driven techniques to analyze large and complex datasets, including electronic health records, medical images, genomic sequences, and data from wearable sensors, machine learning can help doctors make more accurate and timely decisions. It can also uncover new patterns and provide valuable insights. As a result, integrating machine learning into healthcare could improve the quality and efficiency of healthcare services, reduce costs, and minimize the risks associated with human errors [26, 24].

However, integrating machine learning into healthcare also comes with several challenges. These include technical issues related to data quality and availability, the interpretability and explainability of models, and concerns about privacy and security. Medical data can also contain inaccuracies, biases, or sensitive information, which requires careful handling. Ethical issues may also arise, such as the potential for unfair or discriminatory outcomes and questions about responsibility and accountability.

Successfully implementing machine learning in medicine requires a careful balance between the benefits and drawbacks of data-driven approaches. It also requires close collaboration between researchers, healthcare practitioners, and policymakers to ensure that machine learning is used in a safe, ethical, and effective way. By addressing these challenges, the medical community can fully leverage the potential of machine learning to transform healthcare and improve patient outcomes [42].



## Chapter 2

# Kidney Transplantation: Current Practices and Challenges

### 2.1 Overview of Kidney Transplantation

Kidney transplantation serves as the preferred treatment for end-stage renal disease (ESRD), providing superior quality of life and increased survival rates compared to dialysis. The procedure entails the surgical implantation of a healthy kidney from a living or deceased donor into a recipient with ESRD. Despite progress in transplantation techniques and immunosuppressive therapies, challenges persist in enhancing long-term graft survival and addressing the rising demand for donor kidneys [33].

### 2.2 Factors Affecting Kidney Transplant Outcomes

A variety of factors influence kidney transplant outcomes, including the human leukocyte antigen (HLA) system, blood type compatibility, recipient and donor demographics, comorbidities, and waiting times [47]. Machine learning algorithms can incorporate these factors to optimize donor-recipient matching and allocation strategies.

#### 2.2.1 Human Leukocyte Antigen (HLA) System

The HLA system, or Human Leukocyte Antigens, consists of a group of genes that encode proteins vital for the regulation of our immune response. These proteins are found on the surfaces of cells and help the immune system distinguish between self and foreign cells. The HLA system is divided into two primary classes: class I (HLA-A, HLA-B and HLA-C) and class II (HLA-DP, DQ and DR).

Historically, in kidney transplantation, the HLA antigens HLA-A, HLA-B, and HLA-DR have been considered the most relevant. However, recent research [1] suggests that the importance of these antigens may not be as straightforward as previously thought. This is a topic of ongoing investigation.

HLA proteins are incredibly diverse, contributing to more than 10% of our genetic diversity [17], making it the most gene-dense region of our genome. This diversity is crucial as it affects how well our immune system can recognize and respond to foreign substances.

Achieving a high degree of HLA matching between the donor and recipient is essential for the long-term success of kidney transplants. Better HLA matching decreases the likelihood of graft rejection, as the recipient's immune system is less likely to identify the transplanted organ as foreign.

## **2.2.2 Other Medical Factors Affecting Transplant Outcomes**

Besides HLA matching, numerous other medical factors contribute to kidney transplant success.

- **Blood type compatibility (ABO system):** Ensuring blood type compatibility between the donor and recipient is vital for a successful transplant. ABO-incompatible transplants have a higher risk of graft rejection and inferior outcomes.
- **Panel reactive antibodies (PRA) and donor-specific antibodies (DSA):** Preformed antibodies against donor HLA antigens in the recipient's bloodstream can result in antibody-mediated rejection and graft loss. PRA reflects the percentage of potential donors to whom a recipient has antibodies, while DSA are antibodies specific to the donor's HLA antigens.
- **Age, sex, and other demographic factors:** Donor and recipient demographics, such as age and sex, can impact transplant outcomes. Generally, younger donors and recipients exhibit better graft survival rates.
- **Comorbidities:** The existence of comorbid conditions in the donor or recipient, like diabetes or cardiovascular disease, can affect graft survival and patient outcomes.
- **Physical size:** The physical size of the donor and recipient also plays a significant role in transplant success. The size of the donor organ needs to be compatible with the recipient's body size to ensure proper fit and function.
- **Cold Ischemic Time:** This is the time from when the organ is removed from the donor's body and cooled to slow metabolism, until it is transplanted into the recipient. Shorter cold ischemic times are generally associated with better organ function post-transplant.
- **Creatinine Level:** This is a waste product in the blood created by the normal wear and tear on muscles of the body and kidneys filter creatinine from the blood. High levels of creatinine can indicate kidney damage or failure.
- **Primary diagnosis:** The primary diagnosis of kidney disease in the recipient, which led to the need for a transplant, can significantly influence the success of the transplant.
- **Functional health status at transplant:** The recipient's functional status at the time of the transplant can affect their post-transplant recovery and overall outcomes.

## **2.2.3 Waiting Time**

Waiting times for kidney transplantation can significantly impact transplant outcomes. These waiting times differ between deceased donor and living donor transplants, with living donor transplants generally having shorter waiting times. Longer waiting times can lead to a decline in the recipient's health and an increased risk of mortality before transplantation.

## **2.2.4 Types of Kidney Transplants**

### **Deceased donor transplants**

These transplants involve the use of kidneys from donors who have been declared brain dead or experienced cardiac death. The organs are procured following a strict consent and medical evaluation process.

## **Living donor transplants**

In this type of transplant, a healthy individual voluntarily donates one of their kidneys to the recipient. Living donor transplants often have better outcomes due to shorter waiting times and improved HLA matching.

## **Paired exchange programs**

Paired exchange programs: These programs facilitate kidney transplantation for incompatible donor-recipient pairs by identifying other pairs in a similar situation and swapping donors, allowing for compatible transplants.

## **2.3 Donor-Recipient Pairing Strategies Across Regions**

### **2.3.1 United States**

The Organ Procurement and Transplantation Network (OPTN) manages kidney allocation in the United States. The OPTN faces the challenge of ensuring equitable and efficient organ distribution, necessitating continuous adaptations to maximize organ utility. The key tool to achieve this is the Kidney Allocation System (KAS), which takes into account factors such as HLA typing, Kidney Donor Profile Index (KDPI), waiting time, Estimated Post Transplant Survival (EPTS) score, medically urgent status, and deceased donor classifications [36].

#### **2.3.1.1 Core Elements of the Kidney Allocation System**

In the United States, the Organ Procurement and Transplantation Network (OPTN) implemented the Kidney Allocation System (KAS) in December 2014. The KAS was developed to address issues such as high kidney discard rates, inequitable access to transplants for candidates with special health conditions, and a matching system that resulted in unrealized life years and high re-transplant rates. The key elements of KAS are:

1. **HLA Typing:** This identifies genetic markers that affect the immune response, aiming to maximize compatibility and minimize the risk of organ rejection.
2. **Waiting Time:** The duration a candidate has been on the waiting list is considered, with unique rules for adults and children.
3. **Estimated Post Transplant Survival (EPTS) Score:** A comparative measure, applicable to adult candidates, which estimates post-transplant survival time against national benchmarks.
4. **Medically Urgent Status:** Candidates meeting specific criteria are assigned medically urgent status and receive priority.
5. **Deceased Donor Classifications:** Kidneys from deceased donors are categorized based on their Kidney Donor Profile Index (KDPI) score, which takes into account various medical factors of the deceased donor.

Special circumstances are also taken into account under KAS. These include scenarios involving previous organ donors and highly sensitized individuals. As the allocation process unfolds, factors such as blood type compatibility, multi-organ combinations, and dual kidney exchanges become increasingly significant.

### **2.3.1.2 Kidney Donor Risk Index (KDRI) and Kidney Donor Profile Index (KDPI) Calculation**

The KDRI and KDPI [35] are numerical scales used to evaluate the quality of a donated kidney for transplantation, playing a key role in kidney allocation decisions.

#### **KDRI Calculation**

The KDRI is a sum of ten donor characteristics known to impact graft survival. These include factors such as age, height, weight, ethnicity, medical history, cause of death, serum creatinine, Hepatitis C Virus (HCV) status, and Donation after Cardiac Death (DCD) status. The total value represents a quantitative assessment of donor risk associated with the kidney.

#### **From KDRI to KDPI**

The Kidney Donor Profile Index (KDPI) is derived from the Kidney Donor Risk Index (KDRI) by normalizing the KDRI value relative to a reference population of donors from the previous year. For example, a KDPI of 90% means the donor's KDRI (which indicates the relative risk of graft failure) is greater than 90% of recovered kidneys.

#### **KDRI and Kidney Pairing**

KDRI and KDPI help match donors with appropriate recipients. For example, kidneys with a higher KDRI (indicating higher risk) might be allocated to older recipients or those with fewer alternatives, while kidneys with a lower KDRI (lower risk) might be allocated to younger recipients or those expected to live longer. The continued refinement of KDRI/KDPI calculations could include additional medical and demographic factors to enhance the accuracy of donor-recipient matches.

### **2.3.1.3 Calculating Estimated Post-Transplant Survival (EPTS)**

The Estimated Post-Transplant Survival (EPTS) score [27] is a crucial component in kidney allocation decisions. Ranging from 0% to 100%, it aims to predict the longevity of a kidney graft in a specific patient.

The calculation of the EPTS score involves four primary factors: the candidate's age, duration on dialysis, presence of diabetes, and previous solid organ transplants. The formula developed by the Scientific Registry of Transplant Recipients (SRTR) integrates these parameters to generate a raw EPTS, which is then converted to an EPTS score using an annually updated mapping table.

Implemented in 2014 with the new Kidney Allocation System, the EPTS score, in conjunction with the Kidney Donor Profile Index (KDPI), facilitates longevity matching to optimize the use of donated organs.

## **2.3.2 European Union and Related Organ Allocation Systems**

Within Europe, the process of organ allocation, specifically kidney transplantation, remains largely decentralized with separate systems employed within individual member states. One of the more noteworthy systems in operation is the Eurotransplant Kidney Allocation System (ETKAS), which oversees organ allocation across Austria, Belgium, Germany, Luxembourg, the Netherlands, and Slovenia, serving over 12,000 patients suffering from end-stage renal disease [25, 13].

The fundamental objective of ETKAS is to maximize the use of available donor organs while maintaining a transparent selection process for recipients. In order to accomplish this, it utilizes sophisticated algorithms that consider various aspects:

- **Medical Urgency and Transplantability:** Patients are categorized based on the severity of their condition and their suitability for a transplant. Those deemed as high urgency, meeting specific criteria like lack of dialysis access or severe neuropathy, are awarded an extra 500 points in the scoring system.
- **Special Programs:** ETKAS offers specific programs like the Acceptable Mismatch (AM) program for highly sensitized patients and the Eurotransplant Senior Program (ESP) to match elderly donors and recipients.
- **Blood Group Compatibility:** ETKAS strictly adheres to ABO blood group compatibility rules to minimize the risk of organ rejection.
- **Point-based Ranking System:** In cases of multiple suitable recipients, ETKAS uses a point-based system to rank them. The scoring considers factors such as urgency status, HLA match grade, mismatch probability, waiting time, and distance from the donor hospital. Additional points are awarded for high urgency, pediatric patients, patients receiving a kidney after another organ transplant, and patients with end-stage renal disease who previously donated a kidney. The point distribution varies among the participating countries to ensure fair organ allocation. HLA matches play a critical role in this system, with perfect matches receiving the maximum points and each additional mismatch leading to a decrease in points.
- **Pediatric and Preemptive Patients:** Points for HLA antigen mismatches are doubled for pediatric patients, who also receive a gradually phased-out bonus between the ages of 18 to 30. Preemptive patients, however, do not receive points for waiting time.
- **Allocation Algorithms:** Different allocation algorithms are applied depending on the donor's age. These prioritize AM program patients, zero HLA mismatch patients, and then rank other patients according to their point score.

In essence, ETKAS is a comprehensive organ allocation system that balances multiple considerations to maximize the use of available kidneys and ensure equitable organ distribution. It showcases significant progress in coordinating organ transplantation across multiple countries, even though it doesn't cover all of Europe.

### **2.3.3 Czech Republic**

The transplantation practice in the Czech Republic, though smaller in comparison to countries like the United States, has consistently achieved significant success in kidney transplantation. The first successful kidney transplantation in the country dates back to 1961, with the first living donor kidney transplantation carried out at the Institute for Clinical and Experimental Medicine (IKEM) in Prague in 1966 [43].

The Czech Transplantations Coordinating Center (Koordinační středisko transplantací, or KST) oversees the complex process of organ transplantation in the Czech Republic. With IKEM and Motol at the helm, the country has seen encouraging graft survival rates, reporting 92% to 95% and 81% to 84% at 1 and 5 years post-transplant, respectively.

The organ allocation system in the Czech Republic is regulated by the KST, underpinned by a meticulous process focusing on both medical and non-medical criteria. The key criteria for kidney allocation

are blood group, Panel Reactive Antibodies (PRA), Human Leukocyte Antigens (HLA) and time on the waiting list. The level of PRA reflects patient sensitivity and based on these levels, patients are divided into three categories: hypersensitized, moderately sensitized, and non-sensitized. HLA antigens play a pivotal role in transplantation with a compatibility index, based on 27 levels of compatibility, used to select suitable donors. The active waiting time is also a significant criterion for kidney allocation. In addition, a crossmatch test is conducted pre-transplant to ensure the recipient does not have antibodies against the donor. Non-medical factors, such as ensuring equitable contribution and utilization of organs among transplant centers, are also considered. This criterion ensures that transplant centers not only transplant organs but also contribute to the organ donation pool. Thus, patients from centers that have a higher donation-to-transplantation ratio may receive a certain level of priority in the allocation process [45].

What stands out about the Czech system is its active participation in international kidney exchange programs, evidenced by its partnerships initiated with Vienna, Austria, in 2012 and with Israel in 2019. In addition, the integration of a novel tool, TX Matching, developed by Mild Blue, has been a pivotal development in enhancing the effectiveness of paired kidney exchange programs. This tool uses an open-source algorithm that accommodates various parameters, providing flexibility in searching for suitable donors and improving transplant success rates [7, 44].

## **2.4 Current Limitations, Challenges, and Prospects for Improvement in Allocation Systems**

### **2.4.1 Opportunities for Enhancement**

- **Data Availability and Collection:** Leveraging advances in data collection can expose potential areas of improvement and enable the creation of robust machine learning models that aid complex decision-making processes.
- **International Collaborations:** Enhancing international partnerships for paired kidney exchange programs can expand the donor pool, potentially improving match outcomes.
- **Scoring System Enhancement:** Broadening the variables incorporated into scoring systems like KDPI and EPTS can improve their predictive capacity. In this context, the models developed in this thesis represent a significant advancement, as they propose a more comprehensive and inclusive approach. This involves considering a wider array of donor and recipient factors, potentially leading to a significant enhancement in medical decision-making and transplant success rates.

While these enhancements hold potential, it's critical to note that significant research and validation are required before implementation. Nevertheless, their exploration could lead to improved efficiency and efficacy in kidney transplantation, resulting in improved patient outcomes.

### **2.4.2 Persistent Challenges**

- **Graft Rejection and Immunosuppression:** Balancing the need to prevent graft rejection with immunosuppressive medications while avoiding over-immunosuppression remains a major challenge in kidney transplantation. Machine learning algorithms hold promise in personalizing immunosuppression regimens and tracking patient responses [39].
- **Disparities in Access and Outcomes:** Inequalities in access to kidney transplantation and the resulting outcomes persist among different racial, ethnic, and socioeconomic groups. Geographical

disparities also contribute to variations in organ availability and transplant center performance [15].

- **Organ Shortage and Allocation:** The demand for kidney transplants far exceeds the available donor organs. Strategies to address this issue could include public awareness campaigns, and innovations in organ preservation and transplantation technologies.

## Chapter 3

# Survival Analysis and Machine Learning in Kidney Transplantation

### 3.1 Survival Analysis

Survival analysis, a specialized branch of statistics, is indispensable for studying kidney transplant survival time. Unlike standard statistical approaches, survival analysis is uniquely equipped to handle time-to-event data, which is central to medical studies like ours where the event of interest is kidney graft failure.

This type of data often includes cases where the event has not yet occurred for some subjects at the end of the study period, a situation referred to as "censoring". Without survival analysis, these censored observations would need to be excluded or approximated, potentially leading to biased results. Survival analysis allows us to incorporate this censored data effectively, providing a more accurate and comprehensive analysis of survival times [9].

#### 3.1.1 Censoring

Censoring can be classified into right censoring, left censoring, and interval censoring. Right censoring occurs when a subject exits the study before the event happens, or the study concludes before the event takes place. There are two types of independent right censoring: Type I and Type II. Type I censoring is associated with a fixed study end time with no event occurrence. On the other hand, Type II censoring happens when the study concludes after a predetermined number of events among the subjects have occurred.

Left censoring, which is rare, happens when the event of interest has occurred before the study begins. Interval censoring arises when subjects enter and exit observation, and the event of interest occurs within a certain time interval, but the exact time is unknown.

Truncation, another concept in survival analysis, results from the study design and leads to the exclusion of certain subjects. Right truncation occurs when all subjects have already experienced the event of interest (e.g., a historical survey of patients on a cancer registry), while left truncation happens when subjects have been at risk before the study begins (e.g., life insurance policyholders where the study starts on a fixed date, and the event of interest is age at death).

For our research, we concentrate predominantly on right censoring, specifically Type I censoring. This form of censoring becomes particularly relevant in the context of our study on UNOS data on kidney transplants. Given the nature of the dataset, the majority of censoring is Type I and occurs due to the study cut-off in 2022, before the failure of many transplanted kidneys could be observed. This is common



in medical studies like ours, where the objective is to observe long-term outcomes. In these scenarios, the study often concludes before the event of interest (e.g., graft failure) happens for all subjects [9].

For our analysis, each observation is represented as a tuple containing the time to event and the censoring indicator.

$$D = (t_1, \delta_1), (t_2, \delta_2), \dots, (t_n, \delta_n) \quad (3.1)$$

where  $t_i$  represents the time to event for the  $i$ -th observation, and  $\delta_i$  is the censoring indicator for the  $i$ -th observation. The censoring indicator  $\delta_i$  takes the value 1 if the event of interest occurred for the  $i$ -th observation and 0 if the observation was censored.

### 3.1.2 Terminology and Notation

In this section, we will discuss the statistical foundation of survival analysis, which involves three essential functions that are related to the survival distribution. These functions are the survival function, probability density function, and hazard function, and they all have unique aspects that contribute to the analysis.

The survival function, denoted as  $S(t)$ , is the probability that an individual will survive beyond a given time  $t$ , where  $t \geq 0$ , and  $S(t) = P(T > t)$ , with  $T$  representing the survival time of a random variable. This function is derived from the cumulative distribution function of  $T$ , which is expressed as shown in equation:

$$S(t) = 1 - P(T \leq t) = 1 - F(t) \quad (3.2)$$

It is commonly used to identify the median, which is more informative than the mean when there are outliers present. The survival function is a non-increasing function where  $S(t) = 1$  when  $t = 0$  and  $S(t) = 0$  when  $t = \infty$ , indicating that the probability of survival decreases as time goes on [9]. An illustrative example of a survival curve is provided in Figure 3.1.

In survival analysis, the Probability Density Function (PDF)

$$f(t) = \frac{dF(t)}{dt} \quad (3.3)$$

describes the relative likelihood of observing a particular value of a random variable  $T$ , which represents the time to event of interest. The PDF is defined for  $t \geq 0$  and it is a non-negative function, meaning that  $f(t) \geq 0$  for  $t \geq 0$ . The probability of  $T$  falling in the interval  $(t, t + \Delta t)$  is approximately equal to  $f(t)\Delta t$ , for a small  $\Delta t$ . Since the time to event of interest cannot be negative, the PDF is zero for  $t < 0$ .

The hazard function  $h(t)$  is the instantaneous rate at which events occur at a given time, given that an individual has survived up to that time.

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{P(t \leq T < t + \Delta t)}{\Delta t \cdot S(t)} = \frac{f(t)}{S(t)} = -\frac{S'(t)}{S(t)} \quad (3.4)$$

The cumulative hazard function  $H(t)$  describes the accumulated risk up to time  $t$ .

$$H(t) = \int_0^t h(u)du \quad (3.5)$$

Given the relationships established between the survival function  $S(t)$ , cumulative hazard function  $H(t)$ , hazard function  $h(t)$ , and probability density function  $f(t)$ , we can derive additional connections among these functions. For instance, we can express the cumulative hazard function  $H(t)$  in terms of the survival function  $S(t)$  for continuous instances:

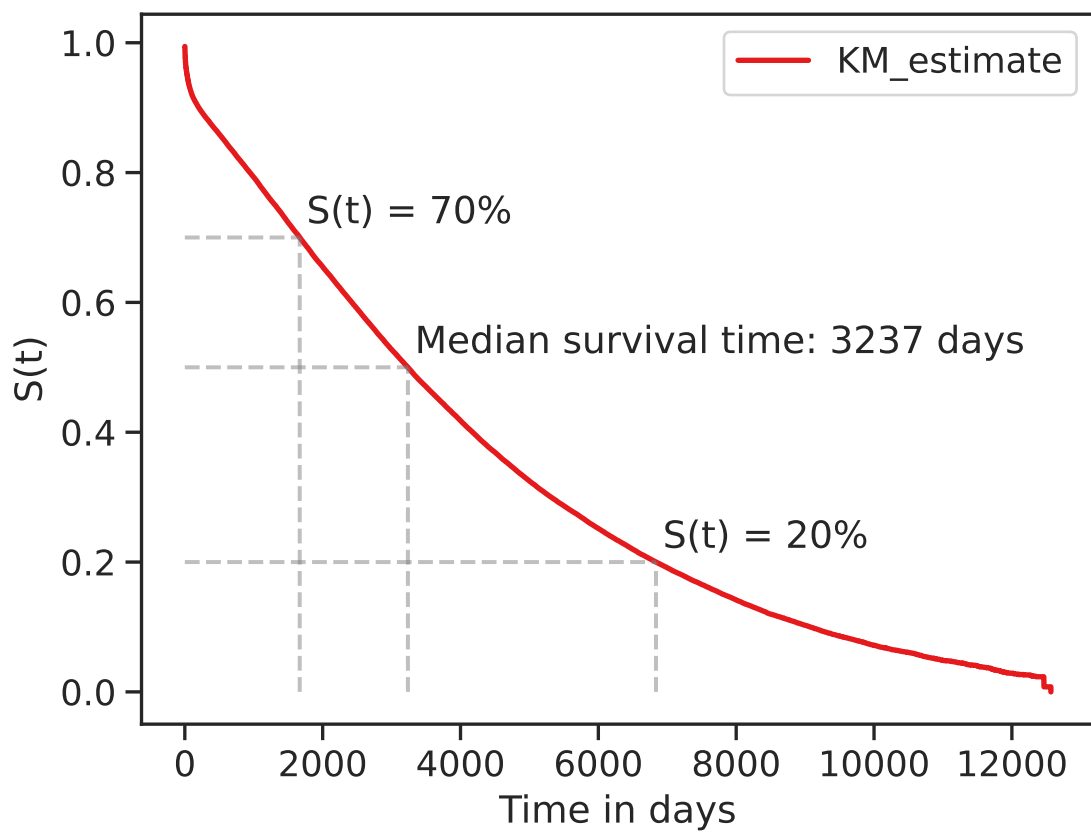


Figure 3.1: Survival Curve Graph Example

$$S(t) = \exp[-H(t)] \quad (3.6)$$

Estimating survival or cumulative hazard functions in survival analysis can be challenging; hence, a risk score  $\eta$  with an arbitrary scale is defined. This risk score is used for ordering individuals based on their likelihood of experiencing the event of interest, providing valuable insights and a practical alternative to the actual prediction of time for various applications.

### 3.1.3 Likelihood and Censoring

In this section, we delve into the construction of the likelihood function for parametric models. Consider independent observations represented as pairs  $(T_i, C_i)$ , where  $i$  ranges from 1 to  $n$ . Here,  $T$  represents the time to event or failure, and  $C$  denotes the censoring time, or the duration of observation.

We define  $X_i$  as the minimum of  $T_i$  and  $C_i$ , symbolizing the observed time, whether it's the failure or censoring time. We also introduce  $\delta_i$ , a binary variable indicating whether the failure time is observed ( $T_i \leq C_i$ ). Our dataset comprises independent pairs  $(X_1, \delta_1), \dots, (X_n, \delta_n)$ .

Assume that  $T_1, \dots, T_n$  are identically distributed with survival function  $S(x; \theta)$ , density  $f(x; \theta)$ , and hazard function  $h(x; \theta)$ , where  $\theta$  is a parameter vector. We also posit that  $T_i$  and  $C_i$  are stochastically independent.  $C_i$  possesses its own survival function  $G_i(x)$  and density  $g_i(x)$ , which can vary across observations.

The likelihood can be expressed using Lemma 2.1, as outlined in the lecture notes from Matfyz [19]:

$$L(\theta) = \prod_{i=1}^n [g_i(X_i)S(X_i; \theta)]^{(1-\delta_i)} [f(X_i; \theta)G_i(X_i-)]^{\delta_i}$$

If the censoring mechanism is independent (sometimes called non-informative) then we can ignore  $g_i$  and  $G_i$ :

$$L(\theta) = C \prod_{i=1}^n S(X_i; \theta)^{(1-\delta_i)} f(X_i; \theta)^{\delta_i}$$

This indicates that censored observations provide information through the survival function, while uncensored individuals contribute information through the density function. Here,  $C$  is a constant that does not depend on the parameter vector  $\theta$ . This constant, a result of disregarding the censoring mechanism, will not influence the estimation of  $\theta$  when maximizing the likelihood.

We can further rewrite this using  $h(t)$ :

$$L(\theta) = C \prod_{i=1}^n S(X_i; \theta)h(X_i; \theta)^{\delta_i}$$

We can maximize this likelihood using methods such as Newton or Fisher scoring [5]. With this model, we can compute all the parametric models mentioned later.

### 3.1.4 Non-Parametric Models

When no event times are censored, the non-parametric estimator of the survival function 3.2 is given by  $1 - \hat{F}(t)$ , where  $\hat{F}(t)$  is the empirical cumulative distribution function.

$$\hat{S}(t) = \frac{\# \text{ subjects with survival time } T > t}{\# \text{ subjects}} \quad (3.7)$$

$$\hat{F}(t) = \frac{\# \text{ subjects with survival time } T \leq t}{\# \text{ subjects}} \quad (3.8)$$

### 3.1.4.1 Nelson-Aalen

In the presence of right-censored data, the Nelson-Aalen [8] estimator can be employed to estimate the cumulative hazard rate function, which is denoted by  $H(t)$ . The estimator is defined as:

$$\hat{H}(t) = \sum_{t_j \leq t} \frac{d_j}{n_j},$$

where  $\hat{H}(t)$  represents the estimated cumulative hazard rate function at time  $t$ ,  $d_j$  is the number of events observed at time  $t_j$ , and  $n_j$  refers to the number of individuals at risk immediately before time  $t_j$ . Consequently, the Nelson-Aalen estimator is an increasing right-continuous step function with increments of  $\frac{d_j}{n_j}$  at the observed failure times.

The Nelson-Aalen estimator is applicable not only to right-censored data but also left-truncated data, which is common in epidemiological studies. In such cases, the number at risk  $n_j$ , is the count of individuals who have entered the study before time  $t_j$  and are still in the study just prior to  $t_j$ .

Using the Nelson-Aalen estimator, we can estimate the survival function indirectly by computing the exponent of the negative Nelson-Aalen estimator using the equation 3.6:

$$\hat{S}(t) = \exp[-\hat{H}(t)]$$

### 3.1.4.2 Kaplan-Meier

When some events are right censored or there are tied event times, we can use the Kaplan-Meier product-limit estimator [9].

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \quad (3.9)$$

where  $\hat{S}(t)$  is the estimated survival probability at time  $t$ ,  $d_i$  is the number of events at time  $t_i$ , and  $n_i$  is the number of individuals at risk (i.e. not yet experienced the event and not censored) of experiencing the event at time  $t_i$ . The product is taken over all time points  $t_i$  less than or equal to  $t$ .

## 3.1.5 Parametric Models

Parametric models make assumptions about the distribution of the survival times, which can lead to more efficient estimates when those assumptions hold true. Some of the most important parametric models for survival analysis include the Exponential, Weibull, Gompertz, Log-normal, and Log-logistic models.

### 3.1.5.1 Exponential Model

The Exponential model assumes that the hazard function is constant over time, making it a simple and useful model for situations with a constant failure rate. This model is an example of an Accelerated Life Model (ALM), a class of parametric models in survival analysis that assume the effect of covariates is multiplicative on the scale of the survival time. We will use the same notation and assumptions as

in Section 3.1.3, but now  $T_1, \dots, T_n$  are sampled from  $Exp(h_i)$ . In the Exponential Model, the hazard function  $h(t, Z_i)$  and the survival function  $S(t, Z_i)$  are defined as follows:

$$\lambda = \alpha \exp(\beta^T Z_i)$$

$$h(t, Z_i) = \lambda$$

$$S(t, Z_i) = e^{-ht}$$

Here,  $Z_i$  denotes a vector of covariates for each individual  $i$ , with each vector containing  $p$  different covariates. The term  $\lambda$  by itself represents the hazard rate, which is assumed to remain constant over time in this model.

The parameter  $\alpha$  acts as a scaling factor, modifying the baseline hazard rate. The expression  $\beta^T Z_i$  encapsulates the impact of the covariates  $Z_i$  on the hazard rate. In this model, the covariates are assumed to multiplicatively affect the hazard rate.

Our objective is to estimate the regression parameters, encapsulated in the vector  $\beta$ , and the scaling parameter  $\alpha$ . According to Section 3.1.3 the likelihood function will have the form [5, 19]:

$$L(\alpha, \beta) = C \prod_{i=1}^n [\alpha \exp(\beta^T Z_i)]^{\delta_i} \exp[-\alpha \exp(\beta^T Z_i) X_i]$$

$$\ln L(\alpha, \beta) = \sum_{i=1}^N [\delta_i \ln \alpha + \delta_i (\beta^T Z_i) - \alpha \exp(\beta^T Z_i) X_i] + C$$

where  $X_i$  is the observed time for the  $i$ -th individual.

### 3.1.5.2 Weibull Model

The Weibull model is a versatile and widely used parametric model in survival analysis that generalizes the Exponential model. It allows for hazard functions that can increase, decrease, or stay constant over time, making it suitable for a wide range of applications. The hazard function for the Weibull model is given by:

$$\lambda = \alpha \exp(\beta^T Z_i)$$

$$h(t, Z_i) = \gamma \lambda (\lambda t)^{\gamma-1}$$

$$S(t, Z_i) = \exp(-(\lambda t)^\gamma)$$

where  $\gamma > 0$  is the shape parameter and  $Z_i$  are random covariate vectors with dimension  $p$ . When  $\gamma = 1$ , the Weibull model reduces to the Exponential model, as the hazard function becomes constant over time. For  $\gamma > 1$ , the hazard function increases over time, while for  $\gamma < 1$ , the hazard function decreases over time.

### 3.1.5.3 Gompertz Model

The Gompertz Model is another parametric model with the base hazard being time varying. The shape of the hazard is characterized by the fact that the log is linear in  $t$ . It reflects very well adult mortality in the developed world.

$$\lambda = \alpha \exp(\beta^T Z_i)$$

$$h(t, Z_i) = \lambda \exp(\gamma t)$$

$$S(t, Z_i) = \exp\left(-\frac{\lambda}{\gamma}(e^{\gamma t} - 1)\right)$$

When  $\gamma = 0$  the hazard is constant and the model can be interpreted as an exponential model. If  $\gamma > 0$  the hazard increases over time and for  $\gamma < 0$  the reverse is true.

### 3.1.6 Semi-parametric Models

#### 3.1.6.1 Cox Proportional Hazards Model

The Cox Proportional Hazards Model, also known as the Cox Regression Model, is a popular method used in survival analysis. It's a semi-parametric technique, meaning it doesn't make specific assumptions about the shape of the hazard function, which describes the risk of an event happening at a certain time.

The model is used to understand how different factors (predictor variables) can affect the likelihood of an event happening. However, it doesn't define the entire hazard function itself, it only calculates the effect of these predictor variables on the hazard function.

One key assumption of the Cox model is that the ratio of the hazard functions for any two individuals (or groups) remains constant over time, regardless of the values of their predictor variables. This is known as the 'proportional hazards' assumption.

The ratio itself is called the 'hazard ratio', and it's used to compare the risk of the event happening between two individuals or groups, based on their predictor variables. So, if you're comparing two groups, the hazard ratio tells you how much more (or less) likely the event is to happen in one group compared to the other, taking into account their predictor variables. This proportional hazards assumption is the foundation of the Cox model.

In the Cox model [6], the hazard function is expressed as the product of a baseline hazard function, denoted as  $h_0(t)$ , and an exponential function derived from the linear combination of predictor variables,  $Z_i$ . This can be depicted as:

$$h(t, Z_i) = h_0(t) \times \exp(\beta^T Z_i) \tag{3.10}$$

In this equation,  $\beta$  represents the regression coefficients associated with each predictor variable. The baseline hazard function, represented as  $h_0(t)$ , signifies the risk for a subject when all covariate components are equal to zero. Since the baseline hazard function remains unspecified, it contributes to the Cox model's semi-parametric attribute.

Moving forward, the survival function in the Cox model is given by:

$$S(t, Z_i) = \exp\left(-\int_0^t h(t, Z_i)\right) = \left[\exp\left(-\int_0^t h_0(t)\right)\right]^{\exp(\beta^T Z_i)} = [S_o(t)]^{\exp(\beta^T Z_i)}$$

The survival function,  $S(t, Z_i)$ , is defined by this equation as the baseline survival function,  $S_o(t)$ , raised to the power of the exponential of the predictor variables' linear combination,  $\exp(\beta^T Z_i)$ . Here, the baseline survival function,  $S_o(t)$ , represents the survival probability when all predictor variables are zero.

Unique to the Cox proportional hazards model is its approach of determining regression coefficients by optimizing the partial likelihood function, as opposed to maximizing the traditional likelihood function. This function is devised by acknowledging the event sequence rather than the specific times of their occurrence. Each individual in the risk set contributes to this function at every event time, and each contribution is proportionate to the individual's estimated hazard rate. The process of optimizing the partial likelihood involves identifying the coefficient values that make the observed event order most probable, considering the covariate values. This method is efficient in using data and respecting the event order, and it sidesteps unnecessary assumptions about the baseline hazard function [19]. The detailed derivation of this partial likelihood is complex and beyond the scope of this explanation [21]:

$$L(\beta) = \prod_{i=1}^n \left( \frac{\exp(\beta^T Z_i)}{\sum_{j:t_j \geq t_i} \exp(\beta^T Z_j)} \right)^{\delta_i} \quad (3.11)$$

Here,  $Z_i$  is the vector of predictor values for the  $i$ -th individual, and  $\delta_i$  is the indicator variable that takes the value of 1 if the individual experiences the event and 0 if the individual is censored. The likelihood function above applies only when there are no tied survival times in the data. When tied times are present, the likelihood function must be adjusted using methods such as the Breslow or Efron method [5].

The Cox model does not estimate the baseline hazard function or the survival function directly. To obtain these functions, additional methods, such as Breslow's method for estimating the baseline hazard function [22], need to be applied.

It is important to note that the Cox model does not include an intercept term, as the baseline hazard function effectively serves this purpose. There are also extensions of the Cox model that accommodate time-varying covariates. In these cases, the covariate vectors  $Z_i$  change over time, allowing the model to capture the dynamic effects of predictor variables on the hazard rate.

### 3.1.6.2 Regularization in the Cox Model

Regularization in the Cox model is a method used to enhance the stability and interpretability of the model, especially when dealing with complex data structures. It's particularly useful when the predictors in the model are highly correlated, a condition known as multicollinearity. Regularization is applied to the partial likelihood function 3.11 and works by shrinking the estimated coefficients towards zero. This helps to mitigate the effects of multicollinearity, making the model more reliable. Common types of regularization techniques used in the Cox model include L1, L2, and Elastic Net regularization.

#### L1 Regularization (Lasso)

L1 regularization, also known as Lasso (Least Absolute Shrinkage and Selection Operator), adds an L1 penalty term to the partial likelihood function of the Cox model. The L1 penalty term is the sum of the absolute values of the coefficients multiplied by a tuning parameter ( $\lambda$ ). The L1-regularized Cox model is formulated as:

$$L1(\beta) = PL(\beta) - \lambda * \sum |\beta_j|$$

where  $PL(\beta)$  is the partial likelihood of the Cox model,  $\beta_j$  represents the  $j$ -th coefficient, and  $\lambda$  is the tuning parameter that controls the amount of shrinkage.

L1 regularization has the property of producing sparse coefficient estimates, effectively performing feature selection by setting some coefficients to zero.

### L2 Regularization (Ridge)

L2 regularization, also known as Ridge regularization, adds an L2 penalty term to the partial likelihood function of the Cox model. The L2 penalty term is the sum of the squared values of the coefficients multiplied by a tuning parameter ( $\lambda$ ). The L2-regularized Cox model is formulated as:

$$L2(\beta) = PL(\beta) - \lambda * \sum (\beta_j^2)$$

where  $PL(\beta)$  is the partial likelihood of the Cox model,  $\beta_j$  represents the  $j$ -th coefficient, and  $\lambda$  is the tuning parameter that controls the amount of shrinkage.

L2 regularization does not produce sparse estimates like L1 regularization but tends to shrink all coefficients towards zero uniformly.

### Elastic Net Regularization

Elastic Net regularization is a combination of L1 and L2 regularization techniques. It adds a linear combination of an L1 penalty term and an L2 penalty term to the partial likelihood function of the Cox model, controlled by a mixing parameter  $\alpha$ . The Elastic Net-regularized Cox model is formulated as:

$$EN(\beta) = PL(\beta) - \lambda \left[ \alpha * \sum |\beta_j| + (1 - \alpha) * \sum (\beta_j^2) \right]$$

where  $PL(\beta)$  is the partial likelihood of the Cox model,  $\beta_j$  represents the  $j$ -th coefficient,  $\lambda$  is the tuning parameter that controls the overall amount of shrinkage, and  $\alpha$  is the mixing parameter ( $0 \leq \alpha \leq 1$ ) that determines the balance between L1 and L2 regularization. When  $\alpha = 1$ , the Elastic Net is equivalent to L1 regularization (Lasso), and when  $\alpha = 0$ , it is equivalent to L2 regularization (Ridge).

Elastic Net regularization combines the advantages of both L1 and L2 regularization. It can produce sparse coefficient estimates while also handling multicollinearity among predictors. By adjusting the mixing parameter  $\alpha$ , the Elastic Net can be fine-tuned to strike the right balance between L1 and L2 regularization, depending on the specific problem and dataset at hand.

## 3.1.7 Machine Learning Techniques for Predictive Analysis

### 3.1.7.1 Random survival forest (RSF)

RSFs are an advanced version of Random Forests 1.4.1, specifically designed for survival analysis. They employ decision trees, which are built using a splitting criterion that optimizes the disparity in survival probabilities among subgroups. This section delves into several splitting criteria proposed and utilized for RSFs, and evaluates their efficacy in diverse scenarios.

#### Splitting Criteria in Random Survival Forests [40, 46]:

1. Log-rank Splitting Rule: Incorporated in the original RSF, this rule is grounded on the log-rank test statistic. It evaluates the survival disparities between two groups at each split, choosing the split with the most significant difference. While generally efficient, its performance may decline in the face of high censoring rates or noisy covariates.



2. **C-index Splitting:** This splitting criterion, derived from Harrell’s concordance index 3.1.8.1, has proven to surpass the log-rank statistic under certain conditions: when the signal-to-noise ratio is high, when the count of informative continuous covariates outweighs that of categorical covariates, and when the data exhibits a high censoring rate. However, due to its computational intensity, it is more appropriate for smaller datasets.
3. **AUC Splitting:** Drawing inspiration from the concordance index, the AUC-based 3.1.8.3 splitting criterion employs the area under the Receiver Operating Characteristic (ROC) curve to identify the optimal split. Generally, AUC splitting slightly outperforms log-rank splitting. However, for datasets with high censoring rates or a multitude of noise covariates, AUC splitting significantly outshines log-rank.

### 3.1.8 Evaluation Metrics in Survival Analysis

In survival analysis, the evaluation of model performance is crucial for assessing the quality of predictions and the usefulness of the models in practical applications. Two important aspects of model performance are calibration and discrimination [16].

Calibration refers to the agreement between the predicted probabilities of survival and the observed probabilities in the data. A well-calibrated model will produce accurate survival estimates, which means that the predicted survival probabilities closely match the observed survival rates at different time points. Calibration can be assessed using methods like the Brier score or by plotting calibration curves.

Discrimination, on the other hand, measures the ability of a model to distinguish between individuals with different survival times. A model with high discrimination power can accurately rank individuals according to their risk of experiencing the event of interest. Common metrics for assessing discrimination in survival analysis include the concordance index (C-index) and time-dependent area under the receiver operating characteristic (ROC) curve.

#### 3.1.8.1 Harrell’s C-index

Harrell’s C-index is a crucial tool for evaluating the effectiveness of a risk model in predicting the sequence of events of interest. For a given patient  $i$ , the risk model assigns a risk score, symbolized as  $\eta_i$ . An efficient model should indicate that patients with shorter time-to-event durations have higher risk scores. This implies that when comparing two patients, the one with a higher risk score should encounter the event of interest sooner.

The C-index can be computed by examining the risk scores and times-to-event for every pair of patients  $i$  and  $j$ . It calculates the proportion of risk scores correctly ordered among all comparable pairs of patients, with one patient experiencing the event of interest before the other [16].

$$c = \frac{\sum_{i \neq j} I(\eta_i < \eta_j) I(X_i > X_j) d_j}{\sum_{i \neq j} I(X_i > X_j) d_j}$$

In this equation, the numerator represents the number of concordant pairs, while the denominator is the total number of comparable patient pairs. The variable  $d_j$  is an indicator function that equals 1 if patient  $j$  experiences the event of interest and 0 otherwise. The C-index value ranges from 0 to 1, with higher values signifying better predictive accuracy. A value near 0.5 implies predictions are as random as a coin toss, while values near 1 suggest the models’ predictions are good. This method accommodates both censored and uncensored event data.

The statistical properties of the C-index are important to consider when evaluating its performance in survival analysis. The C-index is known to be sensitive to censoring, which may introduce bias into the

performance assessment of a risk model. To address this concern, an alternative version of the C-index, known as the IPC (inverse probability of censoring) weighted C-index [12] has been proposed. The IPC-weighted C-index is an unbiased estimator, as it takes censoring into account by applying weights based on the inverse probability of censoring at different times, leading to a more accurate assessment of the model's predictive ability.

For models that provide hazard function, survival function, or scores for different time points, a time-dependent version of the C-index can be used [3].

### 3.1.8.2 Brier Score

The Brier score is an additional metric employed to evaluate the calibration of survival models. It quantifies the average squared discrepancy between the predicted survival probabilities and the actual observed outcomes at a specific time point. Lower Brier scores are indicative of superior model calibration, as they represent smaller deviations between the predicted probabilities and the actual outcomes.

In many cases, researchers opt for the Integrated Brier Score (IBS), which provides an overall assessment of model performance over the entire time range rather than a specific time point.

To understand this in a mathematical context, the Brier score can be computed using the following formula:

$$BS^c(t^*) = \frac{1}{n} \sum_{i=1}^n \left[ I(t_i \leq t^* \wedge \delta_i = 1) \frac{(0 - \hat{S}(t^* | \mathbf{z}_i))^2}{\hat{G}(t_i)} + I(t_i > t^*) \frac{(1 - \hat{S}(t^* | \mathbf{z}_i))^2}{\hat{G}(t^*)} \right]$$

In this equation,  $n$  represents the total number of individuals in the dataset. The observed time for individual  $i$  is denoted by  $t_i$ , while  $\delta_i$  is the censoring indicator, which equals 1 if the event of interest has occurred and 0 otherwise. The term  $\hat{S}(t^* | \mathbf{z}_i)$  signifies the predicted survival probability for individual  $i$  at time  $t^*$ . Lastly,  $\hat{G}(t)$  is the estimated censoring survival function at time  $t$ , which provides the probability of an individual being observed (not censored) up to time  $t$  [16].

In simpler terms, the Brier score calculates the average of the squared differences between the actual outcomes and the predicted probabilities of survival for each individual in the dataset. The score is adjusted by the estimated censoring survival function to account for potential bias introduced by censored data. This adjustment ensures that the Brier score provides a fair evaluation of the model's predictive performance, even in the presence of censored observations.

### 3.1.8.3 Time-Dependent AUC

The time-dependent Area Under the Receiver Operating Characteristic (ROC) curve, or time-dependent AUC, is another metric used to assess the discrimination performance of survival models. It measures the ability of a model to rank individuals according to their risk of experiencing the event of interest at a specific time point  $t$ . The time-dependent AUC is particularly useful when the interest lies in evaluating the model's predictive accuracy at specific time points.

The time-dependent AUC for a given time  $t$  can be calculated as follows:

$$\widehat{AUC}(t) = \frac{\sum_{i=1}^n \sum_{j=1}^n I(y_j > t) I(y_i \leq t) \omega_i I(\eta_j \leq \eta_i)}{(\sum_{i=1}^n I(y_i > t)) (\sum_{i=1}^n I(y_i \leq t) \omega_i)}$$

where  $n$  is the number of individuals in the dataset,  $y_i$  and  $y_j$  are the observed times for individuals  $i$  and  $j$ , respectively,  $\omega_i$  represents the censoring weight [16] for individual  $i$ , and  $\eta_i$  and  $\eta_j$  are the predicted risk scores for individuals  $i$  and  $j$ , respectively.

The time-dependent AUC ranges from 0 to 1, with higher values indicating better discrimination performance. A value of 0.5 suggests that the model's predictions are no better than random chance, while values close to 1 indicate excellent predictive accuracy. By evaluating the AUC at different time points, the time-dependent AUC allows for a more comprehensive assessment of the model's discrimination ability over time.

## Chapter 4

# Data Processing and Analysis

This chapter delves into the methodological approach undertaken for the processing and analysis of the dataset used in this study. It outlines the computational tools and Python libraries employed, the process of data acquisition, and the subsequent steps of data cleaning, feature selection, partitioning, imputation, and transformation. These rigorous procedures ensure the robustness and reliability of the survival analysis models developed, thereby enhancing the validity of the study's findings.

### 4.1 Hardware and Software Configuration, Libraries, and Packages Used

In this section, we discuss the hardware and software configuration, as well as the libraries and packages employed for survival analysis in this thesis. The models were trained on the school cluster Helios, which provided the necessary computational resources and parallel processing capabilities to efficiently handle large-scale data and complex algorithms.

#### 4.1.1 Python Libraries and Packages:

A variety of Python libraries and packages were used for training, and evaluating the survival analysis models. The key libraries used are:

- Scikit-survival [34] is a machine learning library specifically designed for survival analysis and time-to-event data. It extends scikit-learn, a popular machine learning library in Python, by adding support for survival analysis techniques, such as Cox regression, random survival forests, and gradient-boosted survival trees.
- Lifelines [10] is a Python library for survival analysis that focuses on providing an easy-to-use interface and robust statistical methods for analyzing time-to-event data. It supports Kaplan-Meier and Nelson-Aalen estimators, Cox proportional hazards regression, and parametric survival models. This library was utilized primarily for nonparametric models, such as the Kaplan-Meier estimators.
- PySurvival [14] is a Python package optimized with PyTorch that offers various survival analysis models, along with tools for model evaluation, feature selection, and data preprocessing. It includes implementations of popular models such as the Cox proportional hazards model, accelerated failure time model, and random survival forests.

Although not utilized in this thesis, several R and other Python packages are available for survival analysis and may be considered for further research or comparison purposes. Some of these packages include:

- PyCox [20] is a Python library designed for deep learning-based survival analysis using PyTorch. Despite attempts, it was difficult to get it working effectively due to its less intuitive documentation and API implementation. It allows for the integration of deep learning models, such as neural networks, with traditional survival analysis techniques like Cox proportional hazards regression.
- 'survival': One of the most widely-used R packages for survival analysis, it provides a comprehensive set of statistical tools for time-to-event data, including Kaplan-Meier estimators, Cox proportional hazards regression, and parametric survival models. The Cox model from this package was tested and yielded promising results. However, it was ultimately not included as the chosen implementation of Cox was sourced from the Python libraries.
- 'randomForestSRC': This package implements random survival forests, a powerful ensemble learning method for survival analysis that builds on the principles of decision trees and bagging. This implementation of Survival forests was also tested and showed encouraging outcomes, but its lengthy training times rendered it not feasible for this study.
- Statsmodels is a Python library that offers a comprehensive suite of statistical models, tests, and data exploration tools. It provides classes and functions for estimating a wide range of statistical models, and its results are tested against existing statistical packages to ensure accuracy.
- 'glmnet': This package provides tools for fitting generalized linear models with Lasso or Elastic Net regularization, which can be applied to survival analysis through the use of Cox proportional hazards models.

These libraries and packages, along with the computational resources provided by the Helios cluster, enabled the development and evaluation of various machine learning models for survival analysis, as presented in this thesis.

## 4.2 Data Collection

The process of data acquisition for this study was multifaceted, involving several attempts to secure a comprehensive and reliable dataset. Initially, the dataset from the Institute for Clinical and Experimental Medicine (IKEM) was considered. However, it was found to be inadequate for the purposes of this study due to its limited size, substantial missing values, lack of data on other patient indicators, and outdated patient visit information.

Efforts were then directed towards securing data from organizations known for their extensive transplant data repositories, including the Scientific Registry of Transplant Recipients (SRTR), the Australia and New Zealand Dialysis and Transplant Registry (ANZDATA) and the National Health Service (NHS) in the UK. However, these attempts were met with challenges such as prohibitive data acquisition costs, the necessity for collaboration with local research groups and specific citizenship or student status requirements. Similar challenges were encountered with organizations in Canada, France, Germany, and Spain.

Ultimately, the United Network for Organ Sharing (UNOS) agreed to provide their dataset free of charge, following the signing of a data sharing agreement. This dataset is comprehensive, containing information about patients across the United States, and offers a wide variance in data, including geographical location, race, age, and other factors.

Although the IKEM dataset will not be used for model training due to its limitations, it will be utilized for statistical analysis to provide additional context and insights.

### 4.3 Inclusion Criteria and Data Cleaning

The original dataset for this study spanned a vast array of transplant cases, encompassing 1,108,884 entries across 468 variables. To enhance the model’s specificity and relevance, the dataset was meticulously refined using several criteria.

The study included only those patients who had received a single kidney transplant, to eliminate possible confounding effects associated with multiple or various organ transplants. Moreover, only adult patients, defined as individuals aged 18 years and above, were considered, owing to the unique physiological and medical aspects associated with pediatric patients.

The model was trained utilizing both living and deceased donor data. While training separate models for each donor type was tested, it led to inferior performance. Transplant cases involving foreign donors, defined as organs sourced from donors outside the United States, were excluded due to their limited representation in the dataset and potential for outcome variability.

Entries with negative event times, presumably errors as a kidney cannot fail pre-transplantation, were omitted. Other instances of likely errors, such as negative values in features that should inherently be positive, were also excluded. A temporal cut-off was set, incorporating only transplants executed after the year 2000 to reflect the significant advancements in transplant procedures and corresponding medical treatments over the years.

Post-refinement using these filters, the dataset was narrowed down to include 326,440 entries.

### 4.4 Feature Selection and Engineering

The initial selection of variables for this study was broad, encompassing a wide range of potential predictors identified from existing research in the field of kidney transplantation. However, to create a more efficient and interpretable model, it was necessary to reduce the number of variables. This was achieved through a process of feature selection, which involved evaluating the contribution of each variable to the model’s performance using permutation importance from scikit-learn [2] applied to Cox and Random Survival Forests.

Variables that did not contribute to an improvement in model performance were excluded. Variables with a high proportion of missing values were also more likely to be excluded, especially if their inclusion did not result in a significant performance gain. Through this process, the number of variables was reduced to a more manageable set of 28 covariates which can be seen in Table 4.1.

It’s worth noting that we also experimented with Principal Component Analysis (PCA) as an alternative method for dimensionality reduction. However, the models using PCA components as predictors resulted in worse performance compared to the models with the selected covariates, leading us to reject PCA for this particular application.

The principal objective of this research is to forecast kidney graft failure. In order to accomplish this, we utilized two critical indicators, specifically, the graft status (GSTATUS\_KI), and the time until graft failure or the last follow-up (GTIME\_KI). These indicators are elaborated on in Table 4.2. The graft status acts as a binary measure, with '1' representing graft failure and '0' symbolizing ongoing functionality (i.e., censored).

Within our dataset, we engineered two crucial features. One reflects the duration a patient underwent dialysis before transplantation (DIAL\_LEN), and the other pertains to the primary diagnoses of the kidney recipient (DIAG\_KI). The dialysis duration feature was derived from two datetime variables the initiation of dialysis and the date of transplantation. Despite not providing an exact count of dialysis days due to inherent constraints, permutation importance rankings reaffirmed its significance in our model, thereby proving its value as a worthwhile approximation.

The primary diagnosis feature in our dataset initially consisted of a complex structure with 75 categories. To make this more manageable and effective, we applied a method inspired by the Cox proportional hazards model as referenced in the study [23]. This method helped us restructure the categories based on the severity of the conditions, simplifying them to just eight. This reengineering operation, while challenging, led to a significantly improved feature, revealing a category with strong predictive potential that substantially contributes to our study’s outcomes.

We should note that our dataset also includes variables pertinent to patient survival outcomes, namely patient status (PSTATUS) and patient survival time (PTIME). The patient status is a Boolean variable, indicating the most recent status of the patient. In contrast, patient survival time represents the duration of survival post-transplant in days.

Even though the task of predicting patient survival may be comparatively simpler and the models might yield more accurate predictions for this task, it does not constitute the primary focus of our study, which is predicting kidney graft failure. For the purpose of comparison and a comprehensive understanding of the models’ capabilities, we have examined their performance on this alternate prediction task.

Table 4.1: Summary of variables used in the analysis

Variable Name	Description	Variable Type
AGE	Recipient’s age at transplant	Numeric
AGE_DON	Donor’s age at donation	Numeric
BMI_CALC	Recipient’s Body Mass Index	Numeric
BMI_DON_CALC	Donor’s Body Mass Index	Numeric
DAYSWAIT_CHRON_KI	Days recipient waited for transplant	Numeric
COLD_ISCH_KI	Kidney cold ischemic time	Numeric
KDRI_RAO	Kidney Donor Risk Index	Numeric
CREAT_TRR	Creatinine level of the recipient at the time of transplant	Numeric
DIAL_LEN	Number of days the patient was on dialysis	Numeric
ETHCAT	Recipient’s ethnic category	Categorical
ETHCAT_DON	Donor’s ethnic category	Categorical
GENDER	Recipient’s gender	Categorical
GENDER_DON	Donor’s gender	Categorical
ABO_MAT	Blood type match between recipient and donor	Categorical
AMIS	A locus mismatch score	Categorical
BMIS	B locus mismatch score	Categorical
DRMIS	DR locus mismatch score	Categorical
DIAB	Recipient’s diabetes status	Categorical
DON_TY	Type of donor	Categorical
DIAL_TRR	Recipient’s dialysis status at transplant	Categorical
ON_DIALYSIS	Was the recipient on dialysis at somepoint before transplant	Categorical
REGION	Region of the transplant center	Categorical
HCV_SEROSTATUS	Recipient’s Hepatitis C status at the time of transplant	Categorical
DEATH_MECH_DON	Deceased Donor’s Cause of Death	Categorical
DIAG_KI	Primary Diagnosis of the Kidney Recipient	Categorical
FUNC_STAT_TRR	Recipient’s Functional Health Status at the Time of Transplant	Categorical
COD_CAD_DON	Mechanism of Death for the Deceased Donor	Categorical
HIST_HYPERTENS_DON	Donor history of hypertension	Categorical

Table 4.2: Predictor and Outcome Variables

Variable Name	Description	Variable Type
GSTATUS_KI	Graft status of the kidney transplant	Boolean
GTIME_KI	Time to graft failure or last follow-up	Numeric

## 4.5 Data Partitioning

The dataset was split into training (60%), validation (20%), and testing (20%) sets. The training set was used to fit the models, while the validation set assisted in hyperparameter tuning and model selection. The testing set, held out until the final model was selected, provided an unbiased performance estimate. A consistent random seed ensured study replicability.

## 4.6 Data Imputation

Addressing the issue of missing values within predictor variables necessitated the use of an imputation technique, facilitated by the `SimpleImputer` function from the Python `scikit-learn` library. This function is specifically designed for handling missing data, applying distinct strategies for numerical and categorical variables.

For numerical variables, the `SimpleImputer` function adopted a "median" strategy. In this method, the median value was calculated from the available data points of each respective variable, and this median value was then used to replace any missing entries. On the other hand, for categorical variables, a "most\_frequent" strategy was employed. Here, the `SimpleImputer` function identified the category that occurred most frequently within each variable, and used this value to substitute any missing data points.

Such robust handling of missing data ensures the effective operation of machine learning algorithms, which typically require complete datasets to perform optimally.

## 4.7 Data Transformation

Following the imputation phase, the dataset's variables were distinguished into two categories: categorical and numerical. Specific preprocessing strategies were implemented for each category to optimally prepare them for inclusion in the predictive model.

In dealing with categorical variables, we utilized the `OneHotEncoder` function. This function restructures each categorical variable by converting it into a binary vector representation. More specifically, it takes a categorical variable with 'n' distinct categories and converts it into 'n' separate binary features. Each of these binary features corresponds to a unique category of the original variable. For any given record, only one of these binary features will have a value of 1 (indicating that the record belongs to that category), while the rest will have a value of 0 (indicating that the record does not belong to those categories). This transformation effectively transforms each unique category of a variable into a new, standalone feature within the dataset.

An important parameter we used in this process is the 'drop' parameter with 'if\_binary'. This parameter is used to avoid the redundancy that can occur when one-hot encoding binary variables. In a binary variable, there are only two categories, and therefore, when one-hot encoded, it would create two identical but opposite binary features. This could lead to multicollinearity, which can negatively impact some machine learning models.



For numerical variables, we employed the `StandardScaler` function. This function standardizes each variable by deducting the mean value and then scaling it to unit variance. Unit variance indicates that the spread or distribution of values has a standard deviation of 1, and subtracting the mean ensures that the resulting distribution is centered around 0. The application of the `StandardScaler` function effectively ensures that all numerical variables are converted to a common scale. This standardization process is indispensable for mitigating the impact of any single variable having an unduly large influence on the model simply by virtue of its numerical scale.

We also experimented with Min-Max Scaling for the numerical variables, another popular normalization technique that transforms the data to fit within a specified range, typically between 0 and 1. However, in our testing, we found that the use of Min-Max Scaling did not yield significantly better or worse results compared to the `StandardScaler`.

The decision to retain `StandardScaler` was also supported by its wider applicability and versatility. It can handle outliers more effectively than Min-Max Scaling and does not distort the distances between the values for each feature. Furthermore, its use is endorsed by the Scikit-Learn library's documentation for survival analysis, further reinforcing its appropriateness in this context.

## Chapter 5

# Descriptive and Comparative Analysis

In this chapter, we conduct a comprehensive statistical analysis of our dataset, a crucial step in our study. This analysis serves a dual purpose: it not only offers a detailed understanding of the variables that are integral to our research but also uncovers patterns and relationships within the data. These insights are invaluable as they inform and shape our machine learning models, enhancing their ability to predict survival following kidney transplants. For the non-parametric models utilized in this section, we opted for the Lifelines 4.1.1 library, primarily due to its superior support for confidence intervals.

### 5.1 Descriptive Statistics of the Dataset

The descriptive statistics of our dataset provide a detailed understanding of the numerical and categorical variables that are crucial to the study.

#### 5.1.1 Numerical Variables

Table 5.1.1 provides a comprehensive overview of the numerical variables in our dataset. The recipient's age at transplant 'AGE' ranges from 18 to 96 years, with a mean age of 52 years, indicating that kidney transplantation is not limited to a specific age group and is common among middle-aged individuals.

The waiting time for a kidney transplant 'DAYSWAIT\_CHRON\_KI' varies significantly, with some recipients receiving a transplant immediately, while others wait for up to 8554 days. This high standard deviation underscores the unpredictable nature of organ availability and the urgent need for more organ donors.

#### Correlation Matrix

As part of the descriptive analysis, a correlation matrix was computed to examine the relationships between the numerical variables in the dataset. The correlation matrix is presented in Figure 5.1 The matrix provides a visual representation of the correlation coefficients between each pair of variables. A positive correlation indicates that as one variable increases, the other also increases, while a negative correlation indicates that as one variable increases, the other decreases. The highest correlation is between KDRI and AGE\_DON because age is one of the input variables from which KDRI is calculated from. This high correlation could potentially cause multicollinearity in our model, which is something we need to consider in our analysis.

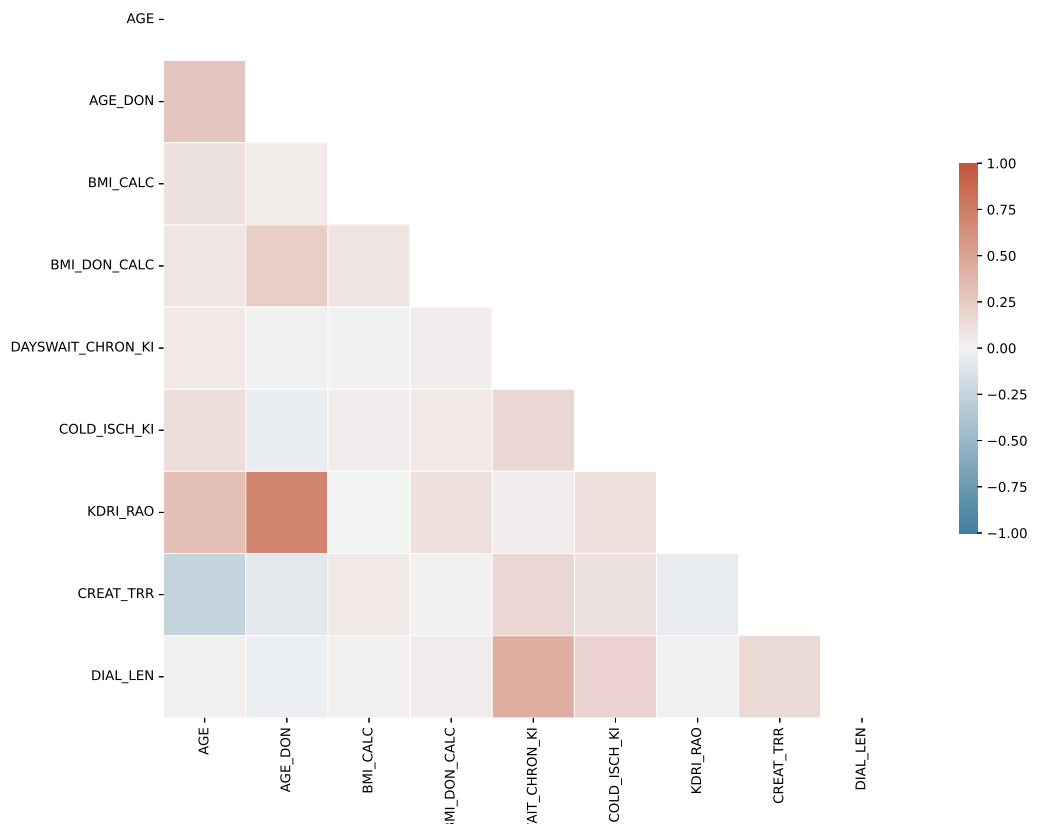


Figure 5.1: Correlation matrix for all covariates

Table 5.1: Descriptive statistics for numerical variables

Variable	Mean	Std	Min	25%	50%	75%	Max	Missing (%)
AGE	52.0	13.5	18.0	43.0	54.0	62.0	96.0	0.0
AGE_DON	39.9	15.0	0.0	29.0	41.0	51.0	88.0	0.0
BMI_CALC	28.1	5.4	15.0	24.0	27.6	31.7	74.2	0.4
BMI_DON_CALC	27.4	6.3	7.7	23.2	26.6	30.5	74.4	1.4
DAYSWAIT_CHRON_KI	685.2	718.9	0.0	138.0	427.0	1032.0	8554.0	0.0
KDRI_RAO	1.26	0.394	0.599	0.960	1.19	1.49	4.24	30.7
CREAT_TRR	7.901	3.619	0.1	5.28	7.4	9.95	36.02	1.9
DIAL_LEN in days	1335.3	1099.3	0	501	1096	1899	15395	20.5
COLD_ISCH_KI in hours	13.7	10.6	0.0	3.4	13.1	20.6	99.0	7.5

Table 5.2: Category percentages in GENDER and GENDER\_DON

Category	GENDER (%)	GENDER_DON (%)
Male	61.1	53.9
Female	38.9	46.1

### 5.1.2 Categorical Variables

Our dataset offers intriguing insights into the categorical variables, providing an overview of donors' and recipients' gender and ethnicity, the degree of mismatch across the A, B, and DR loci, and other key factors.

An examination of Table 5.2 reveals a pronounced dominance of male recipients and donors. This aligns with the established pattern of End-Stage Kidney Disease (ESKD) being more prevalent among men. This imbalance, though anticipated, necessitates further scrutiny to fully grasp its underlying triggers and repercussions.

In terms of ethnicity, as displayed in Table 5.3, the white non-Hispanic population forms the majority of both donors and recipients, mirroring the demographic distribution of the United States.

Our dataset, via Table 5.4, also emphasizes the frequent occurrence of at least one mismatch in each transplant locus. This underlines the inherent difficulty in finding an exact match for kidney transplants but also the potential for success despite some degree of mismatch.

Table 5.1.2 offers valuable insights into patients' medical histories. A considerable majority under-

Table 5.3: Category percentages in ETHCAT and ETHCAT\_DON

Category	ETHCAT (%)	ETHCAT_DON (%)
White, Non-Hispanic	48.2	68.6
Black, Non-Hispanic	27.1	12.8
Hispanic/Latino	16.6	14.5
Asian, Non-Hispanic	6.3	2.9
Amer Ind/Alaska Native, Non-Hispanic	0.9	0.5
Native Hawaiian/other Pacific Islander, Non-Hispanic	0.4	0.3
Multiracial, Non-Hispanic	0.5	0.4

Table 5.4: Category percentages in AMIS, BMIS and DRMIS

Category	AMIS (%)	BMIS (%)	DRMIS (%)
0	15.2	10.6	17.2
1	41.6	31.4	47.0
2	42.5	57.2	35.1
Null or Missing	0.7	0.7	0.7

Table 5.5: Dialysis and Medical Status Category Percentages

Category	DIAL_TRR	ON_DIALYSIS	HIST_HYPERTENS_DON	HCV_SEROSTATUS
No / Negative	18.2	26.7	74.8	90.6
Yes / Positive	81.2	73.3	20.0	4.8
Null or Missing	0.6	0.01	5.1	4.6

went dialysis before the transplant, and a lesser percentage of donors had a history of hypertension, both factors that could potentially impact post-transplant outcomes.

Regarding geographical distribution, Table 5.6 indicates that the dataset evenly spans across UNOS regions, suggesting geographical diversity and representation of the entire United States.

Another crucial observation is the prevalence of cadaverous donors, accounting for nearly 70% of the dataset. This observation is particularly noteworthy as survival rates typically dip for transplants from cadaverous donors compared to living donors.

Table 5.7 illustrates that almost a quarter of the population battles type II diabetes, which is associated with a multitude of health complications.

Among the deceased organ donors in our dataset, the primary causes of death are head trauma and anoxia, accounting for 23.8% and 21.4% of cases, respectively. Specific category labeled 'Not Applicable' has been introduced to accommodate living donors. Since the cause of death is irrelevant for these individuals, this category covers 30.4% of the data, providing a meaningful way to distinguish between the different types of donors.

## 5.2 Survival Analysis Using Kaplan-Meier and Nelson-Aalen Models

In this section, we present the results of the survival analysis conducted on the dataset. The Kaplan-Meier (KM) and Nelson-Aalen (NA) models were used to estimate the survival function and cumulative hazard function, respectively.

Table 5.6: Category percentages in REGION variable

Category	1	2	3	4	5	6	7	8	9	10	11
REGION (%)	3.97	13.00	13.42	9.41	16.32	3.47	9.03	6.10	7.16	8.35	9.79

Table 5.7: Category percentages in DIAB

Description	Percentage (%)
No	65.3
Type I	3.3
Type II	24.6
Type Other	0.5
Type Unknown	6.4

### 5.2.1 Kaplan-Meier Survival Curve

Figure 5.2 displays the Kaplan-Meier survival curve for our entire research cohort. This curve estimates the probability of survival over time. The median survival time, illustrated by the dashed line, is 3942 days. When calculated without accounting for censored data, the median survival time significantly reduces to 1583 days. This stark difference underscores the necessity of survival analysis techniques in handling censored data. Indeed, in our dataset, a substantial 65% of the data is censored, reinforcing this argument.

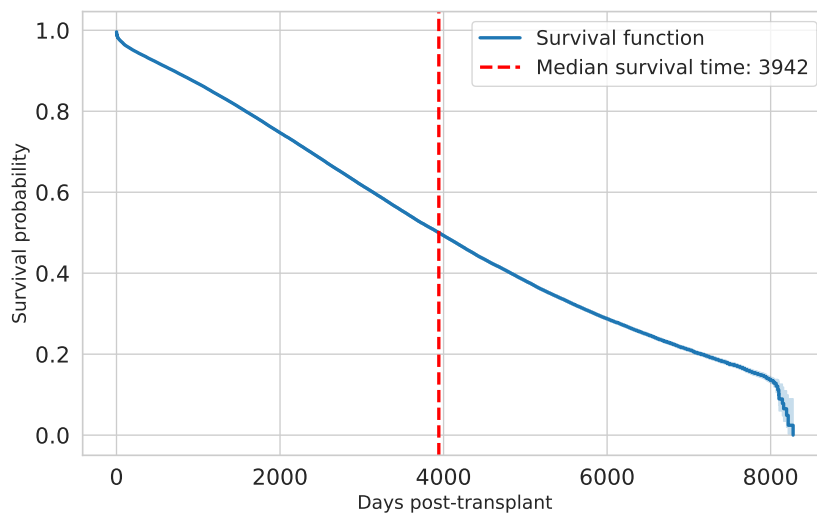


Figure 5.2: Kaplan-Meier Survival Curve UNOS

Figure 5.3 portrays the Kaplan-Meier survival curve derived from a sample of 566 living donors at IKEM. It's crucial to acknowledge that a high percentage (90%) of the observations are censored. This implies that the majority of patients were still alive at the end of the observation period, which is why the survival curve doesn't reach the 0.5 probability threshold and a median survival time cannot be determined.

The IKEM dataset has a relatively small sample size and its patient selection process significantly varies from standard procedures in the United States. These factors limit the ability to directly compare survival outcomes between the two groups. Consequently, while this graph provides insights into the survival time of living donors at IKEM, it doesn't necessarily reflect survival trends in a broader or different demographic context. Any attempt to compare survival times across different populations should entail a comprehensive analysis that acknowledges and adjusts for these discrepancies.

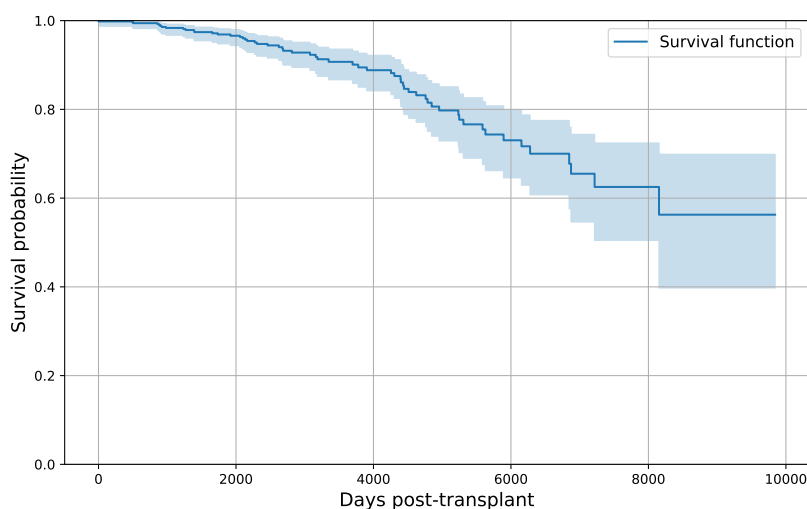


Figure 5.3: Kaplan-Meier Survival Curve IKEM

### 5.2.2 Nelson-Aalen Cumulative Hazard Curve

The Nelson-Aalen cumulative hazard curve for the entire cohort is shown in Figure 5.4. The curve provides an estimate of the cumulative hazard over time. The shaded area represents the 95% confidence interval. It is important to note that the reliability of the curve decreases after about 8000 days due to the reduced number of individuals at risk.

### 5.2.3 Histogram of Transplant Year

Figure 5.5 presents a histogram of the transplant year for both censored and non-censored data. The histogram shows an increase in the number of censored operations over time, reflecting the fact that more recent transplants have had less time to observe the event of interest.

## 5.3 Comparative Analysis of Key Variables Across Groups

In this section, we compare the survival probabilities across different groups.

### 5.3.1 Kaplan-Meier Survival Curves by Ethnicity

Figure 5.6 presents the Kaplan-Meier survival curves for each ethnicity category. The curves reveal differences in survival probabilities across ethnicities. The median survival times for each group can be observed in Table 5.8. For instance, the Asian American population shows the highest survival time, while the Indigenous population shows the lowest. However, it is important to note that these differences cannot be attributed to specific causes without further investigation.

### 5.3.2 Kaplan-Meier Survival Curves by Age

Figure 5.7 displays the Kaplan-Meier survival curves grouped by age. For this analysis, patients are divided into age groups of ten-year intervals, except for the youngest group (18-29 years) and the oldest

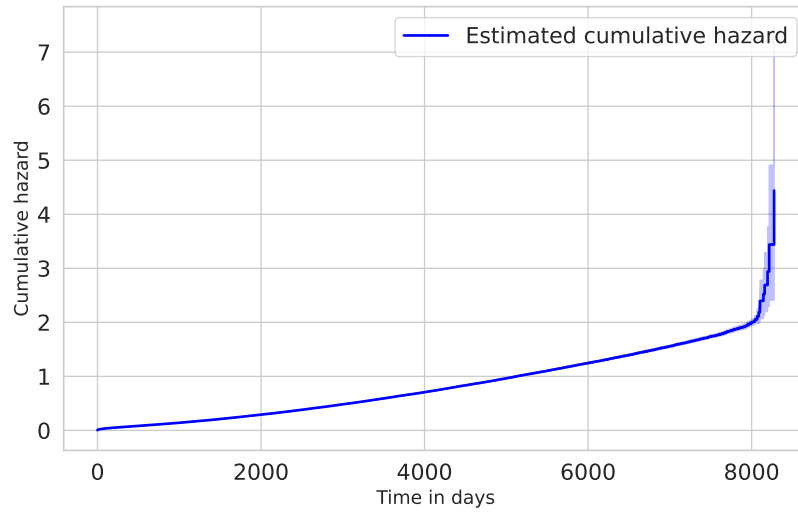


Figure 5.4: Nelson-Aalen Cumulative Hazard Curve

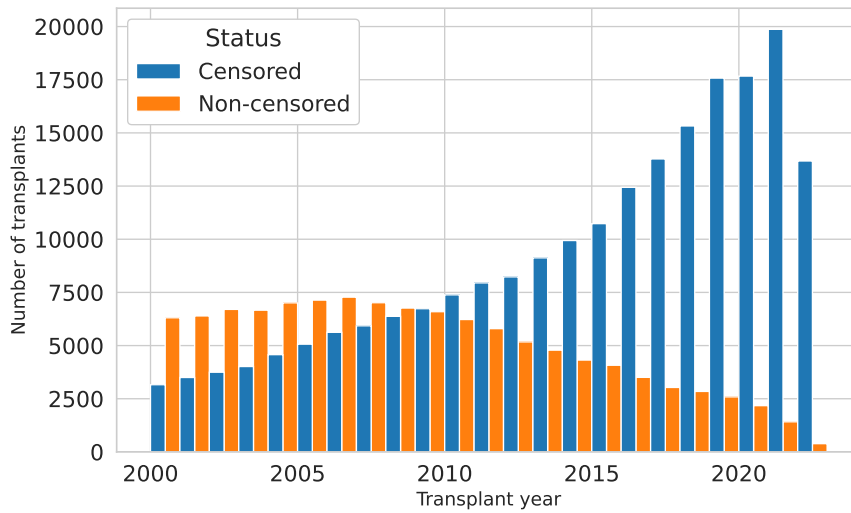


Figure 5.5: Histogram with Transplant Year - Censored and Non-Censored Data

Table 5.8: Ethnicity Median Survival Time

Ethnicity	Median Survival Time
White, Non-Hispanic	4108.0
Black, Non-Hispanic	3295.0
Hispanic/Latino	4355.0
Asian, Non-Hispanic	4903.0
Amer Ind/Alaska Nat.	3212.0
Native Hawaiian	3885.0
Multiracial	4085.0



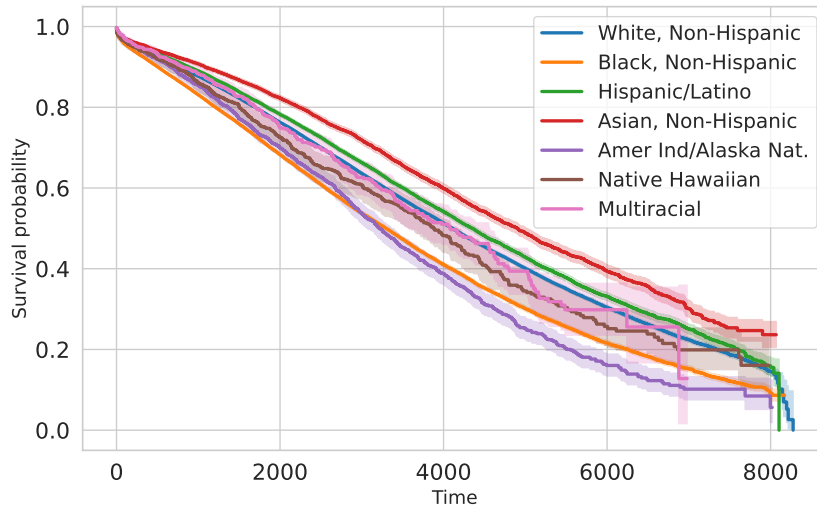


Figure 5.6: Kaplan-Meier Survival Curve by Ethnicity

Table 5.9: Median Survival Time by Donor Type

Donor Type	Median Survival
Cadaverous	3489.0
Living	4999.0

Table 5.10: Median Survival Time by Gender

Gender	Median Survival
Male	3784.0
Female	4180.0

group (80+ years). Interestingly, the data does not show a consistent decrease in survival times with increasing age. In particular, the median survival time of the 30-39 age group, which is 5004 days, is longer than that of the younger 18-29 age group, which is 4554 days.

This finding suggests that a younger age does not always result in a longer graft survival time, and there might be other factors at play. As expected, the oldest recipients, especially those aged 80 or above, tend to have shorter survival times, with a median survival time of 1843 days. This reflects the complex relationship between age and other medical and biological factors in the transplantation process.

Being an inherent patient characteristic that cannot be adjusted, age plays a crucial role in the transplantation process and can influence several other factors. This adds a layer of complexity to the analysis. As such, when interpreting survival probabilities, it's crucial to consider age alongside other variables.

### 5.3.3 Survival Analysis by Donor Type and Gender

In this section, we present the results of the survival analysis conducted on the dataset, segmented by donor type and gender. The median survival times for each category, calculated using the Kaplan-Meier estimator, are presented in Tables 5.9 and 5.10.

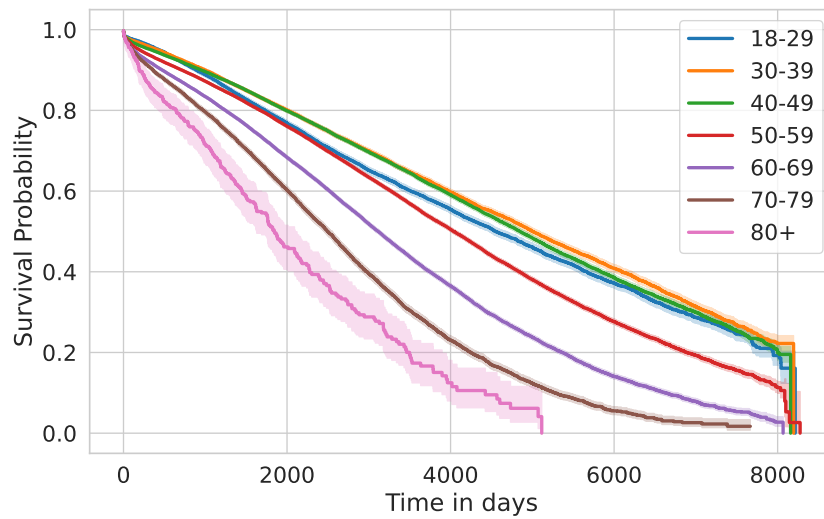


Figure 5.7: Kaplan-Meier Survival Curves by Age Group

In this chapter, we conducted a comprehensive statistical analysis of our dataset. We provided descriptive statistics of our numerical and categorical variables, and examined the relationships between our variables using a correlation matrix. We also conducted a survival analysis using the Kaplan-Meier and Nelson-Aalen models, and compared the survival probabilities across different groups. Our findings revealed interesting patterns and relationships in our data, and highlighted the importance of considering multiple variables when interpreting survival probabilities. In the next chapter, we will use this analysis to inform our machine learning models for survival prediction in kidney transplants.

## Chapter 6

# Model Development and Evaluation

### 6.1 Model Selection and Evaluation

This study includes a varied selection of models chosen based on their applicability and performance in survival analysis. These models have been derived from an expansive array discussed previously, with focus placed on models capable of producing individualized hazard rates or survival functions. This section delves into the specific implementations of the models mentioned theoretically in Section 3.1.

#### 6.1.1 Parametric Models

These models were included due to their interpretability and straightforward explanation. The ExponentialFitter, WeibullFitter and Gompertz models from the PySurvival package were utilized, corresponding to the models detailed in Section 3.1.5. These models presume a particular distribution for survival times, providing a satisfactory fit to the data given that the distributional assumption is correct. All models utilize PyTorch and gradient descent to minimize their loss function or the negative log-likelihood as discussed in 3.1.3.

#### 6.1.2 Semi-Parametric Models

The Cox Proportional Hazard model's usefulness in survival analysis prompted us to select three distinct implementations for our study. Despite their differences, they all adhere to the same fundamental principles as outlined in Section 3.1.6.1. These include CoxnetSurvivalAnalysis from scikit-survival, CoxPHModel from PySurvival, and NonLinearCoxPHModel from PySurvival.

##### CoxnetSurvivalAnalysis

The CoxnetSurvivalAnalysis, equipped with Elastic Net regularization, ensures computational efficiency through its C++ implementation of coordinate descent. This method, which optimizes one variable at a time while keeping others constant, is efficient for large, sparse datasets. However, it may fall short by converging to a local minimum in non-convex problems, potentially resulting in sub-optimal model performance. To handle tied events robustly, the model employs Breslow's estimation, making it suitable for survival analysis where data sparsity is prevalent and computational efficiency is crucial.

## CoxPHModel

Conversely, the CoxPHModel provides only L2 regularization, using Newton's method for optimization - a technique that relies on the second derivatives of the objective function to achieve quadratic convergence. It employs the Efron method for handling tied events. This model may operate slower than the previous implementation but has less risk of failing to find the global minimum due to the inherent shortcomings of coordinate descent.

## NonLinearCoxPHModel

An interesting choice is the NonLinearCoxPHModel, a model that incorporates Neural Networks to model the Cox model coefficients, thus capturing non-linear relationships. This model uses Efron's log-likelihood and PyTorch to compute the gradient and carry out first-order optimization. It serves as an implementation of the DeepSurv model, a concept propagated by Katzman in [18].

The DeepSurv model, a deep learning-based survival analysis approach, enhances the traditional Cox proportional hazards model. It includes a Multi-Layer Perceptron (MLP) described in Section 1.4.3.1, and introduces non-linear activation functions to handle complex feature interdependencies. Constructed with an input layer, multiple hidden layers, and an output layer built on the principles of the Cox proportional hazards model, the loss function is derived from the Cox model's partial likelihood, prioritizing the optimization of risk score prediction.

PySurvival, a Python library, facilitates the implementation of DeepSurv by encapsulating the model within the NonLinearCoxPHModel class. PySurvival allows for the configuration of the model structure by defining a list of dictionaries detailing the activation function and neuron count per layer. It supports multiple activation functions such as ReLU, sigmoid, Atan, BentIdentity, and CosReLU, among others.

The PySurvival loss function employs Efron's approximation of Cox's partial likelihood with L2 regularization, offering a robust solution to prevent overfitting. It supports various optimization algorithms like Adam, RMSprop, and SGD, with added features like dropout and batch normalization that help enhance model generalization and accelerate the training process.

### 6.1.3 Other Machine Learning Models:

RandomSurvivalForests from scikit-survival: This tree-based ensemble method aptly handles censored data. Significantly, Random Survival Forests use the log-rank test as their splitting rule. The Scikit Survival implementation is a close approximation of the method used in the R package 'randomSurvivalForest'.

### Models that were tested but not used for comparison

A number of models underwent testing but were not used in the comparative study due to various factors. For instance, the Survival Support Vector Machines from scikit-survival were evaluated but ultimately excluded from the research. Despite their capability to rank patients, these models lack the ability to predict survival functions, which is a critical feature for our investigation.

Likewise, gradient boosting models from scikit-survival were considered but their lengthy training times and non-parallelizable training process made them unsuitable for inclusion. Given the required number of base learners, these models were determined to be impractical.

In addition to these, we assessed numerous models from a variety of packages and libraries. The goal was to identify those that provide a combination of strong performance on the dataset and innovative strategies in survival analysis. Some models were discounted due to subpar performance, excessive

training durations, or simply being redundant. This culling process allowed us to focus on the most potent and relevant methods for our study.

Ultimately, the objective of this thorough selection process was to present a focused yet comprehensive evaluation of survival prediction models. This method of evaluation provides valuable insights into the relative strengths of the models and their applicability to diverse datasets and clinical contexts.

## 6.2 Model Training

As previously stated, the models were trained on the Helios cluster. Given the substantial size of the dataset, training times proved to be especially challenging for machine learning models, with durations varying significantly across different models. To tackle these challenges and accelerate the process, we leaned heavily on automation.

In our approach, we developed two separate, yet efficient pipelines for training models from the two different libraries we used, necessitated by their unique APIs. Despite the variance in the training pipelines, we maintained uniformity in the data processing stage, ensuring consistent pre-processing of the dataset across different models.

Furthermore, these pipelines automated a series of operations including data preprocessing, model training or grid search, model evaluation, and saving of the models. All these tasks were executed within the pipelines, accompanied by comprehensive logging of all processes. This level of automation not only increased the efficiency of the process but also helped overcome the limitations of using just Jupyter notebooks, which proved unsustainable given the scale of the task. For transparency and reproducibility, the Python code is available at: <https://github.com/peterstran/ml-unos2>.

## 6.3 Hyperparameter Tuning

We conducted hyperparameter tuning separately for each model to ensure fair conditions. For models using the scikit-learn API, we utilized existing functions for parameter space search, such as GridSearchCV and RandomSearch. For PySurvival models, we implemented a simple grid search. The inconsistency in PySurvival's API and varied model implementations complicated more extensive optimization and cross-validation.

## 6.4 Model Evaluation

For evaluating our models, we employed a combination of discrimination and calibration measures. Harrell's C-Index, C-index IPCW and Mean Area Under the Curve (AUC) served as our discrimination metrics, providing insights into the models' ability to correctly distinguish between different outcomes.

For calibration - which assesses the agreement between the predicted probabilities of an outcome and the actual observed frequencies - we used the Integrated Brier Score. Notably, this score, as well as the Mean AUC, was calculated from the 10th to the 90th percentile of the uncensored survival times.

All these metrics, which were drawn from the Scikit Survival package, together delivered a comprehensive assessment of the models' predictive accuracy and ability to discriminate outcomes.

Importantly, we based our results on test data, and refrained from further model tuning to prevent overfitting. This approach ensures that our model evaluations are unbiased and reflective of genuine performance on unseen data.

# Chapter 7

## Results and Previous Research

### 7.1 Performance Evaluation of the Machine Learning Models

This section focuses on the evaluation of various machine learning models' performance, exploring the results of key metrics such as C-Index IPCW, Harrell's C-Index, the Integrated Brier Score (IBS), and Mean AUC.

Table 7.1 encapsulates the performance outcomes of all models incorporated in this study, and the ensuing sections will offer detailed insight into the individual performance of each model.

The C-Index emerges as our primary metric for comparison, owing to its wide utilization in numerous studies, making it an optimal point of reference.

One general observation across all models is that the C-Index usually supersedes C-Index IPCW, as it adjusts to the censoring distribution. Notably, Mean AUC, a distinct measure of discrimination, doesn't correspond directly with the ranking dictated by the C-Index. This discrepancy implies that hyperparameter tuning may inadvertently overlook other performance metrics.

#### Parametric Models:

Parametric models, though reliant on data fitting, offer the advantage of simplicity and understandability. Among our seven models, the Exponential model recorded the least impressive results, with the IBS revealing a mismatch between the shape of the underlying function and the data.

The Weibull model performed slightly better in terms of the C-Index but demonstrated a worse IBS despite having more degrees of freedom. This outcome might be linked to the use of C-Index IPCW in the Grid Search.

Table 7.1: Performance Metrics of Machine Learning Models

Model	C-Index	C-Index IPCW	Integrated Brier Score	Mean AUC	C-index Death
Exponential	0.6232	0.6025	0.4189	0.6425	0.6458
Weibull	0.6233	0.6064	0.4979	0.6359	0.6488
Gomperetz	0.6543	0.6422	0.1872	0.6927	0.7134
CoxPHModel	0.6544	0.6420	0.1692	0.6831	0.7227
CoxnetSurvivalAnalysis	0.6533	0.6425	0.1689	0.6898	0.7203
NonLinearCoxPHModel	0.6645	0.6526	0.1668	0.7107	0.7245
RandomSurvivalForest	0.6514	0.6446	0.1678	0.6859	0.7199

Table 7.2: Hyperparameters of the Pysurvival models

Model	Initialization method	L2 regularization	Learning Rate	# Epochs
Exponential	Glorot uniform	0.900	0.00079	2200
Weibull	Glorot uniform	5.62	0.0010	3000
Gompertz	Glorot uniform	5.62	0.00019	5000
CoxPHModel	Zeros	0.000100	1.0	-
NonLinearCoxPHModel	Glorot uniform	0.00100	0.010	100

Interestingly, the Gompertz model, with a C-index of 0.6543, surpassed both the CoxnetSurvival-Analysis and RandomSurvivalForest in the C-Index. This victory proves that when the correct distribution shape is chosen, even simple models can outperform more complex counterparts. Table 7.2 details the best hyperparameters for these models.

### Semiparametric Models

The widely referenced Cox linear model, renowned for its stability and ubiquity in research, provides a robust benchmark for model performance evaluation.

In our study, the Scikit Survival Cox model, subject to an ElasticNet penalty (L1 ratio) of 0.12, delivered a C-Index of 0.6533. This performance was marginally suboptimal relative to other models, a discrepancy potentially attributable to its coordinate descent implementation.

The CoxPHModel, leveraging a fast and reliable Newton optimization, manifested as the second most proficient model with a C-Index of 0.6544.

Remarkably, the NonLinearCoxPHModel (DeepSurv) presented a dominant performance with a significantly elevated C-Index of 0.6645. This model exhibited superior performance not only with respect to the C-Score but across all other metrics, especially the IBS, indicating successful model calibration. The training of this model necessitated a sophisticated exploration of a myriad of hyperparameters, ranging from the selection of the activation function to the determination of the network size. The top-performing model featured two fully connected hidden layers, each comprising 64 neurons, and employed the ReLU activation function. Our experiments with larger network sizes did not yield any improvements.

The optimal hyperparameters for the PySurvival models are listed in Table 7.2. A comprehensive investigation is imperative to maximize this model’s potential. Implementing this model beyond the limitations of the PySurvival API could offer additional insights and avenues for enhancement.

### Machine Learning Models

We chose the Random Survival Forest (RSF) model for detailed study because we thought it was the best among different machine learning models. However, despite the time spent on training it, the model’s performance was lower than expected. The performance score, or C-Index, was 0.6514, which was lower than our Cox linear models. This was surprising because RSF models are often used in these types of analyses.

Training these models took a long time, sometimes up to 24 hours, and often caused issues like memory shortages and crashes on a powerful computer with 370GB of RAM. We were limited to a maximum of 800 estimators and found that a tree depth of 12 was enough. We used log2 for feature splits in the decision tree.

Improvements to the model’s performance might come from better hardware, new data splitting methods, or different model implementations. However, when we compare this with linear and neural network models, which require less computing power, RSF models seem costly. Other versions of the algorithm took even longer to train, except for Extra Forest models which were faster but performed worse. These findings suggest that we need to find better ways to improve these models, while considering their high resource requirements.

### **Patient survival prediction results**

Each model was trained and evaluated on a patient’s survival indicator, mentioned in 4.4, for comparative analysis. The NonLinearCoxPHModel, employing the same hyperparameters, achieved the highest score with a C-Index of 0.7245, showcasing the performance disparity among the models. The results for other models are demonstrated in Table 7.1 under the C-Index Death column.

## **7.2 Feature Importance Analysis**

The predictive power of a machine learning model is largely dependent on the relevance and significance of the features it uses. In our study, we leveraged the CoxPHModel and identified the top 10 features through the analysis of absolute coefficient values, as represented in Table 7.3.

Initially, our goal was to calculate permutation feature importance [2] using our most refined neural network model. We created a parallelized permutation importance function for this purpose, as our model was not compatible with the sklearn’s default implementation, and PyTorch lacked a native solution. This function assessed feature importance by employing the C-Index, while also incorporating standard deviation measurements. Regrettably, this approach fell short of our expectations, as permutation importance can sometimes skew the true significance of features, particularly within neural network models. A recurring pattern we observed was that highly important features were frequently those with lower representation in the categorical data. This phenomenon might stem from the model concentrating on particular data subsets, but it doesn’t necessarily denote universal importance.

Despite these limitations, the output from our model can be leveraged for anomaly detection due to the highlighted importance of infrequent features. While this information is valuable, translating these insights into practical applications, like in a kidney scoring/allocation system, presents significant challenges due to the rarity of these anomalous instances and the complexities associated with integrating them into the system.

Another complexity encountered was that there is no straightforward method to access the values of the Cox coefficients in a DeepSurv type model. The coefficients are dictated by the output layer within a deep network, and the model weights are notoriously challenging to interpret.

Furthermore, evaluating feature importance only illuminates the magnitude of the feature’s importance, not whether the feature positively or negatively affects outcomes. Hence, we opted for a more encompassing approach to our analysis, which can better inform the enhancement or development of numerical scoring systems such as KDPI and EPTS.

It’s critical to clarify that a positive coefficient value suggests an increase in hazard, thereby reducing the anticipated survival time, whereas a negative coefficient value implies the opposite. Significantly, the recipient’s age ‘AGE’ emerged as the most consequential feature, supporting our initial hypothesis given its well-documented correlation with kidney graft survival. ‘AGE\_DON’ ranked seventh, indicating that the quality of the kidney decreases with donor age.

Another pivotal feature is ‘KDRI\_RAO’, which shows the impact of kidney quality on graft survival. Our engineered feature, ‘DIAL\_LEN’, ranked fifth, outperforming ‘DAYSWAIT\_CHRON\_KI’ - the



Table 7.3: Top 10 Features Used in the CoxPHModel with their Coefficient Values

Feature name	Coef
AGE	0.14
KDRI_RAO	0.11
DIAL_LEN	0.082
ETHCAT - Asian, Non-Hispanic	-0.079
DIAB - No	-0.072
DIAG_KI - Group 2	-0.070
AGE_DON	0.066
CREAT_TRR	-0.058
ETHCAT - Hispanic/Latino	-0.057
DIAG_KI - Group 1	-0.054

duration the recipient waited for a transplant - thereby demonstrating the effectiveness of our feature engineering strategies.

Interestingly, the categories 'ETHCAT' for 'Asian, Non-Hispanic' and 'Hispanic/Latino' ranked fourth and ninth respectively. This suggests that ethnicity plays a role in transplant predictions, although the ethical implications of incorporating this into a scoring system or model remain debatable. Both categories had negative values, suggesting an improvement in expected survival time from the baseline.

The absence of diabetes 'DIAB - No' emerged as the fifth-ranking feature. This emphasizes the influence of diabetes on renal graft outcomes, highlighting the complications associated with diabetes and its detrimental impact on kidney graft survival.

Our self-created diagnosis categories 'DIAG\_KI' took the sixth and tenth place, signifying the crucial role of diagnosis severity and type in outcome prediction. Representing the least severe chronic kidney disease diagnoses, these categories could be readily integrated into a new scoring system.

Finally, it's noteworthy that three of the top ten features were results of our innovative engineering, underscoring the importance of ongoing exploration and development of new features to enhance the model's performance.

### 7.3 Previous Research on Kidney Transplant Survival Prediction

In this segment, we will conduct a comparative analysis of three influential studies in the field of kidney transplantation survival prediction. Each research focuses on a different aspect of survival prediction, thus highlighting the complex nature of the process. It's crucial to emphasize that graft survival prediction is a distinct and typically more intricate task than patient survival prediction, due to the myriad factors affecting graft longevity. Two of the chosen studies primarily focus on graft survival, while the third one provides a comparative angle by focusing on mortality prediction. This selection, while not exhaustive, offers a comprehensive view of various methodologies in the field.

Senanayake et al. [38] used an Australian dataset to develop and validate predictive models for graft failure after deceased donor kidney transplantation. Their models included random survival forest, survival support vector machine, and Cox proportional regression. Notably, the Cox regression and random survival forest models, with a C-index of 0.67, exhibited superior discrimination of graft failure, suggesting their potential utility in pre-transplant decision-making.

In Paquette et al.'s study [30], the results indicate that the neural network-based models, namely DeepSurv, DeepHit, and RNNs, exhibited superior discriminative ability in comparison to traditional

models like the Cox model and random survival forest model. This superiority was observed in the C-index metrics, where the neural network-based models scored 0.650, 0.661, and 0.659, respectively, compared to the Cox model's 0.646 and the random survival forest model's 0.644.

Finally, the study by van Walraven et al. [41] shifts focus to predicting mortality among end-stage renal disease patients considering kidney transplantation. Their predictive model, a prognostic index derived from twelve variables, effectively divided patients into 26 distinct risk categories, each with a unique five-year survival rate. The index's concordance probability was 0.746 (95% CI 0.741–0.751), demonstrating its strong discriminative power in predicting mortality. This study broadens the scope of discussion by considering patient survival separate from graft survival.

## **7.4 Comparison to Previous Research**

The comparison of our study with previous pivotal research within the field of kidney transplantation survival prediction reveals some key insights. The performance of our models aligns closely with the models presented by Senanayake et al. [38] and Paquette et al.'s study [30]. The difference in the C-index between the top-performing models across these studies is around a hundredth, which is negligible. This consistency is encouraging as it indicates the validity and reproducibility of the findings. Notably, there is a larger gap in the C-index when death is used as the indicator. This is expected as variable selection and hyperparameter tuning were optimized for the graft survival indicator in our study.

# Chapter 8

## Conclusion

### 8.1 Limitations of the Study and Future Work

Our study, while offering meaningful insights into kidney graft survival prediction, does have certain constraints that present opportunities for future investigations. Our models have shown promising results consistent with previous landmark studies, however, more rigorous validation using independent datasets is required to confirm their reliability across varied patient groups. Additionally, the extensive computational time and resources required for training specific models, like the Random Survival Forest, may pose practical implementation challenges. Future research could focus on optimizing these models for greater efficiency and less computational demand.

Our models primarily targeted graft survival, but there is potential for refinement to improve patient mortality prediction, as underscored by van Walraven et al.'s study [41]. Our neural networks' notable performance in graft survival prediction suggests that subsequent research could expand upon the development of larger, more practical neural network models.

An additional area for future exploration involves a closer collaboration with clinicians. Their practical experience and expertise can help ascertain the deployability and real-world applicability of these machine learning approaches, ensuring alignment with the complexities of clinical practice. This collaboration could deepen our understanding of transplantation medicine, thereby refining the predictive power and applicability of future models.

### 8.2 Practical Implications and Applications

The findings from our research carry considerable practical significance, particularly in the Czech Republic, where no model like ours has been deployed nor its feature importance used to develop a scoring method akin to KDRI/EPTS. The improved accuracy of our survival prediction models could inform pre-transplant decisions, helping physicians assess different graft options, and in turn, enhancing patient outcomes.

A web-based application modelled after the IChooseKidney study [32] could equip patients with crucial information about transplantation. By providing mortality risk or survival benefit data based on individual patient profiles, the application could aid in informed decision-making. If UNOS-like data were made publicly available in the Czech Republic, our models could be tailored to reflect local demographics more accurately, a concept extendable to other regions not covered in our study.

Our effective application of machine learning techniques, including neural networks, suggests their potential for broader integration in healthcare and transplantation medicine. Additionally, these models

could be translated into user-friendly clinical tools offering evidence-based guidance, thus assisting clinicians and patients in making informed decisions, improving graft survival rates and patient mortality outcomes.

### **8.3 Summary of the Main Findings**

The goal of this thesis was to apply machine learning techniques for predicting kidney graft survival, an intersection of computer science and medical science. We began with a comprehensive review of contemporary machine learning methods, providing a solid foundation for practical application.

Our systematic exploration of kidney transplantation and the current methods of donor-recipient compatibility assessment revealed potential weaknesses, thus identifying areas for improvement. We also delved into survival analysis and the mathematics of predicting time-to-event data, a crucial step that empowered us to create accurate predictions.

Using software tools such as scikit-learn, PyTorch, scikit-survival, and PySurvival, we analysed real data from kidney transplants. This allowed us to understand graft survival comprehensively. In our application phase, we compared various models for predicting graft survival, with Cox linear models, Gomperetz, and a neural network-based model, the DeepSurv, performing exceptionally well.

Despite the computational demands and the need for substantial validation on independent datasets, our findings show promising trends. We highlighted the effectiveness of neural networks and suggest future research for larger, more deployable models. These models can be refined for better patient mortality prediction and increased computational efficiency. It is also recommended that medical professionals be involved in validating the practicality and deployability of such approaches.

In conclusion, our study not only presents promising avenues for immediate application in kidney transplantation but also proposes a pathway for further investigation. As machine learning continues to evolve, it promises to revolutionize patient care in kidney transplantation, creating a more hopeful future for those in need.

# Data Source, Code, and Reproducibility

In this study, the data was 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 and in no way should be seen as an official policy of or interpretation by the OPTN or the U.S. Government. The source of the data is OPTN as of September 2022.

In an effort to uphold transparency and promote reproducibility, the Python code utilized for data pre-processing, model training, and evaluation has been made publicly accessible. All related resources, including the aforementioned code, can be retrieved from the corresponding GitHub repository at: <https://github.com/peterstran/ml-unos2>.

# Bibliography

- [1] Tilahun Alelign, Momina M. Ahmed, Kidist Bobosha, Yewondwossen Tadesse, Rawleigh Howe, and Beyene Petros. Kidney transplantation: The challenge of human leukocyte antigen and its therapeutic strategies. *Journal of Immunology Research*, 2018:1–18, 2018.
- [2] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26:1340–1347, 5 2010.
- [3] Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24:3927–3944, 12 2005.
- [4] Géron Aurélien. *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools, and techniques to build Intelligent Systems*. O’Reilly, 2023.
- [5] Ryan Bakker. An introduction to event history analysis: Regression models for survival data. Lecture notes for Oxford Spring School, June 18-20, 2007, Accessed: 2023-06-30, 2007.
- [6] 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*, 52:50–69, 2 2010.
- [7] Péter Biró, Bernadette Haase-Kromwijk, Tommy Andersson, Eyjólfur Ingi Ásgeirsson, Tatiana Baltetsová, Ioannis Boletis, Catarina Bolotinha, Gregor Bond, Georg Böhmig, Lisa Burnapp, Katarína Cechlárová, Paola Di Ciaccio, Jiri Fronek, Karine Hadaya, Aline Hemke, Christian Jacquelinet, Rachel Johnson, Rafal Kieszek, Dirk R. Kuypers, Ruthanne Leishman, Marie-Alice Macher, David Manlove, Georgia Menoudakou, Mikko Salonen, Bart Smeulders, Vito Sparacino, Frits C.R. Spieksma, María Oliva Valentín, Nic Wilson, and Joris van der Klundert. Building kidney exchange programmes in europe—an overview of exchange practice and activities. *Transplantation*, 103:1514–1522, 7 2019.
- [8] Ørnulf Borgan. *Nelson–Aalen Estimator*. John Wiley & Sons, Ltd, 2014.
- [9] T G Clark, M J Bradburn, S B Love, and D G Altman. Survival analysis part i: Basic concepts and first analyses. *British Journal of Cancer*, 89:232–238, 7 2003.
- [10] Cameron Davidson-Pilon. *lifelines, survival analysis in python*, May 2023.
- [11] K Sai Dhanush and Sv Sudha. Gene expression analysis using svm and knn classifiers on various datasets. pages 1325–1332. IEEE, 1 2023.
- [12] Gaohong Dong, Lu Mao, Bo Huang, Margaret Gamalo-Siebers, Jiuzhou Wang, GuangLei Yu, and David C. Hoaglin. The inverse-probability-of-censoring weighting (ipcw) adjusted win ratio statistic: an unbiased estimator in the presence of independent censoring. *Journal of Biopharmaceutical Statistics*, 30:882–899, 9 2020.

- [13] Eurotransplant. Eurotransplant manual. Available at: <https://www.eurotransplant.org/allocation/eurotransplant-manual>, 2023. Accessed: 2023-07-15.
- [14] Stephane Fotso et al. PySurvival: Open source package for survival analysis modeling, 2019. Accessed: 2023-06-30.
- [15] Elisa J. Gordon, Daniela P. Ladner, Juan Carlos Caicedo, and John Franklin. Disparities in kidney transplant outcomes: A review. *Seminars in Nephrology*, 30:81–89, 1 2010.
- [16] Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *ArXiv*, abs/1811.11347, 2018.
- [17] Malek Kamoun, Jill A. Hollenbach, Steven J. Mack, and Thomas M. Williams. Molecular HLA typing. pages 867–885. Springer International Publishing, 2016.
- [18] Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. DeepSurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18:24, 12 2018.
- [19] M. Kulich. Censored data analysis. Course notes for NMST531, Fall 2021/22, last updated Jan. 2, 2022, Available at: [https://www.karlin.mff.cuni.cz/~kulich/vyuka/archive/cens2021/doc/cens\\_notes\\_ext\\_210105.pdf](https://www.karlin.mff.cuni.cz/~kulich/vyuka/archive/cens2021/doc/cens_notes_ext_210105.pdf), Accessed: 2023-06-30, 2022.
- [20] Håvard Kvamme and Ørnulf Borgan. Time-to-event prediction with pytorch. <https://github.com/havakv/pycox>, 2019. Accessed: 2023-06-30.
- [21] D. Y. Lin. On the breslow estimator. *Lifetime Data Analysis*, 13:471–480, 12 2007.
- [22] Mary Lunn. Undergraduate lecture notes bs3b: Statistical lifetime models. Changed 18 January 2007, Accessed: 2023-06-30, 2007.
- [23] Ethan Mark, David Goldsman, Brian Gurbaxani, Pinar Keskinocak, and Joel Sokol. Using machine learning and an ensemble of methods to predict kidney transplant survival. *PLOS ONE*, 14:e0209068, 1 2019.
- [24] Mike May. Eight ways machine learning is assisting medicine. *Nature Medicine*, 27:2–3, 1 2021.
- [25] Gert Mayer and Guido G. Persijn. Eurotransplant kidney allocation system (etkas): rationale and implementation. *Nephrology Dialysis Transplantation*, 21:2–3, 1 2006.
- [26] Anand Nayyar, Lata Gadhavi, and Noor Zaman. Machine learning in healthcare: review, opportunities and challenges, 2021.
- [27] Organ Procurement and Transplantation Network (OPTN). A guide to calculating and interpreting the estimated post-transplant survival (epts) score used in the kidney allocation system (kas). [https://optn.transplant.hrsa.gov/media/1511/guide\\_to\\_calculating\\_interpreting\\_epts.pdf](https://optn.transplant.hrsa.gov/media/1511/guide_to_calculating_interpreting_epts.pdf), April 2020. Accessed: 2023-06-30.
- [28] Arjun Panesar. *Evaluating Machine Learning Models*, pages 189–205. Apress, Berkeley, CA, 2021.
- [29] Arjun Panesar. *Machine Learning Algorithms*, pages 85–144. Apress, Berkeley, CA, 2021.

- [30] François-Xavier Paquette, Amir Ghassemi, Olga Bukhtiyarova, Moustapha Cisse, Natanael Gagnon, Alexia Della Vecchia, Hobivola A Rabearivelo, and Youssef Loudiyi. Machine learning support for decision-making in kidney transplantation: Step-by-step development of a technological solution. *JMIR Medical Informatics*, 10:e34554, 6 2022.
- [31] R M Parry, W Jones, T H Stokes, J H Phan, R A Moffitt, H Fang, L Shi, A Oberthuer, M Fischer, W Tong, and M D Wang. k-nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *The Pharmacogenomics Journal*, 10:292–309, 8 2010.
- [32] Rachel E. Patzer, Mohua Basu, Christian P. Larsen, Stephen O. Pastan, Sumit Mohan, Michael Patzer, Michael Konomos, William M. McClellan, Janice Lea, David Howard, Jennifer Gander, and Kimberly Jacob Arriola. ichoose kidney. *Transplantation*, 100:630–639, 3 2016.
- [33] Todd E. Pesavento. Kidney transplantation in the context of renal replacement therapy. *Clinical Journal of the American Society of Nephrology*, 4:2035–2039, 12 2009.
- [34] 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.
- [35] Organ Procurement and Transplantation Network. A guide to calculating and interpreting the kidney donor profile index (kdpi). Available at: [https://optn.transplant.hrsa.gov/media/1512/guide\\_to\\_calculating\\_interpreting\\_kdpi.pdf](https://optn.transplant.hrsa.gov/media/1512/guide_to_calculating_interpreting_kdpi.pdf), 2020. Accessed: 2023-06-30.
- [36] Organ Procurement and Transplantation Network (OPTN). Optn policies. Available at: [https://optn.transplant.hrsa.gov/media/eavh5bf3/optn\\_policies.pdf](https://optn.transplant.hrsa.gov/media/eavh5bf3/optn_policies.pdf), 2023. Accessed: 2023-06-30.
- [37] Bharadhwaj Ravindhran, Pankaj Chandak, Nicole Schafer, Kaushal Kundalia, Wochan Hwang, Savvas Antoniadis, Usman Haroon, and Rhana Hassan Zakri. Machine learning models in predicting graft survival in kidney transplantation: meta-analysis. *BJS Open*, 7, 3 2023.
- [38] Sameera Senanayake, Sanjeeva Kularatna, Helen Healy, Nicholas Graves, Keshwar Baboolal, Matthew P. Sypek, and Adrian Barnett. Development and validation of a risk index to predict kidney graft survival: the kidney transplant risk index. *BMC Medical Research Methodology*, 21:127, 12 2021.
- [39] Nurhan Seyahi and Seyda Gul Ozcan. Artificial intelligence and kidney transplantation. *World J Transplant*, 11(7):277–289, 2021. Published online 2021 Jul 18.
- [40] Asanao Shimokawa, Yohei Kawasaki, and Etsuo Miyaoka. Comparison of splitting methods on survival tree. *The International Journal of Biostatistics*, 11, 1 2015.
- [41] C. van Walraven, P. C. Austin, and G. Knoll. Predicting potential survival benefit of renal transplantation in patients with chronic kidney disease. *Canadian Medical Association Journal*, 182:666–672, 4 2010.
- [42] Effy Vayena, Alessandro Blasimme, and I. Glenn Cohen. Machine learning in medicine: Addressing ethical challenges. *PLOS Medicine*, 15:e1002689, 11 2018.
- [43] Ondrej Viklicky, Jiri Fronek, Pavel Trunecka, Jan Pirk, and Robert Lischke. Organ transplantation in the czech republic. *Transplantation*, 101:2259–2261, 10 2017.



- [44] Ondrej Viklicky, Sebastian Krivanec, Hana Vavrinova, Gabriela Berlakovich, Tomas Marada, Janka Slatinska, Tereza Neradova, Renata Zamecnikova, Andreas Salat, Michael Hofmann, Gottfried Fischer, Antonij Slavcev, Pavel Chromy, Rainer Oberbauer, Tomas Pantoflicek, Sabine Wenda, Elisabeth Lehner, Ingrid Fae, Paolo Ferrari, Jiri Fronek, and Georg A. Böhmig. Crossing borders to facilitate live donor kidney transplantation: the czech-austrian kidney paired donation program – a retrospective study. *Transplant International*, 33:1199–1210, 10 2020.
- [45] Ondřej Viklický, Libor Janoušek, and Peter Baláž. *Transplantace ledviny v klinické praxi*. Grada, 2008.
- [46] Hong Wang and Gang Li. A selective review on random survival forests for high dimensional data. *Quantitative Bio-Science*, 36:85–96, 11 2017.
- [47] Jeffrey H. Wang, Melissa A. Skeans, and Ajay K. Israni. Current status of kidney transplant outcomes: Dying to survive. *Advances in Chronic Kidney Disease*, 23:281–286, 9 2016.