ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS A DADOS CENSORIADOS PARA PREVISÃO DE MORTALIDADE DE PACIENTES COM DOENÇA ARTERIAL CORONARIANA

Gabriel Cesario Buginga

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Edmundo Albuquerque de Souza e Silva, Ph.D.

Rio de Janeiro
Setembro de 2023

ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS A DADOS CENSORIADOS PARA PREVISÃO DE MORTALIDADE DE PACIENTES COM DOENÇA ARTERIAL CORONARIANA

Gabriel Cesario Buginga

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Edmundo Albuquerque de Souza e Silva, Ph.D.

Aprovada por: Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.
Prof. Valmir Carneiro Barbosa, Ph.D.
Prof. Christina G. de Souza e Silva, MD

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2023

Cesario Buginga, Gabriel

Algoritmos de Aprendizado de Máquina Aplicados a Dados Censoriados para Previsão de Mortalidade de Pacientes com Doença Arterial Coronariana/Gabriel Cesario Buginga. – Rio de Janeiro: UFRJ/COPPE, 2023.

X, 63 p.: il.; 29, 7cm.

Orientador: Edmundo Albuquerque de Souza e Silva, Ph.D.

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2023.

Referências Bibliográficas: p. 46 – 54.

1. Dados médicos. 2. Análise de sobrevivência. 3. Aprendizado de Máquina. I. Albuquerque de Souza e Silva, Ph.D., Edmundo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS A DADOS CENSORIADOS PARA PREVISÃO DE MORTALIDADE DE PACIENTES COM DOENÇA ARTERIAL CORONARIANA

Gabriel Cesario Buginga

Setembro/2023

Orientador: Edmundo Albuquerque de Souza e Silva, Ph.D.

Programa: Engenharia de Sistemas e Computação

A aprendizagem de máquina probabilística está sendo cada vez mais utilizada na área da saúde para processar dados e melhorar a eficácia dos processos de tomada de decisão dos profissionais. Pacientes recebem inferências precisas, aprimoradas por distribuições completas de probabilidade. Um aspecto crucial da análise de saúde é estudar a morte por qualquer causa e identificar os fatores que mais a influenciam. No entanto, trabalhos anteriores tendiam a explorar insuficientemente informações sobre pacientes que sobreviveram ou não realizaram uma análise completa de aprendizado de máquina. Neste estudo, analisamos um conjunto de dados de pacientes com doença arterial coronariana que foram encaminhados para reabilitação cardíaca (CR), com o objetivo de prever a morte por qualquer causa. Para 88% dos pacientes, suas informações de morte foram censuradas, ou seja, só temos um limite inferior de seu tempo de morte, tornando difícil fazer previsões precisas. Para resolver esse problema, aplicamos algoritmos da literatura de análise de sobrevivencia. Também usamos métodos de seleção de variáveis para reduzir o seu número em 92%, identificando apenas duas variáveis que melhor predizem a morte. Posteriormente, avaliamos um grupo diversificado de modelos e descobrimos que o modelo Survival Tree apresentou excelente desempenho e interpretabilidade, podendo ser utilizado por médicos apenas inspecionando um único diagrama. Além disso, desenvolvemos um novo algoritmo de clusterização para dados de sobrevivência, denominado **SurvMixClust**, para ajudar a modelar situações semelhantes ao nosso conjunto de dados e, ao mesmo tempo, encontrar grupos de pacientes com perfis de sobrevivência semelhantes.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

MACHINE LEARNING ALGORITHMS APPLIED TO CENSORED DATA TO PREDICT MORTALITY OF PATIENTS WITH CORONARY ARTERY DISEASE

Gabriel Cesario Buginga

September/2023

Advisor: Edmundo Albuquerque de Souza e Silva, Ph.D.

Department: Systems Engineering and Computer Science

Probabilistic machine learning is increasingly being used in healthcare to process data and improve the effectiveness of practitioners' decision-making processes. Patients receive precise inferences, enhanced by full probability distributions. One crucial aspect of healthcare analysis is to study death from any cause and identify the factors that influence it the most. However, past works tended to insufficiently explore information about patients that survived or did not conduct a full machine-learning analysis. In this study, we analyzed a dataset of patients with coronary artery disease who were referred to cardiac rehabilitation (CR), aiming to predict death from any cause. For 88% of patients, their death information was censored, i.e., we only have a lower bound of their time of death, making it challenging to make accurate predictions. To address this issue, we applied algorithms from the survival analysis literature. We also used feature selection methods to reduce the number of features by 92%, identifying only two features that best predict death. Afterward, we evaluated a diverse group of models and found that the Survival Tree model had excellent performance and interoperability, being capable of being used by medical practitioners by just inspecting a single diagram. Additionally, we developed a novel clusterization algorithm for survival data, named **SurvMixClust**, to help model situations similar to our dataset while also finding groups of patients having similar survival profiles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The use of artificial intelligence, particularly machine learning, is becoming more prevalent in healthcare [1, 2]. It provides an opportunity to enhance the effectiveness of the decision-making process of healthcare practitioners by processing vast amounts of data and generating valuable insights. With new data-rich technologies such as genomics, proteomics, and device biometrics, there is an increasing demand for these methods. These technologies generate more data than one individual can interpret alone. Additionally, patients are starting to request care that possesses a capability of being precise in relation to the patient's own characteristics [3, 4].

In medicine, It is often desired to acquire knowledge about time until an event, like death or the onset of cancer. The set of approaches that deal with these problems is called *survival analysis* [5, 6]. They are probability-based models which have a long history dating back to the 50s [7]. Overall, probabilistic models enrich the inferences [2, 8] because if a model only outputs average survival, like most regressions, and not a full survival function, It can create misleading results. For instance, two patients with the same average survival can have different early and late survival probabilities. This disparity creates a mistake if we treat them equally. Moreover, this enhanced view can provide a chance for a better plan for the future by naturally integrating uncertainty into statements that can then be used in the clinical decision process.

Survival prediction in cardiovascular medicine has not been fully explored through a non-linear probabilistic approach in previous research studies. These models have relied on linear regression techniques, which can oversimplify the complex, nonlinear interactions between potential prognostic factors. Consequently, these models may not be suitable for uncovering the patterns necessary for predicting an individual's risk accurately [9].

There has been a recent surge in the development of survival models, thanks to advancements in deep learning [10] and the increasing availability of data. Some of the notable ones include: DeepSurv [11], which is a generalized version of a

traditional method; DeepHit [12], a deep learning model that deals with competing risks; models for predicting the occurrence of oral cancer [13]; models for predicting cancer from histology and genomics [14], or from genomics and clinical data [15]; the SALMON algorithm, which integrates multi-omics data for breast cancer prediction [16]; the SurvivalNet model [17]; and a clusterization method specifically designed for survival analysis [18].

The aim of this study is to analyze potential predictors of survival for individuals with coronary artery disease using data from a retrospective cohort study named PROtECT (Prognostic Relevance Of data from patients Entering Cardiac Rehabilitation Training) [19]. Additionally, the study aims to develop a full-fledged survival model that can serve as an important tool for clinical decision support and provide evidence for further research in this area.

Throughout the production of this dissertation, our approach has been strictly focused on the use of machine learning and statistics. The text does not include any discussions or conclusions that require applied medical knowledge, even from cardiology. However, we have had systematic and productive interactions with healthcare providers during the process. Every step forward included a discussion with them since they are the final model users. Key inputs were feature pre-selection and explicit model interpretability necessity.

Our main contributions are summarized below:

- A long-term, interpretable, and lightweight model for overall survival prediction for individuals with known CAD (coronary artery disease) who are referred to a cardiac rehabilitation program.

- A published article available at [20]. In addition to the machine learning analysis presented here, it offers a complete investigation of the medical implications of our findings.

- Evidence for the feature "peak metabolic equivalents" being one the most critical pieces of information for survival prediction for individuals with known CAD.

- We created the **SurvMixClust** algorithm. It clusters survival data and can also be used to make predictions. It is competitive in relation to purely predictive algorithms and has better performance when compared to other clusterization algorithms, helping identify groups with similar survival profiles. The code for the model is publicly accessible and available at https://github.com/buginga/SurvMixClust

The manuscript is organized as follows. Chapter 2 presents the necessary data processing steps and assumptions. In Chapter 3, we provide background information

on survival analysis, including explanations of more specialized methods. Chapter 4 covers the selection of the most important features. Chapter 5 explains the rationale and benchmarks for the final proposed model. In Chapter 6, we gather the results and limitations from the proposed model. Chapter 7 presents the **SurvMixClust** algorithm and its results. Finally, Chapter 8 concludes the manuscript and restates the takeaway messages.

Chapter 2

Data

The dataset was obtained from a retrospective cohort study named PROtECT[1] [19]. Data was collected between September 1995 and March 2016 and was obtained by linking clinical registries from the following two sources:

- Alberta Provincial Project for Outcomes Assessment in Coronary Heart Disease (APPROACH)[21–23].
- The TotalCardiology Rehabilitation Network [24]. It is a Calgary-based cardiac rehabilitation provider for Alberta Health Services.

They included data from Calgary patients referred to the CR (cardiac rehabilitation) program at TotalCardiology following coronary angiography. There were two occasions in which data was obtained:

- *At the time of catheterization*: demographic characteristics, clinical risk factors, comorbidities, indication for coronary angiography and its results which are composed of coronary anatomy, left ventricular ejection fraction (LVEF), and the therapeutic management strategy.
- *Before the CR program enrollment*: a baseline clinical assessment that included a complete physical examination, anthropometric measurements (height and weight), a graded ET, and blood sample collection for analysis, such as complete lipid profile and hemoglobin A1c. Then, peak metabolic equivalents (METs), peak and resting heart rate (HR), peak and resting systolic blood pressure (SBP), and peak and resting diastolic blood pressure (DBP) were recorded via a peak graded exercise test[2].

[1] PROtECT Study: Prognostic Relevance Of data from patients Entering Cardiac Rehabilitation Training. The study protocol was approved by the University of Calgary's ethics review board.

[2] The exercise test (ET) follows the methodology from [25].

It resulted in a raw dataset of shape $(23215, 260)$, where each patient is represented by a single row. Figure 2.1 depicts, via a Sankey plot, the inclusion criteria from the original 23215. We dropped ten patients because they were missing information on death. 4001 did not possess obstructive coronary artery disease, equally dropped. Lastly, 5842 patients missed information on aerobic fitness, which in our case means that `Peak METs` was NaN (not a number).

After this, the dataset has shape $(13362, 260)$. We will keep the number of patients at 13362 and drop some features using the following manually applied heuristics:

- Features representing similar or identical information.
- Features having missing values for over 85% of patients.
- Practitioner feedback identified features outside the main study's objective, like patient ID or unessential dates.

These rules help reduce the number of features by 90%, from 260 to 25 ($\frac{235}{260} = 0.903$), through the elimination of 235 features. The final shape stays at $(13362, 25)$. In Chapter 4, feature selection will further reduce the number of features.

The core measure was death from any cause. This information was obtained through the Alberta Vital Status database [26]. It is important to note that this database is not related to the project that generated our dataset. The cohort entry date was defined by coronary angiography. The study followed the patients, monitoring for censoring or death, until March 31, 2017, with a minimum follow-up time of one year. Figure 2.2 shows the distribution of event times.

To distinguish the feature's name from ordinary words or other defined quantities, we write the feature's name in typewriter font. For example, instead of Peak METs, we write `Peak METs`. We conclude this section by demonstrating how the features can be effectively subdivided for improved organization and understanding:

$$
\begin{aligned}
\mathcal{S}_{\text{Exercise stress test}} = \{&\texttt{Peak METs, Peak DBP, Peak SBP, Peak HR,} \\
&\texttt{Resting DBP,Resting SBP, Resting HR}\} \\
\mathcal{S}_{\text{Comorbidities}} = \{&\texttt{Hypertension, Diabetes, Dyslipidemia,} \\
&\texttt{CHF, COPD, Family History, Current smoking,} \\
&\texttt{PVD, CEVD, Renal insufficiency, Malignancy}\} \\
\mathcal{S}_{\text{Clinical}} = \{&\texttt{Age, BMI, Number of vessels diseased,} \\
&\texttt{Management strategy for CAD, Sex,} \\
&\texttt{Indication for CA, LVEF}\} \\
S_{\text{inital}} = &\mathcal{S}_{\text{Exercise stress test}} \cup \mathcal{S}_{\text{Comorbidities}} \cup \mathcal{S}_{\text{Clinical}}
\end{aligned}
\tag{2.1}
$$

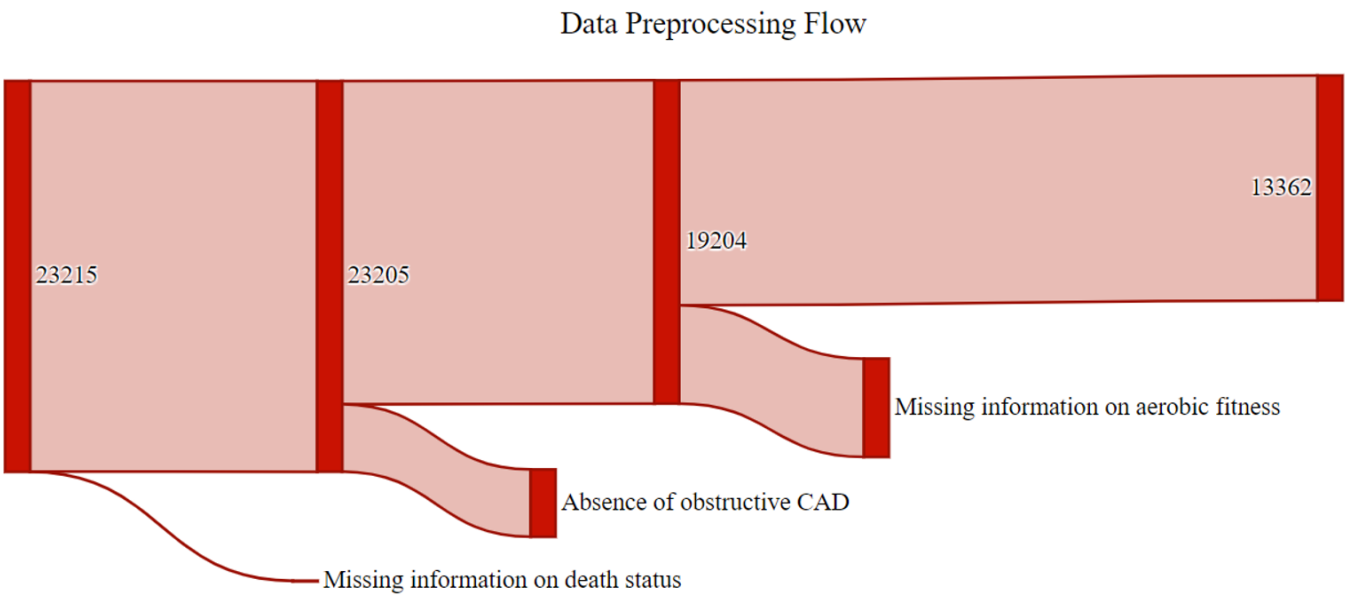

Figure 2.1: Representation of the main steps of patient filtering in a Sankey plot. The sizes of the vertical bars are proportional to the number of patients. CAD stands for coronary artery disease.

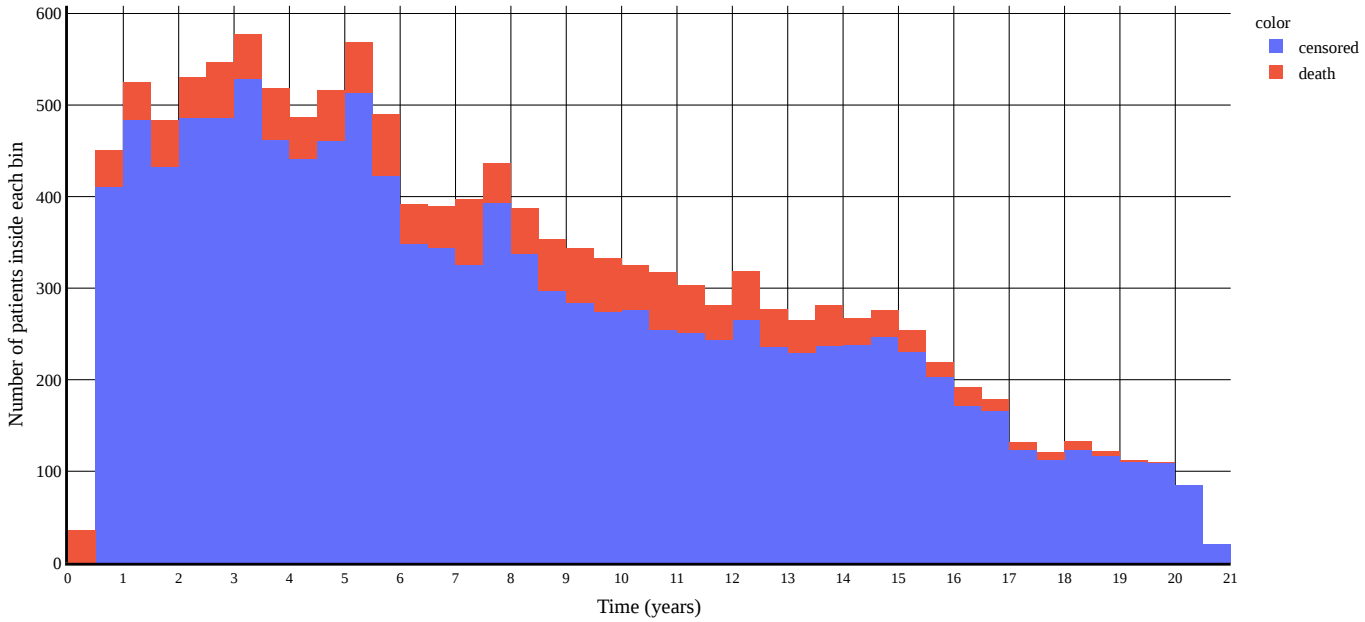

Figure 2.2: Histogram of `Time` stratified by `Event`, totalling 12% of deaths. Baseline of 13362 patients.

In the upcoming subsections, we will discuss the primary data pre-processing steps in detail. To begin with, in section 2.1, we will elaborate on how the outliers

present in the raw dataset were treated. Following that, in section 2.2, we will explain how the missing data was handled and how the scaling was performed. Lastly, we will conclude with section 2.3, which will outline the key empirical assumptions that will govern our entire analysis.

2.1 Outliers' treatment

In this work, a medical practitioner provided expected value ranges for each feature. If any feature took a value outside of the provided range in the "Values" column of Table A, it was considered to be NaN (not a number) and was ready to be imputed. For instance, if a patient's Peak HR was greater than 120, it was replaced with NaN. We did not remove any patients from the dataset due to the presence of outliers.

2.2 NaN imputation and scaling/encoding.

Before any model training, testing, or inference, we imputed and pre-processed the NaN features based on their respective data types. This pre-processing step was **always** a part of the model's pipeline and **not** a one-time process that stored the results. Whenever we needed to train or test a data subset, we initiated the pre-processing step again. Table A displays the data types of all 25 features. The two-step process we followed, separated by data type, is outlined objectively below:

- For categorical and binary features (in this order):
    1. (NaN imputation) Simple imputation by its most frequent mode. The code package used was [27].
    2. (Preprocessing) One-hot encoding, further dropping the first generated column in order to avoid multicollinearity [28]. The code package used was [27].
- Continuous features (in this order):
    1. (NaN imputation) Imputation was done using least-squares regression, taking into account the values of non-missing features, i.e., model-based imputation. The code package used was [29].
    2. (Preprocessing) Standard scaling, i.e. subtracting the mean value and then dividing by the standard deviation. The code package used was used [27].

2.3 Main assumptions

A logistic regression model was used to conduct a test, where `Event` was the dependent feature and `Time` was the independent one. The analysis generated a coefficient of -0.0321 multiplying `Time`, with a 95% CI of $(-0.085, 0.021)$. Furthermore, the likelihood ratio test produced a p-value of 0.2351, assuming the null hypothesis of the intercept-only model. Since the coefficient is close to zero, it suggests that changes in `Time` have a minimal impact on the likelihood of `Event`. Based on the logistic regression analysis, there is insufficient evidence to suggest a significant relationship between these variables. However, it is important to recognize that the absence of a significant relationship in this regression analysis does not conclusively prove independence. Instead, it suggests that the data do not provide strong evidence against the following assumed independence. Starting from Chapter 3, It is assumed the following independence of variables:

$$\texttt{Time} \perp\!\!\!\perp \texttt{Event} \tag{2.2}$$

In conclusion, here are the additional main assumptions about the data:

- $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ had its samples generated by the same process. They are independent and identically distributed, i.i.d.
- Non-informative censoring (section 3.1.2). Evidence at 2.2.
- Independent censoring within `Age` sub-groups (section 3.1.2).

Chapter 3

Background: Survival Analysis

Survival analysis, also known as time-to-event analysis, deals with the problem of modeling and analyzing data wherein the outcome is the time until an event takes place [5, 6]. This approach is widely used in both statistics and machine learning, with numerous applications in fields such as medicine [15, 16, 30], where it is used to determine which treatment has the greatest impact on survival, and customer churn [31], where it helps identify the factors that contribute to early service cancellation.

In this field, a significant challenge arises when, due to specific reasons, some event outcomes become unobservable after a certain point in time. These instances are referred to as *censored*. For example, when a dataset is created to observe the impact of treatment on survival over a period of 10 years, some participants may survive the entire period, and their data on death will not be generated. These participants are known as *censored*. On the other hand, some participants may die within the timeframe of the study, so they are not censored. A common mistake is to remove the censored participants, but this approach can lead to inaccurate estimations, often overestimating the risk of death [9].

The chapter is organized as follows. Section 3.1 introduces the formalization, mostly based on [32]. The survival models are exhibited in section 3.2, while the way to measure their performance is in section 3.3. Lastly, section 3.4 connects some topics previously mentioned to ones appearing here.

3.1 Problem Definition

Let T^* be the random variable indicating the event time, C^* the censoring time[1]. Then, in survival analysis, the relationship between these two random variables dictates the entire investigation. Note that if we only had seen T^*, the problem would

[1]We are going to treat only the right-censored case. Besides accounting for the vast majority of the applications, It fits the nature of our problem. Even so, adaptation to left-censoring or interval-censoring is relatively easy given that one knows the right-censored case.

not be suffering from censoring, degenerating into a standard regression problem. However, what we, in fact, see is the following two random variables:

$$\begin{aligned} T &= \min\{T^*, C^*\} \\ D &= \mathbb{1}\{T^* \leq C^*\} \end{aligned} \tag{3.1}$$

$\mathbb{1}$ is the indicator function, returning 1 when what is inside is true, 0 otherwise. T is the potentially right-censored duration. D is the event indicator, being equal to 1 when we observe T^* and to 0 when we observe C^*, thus hiding the true value of T^*. With these in hands, we can start to define the main objects in survival analysis:

$$\begin{aligned} \text{Survival Function} \Rightarrow S(t) &= P(T^* > t) \\ \text{Hazard Function} \Rightarrow h(t) &= \lim_{\delta t \to 0} \frac{\Pr(t < T^* \leq t + \delta t \mid T^* > t)}{\delta t} = \frac{-S'(t)}{S(t)} \\ \text{Cumulative Hazard Function (CHF)} \Rightarrow H(t) &= \int_0^t h(z)\mathrm{d}z = -\log S(t) \\ \text{Probability Density Function (PDF)} \Rightarrow f(t) &= P(T^* = t) \end{aligned} \tag{3.2}$$

Now, we can finally recognize survival analysis' **main challenge**: receiving T and D, and needing $S(t) = P(T^* > t)$ to infer useful statements. For that, a lot of the models need an explicit quantity to optimize. In other words, we need the likelihood function. We shall first develop an expression for $\mathrm{P}(T = t, D = d)$; let $f_{C^*}(t) = \mathrm{P}(C^* = t)$ and $S_{C^*}(t) = \mathrm{P}(C^* > t)$, then:

$$\begin{aligned} \mathrm{P}(T = t, D = d) &= \mathrm{P}(T^* = t, C^* \geq t)^d \, \mathrm{P}(T^* > t, C^* = t)^{1-d} \\ &= [\mathrm{P}(T^* = t) \, \mathrm{P}(C^* \geq t)]^d \, [\mathrm{P}(T^* > t) \, \mathrm{P}(C^* = t)]^{1-d} \\ &= [f(t) \, (S_{C^*}(t) + f_{C^*}(t))]^d \, [S(t) f_{C^*}(t)]^{1-d} \\ &= \left[f(t)^d S(t)^{1-d}\right] \left[f_{C^*}(t)^{1-d} \, (S_{C^*}(t) + f_{C^*}(t))^d\right] \end{aligned} \tag{3.3}$$

Assuming $f_{C^*}(t)$ and $f(t)$ does not share any parameter. We can build a loss function that is entirely constituted by the distribution of event times, with each data point contributing $L_i = f(t_i)^{d_i} \, S(t_i)^{1-d_i}$, then:

$$\text{Loss} = -\frac{1}{n} \sum_{i=1}^{n} \left(d_i \log\left[f(t_i \mid \mathbf{x}_i)\right] + (1 - d_i) \log\left[S(t_i \mid \mathbf{x}_i)\right]\right) \tag{3.4}$$

3.1.1 Restricted mean survival time (RMST)

The restricted mean survival time (RMST) [33, 34] of T^* is defined as the mean of $X = \min(T^*, t^*)$, where $t^* > 0$ is a previously chosen time upper bound[2]. We shall prove that It is equal to the integral of the survival function; let $f_X(x) = P(X = x)$ be X's probability density, then:

$$\begin{aligned}
E[X] = E[\min(T^*, t^*)] &= \int_0^{t^*} x f_X(x) dx \\
&= -\int_0^{t^*} x P'(X > x) dx \\
&= -[xP(X > x)]_0^{t^*} + \int_0^{t^*} P(X > x) dx, \text{ (integration by parts)} \\
&= -t^* \cdot P(\min(T^*, t^*) > t^*) + 0 \cdot P(\min(T^*, t^*) > 0) + \int_0^{t^*} P(X > x) dx \\
&= -0 + 0 + \int_0^{t^*} P(\min(T^*, t^*) > x) dx \\
&= \int_0^{t^*} P(T^* > x) dx \\
&= \int_0^{t^*} S(x) dx
\end{aligned} \tag{3.5}$$

Notice that RMST is **not** the overall mean survival additional to the patient's age because the experiment has a maximum follow-up time of $T^*_{max} > 0$. Therefore, the model cannot infer survival greater than this number. The code used for RMST was [35].

[2] In most cases, t^* is the maximum follow-up time of the experiment: 21 years in our dataset. However, it is also possible to use narrower time windows, e.g., comparing the "up to 5 years" mean survival, in this case, $t^* = 5$ years.

3.1.2 Censoring assumptions

There are three types of censoring assumptions. Their logical relations are presented in Figure 3.1.

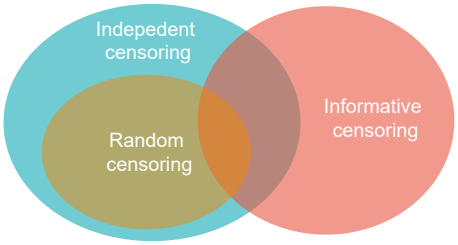

Figure 3.1: Logical relation between censoring assumptions.

- **Random** (vs. non-random) censoring: let the hazard function of the population that was censored ($D = 0$) be $h_{D=0}(t)$. Similarly, for the not censored: $h_{D=1}(t)$. Then, random censoring means $h_{D=0}(t) = h_{D=1}(t)$.
- **Independent** (vs. non-independent) censoring: random censoring holds but only conditional on covariates.
- **Non-informative** (vs. informative) censoring: T provide no information about C, and vice-versa.

3.2 Models

3.2.1 Nonparametric

Kaplan-Meier

The Kaplan-Meier stands as the primary tool in survival analysis [7]. It is commonly utilized to check the accuracy of population estimations by comparing the Kaplan-Meier from a clusterized population against the survival curve generated by a model. Essentially, It aims to estimate the marginal survival function $S(t) = P(T^* > t)$ while having samples from T and D. Let our training data be $(t_1, d_1), (t_2, d_2), \ldots, (t_N, d_N)$, and their unique times of death $k_1, k_2, \cdots, k_U$ (evidently $U \leq N$). We also define the number of deaths at time k_i as $\mathrm{N}_i^{\mathrm{death}}$; and the quantity of subjects still alive at k_i as $\mathrm{N}_i^{\mathrm{risk}}$:

$$\begin{aligned} \mathrm{N}_i^{\mathrm{death}} &= \sum_{j=1}^{N} \mathbb{1}\left\{t_j = k_i\right\} d_j \\ \mathrm{N}_i^{\mathrm{risk}} &= \sum_{j=1}^{N} \mathbb{1}\left\{t_j \geq k_i\right\} \end{aligned} \tag{3.6}$$

At last:

$$\widehat{S}_{\mathrm{KM}}(t) = \prod_{i=1}^{U} \left(1 - \frac{\mathrm{N}_i^{\mathrm{death}}}{\mathrm{N}_i^{\mathrm{risk}}}\right)^{\mathbb{1}\{k_i \leq t\}} \tag{3.7}$$

This model necessitates the random censoring (3.1.2) assumption to be valid. Even so, we can always apply the Kaplan-Meier to progressively stratified subpopulations inside which random censoring holds. In fact, this stratification can be generalized to kernel-based methods [36, 37]. The code implementation used was [35].

Survival Tree

Survival Tree is a method that extends the Tree-based approach to survival analysis. It employs the log-rank splitting technique to evaluate the quality of each split. The population is divided based on this rule until it meets its predefined parameters, such as minimum leaf samples, maximum depth, and minimum samples required for a split. Each leaf node contains a subset of the population that can generate a survival function using the Kaplan-Meier estimator. When we make an inference, we start at the root node and move through the trained thresholds until we reach a leaf node. Finally, we return the Kaplan-Meier function of that leaf node as the survival function for the data point. Finally, this leaf node's Kaplan-Meier is returned as the data point's survival function. The code implementation used was [38].

Random Survival Forest (RSF)

RSF trains a set of Survival Trees based on the same principle as the standard random forest algorithm [39]. It follows the steps outlined in [40]:

1. From the original dataset $\mathcal{D}$, RSF draws N bootstrap samples.
2. Build a Survival Tree, exactly like in section 3.2.1, for each bootstrap sample.
3. Compute a Cumulative Hazard Function (CHF) (details in section 3.1) for each of the N samples. Next, average them, obtaining the *ensemble* CHF.

The implementation used was from [38]. However, it is important to carefully select hyperparameters, as even with our dataset of shape $(13362, 25)$, it was the most time-consuming algorithm among those mentioned in this study.

Logistic Hazard

Logistic Hazard is a discrete-time, neural network-based method that directly models the conditional hazard function $h(t|\mathbf{x}_i)$ [41]. For that, It processes the features $\mathbf{x}$ with

a feed-forward neural network resulting in a vector $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})]$, where m is the index indicating the discrete time-points, i.e. $\tau_j \in \mathbb{R}$ for $j \in \{1, \ldots, m\}$, for instance $\tau_{10} = 5$ years. Knowing that the discrete hazards $h(\tau_j \mid \mathbf{x}_i) \in [0, 1]$, It's easy to transform $\phi(\mathbf{x})$ into $h(\tau_j \mid \mathbf{x}_i)$:

$$h(\tau_j \mid \mathbf{x}_i) = \frac{1}{1 + \exp[-\phi_j(\mathbf{x})]} \tag{3.8}$$

Thus, we now only need a loss function, like in any neural network training. For that, there is the discretized version of the loss function shown at 3.4. Then, It is just a common application of any gradient descent algorithm. Figure 3.2 exhibits the model's overall functioning.

The `pycox` package was used as the implementation [32]. This model is also called Nnet-survival [42].

LifeTableBaseline

We have created a simple model called *LifeTableBaseline*, which serves as a baseline for comparison. It produces the survival function of a patient based on their age, using data from the Canadian Census [43] about life expectancy. Unlike other models, it does not use the dataset for training, making it the simplest model that still provides a complete survival function for each patient. We chose to use it as a benchmark for other models since any survival model that performs worse than *LifeTableBaseline* would be considered inadequate.

3.2.2 Semiparametric

Cox Proportional Hazards (CPH)

The Cox Proportional Hazards (CPH) model is a well-known and widely used method in survival analysis. It is often regarded as the go-to model after the Kaplan-Meier method due to its ease of application and interpretation. The CPH model can be easily adapted to incorporate time-dependent features and competing risks, making it a versatile tool for analyzing medical data. Concretely, the model imposes that, with $\mathbf{x}_i$ being i^{th}'s feature vector:

$$h(t|\mathbf{x}_i) = h_0(t)e^{\sum_{i=1}^{p} \beta_i x_i} \tag{3.9}$$

$h_0(t)$ is the baseline hazard that purposefully does **not** involves the features. It is an unspecified function that makes CPH a semiparametric model. All $\beta_i \in \mathbb{R}$ are learnable parameters. Importantly, the quantity often reported is the **Hazard**

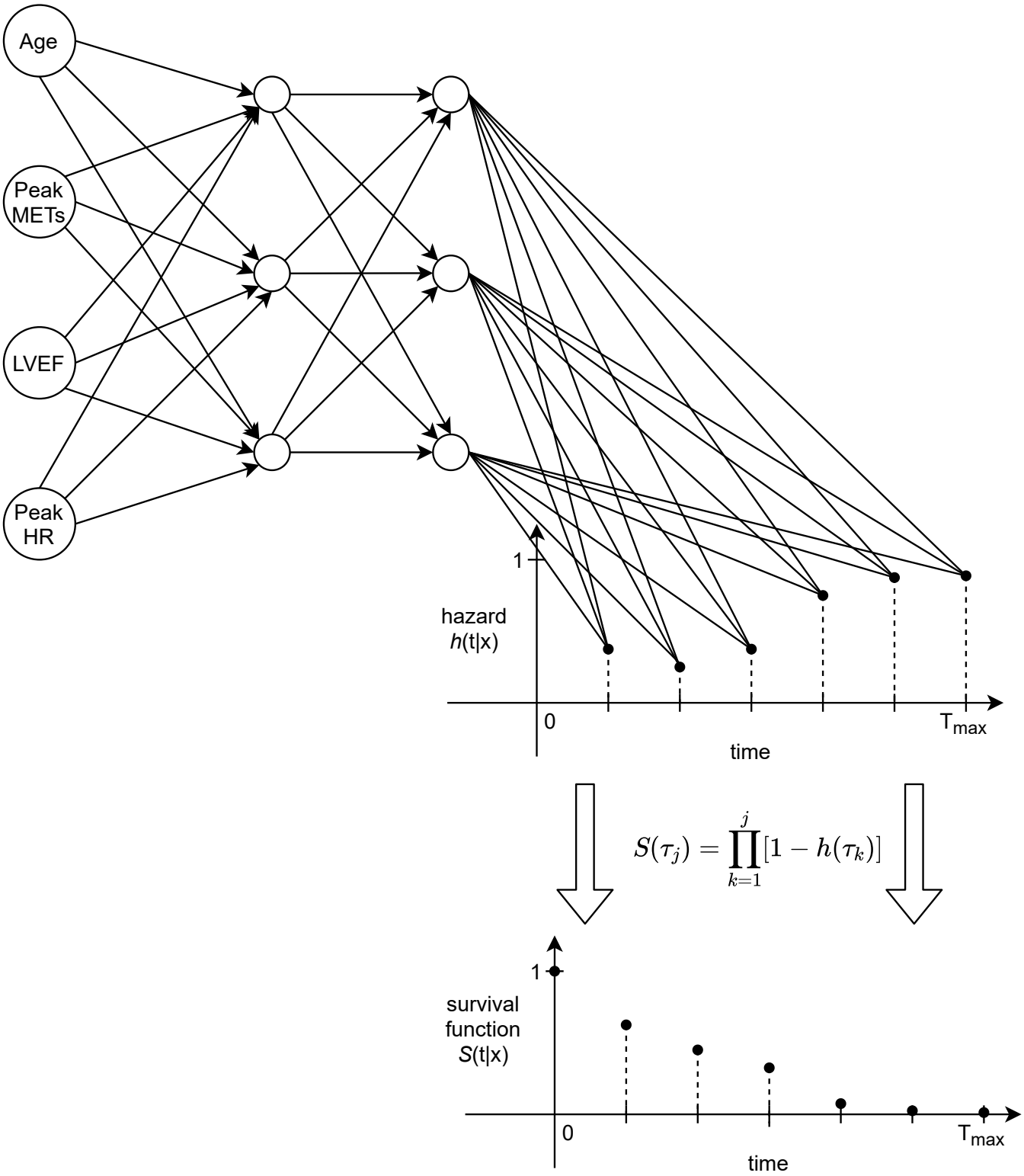

Figure 3.2: A diagram shows LogisticHazard's function with four input features and a 3x3 neural network processing them.

Ratio for a feature. It is defined for a j^{th} feature with its associated coefficient β_j as:

$$\widehat{HR} = e^{\hat{\beta}_j} \tag{3.10}$$

For model training, the parameters are estimated via maximum likelihood. We first estimate β without knowing h_0 using any gradient descent algorithm for:

$$\widehat{\beta} = \arg\min_{\beta \in \mathbb{R}^k} \sum_{i=1}^{n} d_i \left[-\beta^\top x_i + \log \left(\sum_{j=1}^{n} \mathbf{1} \left\{ t_j \geq t_i \right\} e^{\left(\beta^\top x_j\right)} \right) \right] \tag{3.11}$$

Finally, h_0 is found via the Breslow estimator [44], using the newly obtained $\widehat{\beta}$. The code implementation used for this two-step process was [35].

ROYSTON e PARMAR argues that the Hazard Ratio does not provide a clear interpretation of the work's results, with a better option being the RMST. The main reason is that It is not always clear the precise effect on survival if we increase a feature's value when looking at its associated HR. We cannot know from HR alone how many more months/years a patient will have. Worst than that, the effect on mean survival depends on the estimated cumulative hazard H_0, creating a different outcome for different datasets. Contrary to HR, RMST has time units, is directly interpretable, and can be adapted to every other model even if It does not obey the proportional hazards assumption. In any case, HR can be easily translated to RMST. On account of these reasons, we will avoid HR and use RMST instead.

DeepSurv

DeepSurv is an advanced extension of the Cox Proportional Hazards model [11] for assessing non-linear risks. It is inspired by a neural network solution introduced in the '90s [45]. The model uses a configurable neural network to summarize all the features of a patient, which then outputs a risk value. In simpler terms, instead of using the hazard function with a linear combination of features $\hat{h}_\beta(x) = \beta^T x$, DeepSurv replaces $\beta^T x$ with the output of a multi-layer perceptron: $\hat{h}_\theta(x)$, where θ represents the weights of the network whose output is a single node. For training, the loss function is the negative log partial likelihood of equation 3.11, replacing $\beta^T x$ with $\hat{h}_\theta(x)$. The `pycox` package implementation was used [32].

3.3 Metrics

It's important to measure the performance of models. Machine learning offers various metrics based on the problem domain. However, there's no one-size-fits-all

calculation that can encompass every aspect in a single number.

For survival analysis, the metrics can be divided into two groups: concordance and calibration. In detail, all the used metrics are random variables from an algorithm (model) to a real number or another real function, given data. Let S be our set of models[3], i.e. $S = \{\text{CPH}, \text{Survival Tree}, \ldots\}$, and $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$ the dataset. A metric m is a random variable $m : S \to \mathbb{R}$ or $m : S \to \mathcal{F}([0, T_{max}], \mathbb{R})$. $\mathcal{F}([0, T_{max}], \mathbb{R})$ is the set of real-valued functions from the closed interval $[0, T_{max}]$, where T_{max} is the maximum follow-up time in $\mathcal{D}$.

3.3.1 Concordance

Time-dependent C-Index (C^{td})

The time-dependent discrimination index [46] is a modified version of Harrell's C-index [47]. It is a nonparametric statistic, ranging from 0 to 1, that measures the probability of two randomly chosen **comparable** patients of being **concordant** according to model-given risks. Like many other metrics, It returns a number given two objects: a trained survival model and a test set. Let's explain the meanings of **comparable** and **concordant**:

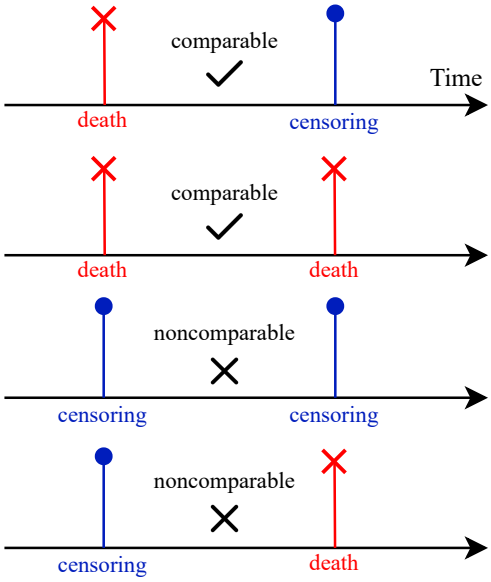

Figure 3.3: Comparable pairs used to calculate the metrics C^{td} and $AUC(t)$.

- In order for a patient's pair to be **comparable**, as shown in Figure 3.3, It needs to have the patient with the smaller event time to have suffered death, i.e. `Event` $= 1$. This is intuitive, as this patient would give us no hint if he died earlier or later than the other patient in case of being censored.

- For a **comparable** pair to be **concordant** according to a model, its patient with the smaller event time needs to be given a higher risk than the patient with the bigger event time. A good model has to assign higher risks to the patient who died earlier. As models return survival functions, the patient's risk is $1 - S(T)$, the probability of *not* living past T. Of course, the risk comparison has to be standardized by calculating it at the same T for *both* patients.

[3]For C-Index and brier score' estimations, the "model" includes inside of it a hyper-parameter search via nested cross-validation. As for AUC(t), because of prohibitory computation time, the "model" does not include a hyper-parameter search. The parameters are fixed.

Concretely, let's say we have a pair of patients with $p_1 = (5 \text{ years, death})$ and $p_2 = (10 \text{ years, censored})$. Our testing model M returns $S_M(t = 5 \text{ years}|p_1) = 0.6$, $S_M(t = 10 \text{ years}|p_1) = 0.3$, $S_M(t = 5 \text{ years}|p_2) = 0.8$ and $S_M(t = 10 \text{ years}|p_2) = 0.6$. Are they **concordant**? Comparing both at $T = 5$ years, $\text{Risk}_{p_1} = 1 - S_M(t = 5 \text{ years}|p_1) = 1 - 0.6 = 0.4$; and $\text{Risk}_{p_2} = 1 - S_M(t = 5 \text{ years}|p_2) = 1 - 0.8 = 0.2$. So, $\text{Risk}_{p_1} > \text{Risk}_{p_2}$. The pair p_1 and p_2 are, in fact, **concordant** according to the model M.

Finally, we repeat this risk ordering test for all **comparable** pairs and calculate the **ratio** of the rightly ordered to the total number, at last acquiring our complete time-dependant C-Index. Note that a C-index of 1 represents a perfect concordance, while a C-index of 0.5 indicates the discrimination was utterly random.

C^{td}'s confidence interval was estimated via nested cross-validation. The code implementation used was [32], plus some adaptations.

Cumulative/Dynamic Area Under the Curve ($AUC(t)$)

The cumulative/dynamic area under the curve or $AUC(t)$ measures the change of the model's performance through time [48]. It returns a score for each desired time t, where t can take values from the start to the end of the follow-up period[4], e.g., $t = 5$ years or $t = 10$ years. At each time t, It calculates its score by estimating the probability that the model generates a **concordant** ordering, via their predicted survival functions, for two randomly and **comparable** pairs of patients. **Concordant** and **comparable** exactly like the C-Index, read subsection 3.3.1 for the explanation. However, there is one additional requirement for the pair to be **comparable** for the $AUC(t)$: one patient needs to have their event with a time smaller than t and the second patient after t. For instance, let's say we want $AUC(t = 5 \text{ years})$. Are the two patients $p_1 = (2 \text{ years, death})$ and $p_2 = (10 \text{ years, censored})$ compatible for AUC purposes? Yes, because they respect the C-Index's **comparable** requirement; also, p_1's 2 years is smaller than 5 years, and p_2's 10 years is greater than 5 years.

Reiterating that $AUC(t)$ is of the form $m : S \to \mathcal{F}([0, T_{max}], \mathbb{R})$, as we explained in the introduction of this section 3.3. Evidently, $AUC(t)$ can be represented with a 2D scatter plot[5] $AUC(t) \times t$. The confidence intervals were generated using bootstrap re-sampling. The code implementation used was [38].

[4]For our case, the 21 years was divided into 300 subdivisions. So we have 300 different t's to estimate 300 different numbers from 0 to 1. The resolution ended up being around 26 days ($\frac{21}{300} \cdot 365 \approx 26$).

[5]Clearly, for similar t, $AUC(t)$ will tend to be close. Then, a simple 2D line plot is feasible and better to visualize.

3.3.2 Calibration

Brier Score (BS(t))

Calibration is a probabilistic concept that measures how close the risk estimate is to the actual risk. A way of expressing this in survival analysis is via the Brier score [49]. Let $\widehat{S}_{p_i}(t|\mathbf{x}_i)$ be the survival function estimate for the patient p_i with feature vector $\mathbf{x}_i$. And $S_{p_i}(t) = P(T_i^* > t)$ be the true survival function for patient i, i.e. what we are trying to estimate. It would be useful if we could calculate the quantity $\text{MSE}(t) = \frac{1}{n}\sum_{i=1}^{n}\left[S_{p_i}(t) - \widehat{S}_{p_i}(t|\mathbf{x}_i)\right]^2$. However, $S_{p_i}(t)$ cannot be known outside simulations because we only receive the event times T_i. An alternative out of this deadlock is, first, to use the information that we get from the dataset, namely, T_i and D_i; second, build a formula having as its expected value the quantity $\text{MSE}(t)$. GRAF *et al.* supplied the answer. Define $G_i(t) = \text{P}\left(C_i^* > t\right) > 0$, then:

$$\text{BS}(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\left[\widehat{S}_{p_i}(t|\mathbf{x}_i)\right]^2 \mathbb{1}\left\{T_i \leq t, D_i = 1\right\}}{G_i\left(T_i-\right)} + \frac{\left[1-\widehat{S}_{p_i}(t|\mathbf{x}_i)\right]^2 \mathbb{1}\left\{T_i > t\right\}}{G_i(t)}\right] \tag{3.12}$$

A straightforward calculation shows that $\text{BS}(t)$ possess our second requirement:

$$\mathbb{E}\left[\text{BS}(t)\right] = \text{MSE}(t) + \frac{1}{n}\sum_{i=1}^{n} S_{p_i}(t)\left[1 - S_{p_i}(t)\right] \tag{3.13}$$

Then, a small value of $\text{BS}(t)$ indicates a better calibration, i.e., a value close to zero indicates better performance. Equation 3.12 is sometimes called IPCW Brier score to differentiate from the uncensored brier score. Henceforth we will always consider 3.12 to be the version used. Some caveats about this metric are presented at [50], while the code implementation used was [32].

Integrated Brier score (IBS)

IBS is a simple summarization of the $BS(t)$ to a single number via integration. Let $T_{max}^* > 0$ and t_0 be the maximum and minimum times that $BS(t)$ is defined, then:

$$\text{IBS} = \frac{1}{T_{max}^* - t_0}\int_{t_0}^{t_{\max}} \text{BS}(t)dt \tag{3.14}$$

3.4 Conclusion

Henceforth, this chapter's T and D correspond to `Time` and `Event` from our dataset. The label is bidimensional: $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$, $\boldsymbol{y}_i \in \mathbb{R}^2$, with $\boldsymbol{y}_i = (t_i, d_i)$ e.g. $\boldsymbol{y}_{50} =$

$(t_{50}, d_{50}) = (4.56 \text{ years}, 0) = (4.56 \text{ years}, \text{censored})$. As for RMST, the upper limit of its integral, t^*, will be the maximum value of `Time`, 21 years. Evidently, all models and metrics presented here will be used at some stage. It's important to mention the purposeful pre-selection of a diversified set of models, carrying distinct inductive biases.

Chapter 4

Feature Selection

4.1 Introduction

Feature selection is an approach to reducing data dimensionality by selecting a subset of important features for model building [51–53].

Consider a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ that has $\boldsymbol{x}_i \in \mathbb{R}^d$, i.e. d features, and $y_i \in \mathbb{R}$ as labels. $\{\boldsymbol{x}_i\}_{i=1}^n$ can be represented as a matrix $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n \times d}$, similarly $\boldsymbol{y} := [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$. Each feature is denoted with a superscript $\boldsymbol{x}_i = \left[x_i^1, \ldots, x_i^d\right]$, where $x_i^j \in \mathbb{R}$. Feature selection's objective is a mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$ where $\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{z} \in \mathbb{R}^p$, with $\{\boldsymbol{z}^j\}_{j=1}^p \subseteq \{\boldsymbol{x}^j\}_{j=1}^d$ and $p \leq d$. For instance, if we end up selecting the first two features from $\boldsymbol{x}_i$, $\boldsymbol{z}_i = [x_i^1, x_i^2]$. Therefore, It forms a new data representation which can be written as $\boldsymbol{Z} := [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] \in \mathbb{R}^{n \times p}$ and used further down the pipeline. Reducing the number of features can lead to a decrease in model complexity, which ultimately helps to improve the model's interpretability [51]. Additionally, it can speed up the model's training, selection, and inference by reducing the number of features to process. Most importantly, reducing the number of features can reduce the risk of overfitting [54–56]. Feature selection is a common practice in various fields such as microarray data analysis [57], bioinformatics [58–60], and network intrusion detection [61].

In this chapter, the algorithms used are displayed in section 4.2. The results obtained are presented in section 4.3, and the final subset of features is discussed in section 4.4.

4.2 Algorithms

Feature selection algorithms can be classified into three categories: filter-based, wrapper-based, and embedded methods [51, 57, 62–64]. Filter-based methods do not involve any model to assist in feature selection. On the other hand, wrapper-

based methods use established models to evaluate the performance of feature subsets by training them and assessing their quality using a performance metric. In the case of embedded methods, the model itself includes a pre-built form of feature selection, such as LASSO [65] or Elastic Net [66]. These methods incorporate a penalty inside the loss function that sets the contribution of unimportant features to zero.

Based on empirical evidence [52], the filter-based option is faster but performs worse on other performance metrics. However, since the computation time for the other two approaches is feasible, we prefer to eliminate filter-based methods. It's worth noting that there are few off-the-shelf embedded methods available for survival analysis, with Cox-LASSO being the notable exception [65]. Therefore, we prefer using wrapper-based methods as they provide us the liberty to mobilize any high-capacity model in section 3.2, and have good performance.

We have utilized the following wrapper-based techniques for our analysis: permutation importance associated with Random Survival Forest (RSF) (section 3.2.1) and Logistic Hazard (section 3.2.1) models; sequential forward and backward search associated with these same models plus the K-Means clustering algorithm. We will provide further explanations for both techniques in the following sections.

4.2.1 Permutation Importance

Permutation importance (PM) [67] is a method used to determine the importance of the features in a dataset according to a model. PM first divides the dataset into a training and test set. Then, the model is trained using the training set and tested using the intact test set, returning a performance metric such as C^{td}. Next, PM shuffles a specific feature on the test set and evaluates the same trained model using this partially shuffled test set. If this newly returned metric is smaller than the original metric, PM outputs this decreased performance difference, and we interpret that the shuffled feature was important after all. However, if this difference is near zero, shuffling the feature does not matter; consequently, we conclude it is unimportant. Finally, we select only the feature found to be important.

The model-agnostic version of this algorithm was the version used, similar to [68]. Detailed pseudo-code is in Algorithm 1, the code implementation used was [27]. The confidence intervals were built via bootstrap re-sampling.

Algorithm 1: Model-agnostic Permutation Importance [68, 69]

Input: Trained model f, feature matrix $X \in \mathbb{R}^{n \times d}$, labels $y \in \mathbb{R}^n$, error measure $L(y, f(X)) \in \mathbb{R}^1$, number of iterations K, training/test fraction splitting q $(0 < q < 1)$

Output: Vector of feature importances $FI \in \mathbb{R}^d$

Split X into training $X_{train} \in \mathbb{R}^{\lfloor qn \rfloor \times d}, y_{train} \in \mathbb{R}^{\lfloor qn \rfloor}$ and test set $X_{test} \in \mathbb{R}^{(n-\lfloor qn \rfloor) \times d}, y_{test} \in \mathbb{R}^{n-\lfloor qn \rfloor}$;

Train a model f using X_{train}, y_{train};

Estimate the original model error $e^{orig} = L(y_{train}, f(X_{train}))$;

`/* L can be C-Index or IBS, and f a survival model. */`

`/* Note that the model f is never retrained. */`

for $j = 1, 2, ..., d$ **do**

  for $k = 1, 2, ..., K$ **do**

    Create $X_{k,j}^{perm}$ from X_{test} by permuting feature j (column j of X_{test});

    Estimate a new error $e_{k,j}^{perm} = L(y_{test}, f(X_{k,j}^{perm}))$;

  end

  Calculate the permutation feature importance $FI_j = e^{orig} - \frac{1}{K} \sum_{k=1}^{K} e_{k,j}^{perm}$;

end

Return the vector $FI \in \mathbb{R}^d$;

4.2.2 Sequential Forward/Backward Search

Sequential Backward Selection (SBS) [70] takes the whole set of features as input. Then, It evaluates how good the model's performance would be in case of removing one feature. The feature that most helps the model performance by being absent is permanently deleted. Subsequently, SBS starts again, now with this smaller set. It keeps deleting until a pre-determined number is reached or up until the subset is empty. Finally, It returns the subset of features that generated the best performance or the first subset that plateaued the performance.

Similar to SBS, Sequential Forward Selection (SFS) [71] starts with a subset containing zero features, then checks which feature most contributes to performance when added. The winner is added to the subset of zero features, making it a one-member set. Afterward, It again tests which feature to add, but this second feature is now added to the one-member set. The winner of this second competition is added, forming a two-member set. This process continues until a pre-determined number of features or a performance plateau is reached, returning this last subset.

The code implementation used was [72] together with some adaptations, and standard deviations were calculated using 5-fold cross-validation. When adding or

deleting a feature, the implementation always manipulates the original features and never its unfolded categories in the case of being categorical. For example, the feature "`Indication for CA`" was added and deleted just like that, and **not** as its constituting categories: "Stable angina", "Unstable angina" and "Myocardial infarction".

K-Means associated with SFS and SBS

K-Means is an unsupervised clusterization algorithm. Therefore, It cannot, on its own, generate a survival function for any patient. It can only give labels to each point in $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{n}$. It does not use the information on the $\boldsymbol{y}_i$s. So how can we use it with SFS and SBS? To use K-Means as a survival analysis model, we apply the Kaplan-Meier estimator (3.2.1) for each clusterized population. In other words, after having trained K-Means in a completely unsupervised way, we generate predictions by returning the Kaplan-Meier of the clusterized population inside which a patient resides.

4.3 Results

We start with the dataset of shape $(13362, 25)$. This shape was attained in chapter 2; check 2.1 for details. Our objective is to reduce S_{inital}, $|\mathcal{S}_{initial}| = 25$.

Hereafter, models are applied and interpreted. The performance metric associated with algorithms used is the time-dependent C-Index 3.3.1. Figures 4.1 and 4.2 provide the results generated by the applications of algorithms presented in section 4.2. Permutation Importance (plot (B) in Figure 4.1) scored `Peak METs` and `Age` as the most important, followed by `Peak HR`, `Resting HR` and `Resting SBP`. These last three are less than a fourth in importance from the first two. The sequential forward search (plot (C) and (D) in Figure 4.1) show that, for both models, most of the C-Index boost was attained with `Peak METs` and `Age`. Some further increases were obtained with `Sex`, `Current smoking`, and `Malignancy`. Then, It reaches a C-Index plateau of about 0.75 for Logistic Hazard and 0.74 for RSF. Further, adding any other features did not result in any gain.

Lastly, results from K-Means associated with SFS and SBS are presented in Figure 4.2. Manifestly, after 3 clusters, no improvement was seen for both. The jump in performance from 2 clusters to 3, followed by no relevant further increase for 4 and 5, openly shows this. In particular, SBS and SFS plateau at two features. The two selected were `Peak METs` and `Age`. Pictorially, the Kaplan-Meier plots are shown at (A) in figure 4.1 exhibit the three subpopulations' Kaplan-Meiers generated by applying K-Means with $k = 3$. The algorithm was applied to the dataset containing

`Peak METs` and `Age`.

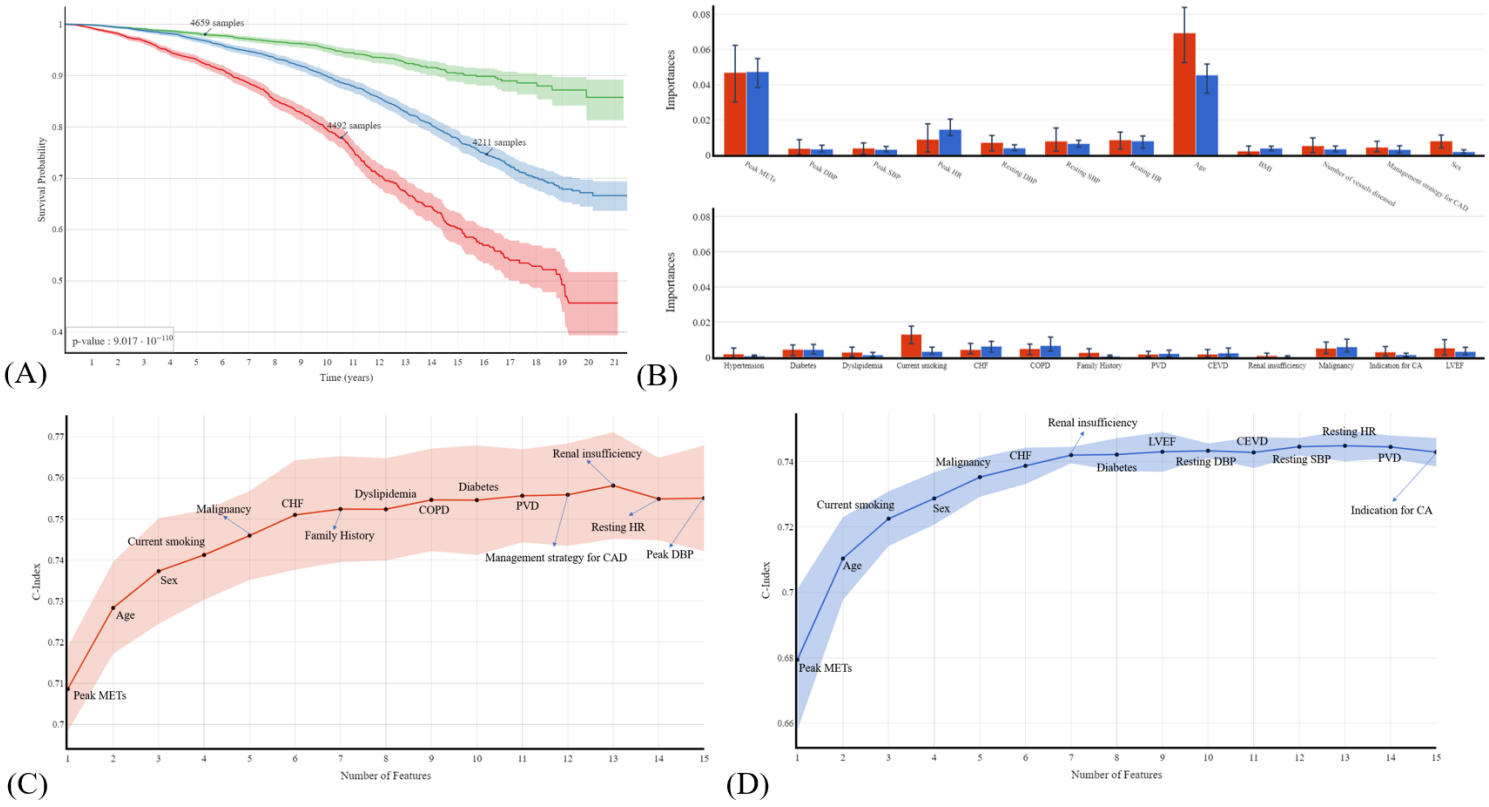

Figure 4.1: Results from feature selection process when applying (A) K-Means (k=3) exclusively with `Peak METs` and `Age`; (B) Permutation Importance with Random Survival Forest (blue) and Logistic Hazard (red); (C) SFS with Logistic Hazard; and (D) SFS with Random Survival Forest. For details about feature names, check table A

.

4.4 Conclusion

`Peak METs` and `Age` were the only consistently important feature across models and methods. Consequently, the subset of selected features is $\mathcal{S}_{final}$={`Peak METs, Age`}, $|\mathcal{S}_{final}| = 2$. Notice that $\frac{|\mathcal{S}_{final}|}{|\mathcal{S}_{initial}|} = \frac{2}{25} = 0.08$, a 92.0% reduction. Remember that the raw dataset came with 260 features, we attained $\frac{|\mathcal{S}_{final}|}{260} = \frac{2}{260} = 0.0076$, a 99.2% overall reduction.

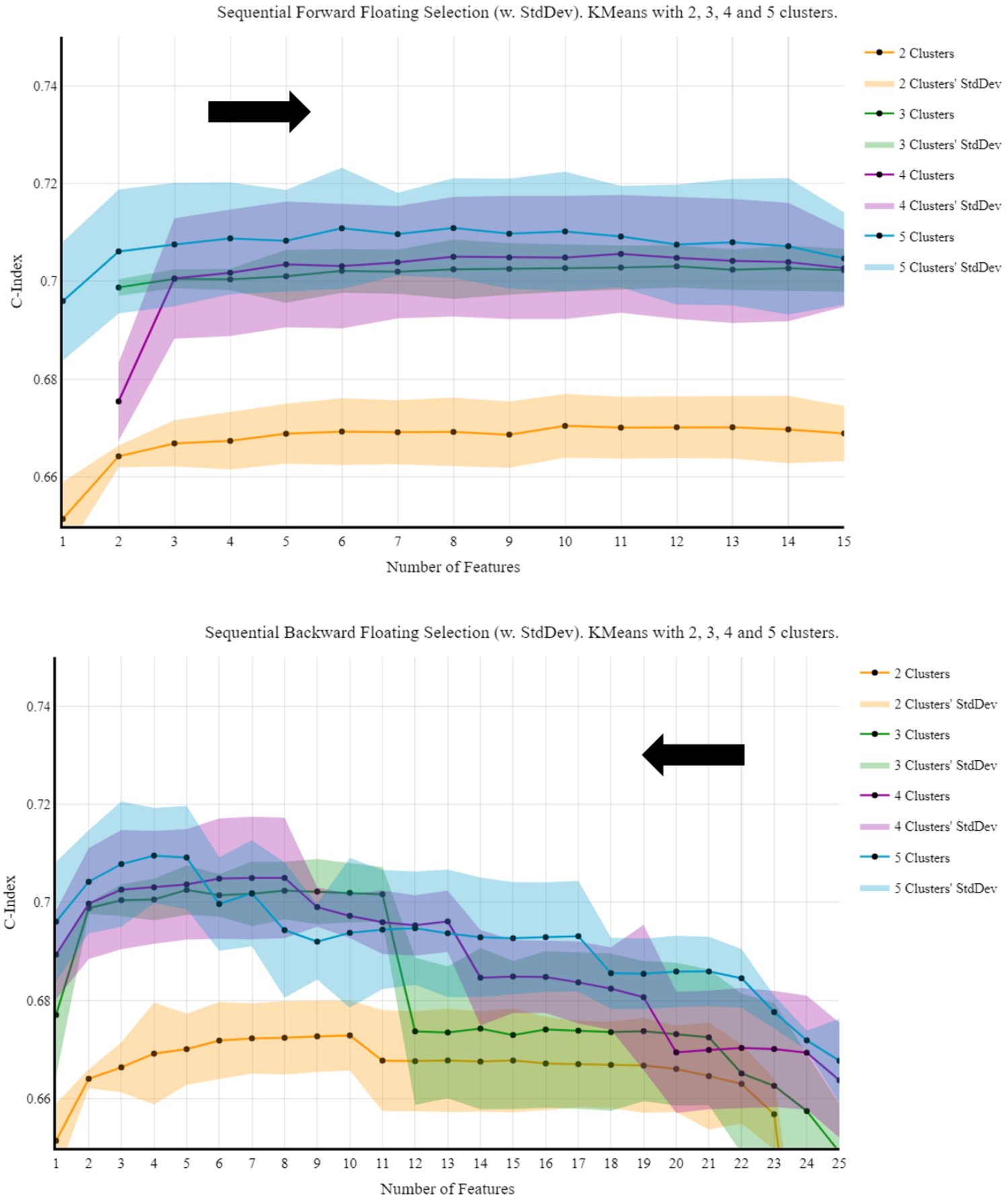

Figure 4.2: Results from K-Means associated with SBS and SFS. Colours indicate the number of clusters.

Chapter 5

Proposed Model

We aim to build an interpretable survival model to predict a patient's death by any cause. The feature selection procedure, chapter 4, greatly facilitated our task by having selected `Age` and `Peak METs`. Now, every algorithmic step is faster to run and interpret.

Currently, the dataset's shape is $(13362, 2)$. We will follow a standard model selection process. In section 5.1, a model benchmark possessing performance estimation using the C-Index and IBS is provided. Then, the results are pondered, and a model is chosen. Next, in section 5.2, the inner workings of the selected model are revealed by utilizing some tools described in chapter 3.

5.1 Model selection

The models selected for the benchmark are the following: Logistic Hazard 3.2.1, Survival Tree 3.2.1, Random Survival Forest (RSF) 3.2.1, Cox Proportional Hazards (CPH) 3.2.2, DeepSurv 3.2.2, and *LifeTableBaseline* 3.2.1. Notice that they are very different models, with completely different inductive biases and inner workings. The neural network-based methods employed almost flat networks with a maximum depth of three. The estimation process for C-Index and IBS is the nested cross-validation procedure as already mentioned in 3.3.1 and 3.3.2, all searched hyperparameters are exposed in Table C.1.

The results are shown in Figure 5.1. All models reached a C-Index performance higher than *LifeTableBaseline*. This is a positive fact, as all of them are better than the baseline. Next, their IBS is very similar. Even the slight difference in Logistic Hazard is relatively small, from 0.0980 to 0.0103. Therefore, we will not use IBS to guide the selection decision.

Logistic Hazard, CPH, and DeepSurv attained similar performance around 0.73. RSF got 0.725, and Survival Tree obtained an average of 0.71. We eliminate the Logistic Hazard and DeepSurv models because they are harder to interpret. Even

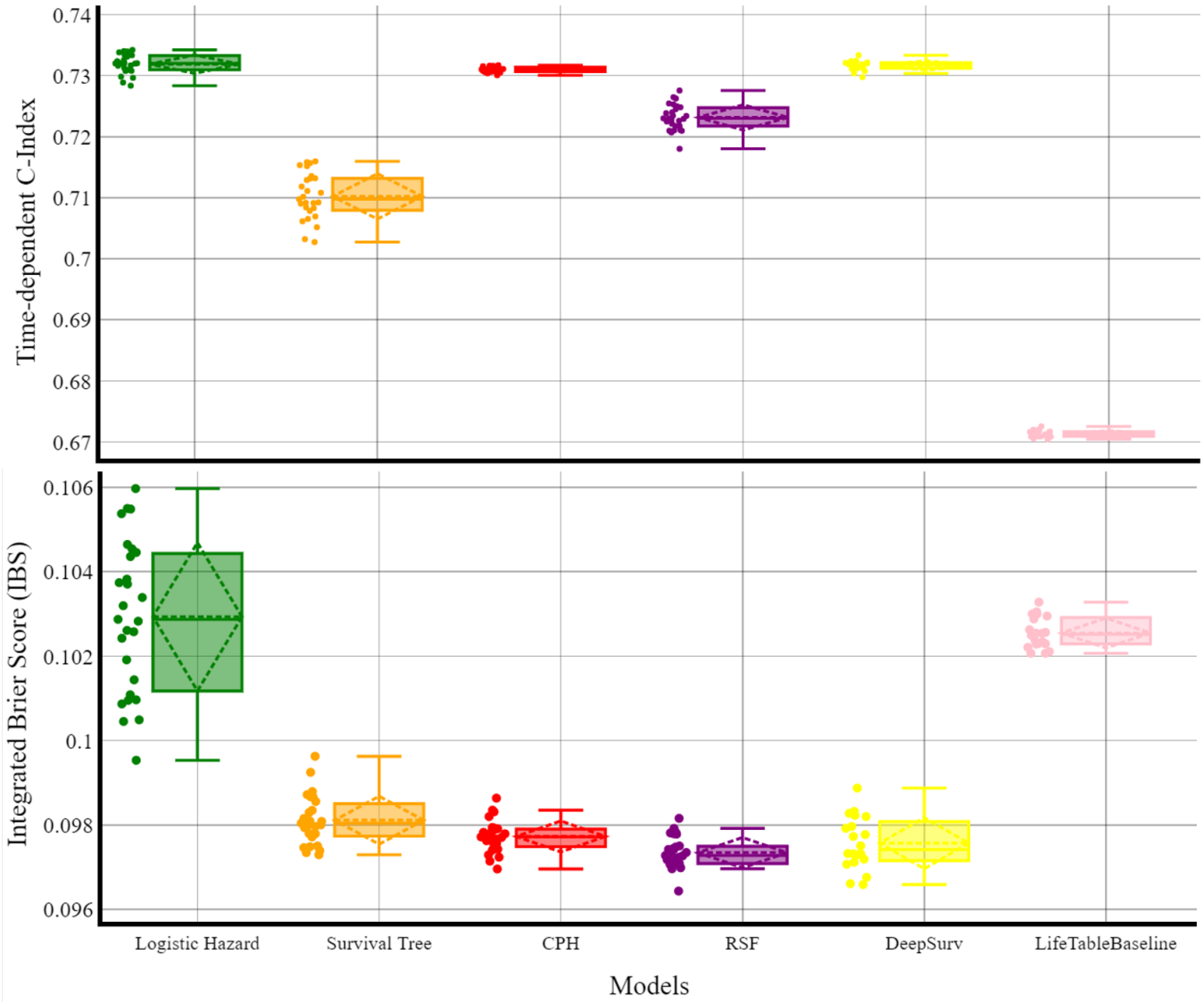

Figure 5.1: This figure presents the models' benchmark measured with C-Index and IBS, estimated with nested cross-validation.

though it is capable of maintaining a neural network depth of 2, It is not worth the mild performance boost. Consequently, we need to decide between CPH and Survival Tree. Albeit simple to interpret, CPH does not stratify patients into risk profiles[1] which is an advantage for medical purposes to guide clinical decisions. Moreover, despite being semiparametric, CPH still forces a strong assumption on data, namely, the proportional hazards assumption. This is unnecessary when we have a nonparametric model working similarly. The third reason to prefer the Survival Tree instead of the CPH is the fact that the inferences can be made in a manifestly clear manner

[1]Of course, this is possible to do by constructing ad hoc thresholds on the resulting summed risk $\sum_{i=1}^{p} \beta_i x_i$. The point is that this procedure is not straightforward, and It is not something automatically done by CPH.

by just utilizing figure 5.2.

On account of these reasons, the model selected is the Survival Tree model 3.2.1.

5.2 Final model

The Survival Tree obtained a (95% CI) C-Index of 0.710 (0.703 − 0.715), and an IBS of 0.097 (0.096 − 0.098). The way to generate inferences via the decision thresholds is displayed in figure 5.2. We need to "answer" the Yes and No questions from the node on the top until it hits a leaf node based on the patient's characteristics. Then, the survival function returned is the Kaplan-Meier of the leaf node's subpopulation.

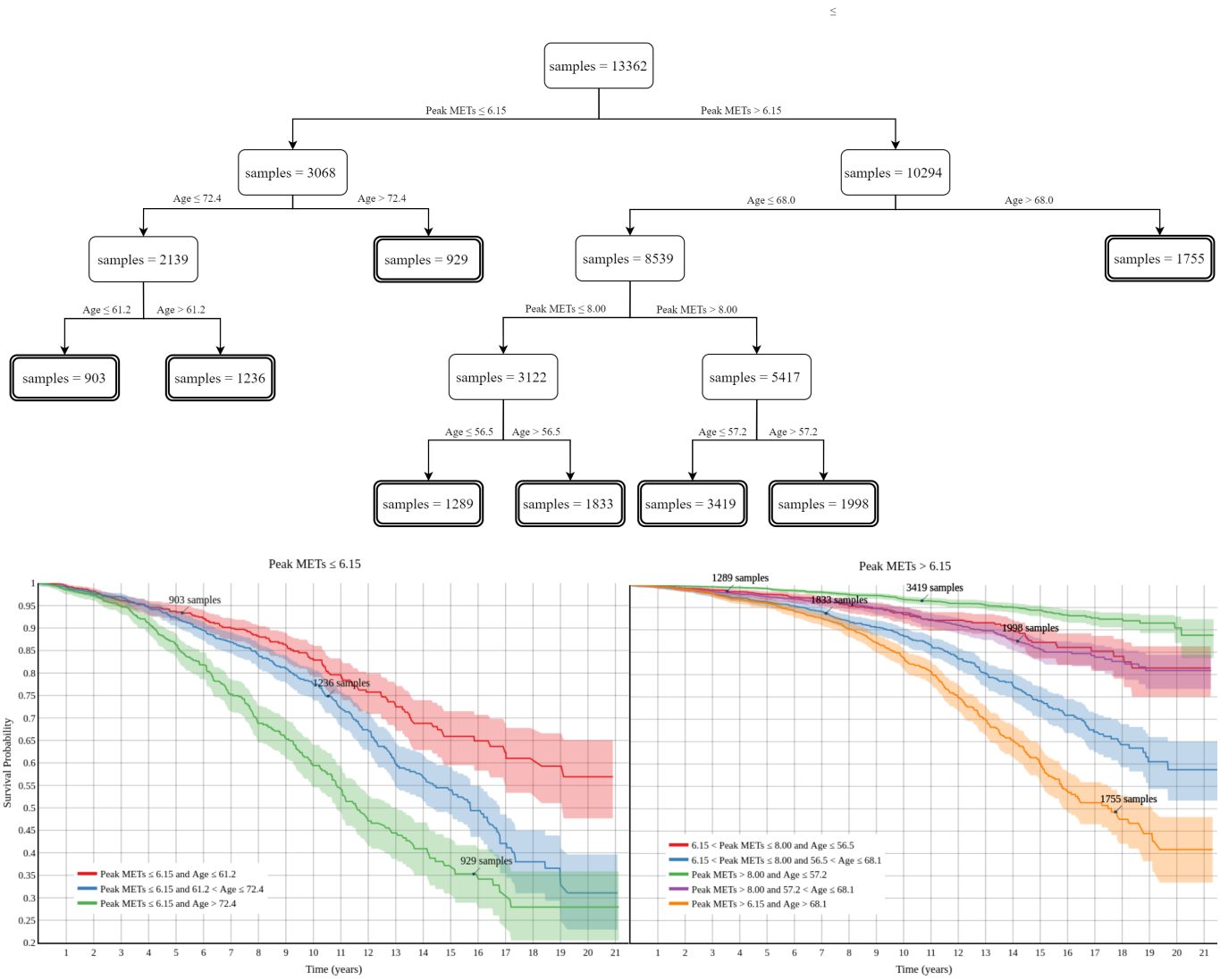

Figure 5.2: This picture shows the decision tree, which dictates how the Survival Tree processes information and returns a survival function.

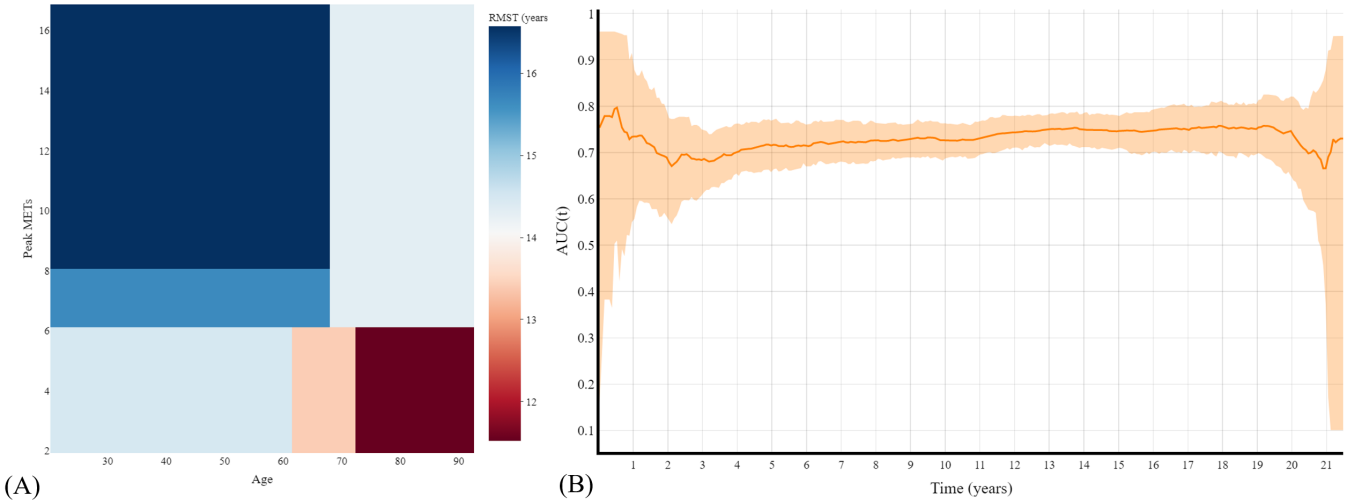

Figure 5.3: (A) the RMST for a grid of values involving `Peak METs` and `Age`. (B) $AUC(t)$ for the Survival Tree.

The $AUC(t)$ (figure 5.3) showed early, up to 1 year, and late, more than 20 years, low performance with huge uncertainty. This unpredictability was caused by the presence of a tiny number of deaths occurring before $t = 1$ year, only 49, and after 20 years, only a single death occurred. Explicitly, before the first year, the $AUC(t)$ was 0.764 $(0.417 - 0.951)$; between 1 and 3 years, the $AUC(t)$ was 0.700 $(0.581 - 0.804)$; between 3 to 20 years, 0.734 $(0.687 - 0.779)$; after 20 years, 0.707 $(0.414 - 0.861)$. Therefore, we advise only to construct inferences for a question involving times after the first year and before 20 years. The RMST in figure 5.3 (A) provides a clear visualization of the risk groups via heatmap with color-coded years.

Chapter 6

Results and Discussions

Results[1]:

- `Peak METs` is the most important feature for long-term overall survival prediction.
- `Peak METs` consistently scored higher than `Age`. This constitutes a surprise as a model that predicts overall survival should be strongly connected with the patient's age.
- A long-term, interpretable survival model was obtained that can be easily used by any practitioner just by inspecting figure 5.2.
- Features usually regarded as important in the literature, like CAD extent and therapeutic management, were outperformed by `Peak METs` and `Age`.

Limitations and shortcomings:

- We cannot infer if the model will generalize to patients with CAD who did not attend cardiac rehabilitation. This could have been curtailed if the feature indicating CR completion had not been corrupted.
- Potential confounding factors like concomitant use of cardiovascular disease medications were not available to help build the model. This problem is lessened in our case because we did not commit to causal analysis and only claimed a prediction approach.
- Each patient only generated a "row" of data at a single point in time. There were no longitudinally repeated measurements throughout which a richer view of the relation between feature and survival could have been undertaken.

[1]Unless stated otherwise, all assertions present here apply only to a population having similar characteristics to our dataset of 13362. Briefly, they consist of individuals with CAD who were referred to a CR program. Detailed information in chapter 2.

Chapter 7

Additional results: SurvMixClust

7.1 Introduction

When dealing with survival problems, it is important to identify subgroups that have similar survival profiles. This is especially crucial in the medical domain, where identifying heterogeneous treatment effects is vital. In section 4.2.2, we discussed the model K-Means associated with SFS and SBS for feature selection 4. However, this model is not suitable for survival data, and it works as an ad-hoc pipeline. Therefore, there is a need for an algorithm that can simultaneously solve the survival problem and cluster it properly. By using such an algorithm, we can enrich our analysis of the PROtECT dataset.

We solved this problem by creating a new clusterization algorithm that is general and can be applied to any survival data with right-censoring[1]. We name it **SurvMixClust**: **Surv** from its capability of returning a survival function, **Mix** from the fact that it uses a finite mixture of components, and **Clust** from its clusterization capability.

In this chapter, the model is explained in a general way matching a stand-alone article describing **SurvMixClust**, plus specific results for the PROtECT dataset. We motivate the algorithm in section 7.2, and expose related works in 7.3. Definition and training are delineated in 7.4. Then, results from the PROtECT and other datasets are shown in section 7.5, followed by a discussion in 7.6. Finally, we complete with a conclusion in 7.7.

[1]Note that other kinds of censoring can be mathematically transformed into the right-censored case. For instance, left-censoring can be transformed into the right-censoring case by multiplying the time labels by -1 [73].

7.2 Motivation

Cluster analysis of survival data can identify similar groups in time-to-event distribution, aiding in disease subtyping, risk stratification, and clinical decision-making. Precision medicine can benefit from these algorithms [74]. There are only a few specialized algorithms available for this task, but they are beginning to receive more attention.

Some models attempt to create clusters without jointly learning clustering and prediction, such as using a pipeline with the help of a CoxPH model [75] or a hierarchical clustering approach [76, 77]. However, these algorithms are not able to provide clusters that are also good at predicting the survival event. Leaving open questions about whether we could not find better groups that are more diversified in regard to survival. Recently, models that learn a latent representation together with the time-to-event prediction problem have been created, such as [18] and [78]. Even so, when compared to normal survival methods, their predictive performance is lacking, so practitioners may have uncertainty about using them as a standalone model. Moreover, their resulting clusterization underwhelms in regards to how diversified the survival functions are.

To address this issue, we offer the following **contributions**:

- We propose the **SurvMixClust**algorithm for clustering survival data. It simultaneously learns a latent representation and solves the time-to-event problem. It can also return a customized survival function for each data point, functioning similarly to standard survival models.

- The **SurvMixClust**algorithm can identify clusters with highly diversified survival functions, each displaying distinctive curves. Additionally, it can find clusters with a balanced number of data points. Compared to other clustering algorithms, it has demonstrated better predictive performance when evaluated using the time-dependent c-index metric across all datasets [79]. Our algorithm also outperformed three out of five datasets when evaluated using the log-rank metric, which measures the quality of the clustering.

- **SurvMixClust**performs as well as survival models that do not cluster like the Random Survival Forest model in terms of predictive performance, as measured by the time-dependent c-index survival metric.

- The code for the model is publicly accessible and can be found at https://github.com/buginga/SurvMixClust. We follow the basic scikit-learn API.

7.3 Related Works

SurvMixClust uses the traditional model-based approach for clustering, as outlined in [80]. The statistical literature contains work using mixtures of parametric distributions for the survival analysis problem, such as [81–84]. However, our main objective is to create a mixture with nonparametric distributions.

In the machine learning literature, [18] provides a Bayesian nonparametric approach under the name Survival Cluster Analysis (SCA), using latent representation and distribution matching techniques. Our method differs in model structure: we include the features with a multinomial logistic regression instead of a neural network; training: **SurvMixClust** uses EM instead of stochastic gradient descent on minibatches like SCA; choice of the number of clusters: in our case, the number of clusters is a hyperparameter, not for SCA. [85] created the DeepCLife model, which finds the clusterization by optimizing the pairwise distance via the logrank score. Similarly to SCA, our model clusters jointly with the predictive time-to-event problem from a latent space point of view, rendering it different from DeepCLife. Also, VaDeSC [78] was proposed as a variational deep survival clustering model. It utilizes a VAE (variational auto-encoder) regularized by a Gaussian mixture to create a latent space, then used to control a survival density function as a mixture of Weibull distributions. The main dissimilarity is, again, the parametric approach.

Our model builds upon [86], using the same basic non-parametric clusterization for the time labels. The difference is that we use the features to model the mixing proportions and build for more than two clusters, among other changes. Moreover, we compare predictive performance with other models.

7.4 Algorithm

The theoretical framework used is the same as in 3.1, with $T = \min\{T^*, C^*\}$, and $D = \mathbb{1}\{T^* \leq C^*\}$. Datasets are assumed to come from an i.i.d. process with samples of the form: $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$, with $\boldsymbol{x}_i \in \mathbb{R}^m$ (m is the number of features), and $\boldsymbol{y}_i = (t_i, d_i)$, with $T \sim t_i$, $D \sim d_i$, and $X \sim \boldsymbol{x}_n$. Z is the discrete latent variable modeling the clusters. Further, we assume random censoring, i.e. T^* is statistically independent of C^*, and figure 7.1 presents a graph model with the rest of the independence assumptions. These assumptions are necessary in order to obtain a feasible data likelihood, to be able to use the Kaplan-Meier estimator [7], and as an intentional modeling constraint.

All relevant notation used throughout this chapter is presented in table 7.1.

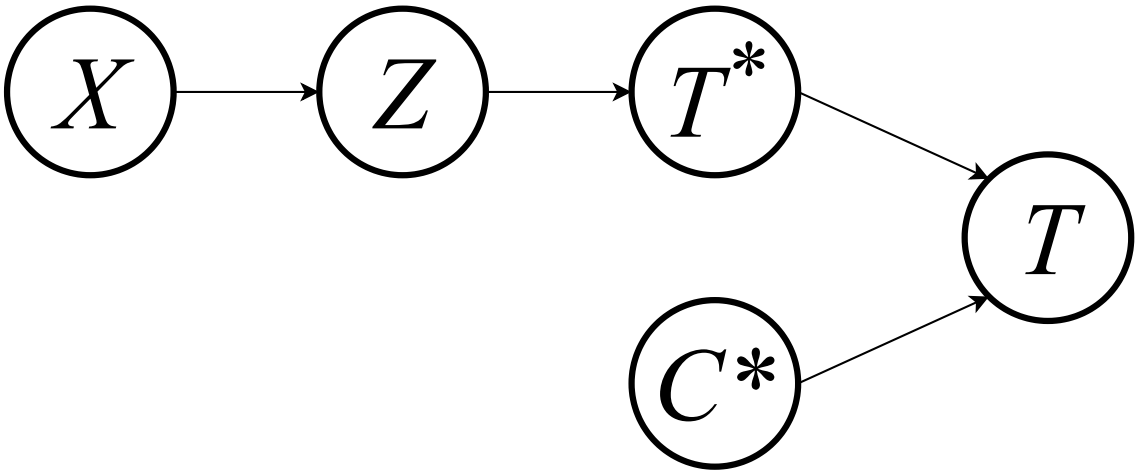

Figure 7.1: Graph model representing independence assumptions for the main model. Notice how the features X can only influence T^* via the clusterization Z.

7.4.1 Definition

The model is a finite mixture of K nonparametric distributions, wherein the mixing weights are calculated via a multinomial logistic regression using the features as input. Figure 7.1 diagrammatically presents the model. Notice how X only influences T^* through Z. Additionally, each cluster possesses its own fixed nonparametric distribution, which models how T^* behaves.

Logistic regression was selected because it's one of the most simple and interpretable models that can be used to model mixing proportions [80]. When using features, the model-based clustering approach often selects it as the primary option. Expectation-maximization training can require dozens of iterations, making a lightweight model desirable. Even so, as section 7.4.2 makes clear, the logistic regression model can be easily exchanged by any other probabilistic model of a finite discrete random variable. This change can be helpful for datasets containing features with an exploitable structure, like images, for which a convolutional neural network might be a better fit.

Writing $f(t) = P(T^* = t)$ and $S(t) = P(T^* > t)$, the model with K clusters can be presented as,

$$f\left(t_i^* \mid \boldsymbol{x}_i\right) = \sum_{k=1}^{K} \tau_k\left(\boldsymbol{x}_i\right) f\left(t_i^* \mid \theta_k\right) \tag{7.1}$$

$$\tau_k\left(\boldsymbol{x}_i\right) = \frac{\exp\left(\beta_k^T \boldsymbol{x}_i\right)}{\sum_{l=1}^{K} \exp\left(\beta_l^T \boldsymbol{x}_i\right)} \tag{7.2}$$

It is possible to obtain an intuitive form that includes the survival function:

$$S\left(t_i^* \mid \boldsymbol{x}_i\right) = \sum_{k=1}^{K} \tau_k\left(\boldsymbol{x}_i\right) S\left(t_i^* \mid \theta_k\right) \tag{7.3}$$

The distributions and parameters to be found are $\boldsymbol{\theta} = \{S(.|\theta_z), f(.|\theta_z)\}_{z=1}^{K} \cup \{\boldsymbol{\beta}\}$.

7.4.2 Training with Expectation-Maximization

The training is done via a standard combination of maximum likelihood estimation and the bound optimization algorithm called Expectation-Maximization [87]. Concretely, we first need to derive the expected complete data log-likelihood equation as a function of the data $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{n}$ and $\boldsymbol{\theta}$. Then, calculate the algorithmic details of the expectation and maximization steps.

Before building the maximum likelihood estimator, we need a helpful formula from [32], derivation at D. In general, we can write:

$$\mathrm{P}(T = t, D = d) = \left[f(t)^{\mathbb{I}(d=1)} S(t)^{\mathbb{I}(d=0)}\right] \cdot \left[f_{C^*}(t)^{\mathbb{I}(d=0)} \left(S_{C^*}(t) + f_{C^*}(t)\right)^{\mathbb{I}(d=1)}\right] \tag{7.4}$$

Let's define $h(t)$ as follows: $h(t) = \left[f_{C^*}(t)^{\mathbb{I}(d=0)} \left(S_{C^*}(t) + f_{C^*}(t)\right)^{\mathbb{I}(d=1)}\right]$. Based on our assumptions presented in figure 7.1, we can recognize that $h(t_n|z_n = l)$ is equal to $h(t_n)$. Then,

$$\begin{aligned} \mathrm{P}(T = t, D = d, Z = z \mid \boldsymbol{\theta}) &= \mathrm{P}_{\boldsymbol{\theta}}(Z = z) f(T = t, D = d \mid Z = z; \boldsymbol{\theta}) \\ &= \mathrm{P}_{\boldsymbol{\theta}}(Z = z) f\left(t \mid \theta_z\right)^{\mathbb{I}(d=1)} S\left(t \mid \theta_z\right)^{\mathbb{I}(d=0)} h(t) \end{aligned} \tag{7.5}$$

Equation 7.5 is going to be used for the calculation of the expected complete data log-likelihood $LL^t(\boldsymbol{\theta})$. We use the following notation for the cluster's labels: $z_n \in [1, ...K]$, and $z_{nk} = \mathbb{I}\left(z_n = k\right)$. For the Expectation step at the iteration (t), it's necessary to calculate the posterior membership probability of cluster k for the datapoint n, details at D:

$$\begin{aligned} r_{nk}^{(t)} &= \mathrm{P}\left(z_n = k \mid t_n, d_n, x_n; \boldsymbol{\theta}^{(t)}\right) \\ &= \frac{\left[f\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=0)}\right] \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} \left[f\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=0)}\right] \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \end{aligned} \tag{7.6}$$

Using the equation $\mathbb{E}\left[z_{nk}\right] = r_{nk}^{(t)}$ and equation 7.5, the Maximization step at (t) can be expressed as follows:

$$
\begin{aligned}
LL^t(\boldsymbol{\theta}) &= \sum_n \mathbb{E}_{q_n^t(z_n)} \left[\log \mathrm{P}(T = t_n, D = d_n, Z = z_n \mid X = x_n; \boldsymbol{\theta})\right] \\
&= \sum_n \sum_k r_{nk}^{(t)} \log \tau_k \left(\boldsymbol{x}_n\right) + \\
&\quad + \sum_n \sum_k r_{nk}^{(t)} \mathbb{I}\left(d_n = 1\right) \log f\left(t_n \mid \theta_k^{(t)}\right) + \\
&\quad + \sum_n \sum_k r_{nk}^{(t)} \mathbb{I}\left(d_n = 0\right) \log S\left(t_n \mid \theta_k^{(t)}\right) + \\
&\quad + \sum_n \sum_k r_{nk}^{(t)} \left(h(t_n)\right)
\end{aligned}
\tag{7.7}
$$

As $\sum_{n,k} r_{nk}^{(t)} h(t)$ does not depend on $\boldsymbol{\theta}$, we will only maximize the remaining three terms of equation 7.7.

7.4.3 Training Algorithm

Utilizing equations 7.7 and 7.6, the model is trained with *stochastic* expectation maximization [88], similar to the one used in section 4.2 of [86]. A single run of the entire algorithm follows the steps:

0. **Hyperparameters**: establish the value of the number K of clusters ($K \in \{2, 3, 4, ...\}$).

1. **Initialization**: For each $i \in \{1, ..., n\}$, z_i is assigned a label from $\{1, ..., K\}$ with equal probability, i.e., completely random assignment.

Then, repeat until convergence the following two steps:

2. **E-Step**: For each $i \in \{1, ..., n\}$:

$$
\begin{aligned}
q_i^{(t)} &= \arg\max_k \left(f\left(t_i \mid \theta_k^{(t)}\right)^{\mathbb{I}(d=1)} S\left(t_i \mid \theta_k^{(t)}\right)^{\mathbb{I}(d=0)} \tau_k^{(t)}\left(\boldsymbol{x}_i\right) \right) \\
r_{iq_i^{(t)}}^{(t)} &= 1 \\
r_{ik}^{(t)} &= 0, \quad \forall k \in \{1, ..., K\} \text{ and } k \neq q_i^{(t)}
\end{aligned}
\tag{7.8}
$$

3. **M-Step**: For each $k \in \{1, ...K\}$, denote $\text{group}_k = \{i \in \{1, ..., n\} \mid r_{ik}^{(t)} = 1\}$, i.e., group_k include all data points that were assigned the exclusive label of k. Repeat the following steps for each k.

   (a) $S\left(t \mid \theta_k^{(l+1)}\right)$ is estimated with the Kaplan-Meier estimator [7] using the data points in group_k.

(b) $f\left(t \mid \theta_k^{(l+1)}\right)$ is calculated using a nonparametric presmoothed estimator [89], its bandwidth is selected via plug-in estimate and fixed for all EM steps in order to reduce computation time.

(c) Finally, for estimating $\tau^{(l+1)}$ we train a multinomial logistic regression classifier. The labels are the $q_i^{(t)}$ calculated at the E-Step. Specifically, the training data for the classifier is $\mathcal{D}^{\text{reg}} = \left\{\boldsymbol{x}_i, q_i^{(t)}\right\}_{i=1}^{n}$. This training is done as a standard supervised learning problem.

7.5 Results

7.5.1 Public Datasets

These experiments compare the proposed algorithm with other survival models, including purely time-to-event and clusterization algorithms. The former group includes the Random Survival Forest 3.2.1, the Cox Proportional Hazards (CoxPH) 3.2.2, Logistic Hazard (neural network model) 3.2.1. The latter group is formed by our proposal, the Survival Cluster Algorithm (SCA) [18], and **K-means Survival**. Further methodological details, including hyperparameters searched, are included in the appendix's subsections C, D, and E. The number of clusters searched for K is $\{2, 3, 4, 5, 6, 7\}$.

K-means Survival is simply the common K-means but interpreted as a fully-fledged survival model, i.e., It can return a survival function for each data point. It manages to do it by training as usual in an unsupervised way, but for the inference, it returns for a data point the Kaplan-Meier of the population inside its inferred cluster.

Table 7.2 lists the publicly accessible datasets used for the experiments. SUPPORT, Study to Understand Prognoses Preferences Outcomes and Risks of Treatment [90]; FLCHAIN, The Assay of Serum Free Light Chain (FLCHAIN) [91]; GBSG, The Rotterdam & German Breast Cancer Study Group [92]; METABRIC, The Molecular Taxonomy of Breast Cancer International Consortium [93]; the Worcester Heart Attack Study (WHAS500), specifically the version with 500 patients [94]. In all experiments, continuous features were imputed with their mean and categorical features with their mode, followed by one-hot encoding. The labels were kept unchanged.

The first metric utilized is the time-dependent C-index 3.3.1, which is calculated by treating all models as purely time-to-event survival models. For an evaluation of clusterization, the logrank score between clusterized populations is used [95].

Figure 7.3 displays the time-dependent C-index outcomes for different datasets and models. In the chart, "**(ours)**" indicates the approach proposed in this paper.

The **SurvMixClust** model outperforms the clustering-based models (SCA and **K-means Survival**) in all datasets. When compared to purely predictive models, it shows a similar performance for the GBSG and WHAS500 datasets. However, for the METABRIC and FLCHAIN datasets, it falls into the second performance tier and is statistically similar to the RSF and CoxPH models, respectively. Finally, for the SUPPORT dataset, **SurvMixClust** ranks third in performance, together with CoxPH.

Similarly, figure 7.4 depicts the logrank results, which show higher log-rank metrics in the SUPPORT, FLCHAIN, and WHAS500 datasets. Additionally, the log-rank metric is similar to SCA for the GBSG dataset but worse for the METABRIC dataset.

Also, an explicit test set's clusterization is shown in figure 7.2 for the SUPPORT dataset. It can be inspected that both SCA and **SurvMixClust** have more different and distributed clusters than pure K-means; this is expected as both use information from labels. The advantage is the fact that **SurvMixClust**has clusters with a more balanced size, and it's more spread out than the other two options. This pattern remained across different experiments.

7.5.2 PROtECT dataset

PROtECT is the dataset presented in chapter 2 and used for the model in chapter 5. We trained **SurvMixClust** with this dataset containing the original 25 features pre-feature selection (check 2.1 for the entire list). After a benchmark comparing the best number of clusters in the set $\{2, 3, 4, 5, 6, 7\}$ using a validation set, the best hyperparameter was K=7. Figure 7.5 reveals a specific clusterization with K=7 and a C-Index of 0.712, slightly better than the Survival Tree model in 5.2. This same clusterization is partially shown in figure 7.6 with the help a scatter plot of `Peak METs` and `Age`. Notice how the clusters have a natural Gaussian-looking format.

7.6 Discussion

The **SurvMixClust**is a model that can produce clusterizations that reveal diverse population distributions with distinct survival profiles, which makes it ideal for discovering different groups. Some positive aspects of this model include: (i) a majority of runs result in balanced clusters; (ii) it possesses better predictive performance regarding the c-index metric when compared to algorithms that cluster across all datasets, and it competes with purely predictive survival models; (iii) it performed better in three out of the five datasets in the log-rank metric against the other clusterization algorithms; (iv) the model's structure and training enable it to be adapted

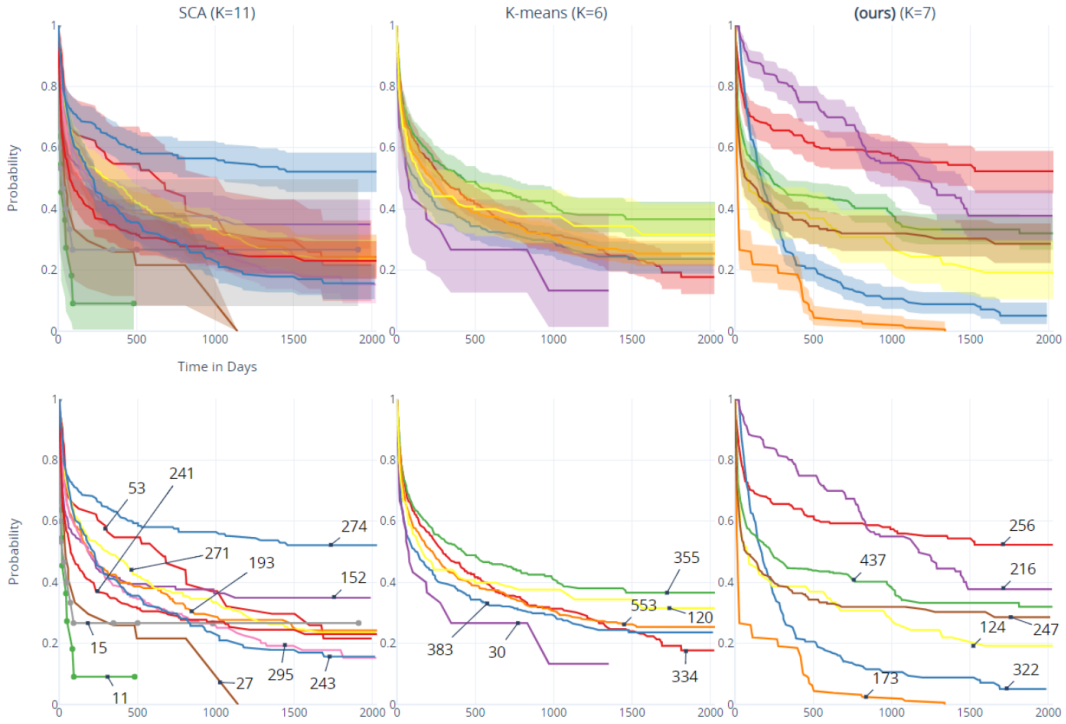

Figure 7.2: Test set's clusterization for the SUPPORT dataset returned by the models: SCA, K-means, and our proposal (**SurvMixClust**). The initial row shows the Kaplan-Meier of the cluster's subpopulations and the calculated confidence intervals. The row below shows the same survival functions, but now without the confidence intervals and the number of data points inside each cluster.

for other types of data by replacing the multinomial regression with other models with the same output.

While using our algorithm, it's important to note that there might be a higher variance in metrics between different runs with the same training data, compared to algorithms that do not use Expectation Maximization in training. Therefore, it may be necessary to run multiple EM runs to select the best hyperparameter K or to find the most suitable trained model, as some degree of exploration is required.

Another pronounced fact is the competitive performance of **K-means Survival** with SCA, as shown in figure 7.3 and 7.4. This can be due to the small to medium dataset sizes used, as SCA uses neural networks that can provide the chance for better scaling.

Finally, there are some guiding principles to keep in mind for proper model use. Firstly, even if a specific trained model acquires high values of C-index, it is advisable to visually check the clusterization on the training and validation sets before making a choice. Often, a more modest C-index (closer to the mean) has better-behaving survival functions. Secondly, if there is a need to return the survival function for a

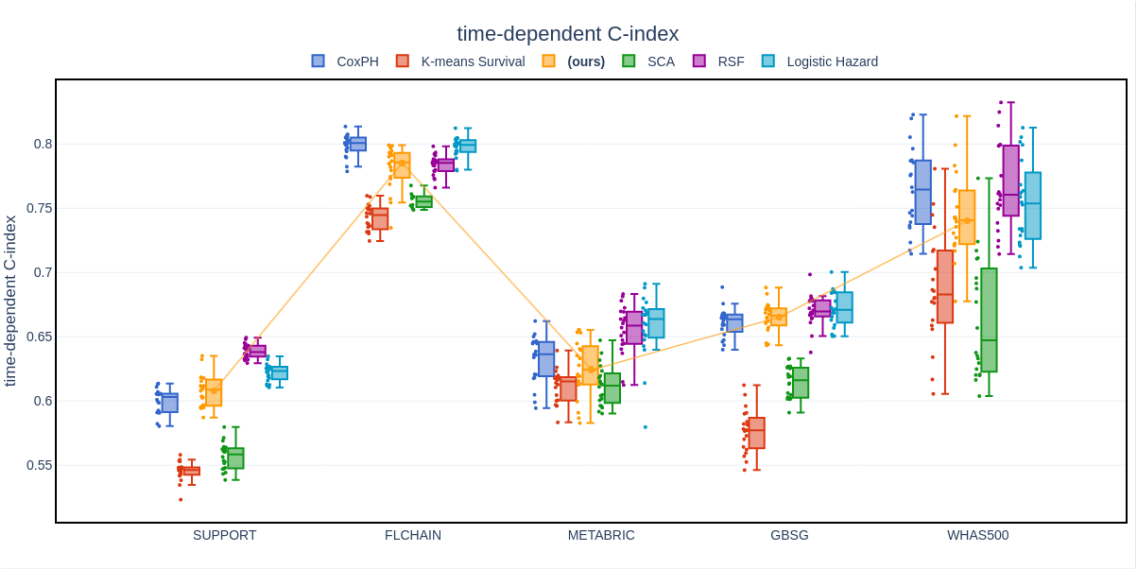

Figure 7.3: Time-dependent C-index across datasets and models. Each boxplot displays 20 samples.

data point, it is overall better to do full inference using equation 7.3 instead of just returning the survival function of its most probable cluster. This is what we do in all metrics calculations that are amenable to this.

For the application of **SurvMixClust** to the PROtECT dataset, we obtained a C-Index performance of 0.712 with K=7 clusters which is competitive with all purely predictive models, as can be inspected from figure 3.3. The main reason that we did not use this model for the final proposed model is that the inference requires the calculation of the posterior, necessarily requiring a computer or deployed model, while the Survival Tree model in chapter 5 only requires checking figure 5.2 in order to do inference. It is worth noting that medical practitioners explicitly communicated the need for the models to be easily usable.

7.7 Conclusion

We developed an innovative algorithm called **SurvMixClust**, which can cluster survival data effectively. The goal behind developing this algorithm was to identify subgroups with different survival profiles while still maintaining a lightweight model that can accurately predict a survival function. To achieve this goal, **SurvMixClust** uses a non-parametric model-based clusterization framework that is specifically adapted for right-censored time-to-event data. Our experiments have shown that this approach is highly effective in identifying subgroups with markedly different survival function formats, even on the PROtECT

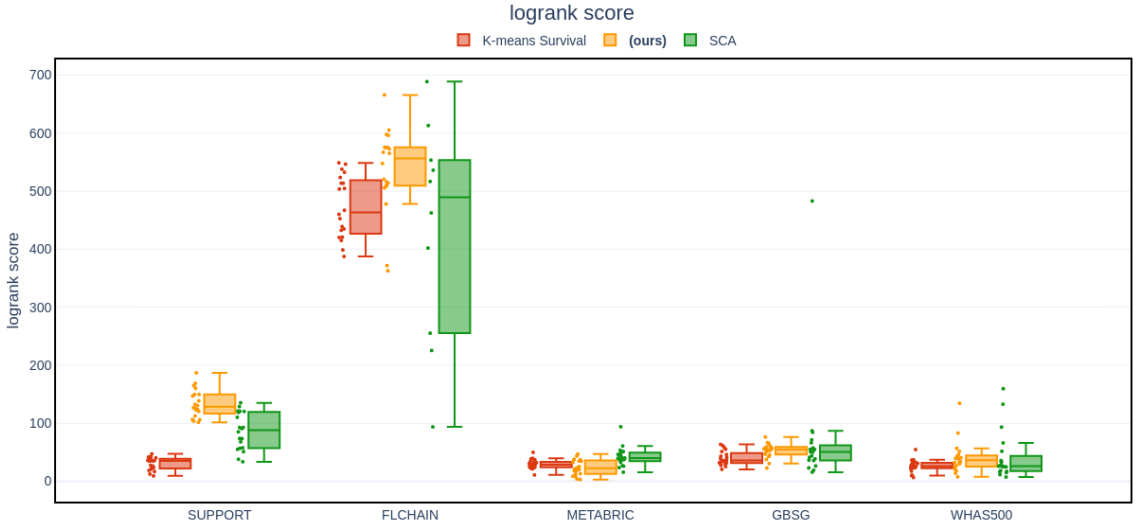

Figure 7.4: Logrank score across datasets and models. Each boxplot displays 20 samples.

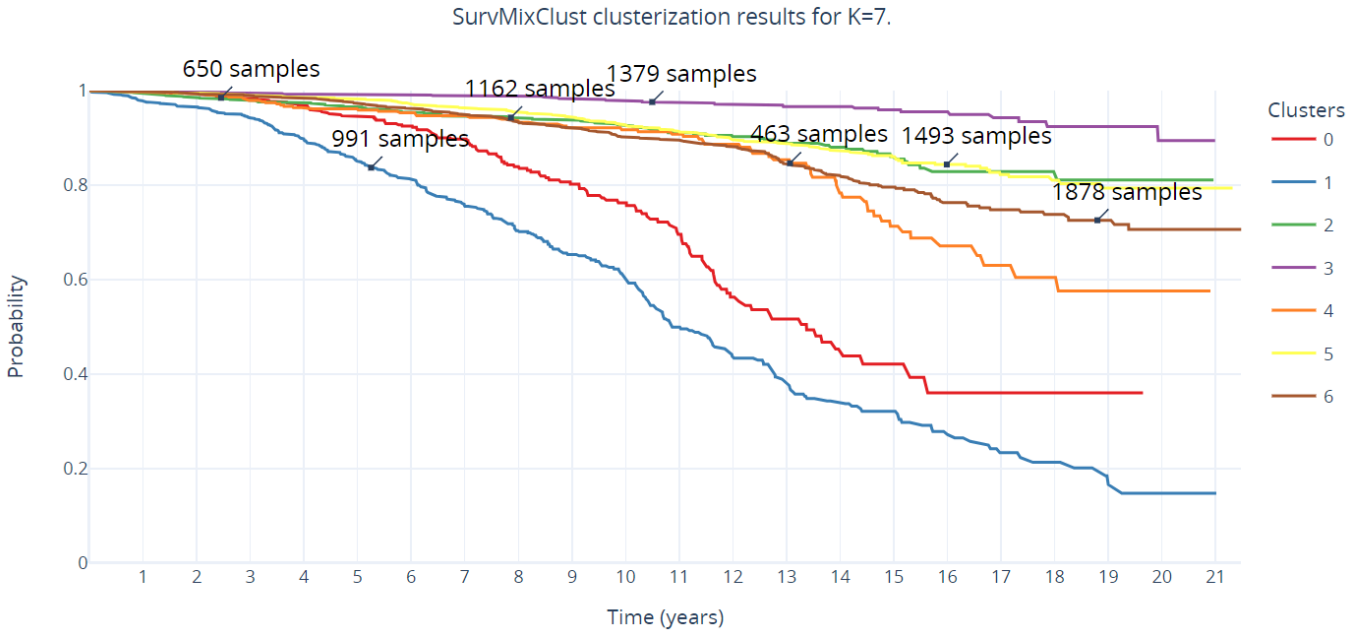

Figure 7.5: **SurvMixClust**'s clusterization results for K=7. Notice how the survival functions are spread out, showing different survival profiles. We can also see the fact that the clusters are balanced. These curves are the seven components of the mixture of the fully-trained model, i.e. the S's in equation 7.3. It obtained a C-index of 0.712.

dataset. This tool can be used by practitioners to detect evidence of heterogeneous treatment effects. The code for the model is publicly accessible and available at https://github.com/buginga/SurvMixClust.

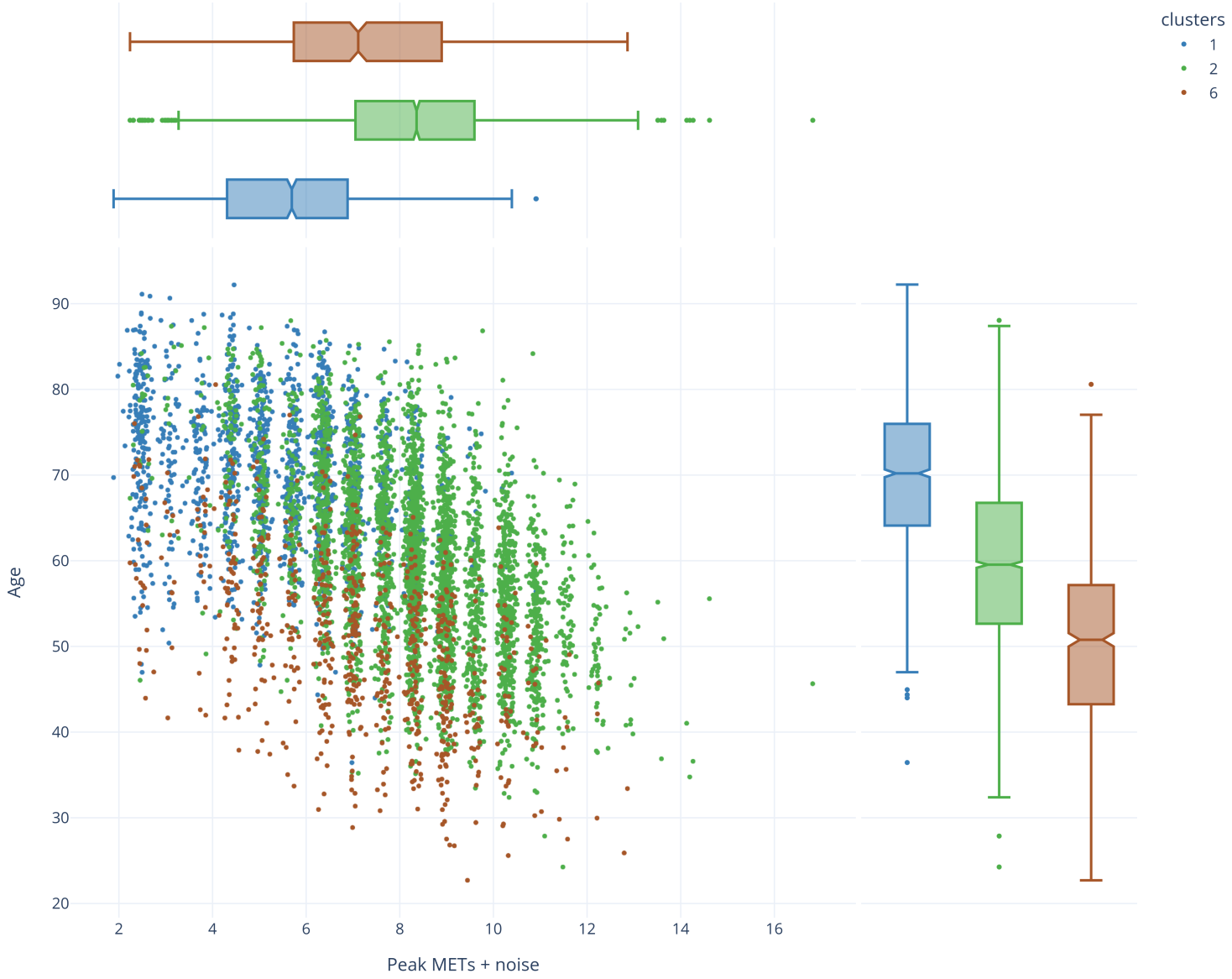

Figure 7.6: **SurvMixClust**'s clusterization results for K=7, it is exactly the same model in figure 7.5, including matching colors. It displays only three clusters inside a scatter plot with features `Peak METs` and `Age`. Some random normal noise was introduced into `Peak METs` in order to facilitate visualization (normal with mean zero and variance 0.1).

Table 7.1: Notations and definitions used throughout chapter 7.

Notation	Description
T^*	random variable for ground-truth time-to-event
C^*	random variable for ground-truth censoring process
$f_{C^*}(t)$	density function for C^*
$S_{C^*}(t)$	survival function for C^*
$f(t)$	density function for T^*
$S(t)$	survival function for T^*
T	random variable for possibly censored time-to-event, defined as $T = \min\{T^*, C^*\}$
t_i	possibly censored time-to-event for datapoint i
D	binary random variable for censoring indication; defined as $D = \mathbb{1}\{T^* \leq C^*\}$, i.e. $D = 1$ represents that the event happened, and $D = 0$ that it was censored
d_i	censoring indication for datapoint i
Z	discrete random variable taking values in $\{1, ..., K\}$, modelling cluster attribution
$\boldsymbol{x}_i$	m-dimensional vector of covariates for datapoint i
$\boldsymbol{y}_i$	2-dimensional label vector for datapoint i, defined as $\boldsymbol{y}_i = (t_i, d_i)$
τ	multinomial logistic regression modeling the mixture proportions
K	integer indicating the number of clusters for the main model
$\boldsymbol{\theta}$	includes all parameters for the EM
θ_k	indicates the non-parametric distribution for cluster k
$\boldsymbol{\beta}$	K-dimensional vector of the coefficients for the multinomial regression

Table 7.2: Publicly accessible datasets used for the benchmark.

Dataset Name	Shape	Censoring ($\frac{\sum_{i=1}^{N} \mathbb{I}(d_i=0)}{N}$)
SUPPORT	$(8873, 14)$	31.9%
FLCHAIN	$(7874, 26)$	72.4%
METABRIC	$(1904, 9)$	42.0%
GBSG	$(2232, 7)$	43.2%
WHAS500	$(500, 14)$	43.0%

Chapter 8

Conclusion

Our work embarked on the opportunity to apply classical and new methods in the realm of survival analysis to generate valuable and actionable knowledge in the vital domain of cardiology.

The PROtECT study dataset had a large number of patients and provided access to the original practitioners who built the dataset, which is not commonly seen in medical applications. Our goal was to build an interpretable model to predict long-term survival using this dataset. To achieve this, we performed extensive data pre-processing and multi-piece feature selection. This process resulted in a remarkable 99.2% reduction in the number of features. We then did a benchmark and analyzed the results, leading us to choose the Survival Tree model. Finally, we presented every aspect of the final model, concluding with a decision tree and Kaplan-Meiers graph that allows for straightforward inspection and production of inferences.

Moreover, we developed **SurvMixClust**, a novel algorithm for clusterization of survival data. It was motivated by the need to identify subgroups with different survival profiles and still have a lightweight model predicting a survival function having competitive performance. **SurvMixClust** solved these prerequisites using a non-parametric model-based clusterization framework that was adapted for right-censored time-to-event data. Experimentation revealed its effectiveness in identifying subgroups with sharply different survival function formats, including for the PROtECT dataset. Practitioners can use this tool to identify evidence of heterogeneous treatment effects. The model's code is publicly accessible and available at https://github.com/buginga/SurvMixClust.

Future works include a Bayesian approach using the survival functions given by the census [43] as a *prior*, as hinted in `LifeTableBaseline`. Another avenue is to obtain information about the extent of the completion of the cardiac rehabilitation program. In this way, a proper causal analysis for observational data can be adopted [96, 97], especially the targeted maximum likelihood estimation method [98]. Finally, **SurvMixClust** could be adapted to deal with the competing risks scenario.

References

[1] JOHNSON, K. W., TORRES SOTO, J., GLICKSBERG, B. S., et al. "Artificial intelligence in cardiology", *Journal of the American College of Cardiology*, v. 71, n. 23, pp. 2668–2679, 2018.

[2] CHEN, I. Y., JOSHI, S., GHASSEMI, M., et al. "Probabilistic machine learning for healthcare", *Annual Review of Biomedical Data Science*, v. 4, 2020.

[3] STEINHUBL, S. R., TOPOL, E. J. "Moving from digitalization to digitization in cardiovascular care: why is it important, and what could it mean for patients and providers?" *Journal of the American College of Cardiology*, v. 66, n. 13, pp. 1489–1496, 2015.

[4] BOELDT, D. L., WINEINGER, N. E., WAALEN, J., et al. "How consumers and physicians view new medical technology: comparative survey", *Journal of medical Internet research*, v. 17, n. 9, pp. e4456, 2015.

[5] KLEINBAUM, D. G., KLEIN, M. *Survival analysis*, v. 3. Springer, Springer, 2010.

[6] WANG, P., LI, Y., REDDY, C. K. "Machine learning for survival analysis: A survey", *ACM Computing Surveys (CSUR)*, v. 51, n. 6, pp. 1–36, 2019.

[7] KAPLAN, E. L., MEIER, P. "Nonparametric estimation from incomplete observations", *Journal of the American statistical association*, v. 53, n. 282, pp. 457–481, 1958.

[8] MURPHY, K. P. "Machine learning: a probabilistic perspective", 2012.

[9] VOCK, D. M., WOLFSON, J., BANDYOPADHYAY, S., et al. "Adapting machine learning techniques to censored time-to-event health record data: A general-purpose approach using inverse probability of censoring weighting", *Journal of biomedical informatics*, v. 61, pp. 119–131, 2016.

[10] LECUN, Y., BENGIO, Y., HINTON, G. "Deep learning", *nature*, v. 521, n. 7553, pp. 436–444, 2015.

[11] KATZMAN, J. L., SHAHAM, U., CLONINGER, A., et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network", *BMC medical research methodology*, v. 18, n. 1, pp. 24, 2018.

[12] LEE, C., ZAME, W. R., YOON, J., et al. "Deephit: A deep learning approach to survival analysis with competing risks". In: *Thirty-second AAAI conference on artificial intelligence*, 2018.

[13] KIM, D. W., LEE, S., KWON, S., et al. "Deep learning-based survival prediction of oral cancer patients", *Scientific reports*, v. 9, n. 1, pp. 1–10, 2019.

[14] MOBADERSANY, P., YOUSEFI, S., AMGAD, M., et al. "Predicting cancer outcomes from histology and genomics using convolutional networks", *Proceedings of the National Academy of Sciences*, v. 115, n. 13, pp. E2970–E2979, 2018.

[15] HAO, J., KIM, Y., MALLAVARAPU, T., et al. "Interpretable deep neural network for cancer survival analysis by integrating genomic and clinical data", *BMC medical genomics*, v. 12, n. 10, pp. 1–13, 2019.

[16] HUANG, Z., ZHAN, X., XIANG, S., et al. "SALMON: survival analysis learning with multi-omics neural networks on breast cancer", *Frontiers in genetics*, v. 10, pp. 166, 2019.

[17] YOUSEFI, S., AMROLLAHI, F., AMGAD, M., et al. "Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models", *Scientific reports*, v. 7, n. 1, pp. 1–11, 2017.

[18] CHAPFUWA, P., LI, C., MEHTA, N., et al. "Survival cluster analysis". In: *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 60–68, 2020.

[19] ARMSTRONG, M. J., SIGAL, R. J., ARENA, R., et al. "Cardiac rehabilitation completion is associated with reduced mortality in patients with diabetes and coronary artery disease", *Diabetologia*, v. 58, n. 4, pp. 691–698, 2015.

[20] E SILVA, C. G. D. S., BUGINGA, G. C., E SILVA, E. A. D. S., et al. "Prediction of mortality in coronary artery disease: role of machine learning and maximal exercise capacity". In: *Mayo Clinic Proceedings*, v. 97, pp. 1472–1482. Elsevier, 2022.

[21] GHALI, W., KNUDTSON, M. "Overview of the Alberta Provincial Project for Outcome Assessment in Coronary Heart Disease. On behalf of the APPROACH investigators." *The Canadian journal of cardiology*, v. 16, n. 10, pp. 1225–1230, 2000.

[22] SOUTHERN, D. A., JAMES, M. T., WILTON, S. B., et al. "Expanding the impact of a longstanding Canadian cardiac registry through data linkage: challenges and opportunities", *International Journal of Population Data Science*, v. 3, n. 3, 2018.

[23] MARTIN, B.-J., HAUER, T., ARENA, R., et al. "Cardiac rehabilitation attendance and outcomes in coronary artery disease patients", *Circulation*, v. 126, n. 6, pp. 677–687, 2012.

[24] "Welcome to TotalCardiology Rehabilitation". https://tcrehab.totalcardiology.ca/, 2021.

[25] RIEBE, D., EHRMAN, J. K., LIGUORI, G., et al. "ACSM's guidelines for exercise testing and prescription". 2018.

[26] "Open Government". https://open.alberta.ca/interact/vital-statistics#death.

[27] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.

[28] FARRAR, D. E., GLAUBER, R. R. "Multicollinearity in regression analysis: the problem revisited", *The Review of Economic and Statistics*, pp. 92–107, 1967.

[29] JOSEPH KEARNEY, S. B. "Collaborative data science". 202. Disponível em: <`https://plot.ly`>.

[30] SPOONER, A., CHEN, E., SOWMYA, A., et al. "A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction", *Scientific reports*, v. 10, n. 1, pp. 1–10, 2020.

[31] LARIVIÈRE, B., VAN DEN POEL, D. "Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services", *Expert Systems with Applications*, v. 27, n. 2, pp. 277–285, 2004.

[32] KVAMME, H., BORGAN, Ø. "Continuous and discrete-time survival prediction with neural networks", *arXiv preprint arXiv:1910.06724*, 2019.

[33] ROYSTON, P., PARMAR, M. K. "Restricted mean survival time: an alternative to the hazard ratio for the design and analysis of randomized trials with a time-to-event outcome", *BMC medical research methodology*, v. 13, n. 1, pp. 152, 2013.

[34] MCCAW, Z. R., YIN, G., WEI, L.-J. "Using the restricted mean survival time difference as an alternative to the hazard ratio for analyzing clinical cardiovascular studies", *Circulation*, v. 140, n. 17, pp. 1366–1368, 2019.

[35] DAVIDSON-PILON, C., KALDERSTAM, J., JACOBSON, N., et al. "CamDavidsonPilon/lifelines: v0.25.11". abr. 2021. Disponível em: <`https://doi.org/10.5281/zenodo.4683730`>.

[36] BERAN, R. "Nonparametric regression with randomly censored survival data", 1981.

[37] CHEN, G. "Nearest neighbor and kernel survival analysis: Nonasymptotic error bounds and strong consistency rates". In: *International Conference on Machine Learning*, pp. 1001–1010. PMLR, 2019.

[38] PÖLSTERL, S. "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn", *Journal of Machine Learning Research*, v. 21, n. 212, pp. 1–6, 2020. Disponível em: <`http://jmlr.org/papers/v21/20-729.html`>.

[39] BIAU, G., SCORNET, E. "A random forest guided tour", *Test*, v. 25, n. 2, pp. 197–227, 2016.

[40] ISHWARAN, H., KOGALUR, U. B., BLACKSTONE, E. H., et al. "Random survival forests", *The annals of applied statistics*, v. 2, n. 3, pp. 841–860, 2008.

[41] KVAMME, H., BORGAN, Ø. "Continuous and discrete-time survival prediction with neural networks", *arXiv preprint arXiv:1910.06724*, 2019.

[42] GENSHEIMER, M. F., NARASIMHAN, B. "A scalable discrete-time survival model for neural networks", *PeerJ*, v. 7, pp. e6257, 2019.

[43] GOVERNMENT OF CANADA, S. C. "Life Expectancy and Other Elements of the Life Table, Canada, All Provinces except Prince Edward Island". https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1310011401, nov. 2020.

[44] BRESLOW, N. E. "Contribution to discussion of paper by DR Cox", *J. Roy. Statist. Soc., Ser. B*, v. 34, pp. 216–217, 1972.

[45] FARAGGI, D., SIMON, R. "A neural network model for survival data", *Statistics in medicine*, v. 14, n. 1, pp. 73–82, 1995.

[46] ANTOLINI, L., BORACCHI, P., BIGANZOLI, E. "A time-dependent discrimination index for survival data", *Statistics in medicine*, v. 24, n. 24, pp. 3927–3944, 2005.

[47] HARRELL, F. E., CALIFF, R. M., PRYOR, D. B., et al. "Evaluating the yield of medical tests", *Jama*, v. 247, n. 18, pp. 2543–2546, 1982.

[48] LAMBERT, J., CHEVRET, S. "Summary measure of discrimination in survival models based on cumulative/dynamic time-dependent ROC curves", *Statistical methods in medical research*, v. 25, n. 5, pp. 2088–2102, 2016.

[49] GRAF, E., SCHMOOR, C., SAUERBREI, W., et al. "Assessment and comparison of prognostic classification schemes for survival data", *Statistics in medicine*, v. 18, n. 17-18, pp. 2529–2545, 1999.

[50] KVAMME, H., BORGAN, Ø. "The brier score under administrative censoring: Problems and solutions", *arXiv preprint arXiv:1912.08581*, 2019.

[51] GHOJOGH, B., SAMAD, M. N., MASHHADI, S. A., et al. "Feature selection and feature extraction in pattern analysis: A literature review", *arXiv preprint arXiv:1905.02845*, 2019.

[52] JOVIĆ, A., BRKIĆ, K., BOGUNOVIĆ, N. "A review of feature selection methods with applications". In: *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 1200–1205. Ieee, 2015.

[53] KUHN, M., JOHNSON, K. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, CRC Press, 2019.

[54] ZHAO, Z., MORSTATTER, F., SHARMA, S., et al. "Advancing feature selection research", *ASU feature selection repository*, pp. 1–28, 2010.

[55] LANGLEY, P. *Elements of machine learning*. Morgan Kaufmann, Morgan Kaufmann, 1996.

[56] LANGLEY, P., OTHERS. "Selection of relevant features in machine learning". In: *Proceedings of the AAAI Fall symposium on relevance*, v. 184, pp. 245–271, 1994.

[57] SAHU, B., DEHURI, S., JAGADEV, A. "A Study on the Relevance of Feature Selection Methods in Microarray Data", *The Open Bioinformatics Journal*, v. 11, n. 1, 2018.

[58] GAO, Y.-F., LI, B.-Q., CAI, Y.-D., et al. "Prediction of active sites of enzymes by maximum relevance minimum redundancy (mRMR) feature selection", *Molecular BioSystems*, v. 9, n. 1, pp. 61–69, 2013.

[59] ALBA, E., GARCIA-NIETO, J., JOURDAN, L., et al. "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms". In: *2007 IEEE congress on evolutionary computation*, pp. 284–290. IEEE, 2007.

[60] RÜCKSTIESS, T., OSENDORFER, C., VAN DER SMAGT, P. "Sequential feature selection for classification". In: *Australasian Joint Conference on Artificial Intelligence*, pp. 132–141. Springer, 2011.

[61] EID, H. F., HASSANIEN, A. E., KIM, T.-H., et al. "Linear correlation-based feature selection for network intrusion detection model". In: *International Conference on Security of Information and Communication Networks*, pp. 240–248. Springer, 2013.

[62] MIAO, J., NIU, L. "A survey on feature selection", *Procedia Computer Science*, v. 91, pp. 919–926, 2016.

[63] GNANA, D. A. A., BALAMURUGAN, S. A. A., LEAVLINE, E. J. "Literature review on feature selection methods for high-dimensional data", *International Journal of Computer Applications*, v. 975, pp. 8887, 2016.

[64] CAI, J., LUO, J., WANG, S., et al. "Feature selection in machine learning: A new perspective", *Neurocomputing*, v. 300, pp. 70–79, 2018.

[65] MA, S., HUANG, J. "Penalized feature selection and classification in bioinformatics", *Briefings in bioinformatics*, v. 9, n. 5, pp. 392–403, 2008.

[66] ZOU, H., HASTIE, T. "Regularization and variable selection via the elastic net", *Journal of the royal statistical society: series B (statistical methodology)*, v. 67, n. 2, pp. 301–320, 2005.

[67] BREIMAN, L. "Random forests", *Machine learning*, v. 45, n. 1, pp. 5–32, 2001.

[68] FISHER, A., RUDIN, C., DOMINICI, F. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously". 2018.

[69] MOLNAR, C. *Interpretable Machine Learning*. Lulu. com, Lulu. com, 2020.

[70] FERRI, F. J., PUDIL, P., HATEF, M., et al. "Comparative study of techniques for large-scale feature selection", *Machine Intelligence and Pattern Recognition*, v. 16, pp. 403–413, 1994.

[71] PUDIL, P., NOVOVIČOVÁ, J., KITTLER, J. "Floating search methods in feature selection", *Pattern recognition letters*, v. 15, n. 11, pp. 1119–1125, 1994.

[72] RASCHKA, S. "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack", *The Journal of Open Source Software*, v. 3, n. 24, abr. 2018. doi: 10.21105/joss.00638. Disponível em: <http://joss.theoj.org/papers/10.21105/joss.00638>.

[73] KLEIN, J. P., MOESCHBERGER, M. L., OTHERS. *Survival analysis: techniques for censored and truncated data*, v. 1230. Springer, 2003.

[74] COLLINS, F. S., VARMUS, H. "A new initiative on precision medicine", *New England journal of medicine*, v. 372, n. 9, pp. 793–795, 2015.

[75] TOSADO, J., ZDILAR, L., ELHALAWANI, H., et al. "Clustering of Largely Right-Censored Oropharyngeal Head and Neck Cancer Patients for Discriminative Groupings to Improve Outcome Prediction", *Scientific Reports*, v. 10, n. 1, pp. 3811, dez. 2020. ISSN: 2045-2322. doi: 10.1038/s41598-020-60140-0. Disponível em: <http://www.nature.com/articles/s41598-020-60140-0>.

[76] CHEN, D., WANG, H., SHENG, L., et al. "An Algorithm for Creating Prognostic Systems for Cancer", *Journal of Medical Systems*, v. 40, n. 7, pp. 160, jul. 2016. ISSN: 1573-689X. doi: 10.1007/s10916-016-0518-1.

[77] AHLQVIST, E., STORM, P., KÄRÄJÄMÄKI, A., et al. "Novel subgroups of adult-onset diabetes and their association with outcomes: a data-driven cluster analysis of six variables", *The lancet Diabetes & endocrinology*, v. 6, n. 5, pp. 361–369, 2018.

[78] MANDUCHI, L., MARCINKEVIČS, R., MASSI, M. C., et al. "A Deep Variational Approach to Clustering Survival Data", *arXiv preprint arXiv:2106.05763*, 2021.

[79] ANTOLINI, L., BORACCHI, P., BIGANZOLI, E. "A time-dependent discrimination index for survival data", *Statistics in medicine*, v. 24, n. 24, pp. 3927–3944, 2005.

[80] BOUVEYRON, C., CELEUX, G., MURPHY, T. B., et al. *Model-based clustering and classification for data science: with applications in R*, v. 50. Cambridge University Press, 2019.

[81] ZELLER, C. B., CABRAL, C. R. B., LACHOS, V. H., et al. "Finite mixture of regression models for censored data based on scale mixtures of normal distributions", *Advances in Data Analysis and Classification*, v. 13, n. 1, pp. 89–116, mar. 2019. ISSN: 1862-5355. doi: 10.1007/s11634-018-0337-y. Disponível em: <https://doi.org/10.1007/s11634-018-0337-y>.

[82] LACHOS, V., LOPEZ, E., CHEN, K., et al. "Finite mixture modeling of censored data using the multivariate Student- distribution", *Journal of Multivariate Analysis*, v. 159, maio 2017. doi: 10.1016/j.jmva.2017.05.005.

[83] DE ALENCAR, F. H. C., GALARZA, C. E., MATOS, L. A., et al. "Finite mixture modeling of censored and missing data using the multivariate skew-normal distribution", *arXiv:2009.10826 [stat]*, set. 2020. Disponível em: <http://arxiv.org/abs/2009.10826>. arXiv: 2009.10826.

[84] WANG, W.-L., CASTRO, L., LACHOS, V., et al. "Model-based clustering of censored data via mixtures of factor analyzers", *Computational Statistics & Data Analysis*, v. 140, jun. 2019. doi: 10.1016/j.csda.2019.06.001.

[85] MOULI, S. C., TEIXEIRA, L., NEVILLE, J., et al. "Deep lifetime clustering", *arXiv preprint arXiv:1910.00547*, 2019.

[86] BORDES, L., CHAUVEAU, D. "Stochastic EM algorithms for parametric and semiparametric mixture models for right-censored lifetime data", *Computational Statistics*, v. 31, n. 4, pp. 1513–1538, 2016.

[87] DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B. "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society: Series B (Methodological)*, v. 39, n. 1, pp. 1–22, 1977.

[88] CELEUX, G., CHAUVEAU, D., DIEBOLT, J. "Stochastic versions of the EM algorithm: an experimental study in the mixture case", *Journal of statistical computation and simulation*, v. 55, n. 4, pp. 287–314, 1996.

[89] DE ULLIBARRI, I. L., JÁCOME, M. A. "survPresmooth: An R Package for Presmoothed Estimation in Survival Analysis", *Journal of Statistical Software*, v. 54, n. 11, pp. 1–26, 2013. doi: 10.18637/jss.v054.i11.

[90] KNAUS, W. A., HARRELL, F. E., LYNN, J., et al. "The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults", *Annals of internal medicine*, v. 122, n. 3, pp. 191–203, 1995.

[91] DISPENZIERI, A., KATZMANN, J. A., KYLE, R. A., et al. "Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population". In: *Mayo Clinic Proceedings*, v. 87, pp. 517–523. Elsevier, 2012.

[92] FOEKENS, J. A., PETERS, H. A., LOOK, M. P., et al. "The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients", *Cancer research*, v. 60, n. 3, pp. 636–643, 2000.

[93] CURTIS, C., SHAH, S. P., CHIN, S.-F., et al. "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups", *Nature*, v. 486, n. 7403, pp. 346–352, 2012.

[94] LEMESHOW, S., MAY, S., HOSMER JR, D. W. *Applied survival analysis: regression modeling of time-to-event data*. John Wiley & Sons, 2011.

[95] MANTEL, N. "Evaluation of survival data and two new rank order statistics arising in its consideration", *Cancer Chemother Rep*, v. 50, pp. 163–170, 1966.

[96] HERNÁN, M. A., ROBINS, J. M. "Causal inference: what if", *Boca Raton: Chapman & Hill/CRC*, v. 2020, 2020.

[97] PEARL, J. "Causal inference". In: *Causality: Objectives and Assessment*, pp. 39–58, 2010.

[98] LUQUE-FERNANDEZ, M. A., SCHOMAKER, M., RACHET, B., et al. "Targeted maximum likelihood estimation for a binary treatment: A tutorial", *Statistics in medicine*, v. 37, n. 16, pp. 2530–2546, 2018.

Appendix A

Features' dictionary

Feature	Group	Data type	Further Description	Values
Peak METs	Exercise stress test	Continuous	Peak metabolic equivalents	2.0-16.8, 7.4±2.1
Peak DBP	Exercise stress test	Continuous	Peak diastolic blood pressure (mmHg)	50-140, 74±11 Missing: 649 (5)
Peak SBP	Exercise stress test	Continuous	Peak systolic blood pressure (mmHg)	90-260, 159±27 Missing: 524 (4)
Peak HR	Exercise stress test	Continuous	Peak heart rate (bpm)	40-110, 68±12 Missing: 510 (4)
Resting DBP	Exercise stress test	Continuous	Resting diastolic blood pressure (mmHg)	50-120, 73±10 Missing: 574 (4)
Resting SBP	Exercise stress test	Continuous	Resting systolic blood pressure (mmHg)	80-180, 118±18 Missing: 607 (5)
Resting HR	Exercise stress test	Continuous	Resting heart rate (bpm)	40-110, 68±12 Missing: 510 (4)
Age	Clinical	Continuous	(years)	21-92, 60±11
BMI	Clinical	Continuous	Body mass index	15-60, 29±5 Missing: 1,414 (11)
Number of vessels diseased	Clinical	Categorical	—	One vessel = 4,811 (36) Two vessels = 3,961 (30) Three vessels = 3,705 (28) Left main = 885 (7)
Management strategy for CAD	Clinical	Categorical	Cardiac referral at time of catheterization	Medical management = 1,535 (11) PCI = 8,159 (61) CABG = 1,724 (13) Missing: 1,944 (15)
Sex	Clinical	Binary	—	Men = 10,962 (82) Women = 2,400 (18)
Indication for CA	Clinical	Categorical	Indication for catheterization	Stable angina = 3,195 (24) Unstable angina = 2,194 (16) Myocardial infarction = 7,973 (60)
LVEF	Clinical	Categorical	Left ventricular ejection fraction	>50% = 9,340 (70) 35-50% = 2,414 (18) 20-34% = 371 (3) <20% = 42 (0) Missing: 1,195 (9)
Hypertension	Comorbidity	Binary	—	Yes = 8,041 (60) No = 5,321 (40)
Diabetes	Comorbidity	Binary	Diabetes mellitus (Type I or Type II)	Yes = 2,721 (20) No = 10,641 (80)
Dyslipidemia	Comorbidity	Binary	—	Yes = 9,086 (68) No = 4,276 (32)
Current smoking	Comorbidity	Binary	Smoking at the time of its measurement	Yes = 3,550 (27) No = 9,812 (73)
CHF	Comorbidity	Binary	Congestive heart failure	Yes = 735 (6) No = 12,627 (95)
COPD	Comorbidity	Binary	Chronic obstructive pulmonary disease	Yes = 1,329 (10) No = 12,033 (90)
Family History	Comorbidity	Binary	Family history of coronary artery disease	Yes = 3,639 (27) No = 8,556 (64) Missing: 1,167 (9)
PVD	Comorbidity	Binary	Peripheral vascular disease	Yes = 587 (4) No = 12,775 (96)
CEVD	Comorbidity	Binary	Cerebral vascular disease	Yes = 481 (4) No = 12,881 (96)
Renal insufficiency	Comorbidity	Binary	—	Yes = 194 (1) No = 13,168 (99)
Malignancy	Comorbidity	Binary	—	Yes = 476 (4) No = 12,886 (96)

Table A.1: Feature's dictionary.

Appendix B

Hyperparameters

Model	Hyperparameter
Logistic Hazard	num_nodes: [[1], [2], [3], [4], [5], [8], [2]*2, [3]*2, [4]*2, [5]*2, [8]*2, [2]*3, [3]*3 , [4]*3, [5]*3, [8]*3], dropout: [0, 0.2, 0.5,0.8], num_durations: [10, 50, 100]
DeepSurv	num_nodes: [[2], [3], [5], [8], [10], [16], [32], [2]*2, [3]*2, [5]*2, [8]*2, [10]*2, [16]*2, [32]*2, [2]*3, [3]*3 , [5]*3, [8]*3 , [16]*3, [32]*3], dropout: [0, 0.2, 0.5, 0.8]
Random Survival Forest (RSF)	max_depth: [5], min_samples_leaf: [50, 150], n_estimators: [50, 100]
Survival Tree	max_depth: [2,3,4,5], min_samples_leaf: [30, 100, 150, 300, 500, 750, 1000], min_samples_split: [4, 8, 50, 100]
Cox Proportional Hazard (CPH)	l2_reg: [1e-4, 1e-2, 1], lr: [1,0.5, 0.01]

Table B.1: Hyperparameters used for the model benchmark in chapter 5

Appendix C

Hyperparameters used in SurvMixClust

Model	Hyperparameters
RSF	max tree depth: [3, 5, 8], min # of samples required to be at a leaf node: [20, 50, 150], number of trees: [50, 100, 200]
CoxPH	l2 regularization factor: [0.0001, 0.01, 1], lr: [1, 0.5, 0.01]
K-means Survival	number of clusters: [2, 3, 4, 5, 6, 7]
SurvMixClust	number of clusters: [2, 3, 4, 5, 6, 7]
Logistic Hazard	neural network architecture: [[1], [2], [3], [5], [8], [2, 2], [3, 3], [5, 5], [8, 8], [2, 2, 2], [3, 3, 3], [5, 5, 5], [8, 8, 8]], dropout: [0, 0.2, 0.5, 0.8], number of divisions of the output time axis: [10, 50, 100]
SCA	default from the paper's code

Table C.1: Hyperparameters used for the benchmark that tested **SurvMixClust**'s performance.

Appendix D

Mathematical Derivations

Using the notation in table 7.1 and the same assumptions, we give the details of the derivations needed to arrive at equations 7.4, 7.5, 7.7 and 7.6. Firstly, for equation 7.4, remember how the pertinent random variables were defined: $T = \min\{T^*, C^*\}$ and $D = \mathbb{1}\{T^* \leq C^*\}$. These probability events are rewritten in a better-equipped format for our aims:

$$\begin{aligned}
\mathrm{P}(T = t, D = d) &= \mathrm{P}\left(T^* = t, C^* \geq t\right)^{\mathbb{I}(d=1)} \mathrm{P}\left(T^* > t, C^* = t\right)^{\mathbb{I}(d=0)} \\
&= \left[\mathrm{P}\left(T^* = t\right) \mathrm{P}\left(C^* \geq t\right)\right]^{\mathbb{I}(d=1)} \left[\mathrm{P}\left(T^* > t\right) \mathrm{P}\left(C^* = t\right)\right]^{\mathbb{I}(d=0)} \\
&= \left[f(t)\left(S_{C^*}(t) + f_{C^*}(t)\right)\right]^{\mathbb{I}(d=1)} \left[S(t) f_{C^*}(t)\right]^{\mathbb{I}(d=0)} \\
&= \left[f(t)^{\mathbb{I}(d=1)} S(t)^{\mathbb{I}(d=0)}\right] \left[f_{C^*}(t)^{\mathbb{I}(d=0)} \left(S_{C^*}(t) + f_{C^*}(t)\right)^{\mathbb{I}(d=1)}\right]
\end{aligned} \tag{D.1}$$

Equation 7.6 for $r_{nk}^{(t)}$ is derived with the help of the fact that $h(t) = \left[f_{C^*}(t)^{\mathbb{I}(d=0)} \left(S_{C^*}(t) + f_{C^*}(t)\right)^{\mathbb{I}(d=1)}\right]$ and $h(t_n|z_n = l) = h(t_n)$:

$$
\begin{aligned}
r_{nk}^{(t)} &= \mathrm{P}\left(z_n = k \mid t_n, d_n, x_n; \boldsymbol{\theta}^{(t)}\right) \\
&= \frac{\mathrm{P}\left(t_n, d_n \mid z_n = k; \boldsymbol{\theta}^{(t)}\right) \mathrm{P}\left(z_n = k \mid x_n; \boldsymbol{\theta}^{(t)}\right)}{\sum_{l=1}^{K} \mathrm{P}\left(t_n, d_n \mid z_n = l; \boldsymbol{\theta}^{(t)}\right) \mathrm{P}\left(z_n = l \mid x_n; \boldsymbol{\theta}^{(t)}\right)} \\
&= \frac{\mathrm{P}\left(t_n, d_n \mid z_n = k; \boldsymbol{\theta}^{(t)}\right) \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} \mathrm{P}\left(t_n, d_n \mid z_n = l; \boldsymbol{\theta}^{(t)}\right) \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \\
&= \frac{\mathrm{P}\left(t_n, d_n \mid z_n = k; \boldsymbol{\theta}^{(t)}\right) \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} \mathrm{P}\left(t_n, d_n \mid z_n = l; \boldsymbol{\theta}^{(t)}\right) \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \\
&= \frac{f\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=0)} h(t_n | z_n = k) \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} f\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=0)} h(t_n | z_n = l) \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \\
&= \frac{f\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=0)} h(t_n) \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} f\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=0)} h(t_n) \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \\
&= \frac{f\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=0)} \cancel{h(t_n)} \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} f\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=0)} \cancel{h(t_n)} \tau_l^{(t)}\left(\boldsymbol{x}_i\right)} \\
&= \frac{\left[f\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k\right)^{\mathbb{I}(d_n=0)}\right] \tau_k^{(t)}\left(\boldsymbol{x}_i\right)}{\sum_{l=1}^{K} \left[f\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_l\right)^{\mathbb{I}(d_n=0)}\right] \tau_l^{(t)}\left(\boldsymbol{x}_i\right)}
\end{aligned}
\tag{D.2}
$$

Equation 7.7 is a direct application of 7.5:

$$
\begin{aligned}
LL^t(\boldsymbol{\theta}) &= \sum_n \mathbb{E}_{q_n^t(z_n)}\left[\log \mathrm{P}(T = t_n, D = d_n, Z = z_n \mid X = x_n; \boldsymbol{\theta})\right] \\
&= \sum_n \mathbb{E}_{q_n^t(z_n)}\left[\log \mathrm{P}_{\boldsymbol{\theta}}(Z = z_n \mid X = x_n) f\left(t_n \mid \theta_{z_n}^{(t)}\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_{z_n}^{(t)}\right)^{\mathbb{I}(d_n=0)} h(t_n)\right] \\
&= \sum_n \mathbb{E}_q\left[\log \tau_{z_n}\left(\boldsymbol{x}_n\right) f\left(t_n \mid \theta_{z_n}^{(t)}\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_{z_n}^{(t)}\right)^{\mathbb{I}(d_n=0)} h(t_n)\right] \\
&= \sum_n \mathbb{E}_q\left[\log \prod_k \left(\tau_k\left(\boldsymbol{x}_n\right) f\left(t_n \mid \theta_k^{(t)}\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k^{(t)}\right)^{\mathbb{I}(d_n=0)} h(t_n)\right)^{z_{nk}}\right] \\
&= \sum_n \sum_k \mathbb{E}\left[z_{nk}\right] \log \tau_k\left(\boldsymbol{x}_n\right) + \\
&\quad + \sum_n \sum_k \mathbb{E}\left[z_{nk}\right] \log f\left(t_n \mid \theta_k^{(t)}\right)^{\mathbb{I}(d_n=1)} S\left(t_n \mid \theta_k^{(t)}\right)^{\mathbb{I}(d_n=0)} + \\
&\quad + \sum_n \sum_k \mathbb{E}\left[z_{nk}\right] \log\left(h(t_n)\right)
\end{aligned}
\tag{D.3}
$$

Finally, the terms that are searched to find the value of $q_i^{(l)}$ were arrived at by the following straightforward relation. Notice that the denominator is not dependent on k, so It can be taken out of the arg max:

$$
\begin{aligned}
q_i^{(t)} &= \arg\max_k \left(r_{ik}^{(t)} \right) \\
q_i^{(t)} &= \arg\max_k \frac{\left[f\left(t_i \mid \theta_k^{(t)} \right)^{\mathbb{I}(d=1)} S\left(t_i \mid \theta_k^{(t)} \right)^{\mathbb{I}(d=0)} \right] \tau_k^{(t)}\left(\boldsymbol{x}_i \right)}{\sum_{l=1}^{K} \left[f\left(t_i \mid \theta_l^{(t)} \right)^{\mathbb{I}(d=1)} S\left(t_i \mid \theta_l^{(t)} \right)^{\mathbb{I}(d=0)} \right] \tau_l^{(t)}\left(\boldsymbol{x}_i \right)} \\
q_i^{(t)} &= \arg\max_k \left(\left[f\left(t_i \mid \theta_k^{(t)} \right)^{\mathbb{I}(d=1)} S\left(t_i \mid \theta_k^{(t)} \right)^{\mathbb{I}(d=0)} \right] \tau_k^{(t)}\left(\boldsymbol{x}_i \right) \right)
\end{aligned}
\tag{D.4}
$$

Appendix E

Visualizing the survival functions generated by SurvMixClust

Figure E.1 displays the survival functions of the clusterized populations which were generated by **SurvMixClust**. Each card corresponds to a dataset, all of which were present inside 7.5.1. A randomly selected trained model for each number of clusters (2, 3, 4, 5, 6, 7) is used to cluster the test set. The survival function of these grouped populations, via Kaplan-Meier, is exhibited inside each card.

As mentioned in the main text, most inferred clusterization displays good qualitative distribution of heterogeneous populations, discovering different survival profiles. A further point is the balanced character of each cluster; no cluster has too few members, as can be immediately seen by the size of the Kaplan-Meier's confidence interval.

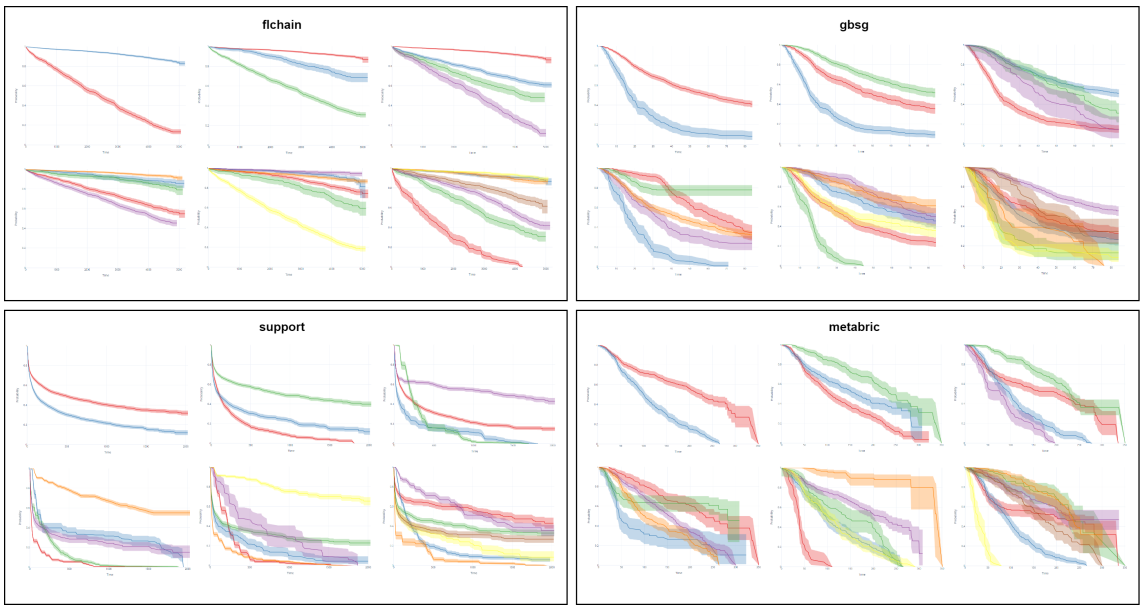

Figure E.1: Inferred clusterizations generated by **SurvMixClust**. Each card corresponds to a dataset. A randomly selected trained model for each number of clusters is used to cluster the test set. The survival function of these populations, via Kaplan-Meier, is exhibited inside each card.