TU WIEN Informatics

A Comparison Study of Parametric and Machine Learning Survival Analysis Models to Predict Customer Churn in the Edtech Sector

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Data Science

by

Benjamin Lee, BA/BE(Hons)
Registration Number 12112693

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

Vienna, 27th January, 2025

Benjamin Lee	Peter Filzmoser

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Declaration of Authorship

Benjamin Lee, BA/BE(Hons)

I hereby declare that I have written this Masters Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 27th January, 2025

Benjamin Lee

Acknowledgements

It is a cliché, and technically not fully accurate to talk here about *nani gigantum humeris insidentes*, but this thesis would not be possible at all were it not for all the people who helped and supported me through this academic journey.

First and foremost, my supervisor Dr Peter Filzmoser whose guidance and expertise were invaluable in writing this thesis. Not only did he support me in finessing the work, but also provided counsel and direction when my literature searches came up blank.

I would also want to take the time to acknowledge and thank the examiners who will be reading this thesis and participating in my defensio. Your time and patience is appreciated.

To GoStudent and the Finance team, especially Alexander Schaffgotsch and Paul Krall, thank you for your support in this endeavour and generosity in providing the data that made this work possible.

Michael Weingartner at the Department of Computational Statistics has been an absolute star at setting me up with the hardware system I used for this thesis and I appreciate his patience at dealing with my many questions, particularly so close to Christmas.

In my literature review, I would particularly like to thank Luca Badolato, Nicholas Irons, and Weichi Yao for their kindness in answering questions seeking clarification and expansion on some aspects of the papers they had written.

Of course, many other people not directly part of this thesis have also made this academic journey far more bearable and even joyful at times. To my friends and everyone with whom I talked about this thesis, thank you from the bottom of my heart.

Special acknowledgements are necessary to the Fachschaft crew, who were my first friends in Vienna and still great friends today. In particular, I would like to give a shoutout to: Alicja, Bettina, Christian, Elisabeth, Fränzi, Gunnar, Ivan, Łukasz, Valentin B, and Valentin M.

Most important of all, I'd like to thank my family: my Mum, my sister Amanda, and my darling wife Marieke. Your love and belief in me have helped me not only in this journey, but through my life. I love you.

Vienna, 27th January 2025

Abstract

This thesis explores the application of survival analysis models to predict customer churn in the edtech sector, an area of growing importance for subscription-based businesses. By leveraging statistical and machine learning techniques, the study aims to improve retention models over existing heuristic methods and identify key variables influencing churn behaviour. The research focuses on using survival analysis, a statistical framework adept at handling censored data, to predict customer churn and retention duration, providing more precise and actionable insights.

Drawing from a dataset comprising several hundred thousand customer records with both time-variant and time-invariant features, this study evaluates classical survival models, including Kaplan-Meier and Cox Proportional Hazards models, as well as advanced machine learning techniques like Random Survival Forests and Gradient Boosting Machines. The incorporation of time-variant data, a novel aspect of this study, enhances model sophistication and predictive capability.

Results demonstrate that machine learning models outperform traditional heuristic approaches, achieving higher concordance index and lower integrated Brier scores. Permutation importance methods highlighted variables and features which strongly affected survival time and its inverse: customer churn. Time-variant data was found to further improve model performance although caution must be exercised to ensure correct interpretation of results.

This work contributes to the literature by extending survival analysis applications to the edtech sector, where customer retention is critical for sustainable growth. The developed models form a basis as a testbed for further analysis as new hypothesised variables come in for testing. However, the lack of readily-available libraries for time-variant analysis, particularly in Python both highlight the cutting edge nature of time-variant survival analysis, as well as the risks of productionising time-variant methodologies.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Problem Statement

In subscription businesses where revenues come from recurring payments by customers, customer retention is considered to be a key component in the sustainability and success of a business. This is because the cost of acquisition is substantially more expensive than that of retaining a customer, with five times being the general rule of thumb[1].

As a result, the prediction of how long a customer would stay (customer retention) as well as what customers are at risk of leaving (customer churn) are crucial in order to reduce costs of marketing for acquisition and maintain a sustainable profit. Even a modest 5% increase in customer retention rate translates into customer lifetime value (LTV) improvements of 35-95%[2].

Due to its focus on predicting time-to-event, survival analysis is a commonly-used framework for predicting customer churn and retention, particularly in situations where the provided data is censored for whatever reason[3]. While there has been a reasonable coverage of academic papers on the subject, the vast majority of papers published on survival analysis in a customer churn context since 2015 have been in the telecommunications, gaming, and finance domains.[4]. It is therefore still an open question on how applicable these models are in other domains.

In the edtech sector, churn and survival models tend to use heuristic and/or simple parametric models which do not perform well when market trends or customer makeup changes. In addition, the little research conducted in this space is focused more on course and degree-level in traditional educational institutions which only looks at demographic data[5, 6].

The sponsor of this project is a leading provider of online tutoring services. Being in the online space, it is able to obtain data in addition to demographic data (e.g. desired goals,

opinions of tutors, etc.) which can potentially provide better indications of customer satisfaction and encourage proactive intervention to improve the service.

It is therefore of interest to find models which both beat the current heuristic models as well as find variables which have an impact of a customer's desire to stay with the subscription.

1.2 Aim of the Work

The goal of this thesis is to determine if a better-performing customer retention model can be built with available variables compared to naïve or heuristic alternatives. In addition, the thesis intends to also explore key variables which affect churn behaviour as well as the extent to which churn behaviour can be influenced.

The research questions asked in this thesis were: -

RQ1 What are the best-performing metrics by Concordance Index (C-index) and Integrated Brier Score (IBS) which are obtainable against a test set by various Survival Analysis models on the provided customer churn data?

RQ2 Can we reject the null hypothesis that the results of these more sophisticated Survival Analysis models are the same as those from simple naïve/heuristic models?

RQ3 What variables best explain variation in customer churn behaviour?

1.3 Structure of this Thesis

Following the introduction in the current Chapter (Chapter 1), we will cover general background of customer analytics and churn, as well as a basic introduction to survival analysis in the Theoretical Background in Chapter 2. Chapter 3 then covers the state-of-the-art in survival analysis as well as the various applications thereof that have been published. In Chapter 4, we will then discuss the dataset and details of the methodological approach and experiment setup to achieve the goals described above in Section 1.2. We then present the results of the experiment in Chapter 5, followed by a discussion in Chapter 6, and concluding remarks in Chapter 7 which will cover the contributions and limitations of this thesis, as well as recommendations for future tasks to build on this thesis.

CHAPTER 2

Theoretical Background

2.1 Customer Analytics in Subscription Businesses

Due to the recent rise of the so-called "sharing economy", where goods and services are shared amongst multiple users instead of being owned or used by a single party, new business models have emerged which work appropriately within this context. One of these business models is the subscription-based business model which has expanded well beyond its 17th century roots in the publications industry [7].

Organisations with subscription-based business models tend to own unique and hard-to-replicate assets, which are then converted into a set of standardised products and services for customers to use. These customers are then encouraged to use the product as frequently as possible to maximise their investment in the value proposition as they tend to be locked in via contractual arrangements or high switching costs [8]. An example of this would be the meal-box company HelloFresh.

In the subscription model, revenue is generated by customers through monthly subscription fees, often structured with different pricing tiers based on product. This monthly recurring revenue (MRR) structure differs fundamentally from traditional businesses as it faces a higher upfront cost in marketing and sales expenses with these customer acquisition costs (CAC) being recouped via subscription fees over the customer's lifetime as the first monthly fee rarely exceeds the CAC [9]. In fact, it can take approximately 12 months for a typical subscription business to recoup its CAC (Refer Figure 2.1) [10].

The subscription business therefore focuses heavily on relationships with their customers as retaining customers is crucial to the profitability and sustainability of the company, especially once a customer passes the breakeven threshold. In order to do this, performance evaluation of customers through the usage of insightful metrics and key performance indicators (KPIs) is essential for the company to make relevant managerial decisions

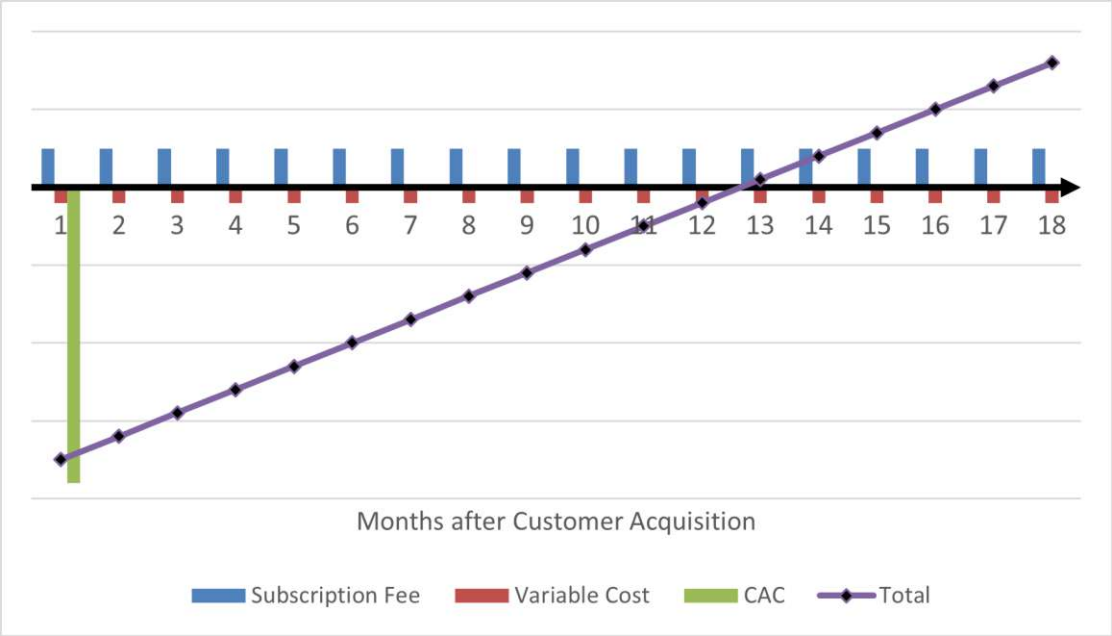

Figure 2.1: Customer cumulative net cashflow after acquisition. An illustrative example showing the various revenue and cost components over time showing a typical 12 month breakeven point. Any churn of a customer prior to this point equates to an overall loss for the company

and maintain existing customers [11]. Some of the key KPIs relevant to this thesis are described in the following sections.

2.1.1 Lifetime value (LTV)

LTV, also known as Customer Lifetime Value (CLV) in the business is a basic metric reflecting the potential value of any given customer to the business. In more specific terms, LTV is *the present value of all future cash flows attributed to a customer relationship* [12].

While the specific calculation can vary based on business context, one can define a customer's expected LTV at the start of the customer relationship by the formula:

$$LTV_i = \sum_{t=1}^{\tau} \left(\frac{E(\widetilde{V}_t)}{(1+d)^{t-1}} \right) - C_0 \tag{2.1}$$

where τ represents the chosen observation timeframe, $\widetilde{V}_t$ represents the customer's net contribution in period t, d represents the discount rate, and C_0 represents the CAC which is effectively paid at $t = 0$.

In terms of the observation timeframe τ, while theoretically one could choose the customer's entire lifetime, i.e. $\tau = \infty$, in practice, many firms consider three years to be a good estimate for a horizon where the business environment would not substantially change [13]. In addition, having a limited time horizon provides mathematically easier calculations as will be described in Section 2.2.

In terms of predicting LTV, the discount and CAC components stay reasonably static while the net contribution component $\widetilde{V}_t$ is affected by many variables which are customer-specific. Ultimately, these variables affect one or more of the three components which make up net contribution: -

1. the expected revenues generated by a customer in period t, usually comprising a MRR component and an ancillary component. In a subscription business model, the monthly recurring component usually dominates and is thus reasonably static
2. the expected costs of serving the customer in period t, usually cost of goods sold (COGS) and general fixed costs
3. the duration of the relationship with the customer

In a subscription business model, the first two components of calculating net contribution $\widetilde{V}_t$ tend not to vary significantly between customers whereas the duration of the relationship can take any value of t from 0 to τ. Obtaining this value requires a customer retention model.

2.1.2 Retention and Churn

Analysing customer retention and its inverse customer churn is a critical focus for companies with a subscription business model. Customer retention is defined as the probability of a customer of still being a customer at any given period t. In customer retention modelling, there are two broad classes:

Always-a-Share

In an always-a-share model, customers can switch between customer states as well as leave the company and return again. These are usually for businesses where switching costs are low and contracts are generally not enforced. An always-a-share model is usually modelled as Markov chains. In the context of subscription-based businesses without contracts, a Markov model has been proven to have a high risk of being substantially wrong [14].

Lost-for-Good

When a customer leaves the company, they are assumed to be lost forever and will never return. This is generally more appropriate for contract-based approaches and is the basis

for the vast majority of classical retention models, including the one we will be looking at in the thesis. While there is an argument that LTV is understated in classical models because it does not take returning behaviour into account [14], we will suggest a possible way around this in Section 4.

2.2 Survival Analysis

In order to analyse and forecast churn, this thesis looks at a class of statistical methods known as survival analysis. The outcome variable of interest is almost always *time until an event occurs* where an event can be any change in state, be it death, recidivism, or more specific for our context, customer churn [15].

At first glance, one may think that normal regression and classification methods would be sufficient to forecast churn and retention; however, customer churn data have some characteristics which make it challenging to address with traditional methods:

1. **Censoring:** Any customer data obtained will inevitably include customers who are still customers with the company; in other words, we *do not know the actual survival time when we train our models.* (Refer Figure 2.2). While various sub-types of censoring exist, in the context of customer churn, only right-censoring (i.e. we do not know the final time of some subjects) matters.*
2. **Skewness:** Times-to-event are rarely distributed normally; instead, customer churn curves tend to be characterised by a disproportionate number of events early on, followed by another disproportionate number of events during contract end periods (Refer Figure 2.3)

The outcomes of survival analysis are quantitatively expressed in a **survival function** $S(t)$ and a **hazard function** $h(t)$.

2.2.1 Survival Function

The survival function $S(t)$ is defined as the probability that a subject does not have an event occurrence at time t. It can also be defined as the complementary cumulative distribution function of the subject's lifetime. Mathematically, it is defined as:

$$\begin{aligned} S_i(t) &= P(T_i > t) \\ &= 1 - F_i(t) \\ &= 1 - \int_0^t f_i(u)du \end{aligned} \tag{2.2}$$

*The other types of censoring are left-censoring, where we do not know the exact time of beginning e.g. in a study of cancer where the start point is when cancer is reported, we do not know exactly when prior to the reporting the cancer occurred; and interval-censoring, where subjects come in and out of a study when an event occurs e.g. in the same cancer study, follow-up is only done at $t = 3$ and $t = 6$ but the event occurred sometime in between.

where T_i represents the survival time, $F_i(t)$ represents the cumulative distribution function, and $f_i(u)$ represents the probability density function of subject i. All survival functions can be modelled as a curve from $t = 0$ to $t = \infty$ with the y-axis starting from 1 at $t = 0$ and reducing downwards towards 0 at $t = \infty$ (Refer Figure 2.4).

2.2.2 Hazard Function

A related function is the hazard function $h(t)_i$ which is the instantaneous potential per unit time for an event to occur, given that subject i has survived up to time t[15]. It can

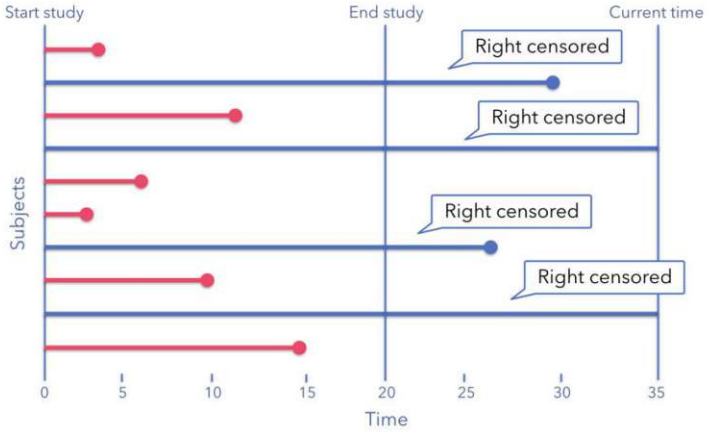

Figure 2.2: The concept of censoring illustrated. Dots show time of event. Red lines represent subjects who have churned, whereas blue lines represent subjects who are still customers at the end study point, i.e. the cut-off point for training data [16]

.

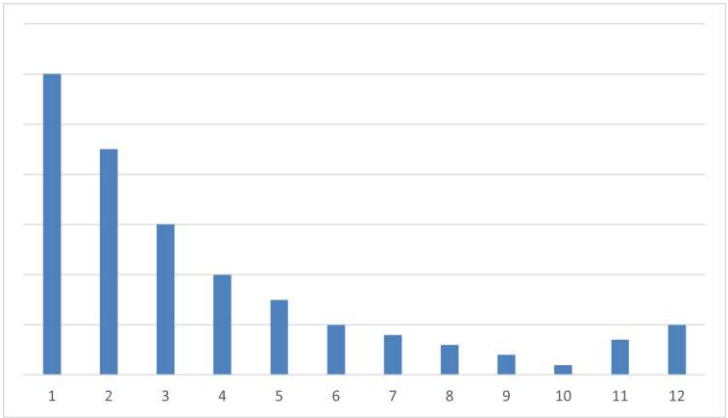

Figure 2.3: Customer churn by month. An illustrative example showing the larger number of early churners in months 1-3, as well as at the end of a 12-month contract in months 11 and 12.

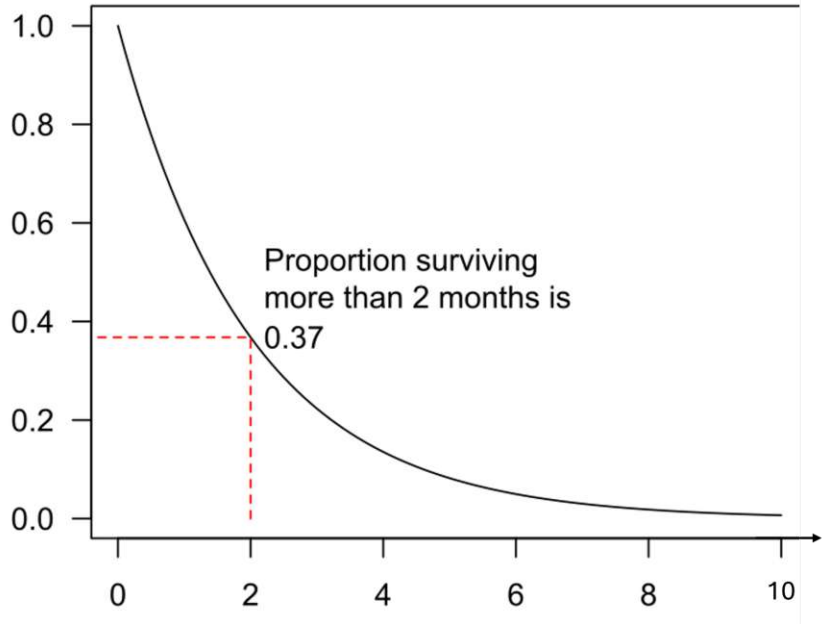

Figure 2.4: A theoretical survival function illustrating the start at 1 at $t = 0$ and approaching 0 as time approaches ∞ [17]

.

also be described as the instantaneous probability that an individual's survival time T lies between time t and $t + \Delta t$, if the survival time is already greater than or equal to t or the conditional failure rate [18]. Mathematically, we define this as:

$$h_i(t) = \lim_{\Delta t \to 0} \frac{P(t < T_i \leq t + \Delta t | T_i \geq t)}{\Delta t} \tag{2.3}$$

where T_i represents the survival time of subject i. It is worth noting that a hazard is not a probability but a probability rate, i.e. the value can change depending on the time unit used (days, weeks, months) and can also be a value greater than 1.

Hazard functions are distinctive with two main characteristics:

1. the value is always greater than zero
2. there is no upper bound

The hazard function can also be alternatively expressed as a cumulative hazard function $H(t)$, which is effectively the build-up of hazard over time, or mathematically:

$$H_i(t) = \int_0^t h_i(u)du \tag{2.4}$$

2.2.3 Linking the Survival and Hazard Functions

When either the hazard or survival functions are known, then one can also derive the other function, which reflects the churn/retention duality. From Equation (2.3), we have:

$$P(t \leq t + \Delta t | T_i \geq t) \tag{2.5}$$

From basic conditional probability, we also know that:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.6}$$

Putting these two together, Equation (2.3) then becomes:

$$\begin{aligned} h_i(t) &= \lim_{\Delta t \to 0} \frac{P(t < T_i \leq t + \Delta t | T_i \geq t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{P(t < T_i \leq t + \Delta t) \cap P(T_i \geq t)}{P(T \geq t) \cdot \Delta t} \end{aligned} \tag{2.7}$$

We know that the probability $P(t < T_i \leq t + \Delta t)$ and $P(T_i \geq t)$ at the same time is exactly the same as $P(t < T \leq t + \Delta t)$ as the latter is a subset of the former. Similarly, we also know from Equation (2.2) that $S_i(t) = P(T_i > t)$. Thus, we can put this and Equation (2.7) in order to produce:

$$\begin{aligned} h_i(t) &= \lim_{\Delta t \to 0} \frac{P(t < T_i \leq t + \Delta t)}{S(t) \cdot \Delta t} \\ &= \lim_{\Delta t \to 0} \frac{F_i(t + \Delta t) - F_i(t)}{\Delta t} \cdot \frac{1}{S_i(t)} \\ &= \frac{dF_i(t)}{dt} \cdot \frac{1}{S_i(t)} \end{aligned} \tag{2.8}$$

where $F_i(t)$ represents the cumulative distribution function of subject i. Put purely in terms of $S_i(t)$ and noting Equation (2.2), Equation (2.8) then boils down to:

$$\begin{aligned} h_i(t) &= \frac{d}{dt}(1 - S_i(t)) \cdot \frac{1}{S_i(t)} \\ &= -\frac{d}{dt} S_i(t) \cdot \frac{1}{S_i(t)} \end{aligned} \tag{2.9}$$

Now, if we apply the chain rule of differentiation to a composite function over a log function, we end up with a familiar looking result:

$$\begin{aligned} \frac{d}{dx} f(g(x)) &= f'(g(x)) \cdot g'(x) \\ \text{So, } \frac{d}{dx} \ln(f(x)) &= \frac{1}{f(x)} \cdot \frac{d}{dx} f(x) \\ &= \frac{d}{dx} f(x) \cdot \frac{1}{f(x)} \end{aligned} \tag{2.10}$$

This means that the hazard function can be written as:

$$\begin{aligned} h_i(t) &= -\frac{d}{dt} S_i(t) \cdot \frac{1}{S_i(t)} \\ &= -\frac{d}{dt} \ln(S_i(t)) \end{aligned} \tag{2.11}$$

And the cumulative hazard function is:

$$H_i(t) = -\ln(S_i(t)) \tag{2.12}$$

In summary, the cumulative hazard rate of subject i at time t can also be defined as the **negative logarithm of the survival function at time** t.

Taken together, the relationships between the various functions discussed above are shown in Figure 2.5.

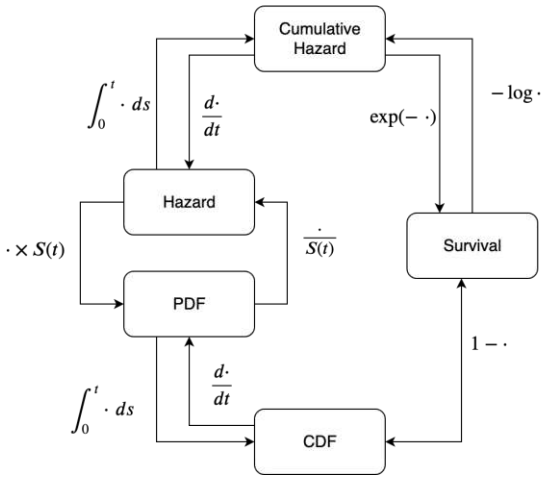

Figure 2.5: Mathematical entities used in survival analysis and the transformations between them [19]. PDF represents the probability distributive function, and CDF represents the cumulative distribution function

2.2.4 Censoring Assumptions

While generally more free from distribution assumptions, survival analysis models instead lean on several assumptions about how the data is censored. The most important of these assumptions is that of **independent censoring**, where the survival time T is independent of the censoring time C. Another way of looking at it is that within any given subgroup of subjects, any subjects censored at time t should be representative of subjects who are at risk at time t.

This assumption is necessary in order to obtain convergence on the likelihood function $L = f_i(t_i)^{(\delta_i)} S_i(C_i)^{1-\delta_i}$ which is the engine of most of the common models used to determine the distribution function of survival time T.

It is also worth noting that a stricter definition of censoring called **random censoring** exists and is commonly used by authors when they actually mean independent censoring [20]. In random censoring, the censoring mechanism is assumed to be completely unpredictable, i.e. censoring occurs by chance rather than any factors about the subjects themselves.

Under certain circumstances of data, the likelihood function might also be potentially used, as described by Lagakos [21], these are:

1. **Nonprognostic Censoring**: The censoring time of any given subject C_i only indicates that the survival time exceeds C_i and gives no further prognostic information about the survival time of the subject or any other subject in the dataset.
2. **Noninformative Censoring**: The censoring of a subject does not change the hazard rate of that subject, or in other words, the distribution of censoring times does not provide any information about the distribution of survival times and vice-versa. This property is also known as the constant-sum property [22].

Although methods have been suggested to test for non-informative censoring [23, 24], we will not go into significant detail as the nature of the dataset used in this thesis allows us to confirm independent censoring without requiring substantial effort in terms of distribution testing.

2.2.5 Typical Data Structure for Survival Analysis

When providing data for survival analysis, the general data layout is as illustrated in Table 2.1. The layout changes slightly when time-variable longitudinal data comes into play and that will be discussed in Section 2.5.2.

2.3 Survival Analysis Models

In general, survival analysis models are analogous to typical multivariate regression and classification problems, adapted to take censored data into account.

Subject	t	e	X_1	X_2	**...**	X_m
1	t_1	e_1	X_{11}	X_{21}	...	X_{m1}
2	t_2	e_2	X_{12}	X_{22}	...	X_{m2}
3	t_3	e_3	X_{13}	X_{23}	...	X_{m3}
...	...	...	...	...	...	...
n	t_n	e_n	X_{1n}	X_{2n}	...	X_{mn}

Table 2.1: The general data input structure for survival analysis. Each subject has an observed survival time t and an event variable e that shows whether the event in question has occurred or if the information has been censored. The subject then has as m explanatory variables (X).

2.3.1 Kaplan-Meier Estimator

The Kaplan-Meier estimator is the primary statistical tool used to estimate a true survival function from available data and can be considered the 'best' estimator of survival probability when no parametric structure is assumed [25]. In comparison to other parametric estimators, Kaplan-Meier has been found to be unbiased and has minimal efficiency losses in most cases, except for extreme situations such as extremely small sample sizes [26].

This estimator is a non-parametric estimator that only requires the time-to-event (or time-to-censoring) t, and the event status e for every subject. With this information, the survival function estimator $\hat{S}(t)$ is given by:

$$\hat{S}(t) = \prod_{j:t_i \leq t} (1 - \frac{e_j}{n_j})[27] \tag{2.13}$$

where t_j is the time at which at least one event e occurred, and n_j is the total number of subjects who have been censored or have not had the event yet at time t_j. To derive this, we take our survival function from Equation (2.2) and express it in terms of the previous time point $t-1$. Therefore:

$$\begin{aligned}
S_i(t) &= P(T_i > t) \\
&= P(T_i > t \cap T_i > t-1), && \text{if } T{>}t, \text{ then } T{>}t{-}1 \text{ must also be true.} \\
&= P(T_i > t | T_i > t-1) \cdot P(T_i > t-1), && \text{from conditional probability} \\
&= 1 - P(T_i \leq t | T_i > t-1) \cdot S_i(t-1) \\
&= 1 - P(T_i = t | T_i \geq t) \cdot S_i(t-1) \\
&= q_i(t) \cdot S_i(t-1)
\end{aligned} \tag{2.14}$$

We now call $1 - P(T_i = t | T_i \geq t-1)$ as $q_i(t)$ for ease of the next steps. By doing recursive expansion on Equation (2.14), we can conclude that:

$$\begin{aligned} S_i(t) &= q_i(t) \cdot S_i(t-1) \\ &= q_i(t) \cdot q_i(t-1) ... \cdot q_i(0) \\ &= \prod_{s=0}^{t} q_i(s) \end{aligned} \tag{2.15}$$

Taking the conditional probability formula again, we can express $q_i(s)$ as:

$$\begin{aligned} q_i(s) &= 1 - P(T_i = s | T_i \geq s) \\ &= 1 - \frac{P(T_i = s \cap T_i \geq s)}{P(T_i \geq s)} \\ &= 1 - \frac{P(T_i = s)}{P(T_i \geq s)} \end{aligned} \tag{2.16}$$

The probability of a subject i having an event at time s, $P(T_i = s)$ is the number of events e which occurred at time $t = s$ divided by the total population. Similarly, the probability that the event is greater than or equal to time s, $P(T_i \geq s)$ is the number of subjects who did not have an event occur until time s or later divided by the total population. Therefore, we can express $q_i(s)$ as:

$$\begin{aligned} q_i(s) &= 1 - \frac{P(T_i = s)}{P(T_i \geq s)} \\ &= 1 - \frac{e_s}{n_s}) \end{aligned} \tag{2.17}$$

Combining equations 2.17 and 2.15, taking into account that we are working off a sample rather than the whole population, and not strictly enforcing the need for time steps to be uniformly distributed, we finally return to the general Kaplan-Meier estimator equation at the start of this subsection:

$$\hat{S}(t) = \prod_{j:t_i \leq t} (1 - \frac{e_j}{n_j}) \tag{2.18}$$

The result of this formula on a given data sample is a step curve that shows the percentage of survival over time of a given sample, as shown in Figure 2.6.

One of the main drawbacks of the Kaplan-Meier model is that it is not able to take any subject covariates into account. In other words, while it is a very good an unbiased estimator of the true survival curve of a given population, it does not provide much explanatory power in terms of different variables within a group. However, as described in Section 2.4, there are ways to compare the survival curves of a small number of categorical variables with this method.

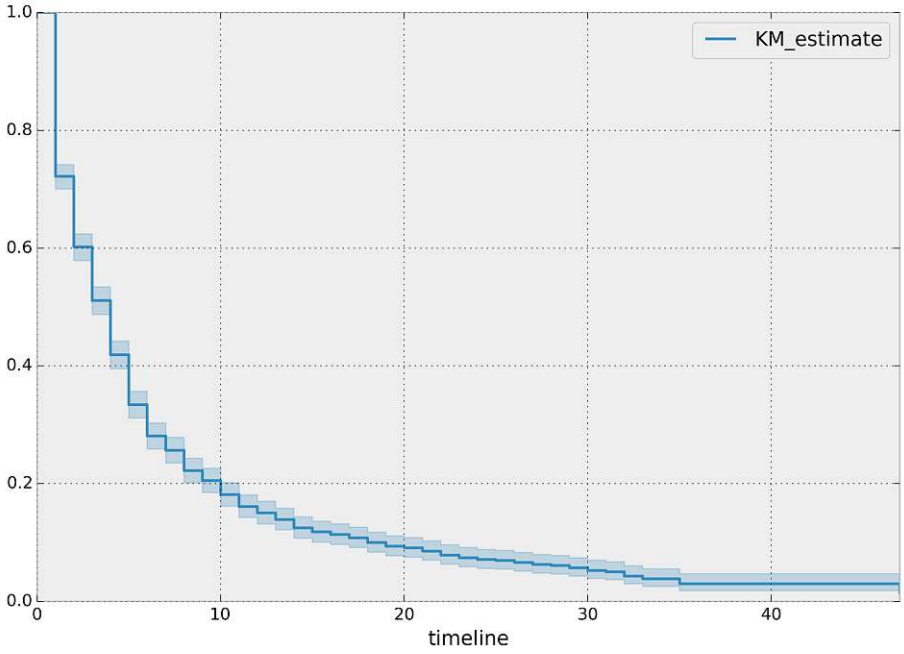

Figure 2.6: A Kaplan-Meier survival curve produced from the lifelines democracy and dictatorship dataset with confidence interval bars [19]

.

2.3.2 Cox Proportional Hazards

The Cox proportional hazards model is the most popular model used in survival analysis [28]. It is a semi-parametric model in that it makes no assumptions about the hazard function itself, but rather regresses covariates such that it has a multiplicative effect on the hazard rate at any point along the timeline.

Mathematically, we can define the Cox proportional hazard model as:

$$h(t, \mathbf{X}) = h_0(t) \times \exp(\mathbf{X}\boldsymbol{\beta}) \tag{2.19}$$

where $h_0(t)$ represents the base hazard (a non-parametric estimation) and the exponential part $\exp(\mathbf{X}\boldsymbol{\beta})$ represents the partial hazard, which is parametric and based entirely on the covariate matrix $\mathbf{X}$. In other words, the only time-sensitive part is the base hazard while the covariates only affect the partial hazard. The partial hazard then has an exponential function applied to it in order to ensure the hazard function remains positive.

As can be seen from Equation (2.19), the log-partial hazard acts in a very similar way to linear regression. In fact, most implementations in Python and R also contain the ability

to include L1 and L2 regularisation [29, 30] in the same way as an ElasticNet based on the work by Simon et al [31].

In the Cox proportional hazards model, one assumes that the relative risk of an event occurring β remains constant over time, i.e. **the proportional hazards assumption**. This is a direct consequence of the structure of the model; if we assume just a single covariate x, then our model will be:

$$h(t, x) = h_0(t) \cdot \exp(\beta x) \tag{2.20}$$

If we assume two subjects (a and b), then the hazards of subjects a and b will always remain constant for all values of t:

$$\begin{aligned} h_a(t, x_a) &= h_0(t) \cdot \exp(\beta x_a) \\ h_b(t, x_b) &= h_0(t) \cdot \exp(\beta x_b) \end{aligned}$$

$$\begin{aligned} \frac{h_a(t, x_a)}{h_b(t, x_b)} &= \frac{h_0(t) \cdot \exp(\beta x_a)}{h_0(t) \cdot \exp(\beta x_b)} \\ &= \frac{\exp(\beta x_a)}{\exp(\beta x_b)} \\ &= \exp\left[\beta \cdot (x_a - x_b)\right] \end{aligned} \tag{2.21}$$

Having a dataset which violates the proportional hazards assumption causes a reduction in overall power of the model as well as the predictive power of other covariates which themselves have constant hazard ratios due to the inferior fit of the model [32]. Various methods exist to test the appropriateness of the Cox model on the data such as assessing the goodness of fit by residuals [33] or using an extended Cox model as described in Section 2.5.3.

Nonetheless, Stensrud and Philos have argued that within certain limits, the hazard output of the model is a useful tool in assessing the general magnitude of a covariate's effects, i.e. one can interpret the hazard ratio as the *weighted average* of true hazard ratios over the time period [34]. Therefore, one must not *strictly* conform to the proportional hazards assumption, but it is always appropriate to have a dataset which approximately meets this assumption [35].

If we do know of variables which cause the dataset to break the proportional hazards assumption, it is also possible to adjust this by dividing the dataset into relatively homogeneous strata. This process is known as **stratification** [36]. This results in multiple base hazard curves for each of the different strata chosen.

In spite of the need for the proportional hazards assumption to at least approximately hold, Cox's proportional hazard model is very powerful particularly in situations where the underlying base hazard function is unknown. If the correct underlying model is some

already-known model, the Cox model is known to provide a reasonable approximation of an already-known model [15].

2.3.3 Parametric Models

When the underlying probability distribution of the dataset is known, one can use parametric models to model the survival function of the dataset. Once the underlying model is specified either in terms of the survival times or the logarithm of survival times, the model can be fitted and estimated using the maximum likelihood estimator.

The most popular of these parametric models are a class of models known as accelerated failure time (AFT) models [37]. In contrast to the Cox model, AFT models assume that **covariates are proportional with respect to survival time**. Mathematically, this is expressed as:

$$S_2(t) = S_1(\gamma t) \quad \text{for all } t \geq 0 \tag{2.22}$$

where $S_1(t)$ is some base known distribution, $S_2(t)$ is the distribution for another group, and γ represents the acceleration factor, typically some regression such as $\gamma = \exp(-\vec{\beta}\mathbf{X})$. A stochastic multiplicative component α based on some distribution $g(\alpha)$ called **the frailty** can also be added to the hazard in order to model unobserved effects, resulting in a survival function of:

$$S_2(t) = S_1(\gamma t)^{\alpha} \quad \text{for all } t \geq 0 \tag{2.23}$$

The benefit of using AFT models are the relatively intuitive nature of the effects of covariates on the survival time. A general review of the literature suggests AFT models work very well with biological ageing research [38, 39] but the major key here is that biological processes are commonly modelled with known statistical distributions such as the log-normal or exponential distributions [40]. On the other hand, clinical and medical epidemiology applications tend to show better performance with non-parametric methods [41]. As we do not want to make any assumptions about the underlying base hazard function which is not particularly linked to biological processes, parametric models are beyond the scope of this thesis.

2.4 Evaluation Metrics for Survival Analysis

2.4.1 Log-Rank Test

The log-rank test is used to compare two or more survival functions with each other [42]. In this sense, it is analogous to the t-test or Pearson's chi-squared test for survival analysis. Like those tests, the log-rank test tests the null hypothesis H_0 that there is no difference between the survival functions being compared in the probability of an event e occurring at any time t.

The observed events are then compared to the expected number of events at any particular point in time if the null hypothesis was true. Given two survival curves a and b, the expected number of events of curve a at time t, $E(a_t)$ is expressed as:

$$E(a_t) = \frac{(e_{a,t} + e_{b,t})(e_{a,t} + s_{a,t})}{n_t} \tag{2.24}$$

where $s_{a,t}$ is the number of subjects still surviving from group a at time t, and n_t is the total number of subjects from both groups. With a calculation at every time point $t \in T$, the test-statistic Z can then be calculated as:

$$Z = \frac{\sum_{t=1}^{T}(e_{a,t} - E(a_t))}{\sqrt{\sum_{t=1}^{T} \mathrm{Var}(e_{a,t})}} \tag{2.25}$$

The test statistic is then treated as any other with the same p-value rules for significance.

2.4.2 Concordance Index

The concordance index or C-index is the most widely used evaluation metric for survival analysis models [43]. The value lies between 0 and 1, with a value of 1 meaning that the survival model perfectly assigns higher risk or hazard scores to subjects who are at a higher risk of experiencing the event in question while 0.5 means it is basically random [44]. In other words, the index is a great metric for the *discriminatory* power of the model, but does not make any explicit claim on the goodness of fit of the model.

The standard formulation of the concordance index is the Harrell's estimator which is the ratio of subject-pairs (i, j) in the dataset with "comparable" (one has a higher risk score than the other) and "concordant" risk scores (i.e. the one with the higher risk score has the event earlier) to the total number of "comparable" pairs.

$$\hat{C} = \frac{\sum_{i \neq j} \mathbb{I}\{T_i^{\mathrm{obs}} < T_j^{\mathrm{obs}}\} \cdot \mathbb{I}\{M_i > M_j\}}{\sum_{i \neq j} \mathbb{I}\{T_i^{\mathrm{obs}} < T_j^{\mathrm{obs}}\}} \tag{2.26}$$

where T_i^{obs} is the observed event time of subject i, M_i is the risk score (usually hazard) of subject i and $\mathbb{I}$ is the indicator function.

While it is a highly popular and in some ways intuitive metric, one needs to be aware of the quirks of C-index. Firstly, as the C-index only uses data with observed event times, datasets with a high proportion of censored data will very likely result in an overestimated C-index value. Uno et al have proposed an index based off the inverse probability of censoring weight (IPCW) which reduces this bias [45].

Secondly, changes and differences in the C-index do not necessarily reflect linearly to overall performance. For example, an improvement in $\hat{C}$ from 0.70 to 0.75 is not

necessarily the same difference in performance as one from 0.90 to 0.95. As Longato et al have shown, improving the C-index at the high end (0.90 to 1.0) requires substantially greater performance improvement compared to improving it at the low end (e.g. 0.50 to 0.60) [43].

Nonetheless, the C-index remains a good metric to use due to its ubiquity and implications in the experiment with the above caveats in mind will be covered in Section 6.

2.4.3 Brier Scores

While the C-index has good discriminatory power, the Brier score is a metric which assesses *calibration* of the model; namely, how accurate a model's predictions are. The Brier score is effectively the analogue to a mean squared error at different time points along the survival curve and takes censored data into account. Noting that it is an equivalent to mean squared error, the *lower* a Brier score, the more accurate a model is. It is based on a proposal by Graf et al [46] and is expressed mathematically as:

$$BS(t) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left[\frac{(\hat{S}(t \mid \mathbf{x}_i))^2}{\hat{G}(T_i)} \cdot \mathbb{I}(T_i < t, \delta_i = 1) + \frac{(1 - \hat{S}(t \mid \mathbf{x}_i))^2}{\hat{G}(t)} \cdot \mathbb{I}(T_i \geq t) \right] \tag{2.27}$$

where $\hat{S}(t \mid \mathbf{x}_i)$ is the model prediction at time t for subject i with subject i's covariates being $\mathbf{x}_i$, $\frac{1}{\hat{G}(\bullet)}$ is the IPCW, N is the total number of subjects, and δ_i is the binary variable representing whether subject i has had an event at time t or not.

In practice, the IPCW $\frac{1}{\hat{G}(\bullet)}$ is estimated on the survival times and events of the training set.

Equation (2.27) only provides the Brier score for a given time point on the model. In order to obtain an overall view of performance for the whole model across the entire time horizon, one would use the Integrated Brier Score (IBS). This simply takes the integral of the Brier Scores obtained for every relevant timestep. In other words:

$$\text{IBS} = \int_{t_1}^{t_{max}} BS(t) dw(t) \quad \text{for any interval } [t_1; t_{max}] \tag{2.28}$$

where the weighting function $w(t) = \frac{t}{t_{max}}$.

2.5 Incorporation of Time-Variant Data

Frequently in survival analysis, subjects may also have other covariates which vary over time. In other words, these covariates are time-dependent and fall into two general categories:

1. **Events:** A singular event occurs at a point in time after the study starts; e.g. a patient receiving a heart transplant, a customer submits an NPS (Net Promoter Score) response.

2. **Continuous Variables:** A measured variable across the study period; e.g. a patient's blood pressure readings at various time points, a customer's utilisation of a product.

Depending on study structures, continuous variables could also potentially be event-like, in that measurements are made at intervals of study time (e.g. patients only do measurements at $t = 1, 3, 5, 9$ while the study incorporates $t = 1, 2, 3, ..., 10$).

The literature also differentiates between *internal* time-varying variables (where the variable changes based on the individual's own characteristics and therefore ends when a subject event occurs) and *external* time-varying variables (where the variable is pre-determined by some external factor unrelated to the subject) [47]. This differentiation appears to only be necessary in terms of specific computer-program treatments of the data; however, in general theoretical terms for algorithms discussed here, there is no differentiation between the two [15].

2.5.1 Challenges and Considerations

One major consideration in the usage of time-variant data is the risk of immortal time bias. This is the situation where definitions in study design are set up in such a way that it is literally impossible for an event to occur before a time-varying variable reaches a certain threshold or occurs [48]. For example, a Texas Heart Institute survival analysis which concluded that heart transplant patients have a substantially longer survival time than non-transplant patients was found by Gail to be biased as the selection criteria for the study was acceptance to a transplant waiting list. This resulted in the 'received transplant' group to have an additional immortal period which is the waiting period before actually receiving the transplant. In other words, it was only possible by definition for 'not received transplant' subjects to have their death event occur during the waiting time period and 'received transplant' patients are by definition immortal until the point they receive their transplant resulting in a significant positive bias in survival time [49].

Similarly, one needs to be careful about cumulative data counts over time. For example, a breast cancer chemotherapy study showed an apparently strong effect for higher dosages and breast cancer survival. However, this was driven by the fact that patients are classified into groups based on the percentage of prescribed dosages. By definition, patients who died before completing their drug regimen were placed into the low dosage groups and therefore resulting in the group given high dosages having a positive survival bias [50].

While there is the risk of substantial bias, these issues can be minimised by correct study design and making sure any groupings are made *a priori*.

2.5.2 Data Structure for Time-Variable Data

The standard structure commonly used is a small modification to that described in Section 2.2.5. Instead of a single row per subject, a subject has multiple rows with covariates changing based on a (start, stop] format [51].

An example is detailed in Table 2.2.

Subject	**start**	**stop**	**event**	X_1	X_2
1	0	4	False	0.1	1.4
1	4	8	False	0.1	1.2
1	8	10	True	0.1	1.6
2	0	7	False	0.5	1.5
2	7	10	False	0.5	1.4

Table 2.2: An example of time-variant data structure. Each subject may or may not have multiple rows with start and stop times where events and variables can change. In this example, X_1 is a time-invariant variables while X_2 changes over time

2.5.3 Extending the Cox Model

Having time-variant variables would naturally break the proportional hazard assumption of the Cox model. Therefore, the model in Equation (2.19) is extended to incorporate these variables as a separate block of predictors:

$$h(t, \mathbf{X}(t)) = h_0(t) \times \exp(\mathbf{X}_{st}\beta_{st} + \mathbf{X}_{tv}(t)\beta_{tv}) \tag{2.29}$$

where matrix $\mathbf{X}(t)$ contains static variables $\mathbf{X}_{st}$ and time-variant variables $\mathbf{X}_{tv}$ and a vector of their respective coefficients β_{st} and β_{tv}.

One can also incorporate a lag-time effect for the time-variant variables. In other words, if one suspects the effect of a time-variant of any variable $X_{tv,j}$ only affects the outcome of the event after a time lag of L_j, one can rewrite Equation (2.29) to say:

$$h(t, \mathbf{X}(t)) = h_0(t) \times \exp(\mathbf{X}_{st}\beta_{st} + \mathbf{X}_{tv}(t - L_j)\beta_{tv}) \tag{2.30}$$

Taking all of this together, we can show the hazard ratio between any two subjects (a and b) as:

$$\frac{h_a(t, x_a)}{h_b(t, x_b)} = \exp\left[\beta_{st} \cdot (x_{st,a} - x_{st,b}) + \beta_{tv} \cdot (x_{tv,a}(t - L_j) - x_{tv,b}(t - L_j))\right] \tag{2.31}$$

While the proportional hazards assumption is broken, it is worth noting that β_{tv} is not time-dependent and therefore reflects the overall effect of variables $\mathbf{X}_{tv}$.

It is also worth noting that the extended Cox model can be used as a way to check if the proportional hazards assumption is correct. This is done by creating an extended Cox version of the original Cox model, with the variables of interest being multiplied by some time function $g(t)$[†]. In other words:

$$h(t, \mathbf{X}(t)) = h_0(t) \times \exp(\mathbf{X}\beta + \mathbf{X}(t)g(t))\beta_{tv} \tag{2.32}$$

If the model fulfils the proportional hazard assumptions, we can reasonably expect β_{tv} to boil down to a zero vector. Therefore, to check the assumption, the two models from equations 2.19 and 2.32 can be compared with the log-rank test as described in Section 2.4.1.

[†] $g(t)$ is often simply t or $\log(t)$

CHAPTER 3

Machine Learning for Survival Analysis

Applying machine learning (ML) techniques to survival analysis represents the state-of-the-art in the field. While research guidelines and general overviews have been around for nearly 10 years [52, 53], a scoping review in 2023 showed only 49 publications in PUBMED and EMBASE describing ML methods for survival analysis* [54].

This chapter thus provides a general overview of ML methods commonly used in the field of survival analysis.

3.1 Classic ML Methods

3.1.1 Regularisation Methods

In linear regression, regularisation or shrinkage methods are commonly used to reduce overfitting of models by penalising complexity in variables and promotes simple models which generalise better [55]. These methods apply just as well onto the standard Cox regression and come in the usual L1/LASSO (least absolute shrinkage and selection operator) and L2/Ridge variations which penalise the absolute and squared sum of regression coefficients respectively.

In most ML applications, both these regularisations will be applied at the same time with varying ratios using the ElasticNet protocol [56].

*It is also worth noting that of these 49, nearly half (21) were rejected from the study due to lacking reporting metrics, ML techniques not being used for survival outcomes, or are reviews rather than new algorithms

3.1.2 Support Vector Machines

SVMs work by finding an optimal hyperplane between data points which separates them linearly with the maximum margin possible. In order to work with data which cannot be separated linearly, kernel functions are applied to the data which transforms them into higher dimensions which can then be separated. These have been successfully applied to many regression and classification problems outside of the survival analysis domain.

In a survival analysis context, SVMs can be applied as either of:

1. A regression problem where the desired outcome is to predict the expected lifetime of a subject. Shivaswamy et al [57] as well as Khan and Zubek [58] have proposed different methods of applying penalties to censored data which take into account the negative bias to ignoring censored data-points if the basic support vector machine (SVM) regression estimator is used.

2. A ranking problem which uses the SVM classifier where subjects with shorter survival times are ranked lower than subjects with longer survival times when all subjects are compared to each other. Algorithms commonly in use have been proposed by Van Belle et al [59] and Evens and Messow [60].

Van Belle et al conducted a comparison between these two approaches and found that the ranking method substantially underperforms regression methods [61]. These creates a challenge as these models are not easily translatable into survival/hazard curves which are used for evaluating other models in this study. In addition, SVMs are considered unsuitable for large data sets [54]. As a result, SVMs will remain out of scope of this thesis.

3.2 Ensemble Methods

Ensemble methods are a collection of models which combine large numbers of so-called weak learners which perform very poorly on their own, but end up with a much stronger-performing model when combined.

3.2.1 Random Forests

Random forests are based on simple decision trees as weak learners. Multiple decision trees are built and trained on a different bootstrap sample (random subset with replacement) as well as a random subset of features from the dataset. These resulting predictions from these trees are then aggregated either by majority vote[†] or by mean prediction depending on the task (Refer Figure 3.1). Breiman proved that if there are sufficient trees in the random forest model, the resulting model will always converge and overfitting is impossible due to the Strong Law of Large Numbers [62].

[†]It is worth noting that sklearn uses mean probability rather than majority vote for classification tasks

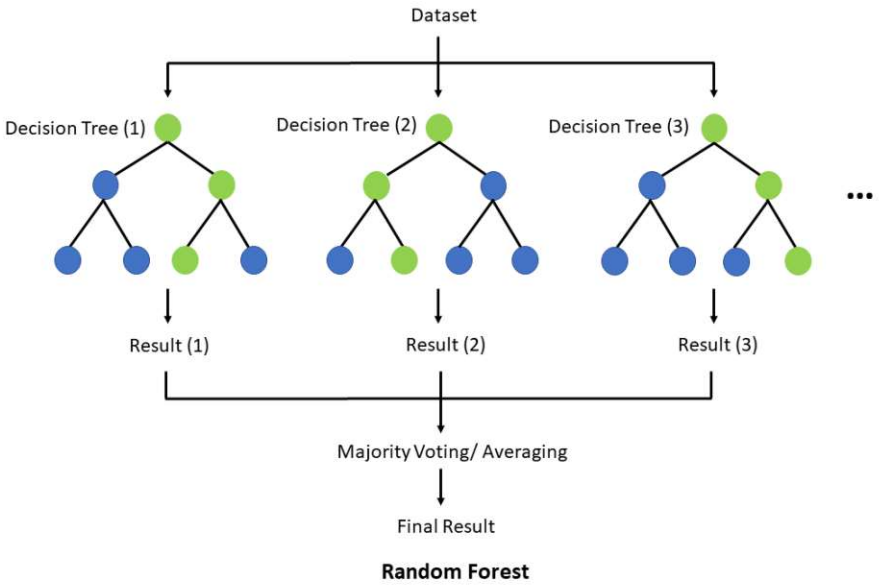

Figure 3.1: A diagram showing how a Random Forest is made up of multiple decision trees; whose results are then aggregated to obtain a final result[63]

While it is well-known for robustness and generally good performance, random forests are computationally expensive in comparison to classical ML algorithms, particularly for large datasets with large numbers of trees. While sold as a method which works well with high-dimensional datasets, studies suggest performance declines significantly when only a small percentage of features are truly informative, meaning in some cases, feature selection may still be a worthy endeavour [64, 65]. It is also worth noting, that in the event that there is a linear relationship between the attributes and the target value, random forests may not be any better than linear estimators in spite of the substantially longer computational times involved [66].

In the context of survival analysis, while many have proposed methods which force the survival analysis problem into either a regression or classification problem which fits within the original paradigm [67, 68, 69], Ishwaran et al presented another approach which strictly adhered to the guidelines laid out by Breiman for Random Forests, namely that all aspects of growing a random forest must take the outcome into account, i.e. splitting criterion during the growing of the tree must take survival time and event occurrence into account [70]. This algorithm is described in algorithm 3.1.

Up to line 5, the algorithm is exactly the same as the standard random forest algorithm. Then, the split is chosen by iterating through all possible variables and split values finding the one which achieves the greatest difference in survival outcomes. Once the node cannot form new daughter nodes as the node does not achieve $n_e > 0$, that node is determined to be terminal.

Algorithm 3.1: Random Survival Forest Algorithm

```
Draw B bootstrap samples from the original data;
for each bootstrap sample do
    while node_size < n_min do
        Grow a random forest tree T_b by recursively repeating the following steps
         at node;
        if n_events ≤ 0 then
            return to parent node and work down another daughter node;
        else
            (1) Select m variables at random from the p variables.;
            (2) Split the node using the variable which maximises survival
             difference between daughter nodes;
        end
    end
    Calculate a cumulative hazard function for the tree H(t)
end
Aggregate the individual tree cumulative hazard functions using a mean function;
Using out-of-the-bag data, calculate prediction error for the aggregated
 cumulative hazard function;
```

For every terminal node h, the cumulative hazard function can be described as the Nelson-Aalen estimator, namely:

$$\hat{H}_h(t) = \sum_{t_{l,h} \leq t} \frac{e_{l,h}}{Y_{l,h}} \tag{3.1}$$

where $e_{l,h}$ represents the number of events at terminal node h, l represents every time point with at least one event occurrence, and $Y_{l,h}$ are the number of individual subjects at risk at time $t_{l,h}$.

Therefore, for a subject i with covariate matrix $\mathbf{x}_i$, the cumulative hazard function is the relevant terminal node determined by $\mathbf{x}_i$, assuming of course that $\mathbf{x}_i \in h$. Mathematically:

$$H(t|\mathbf{x}_i) = \hat{H}_h(t) \tag{3.2}$$

Taking Equation (3.2) and the average of all cumulative hazard functions, we can conclude that the function for a dataset with B bootstrap samples is:

$$H(t) = \frac{1}{B} \sum_{b=1}^{B} H_b(t|\mathbf{x}_b) \tag{3.3}$$

Feature Importance in Random Forests

While the traditional way of obtaining feature importance involves cross-validation, a far more computationally efficient method for random forests is to use Breiman-Cutler variable importance or permutation importance. The way to calculate is described in the Manual On Setting Up, Using, And Understanding Random Forests as:

> "In the OOB cases for a tree, randomly permute all values of the jth variable. Put these new covariate values down the tree and compute a new internal error rate. The amount by which this new error exceeds the original OOB error is defined as the importance of the jth variable for the tree. Averaging over the forest yields variable importance [71]."

By using this method, we can obtain feature importance from large random forests without having to use up precious computational resources.

3.2.2 Relative Risk Forests

As discussed in Section 3.2.1 above, certain forest-like methods exist which do not adhere to the Breiman guidelines for Random Forests. We discuss one such version here as it will be relevant when discussing the incorporation of time-variant data in Section 3.4.2.

The Relative Risk Forest - also proposed by Ishwaran et al - treats the survival analysis problem based on a classification and regression tree (CART) methodology which depends on the equivalence between a survival tree and Poisson tree likelihoods [69]. To grow the tree recursively, the deviance residual below is used as a splitting criteria:

$$d_i = 2\left[\delta_i \log\left(\frac{\delta_i}{\hat{H}_0^1(t_i)\hat{\theta}_b^1}\right) - \left(\delta_i - \hat{H}_0^1(t_i)\hat{\theta}_b^1\right)\right] \tag{3.4}$$

where δ_i represents censoring information for subject i[‡], $\hat{H}_0^1$ represents the one-step Nelson-Aalen estimator at that particular node for that given iteration, and $\hat{\theta}_b^1$ is the one-step estimate for node b under the proposed split.

Using this one-step estimate, a new Nelson-Aalen estimator $\hat{H}_0^2(t)$ is recalculated using:

$$\hat{H}_0^2(t) = \sum_{\{i:t_i \leq t\}} \frac{\delta_i}{\sum_{b \in T^*} n_b(t_i)\hat{\theta}_b^1} \tag{3.5}$$

where n_b represents all individuals in node b who are at risk at time t_i, and T^* represents all terminal nodes in the tree.

[‡]namely, 1 in the event of a customer churning and 0 if the customer was censored

The tree is then resplit again until a fixed number of calculations are complete or a stopping criterion is achieved. Once this occurs, the relative risk tree is fully-grown and ready to take in new data. At this point, a given covariate set $\mathbf{X}$ is then dropped down the tree and the final-step estimate in the relevant terminal node $\hat{\theta}_h$ is the relative risk/hazard rate for the given covariate set.

In the production of the forest, a bootstrap sample of subjects are selected as well as a random sample of covariates from $\mathbf{X}$ are chosen and multiple relative risk trees are grown as the base learners for the forest. Then, the average final-step estimate is taken to represent the hazard rate.

3.2.3 Gradient Boosting Models

Unlike random forests, gradient boosting models (GBMs) combine their weak learners in an *additive* form, where the addition of each new model 'boosts' (improves) the currently existing model. Mathematically, it can be described as:

$$f(\mathbf{X}) = \sum_{m=1}^{M} g(\mathbf{X}; \theta_m)\boldsymbol{\beta}_m \tag{3.6}$$

where $\boldsymbol{\beta}_m$ represents the weights for base learner function $g(\mathbf{X}; \theta_m)$ and θ_m are parameters which change with each learner iteration.

In terms of base learners, the user can choose from various simple algorithms although the most common ones available in relevant Python and R libraries are regression trees, component-wise least squares (which is an analogue of least-squares regression), and the stratified Cox models [29, 72].

Whichever base learner is selected, the gradient boosting model (GBM) learns using a 'greedy stage-wise' process, in which each iteration is fitted such that $\vec{\beta_m}$ minimises a loss function (e.g. partial likelihood). The next iteration is then updated with that of the previous iteration, in other words:

$$f_m(\mathbf{X}) = f_{m-1}(\mathbf{X}) + f(\mathbf{X}; \theta_m)\boldsymbol{\beta}_m \tag{3.7}$$

Often, regularisation (see Section 3.1.1) or subsampling (where each base learner is fitted on a random subset of the dataset) will be employed in order to prevent overfitting [73].

3.3 Deep Learning Methods

In terms of state-of-the-art deep learning applications, a review by Wiegrebe et al shows that the vast majority of the work done in this area are highly-specific applications focused on estimating patient survival based on high-dimensional data such as CT Scan images or multiomics data [74].

The earliest application of deep learning methods is to extend the Cox method beyond having a linear proportional hazard to having a non-linear proportional hazard. Katzman et al proposed a feed-forward deep neural network built from multiple Dense layers called DeepSurv which determines the hazard rate of the subject [75]. Although similar neural network systems had been proposed in the past [76], Katzman claims this is the first version of a neural network structure which provided superior results compared to the standard linear Cox model.

43% of research in deep learning for survival analysis reviewed by Wiegrebe et al expand upon DeepSurv's foundations, either tweaking architecture, experimenting with different loss functions, or incorporating multi-modal input such as unstructured data or image data on top of the standard tabular data available. In other words, the standard stack of Dense layers provided by DeepSurv can also incorporate various types of input data such as X-ray images. See Figure 3.2

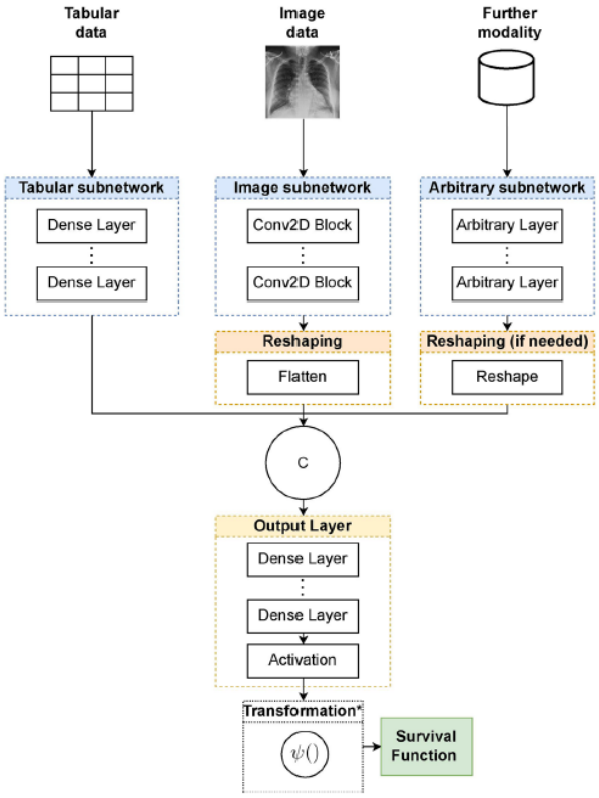

Figure 3.2: DeepSurv architecture which incorporates multi-modal input. Inputs are processed and then fed into the Dense layer stack which outputs the hazard and survival functions [74].

The other class of deep learning applications reviewed which comprised a further 31% treat each time point as a discrete classification problem. Because the problem becomes

a classification one, the methods reviewed are highly variable with a mix of recurrent neural networks (RNNs), transformers, and feed-forward neural networks. In addition to varying architectures, the proposed methods cover many different ways of choosing loss models and parametrising the probability mass function of event occurrences. It is worth noting that all these methods break away from the survival analysis mathematics that have been discussed in Chapter 2.

While interesting and novel, the review also emphasised that 87% of the methods discussed either did not come with code or have been built as bespoke one-shot implementations. In addition, the methods published rarely discussed optimisation and tuning. A combination of these issues mean that deep learning methods will remain out of scope for this thesis, noting that there is already sufficient room for exploration in time-variant data on classical and ML methods.

3.4 Incorporating Time-Variant Data in ML Methods

The incorporation of time-variant data in more advanced ML models is still relatively novel, with most literature found on this only published in the last five years [77, 78, 79, 80].

Methods usually cover three different approaches of dynamic estimation, namely where the hazard function of a subject is continuously updated as new time-varying covariates appear. These methods are: -

1. **Landmark Analysis**: This involves building a new standard model (e.g. Cox PH) at various time points t known as landmark times looking only at subjects which have survived to the landmark times [81].
2. **Joint Modelling**: This process models the time-varying covariates jointly with the event time data process by assuming time-varying and survival processes are underpinned by the same random effects [82].
3. **Counting Process**: The counting process splits the follow-up information (i.e. all covariates after $t = 0$ into non-overlapping interval sections).

When dealing with ML methods, the most literature has been written about Random Forest processes which primarily use the counting process above. In 2020, two different methods were proposed to incorporate time-variant data into the Random Survival Forest model described in Section 3.2.1.

3.4.1 Random Forest for Survival, Longitudinal, and Multivariate data (RF-SLAM)

The RF-SLAM method proposed by Wongvibulsin et al utilises a variation of the standard Random Survival Forest by implementing a preprocessing step to the data prior to creating the tree. This step involves building discrete units called counting process information

units (CPIUs)[83]. In effect, the CPIU is similar to the long-format data described in Section 2.5.2 except all subjects share the same time intervals. Figure 3.3 illustrates how CPIUs are produced prior to being fed into the random forest.

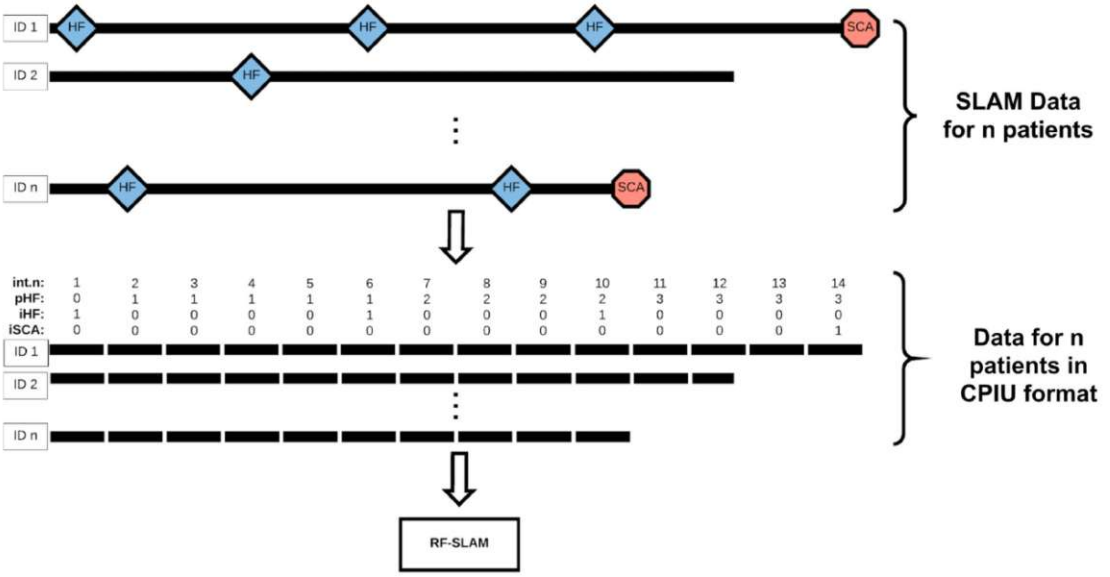

Figure 3.3: An example of CPIU generation for the RF-SLAM algorithm. Wongvibulsin used LV Structural Predictors Registry as the data source which includes hospitalisations due to heart failures (HF in blue diamonds) as a time-varying covariate and SCA (in red octagons) being the event of interest. The timeline of every subject is cut into similar multiple discrete pieces which are the CPIUs. Each CPIU then has variables $int.n$ (CPIU ID), pHF (count of HFs so far), iHF (whether a HF occured during the time point), $iSCA$ (whether the SCA event occurred or not), and ID n (the ID of the subject) [83].

To deal with the clear loss of independence between CPIUs, the RF-SLAM uses a Poisson log-likelihood (instead of the traditional log-rank split statistic which depends on the proportional hazards assumption) as a split criteria. Wongvibulsin describes the Poisson log-likelihood split statistic as:

$$\begin{aligned}\sum_{i\in L}\sum_{t=1}^{T_i}\left[-\hat{\mu}_{it}^{L}+y_{it}\cdot\log\left(\hat{\mu}_{it}^{L}\right)\right]+\sum_{i\in R}\sum_{t=1}^{T_i}\left[-\hat{\mu}_{it}^{R}+y_{it}\cdot\log\left(\hat{\mu}_{it}^{R}\right)\right]\\-\sum_{i\in P}\sum_{t=1}^{T_i}\left[-\hat{\mu}_{it}^{P}+y_{it}\cdot\log\left(\hat{\mu}_{it}^{P}\right)\right]\end{aligned}\tag{3.8}$$

where subject i at time t and a maximum time of T, has an estimated expected number of events $\hat{\mu}$ and the actual number of observed events y, and these subjects can be parts of the left node L, right node R, and parent node P.

The estimated expected number of events $\hat{\mu}$ for subject i at time t at node S is given as:

$$\hat{\mu}_{it}^{S} = \hat{\lambda}_{it}^{S} r_{it} \tag{3.9}$$

where $\hat{\lambda}$ is the estimated event rate and r is the total length of the at-risk time interval.

Each terminal node in the forest is then assigned an estimated event rate based off the training data using the above calculations called $\hat{\lambda}_b$. When applying an observation outside of the training set, the hazard rate is obtained by averaging across all trees in the forest. Mathematically, the hazard rate for a new subject i at time t with covariates $\mathbf{X}_i$ is:

$$\hat{h}_i(t|\mathbf{X}_i) = \frac{\Sigma_{b=1}^{B} \hat{\lambda}_b(\mathbf{X}_i)}{B} \tag{3.10}$$

where b represents any one of B trees in the random forest.

As of the time of writing, no readily available libraries exist for R and for Python.

3.4.2 LTRC Trees

Another group of methods called LTRC trees was proposed Yao et al to estimate a population-level survival function [84]. These are extensions of random-forest-like algorithms; namely relative risk and conditional interference forests, which are variations of random forests which treat the survival question as the standard Classification and Regression Trees (CARTs) problem solved by standard forest algorithms.

The proposed methods follow a similar process, in that the data is split into the counting method format described in Section 2.5.2, and then the CPIUs are processed as if they were independent. Lastly, the survival function estimate is calculated based on the outputs of the forests and dependent on covariate information up to time t.

In order to incorporate the time-variant data, Yao et al amended the split criteria from the standard conditional inference and relative risk forests. For example, in Yao's relative risk forest implementation, the split criteria referenced in Equation (3.4) is replaced by:

$$d_i = \sum_{l \in \mathcal{R}_h} 2 \left[\delta_i \log \left(\frac{\delta_i}{\left[\hat{H}_0(R'_l) - \hat{H}_0(L'_l)\right] \hat{\theta}_b^1} \right) - \left(\delta_i - \left[\hat{H}_0(R'_l) - \hat{H}_0(L'_l)\right] \hat{\theta}_b^1 \right) \right] \tag{3.11}$$

where L'_l and R'_l represents the starting and ending times t of a CPIU l, and $\mathcal{R}_h$ represents the set of possible time observations (essentially an individual row in the data table presented in Table 2.2. This then allows the relative risk forest to accept and process time-variant data and ensures the algorithm only sees data prior to the actual time point.

A CRAN library of these implemented methods was also published by Yao et al in November 2023 [85].

CHAPTER 4

Implementation and Employed Algorithms

4.1 Dataset and Customer Information

Due to commercial requirements, the information on the dataset is kept purposely vague, with the intention to provide an idea of the size and type of the dataset which is used in the experiment.

The dataset consists of customer subjects with n in the 10^5 order of magnitude and contains information on whether they are churned customers or not (the event), the total time they have been a customer (the survival or censored time), and 38 time-invariant variables associated with each individual customer. These variables are a mix of numerical as well as categorical variables. In addition, there are an additional six time-variant variables in the dataset covering continuously measured metrics as well as event occurrences.

The dataset skews towards churned customers with a majority of subjects having had the event occur as it represents several years of the customer base.

This dataset was split into training and test sets, where the test set consists of several chosen customer cohorts which comprises approximately 15% of the total dataset. The test set is only used at the end of the experiment to assess model performance, and a separate validation set is created through cross-validation during the training and hyperparameter tuning processes.

4.2 Preprocessing

The vast majority of the pre-processing work was completed at the source, namely sourcing the appropriate tables, making sure joins are appropriate, and ensuring aggregations

returned the correct values. Clean raw data was then extracted via SQL code from company databases and as much as possible, formatting was fixed at source. This data was then inspected and spot-checked to determine fidelity with the original sources (for example, confirming with source that customer *abc*123 actually had six events of variable x occur) and corrections to the joins were made as needed.

This clean raw data is then further processed as needed to fit the specific model/library format requirements. Noting that the source of data is assumed to be 'complete', missing values (i.e. NAs and inifinity values) were set to zero assuming that either the event did not occur or no value was collected for the given variable.

4.2.1 Categorical Variables

As most of the categorical variables were nominal and not ordinal, it was chosen to one-hot encode all categorical variables, i.e. all unique categorical values are converted to Boolean variables (See Figure 4.1). This did increase the dimensionality of the dataset.

To reduce this effect somewhat, one value from each of the categorical values was chosen to represent a default customer and columns with those categories were removed, under the assumption that all 0s across the relevant columns means a 1 for the default category. To use the example in Figure 4.1, say we select `red` as the default value for `color`. This means column `color_red` was removed from the dataframe. This resulted in the 31 total variables of the dataset growing by 54% to a total of 48 variables. However, this was considered an acceptable growth as the dataset could then be cut down again by using feature selection methods.

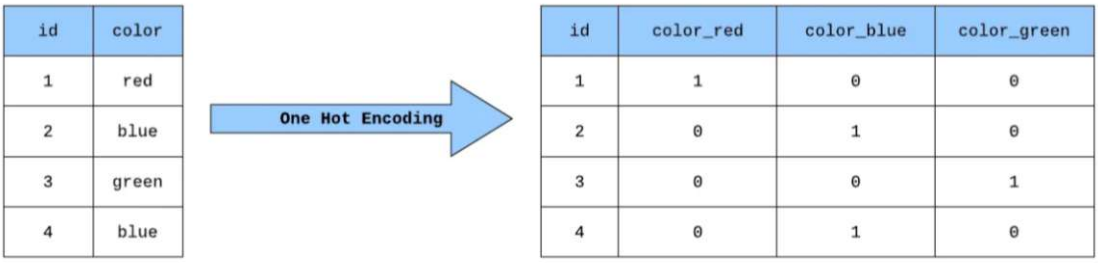

Figure 4.1: An example of one-hot encoding showing how all unique values of `color` are split into columns with Boolean variables [86].

4.3 Feature Selection

Due to the dimensional growth, particularly from the one-hot encoding discussed in Section 4.2.1, consideration was given to the need for feature selection, particularly for computationally-intensive algorithms such as the random forest.

Some algorithms already have feature selection in-built; for example, `sksurv`'s Cox Proportional Hazards estimator comes with ElasticNet capability built in [29], where coefficients are penalised (See Section 3.1.1 for a more detailed description of how regularisation works). However, this does not help with the issue of selecting features for computationally-intensive algorithms.

Another alternative is to implement transformations to the inputs and using these derived inputs for the algorithms. This is commonly done through regression processes such as principal component regression (PCR) or partial least squares regression. Both of these methods construct a set of linear combinations of the original inputs which is then used as an input [55]. Cox variations of this method exist [87]; however, these methods reduce interpretability somewhat as resulting analyses of feature importance, e.g. by looking at the various Cox coefficients, would only show their relation with the principal components rather than the variables in question. This makes it more difficult to assess RQ3 in Section 1.2, particularly to a non-scientific corporate audience.

Therefore, it was decided to manually choose features based on the results from implementing feature importance methods. In order to do this, the results of the basic Cox and Random Forest full models were used to manually decide what should be setup as a reduced model for testing.

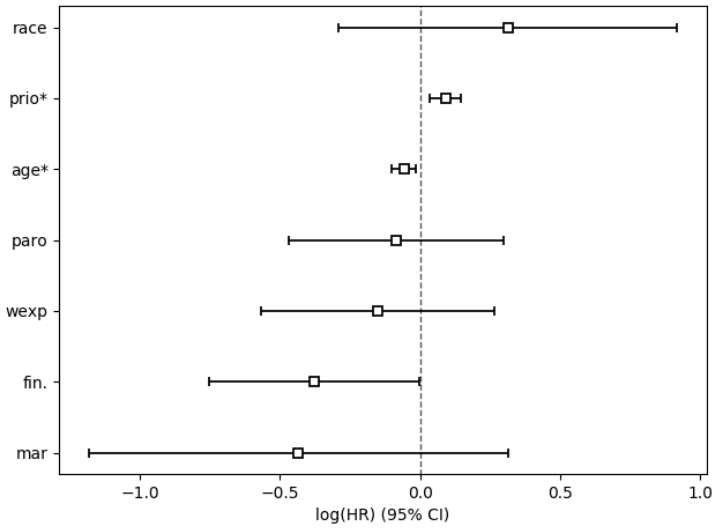

Figure 4.2: An illustrative example of Cox Proportional Hazards coefficients being plotted to assist with manual decision of what features to keep and what features to drop [19].

The basic Cox model conveniently has coefficients which are a sign of feature importance in that they show an estimated effect on the partial hazard of a subject at a given point in time (Refer Figure 4.2). Similarly, standard `scikit-learn` implementations of feature importance permutes through the variables, calculates the impurity of child nodes, and

determines importance of each feature as the magnitude of reduction in impurity due to a forest split.

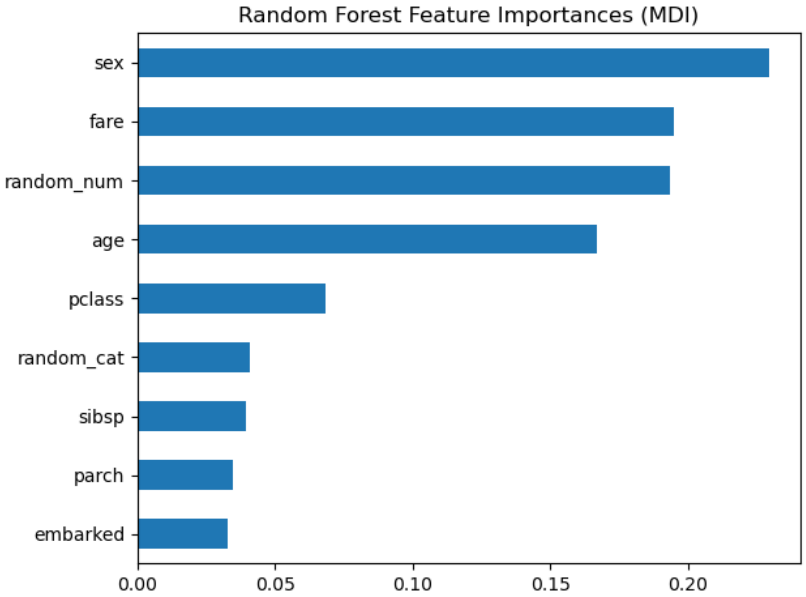

Figure 4.3: An illustrative example of permutation importance values from a random forest being plotted to assist with manual decision of what features to keep and what features to drop [88].

In addition to the reduced dataset, a minimal dataset was also created which only has the variables used in the base Kaplan-Meier as well as the six time-variant variables while half of the subjects were chosen at random to be removed, resulting in a substantially smaller dataset that has 48.9% of the rows and 38% of the columns.

4.4 Hyperparameter Tuning

Most of the various models described in the following sections come with a range of various parameters which allow tuning of the model's performance. Where practical, we have attempted hyperparameter tuning and the specific hyperparameter grids are shared in the relevant algorithm section.

The general principle for hyperparameter tuning was to use a Random Search algorithm with five-fold cross-validation over 50 iterations. This was deemed to be better than an exhaustive grid search process and both Bengio and Bastra as well as Zheng et al have shown that a random search can achieve the same performance as grid search with greater efficiency [89, 90]. This effect and the reasons why this occurs is illustrated in Figure 4.4.

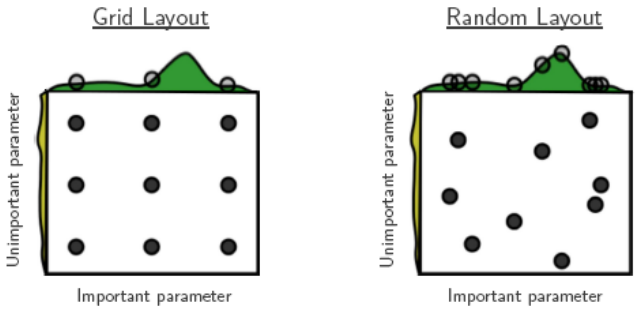

Figure 4.4: An illustrative example comparing Grid Search with Randomised Search, noting that for an important parameter, Randomised Search will look at nine distinct values while Grid Search is only limited to three, which drives improved efficiency at finding optimal parameters [89].

As will be further discussed in Section 5, this principle was occasionally not achieved due to computational issues, and this will also be discussed in the sections pertaining to those algorithms as well as in Section 6.

4.5 Models Implemented

The following section describes the various models and algorithms tested, as well as relevant hyperparameters used during the experiment.

4.5.1 Basic Algorithms

Kaplan-Meier Estimator

The Kaplan-Meier estimator is used as the base simple/heuristic model for comparison with the other following models. Two variables were chosen for stratification purposes creating a total library of 60 individual K-M curves. When a new entry from the test set is introduced, the survival curve chosen would be that which matches the correct variables in the K-M curve library.

The `KaplanMeierFitter` function from Python's `lifelines` library was used to obtain these curves [19].

Cox Proportional Hazards

The `CoxPHFitter` function from Python's `lifelines` library was used. This function conveniently also comes with an ElasticNet implementation which was explored for hyperparameter tuning. The search space consisted of the following hyperparameters:-

- `penalizer`: A sequence of 100 equally spaced numbers starting from 0.001 and going up to 0.1. (`np.linspace(0.001, 0.1, 100)`)
- `l1_ratio`: A sequence of 50 equally spaced numbers ranging from 0 to 1 where 0 represents a full Ridge regularisation and 1 represents full Lasso (`np.linspace(0, 1, 50)`).

The best model was decided by the best mean concordance index from the five cross-validation folds.

In addition, as described in Section 4.3, coefficient summaries from this model are also obtained to decide on reduced models for the more computationally-expensive models in the following sections.

Random Forest Survival

The `RandomSurvivalForest` function from Python's `sksurv` library was used. This implementation is the same one discussed in Section 3.2.1. The search space consisted of the following hyperparameters:

- `n_estimators`: A logarithmic sequence of 10 values from 50 going up to 2000. (`np.geomspace(50, 2000, num=10)`)
- `max_depth`: A sequence of 16 equally spaced numbers ranging from 2 to 32 (`np.linspace(2, 32, 16)`)
- `min_samples_split`: A logarithmic sequence of from 2^1 to 2^7. (`np.logspace(1, 7, base=2, num=7)`)
- `min_samples_leaf`: Also a logarithmic sequence of from 2^1 to 2^7. (`np.linspace(2, 32, 16)`)
- `max_features`: The option between the square root of features or the base-2 logarithm of features .(`["sqrt", "log2"]`)

In addition, permutation importance analysis was included to obtain feature importance values for manual model reduction and feature selection.

Gradient Boosting

While it wasn't intended in the original brief, we also tested the Python's `sksurv` library's implementation of gradient boosting. The hyperparameter search space was as follows:

- `n_estimators`: A logarithmic sequence of 10 values from 50 going up to 1000. (`np.geomspace(50, 1000, num=10)`)

- `max_depth`: A sequence of 16 equally spaced numbers ranging from 2 to 32 (`np.linspace(2, 32, 16)`)
- `min_samples_split`: A logarithmic sequence of from 2^1 to 2^7. (`np.logspace(1, 7, base=2, num=7)`)
- `min_samples_leaf`: Also a logarithmic sequence of from 2^1 to 2^7. (`np.linspace(2, 32, 16)`)
- `learning_rate`: A logarithmic sequence of 50 values from 0.0001 going up to 1. (`np.geomspace(1e-4, 1, num=50)`)
- `loss`: All available options for the loss function to be optimised was given, with `coxph` being the traditional standard using partial likelihood(`['coxph', 'squared', 'ipcwls']`)
- `criterion`: The criterion used to measure the quality of split (`['friedman_mse', 'squared_error']`)

4.5.2 Time-Variant Methods

As these methods are more cutting edge, there is a severe dearth of available libraries that address time-variant survival analysis, especially in Python, which is the primary language used in this work. For example, `lifelines` in Python provides some time-variant capability; however, due to the creator's issues with the potential of immortal time biases, they chose not to implement the production of survival curves or cumulative hazard functions to the time-variant functions.

Because of this, and the fact that the basic `survival` package in R supports some basic time-variant functionality [91], and the most advanced methods have libraries provided in R, the analysis pipelines for time-variant covariates were basically rebuilt in R.

Extended Cox Model

The extended Cox model was built from the `coxph` function from the `survival` package in R. While a Ridge penaliser could be applied to the formula, the `survival` package was not as flexible in allowing a survival ElasticNet. Therefore, the fit for this was the standard Cox model and no hyperparameter tuning was conducted.

Options from the `glmnet` library were also explored to obtain ElasticNet capability as well as hyperparameter tuning; however, challenges in producing survival curves and fits on test curves after training for evaluation purposes meant this was not able to be fully explored by the end of the study period.

The Python application through the `lifelines` library was also implemented to obtain rough performance figures as hyperparameter tuning was a possibility. However, as no straightforward function was provided in the library to obtain survival curves, the R function was the one that was assessed in Section 5.

LTRC Relative Risk Forest

We implemented the `LTRCforests` library provided by Yao et al in their paper on the algorithm using the extended relative risk forest algorithm [84, 85]. Settings were kept at default and while there were hyperparameters to explore, it was quickly determined (as discussed in Section 5.3) that this was not practical even on the reduced and minimal datasets.

Therefore, three experiments (with the full, reduced, and minimal datasets) were conducted with the time-varying relative risk forest algorithm on recommended settings based on the paper produced by Yao et al [84], namely:

- `ntree`: The total number of trees: 100
- `mtry`: Number of variables sampled for each node: $\sqrt{m}$, where m is the total number of variables
- `nodesize`: Average terminal node size: $\sqrt{n}$, where n is the total number of rows in the data
- `nsplit`: The number of random splits to consider for each candidate splitting variable: 10

4.6 Evaluation Metrics

For the purposes of evaluation, a fit was made against the test set, and the Brier Score, time-dependent AUC, and concordance index were obtained. In addition, the mean absolute error, mean percentage error, and RMSEs were calculated. For the purposes of presentation in this thesis, we have limited the fit to the first customer year.

An overall K-M curve for the test set was also drawn and compared to a K-M curve of the predictions to give a general and more interpretable picture of model performance across time. However, this is not shared due to commercial sensitivities relating to the data.

4.7 Hardware

All experiments were conducted on the Fangorn server at TU Wien's Computational Statistics Department of the Institute of Statistics and Mathematical Methods in Economics.

The cluster comprises two AMD Epyc 9654 96-Core processors running at 2.40 GHz with 768GB of RAM. All experiments were run on CPU and used 50 cores in total to maintain consistency and to prevent the hogging of shared resources, particularly for extended runs of the time-variant algorithms.

CHAPTER 5

Results

The results chapter of this thesis will cover the basic time-invariant and time-variant (aka time-dependent) methods separately. This allows us to cover the time-variant results in greater detail and what they imply for future work.

5.1 Basic Methods

Table 5.1 summarises the overall performance of every model across the entire 12-month period examined for the purposes of assessment. Across every summary metric, the Random Forest performed best, and was the only model which performed better compared to the base Kaplan-Meier model.

Model Name	C-Index	Mean AUC	Int. Brier	RMSE
Kaplan-Meier (Base)	0.640	0.651	0.175	0.0967
Cox Elastic-Net	0.457	0.504	0.218	0.0760
Gradient Boost	0.661	0.678	0.168	0.0458
Random Forest	0.651	0.738	0.157	0.0356

Table 5.1: Summary of fully-tuned basic algorithm results against the test set, showing the concordance index, mean area under the curve (AUC), integrated Brier scores, and RMSEs for every model. To recall, a C-index of 0.50 represents complete randomness, and 1 represents perfect discrimination, a higher AUC is better, and an integrated Brier score of 0.25 is considered fully random and lower scores are better.

5.1.1 Findings from General Results

This section will highlight some interesting aspects of the general results in Table 5.1 the table will be pointed out.

Kaplan-Meier RMSE: The Kaplan-Meier base model shows better discriminatory performances in comparison to the Cox Elastic Net as evidenced by the C-index, AUC, and IBS scores. However, it is interesting to point out a higher RMSE value compared to the other models suggesting that the base model performs particularly poorly when dealing with outliers.

ML Methods Show Minor Improvements over Base: The Gradient Boost and Random Forest algorithms showed better results across all metrics in comparison to the base Kaplan-Meier model. However, the improvement levels are not particularly large compared to the base, with the exception of RMSE, which suggests good ability to deal with outliers.

5.1.2 Findings from Time-Dependent General Results

Noting that average values in Table 5.1 do not necessarily show the whole picture, Figures 5.1 to 5.8 show the time-dependent AUC and Brier scores for each model through the first year. These provide a better view as to typical performance of models across the time-period analysed. The observations include:

Short-Term Performance generally better: Across the board (with the Gradient Boost being the only exception - Refer Figure 5.5), we can see that both the AUC and Brier scores of all models show substantially better performance prior to the sixth month of enrolment compared to post-six month. One potential reason for this is that as in any customer dataset, customers are skewed towards the shorter end of the lifetime scale, in other words, there are significantly more customers who are short term as opposed to long term thus providing more data for the model to train on.

Random Forest only algorithm which performed consistently well: Looking at the Brier score graph in Figure 5.8, the Random Forest is the only algorithm which consistently stayed below the 0.25 line across the whole time period. The Gradient Boost showed a similar Brier score curve but stayed generally worse compared to the Random Forest suggesting similar overall performance, except worse.

5.1.3 Feature Selection

Noting the substantially better performance of the Random Forest algorithm as compared to the Cox Elastic-Net, it was decided to use solely the permutation importance values from the Random Forest model in order to generate the reduced dataset for the time-variant methods. Following the selection process (based on variables where the absolute feature importance score was close to zero), a reduced dataset was produced which removed 15 columns resulting in a dataset comprised of 29 variables. This is a reduction of 34% from the full data-set which consisted of 44 variables in total.

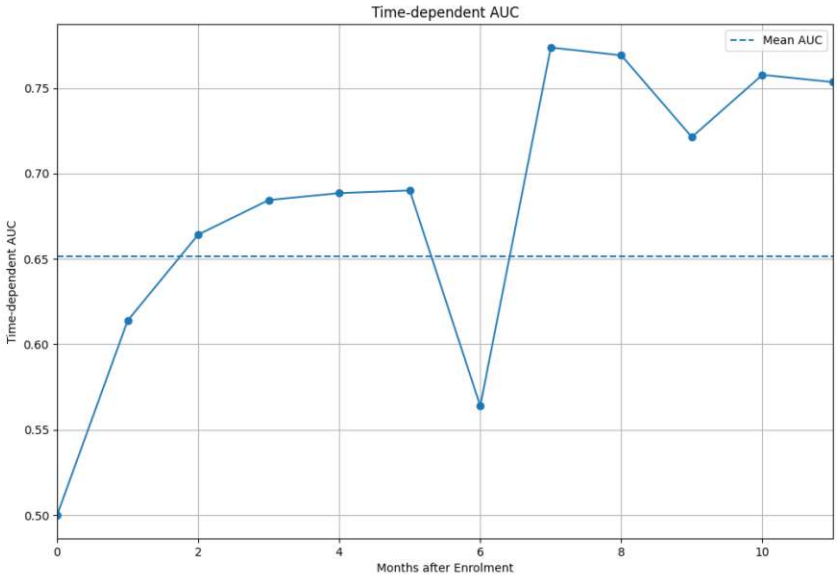

Figure 5.1: Time-Dependent AUC Curve for the Base Kaplan-Meier

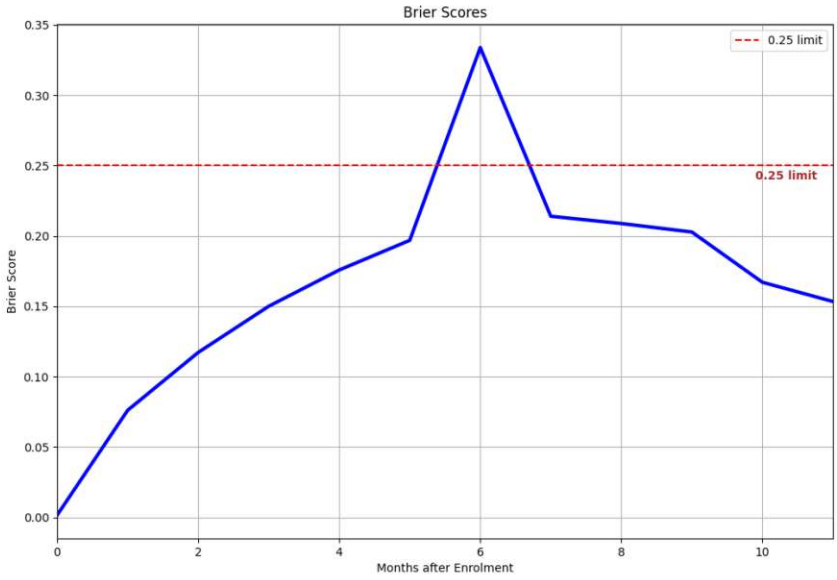

Figure 5.2: Time-Dependent Brier Scores for the Base Kaplan-Meier

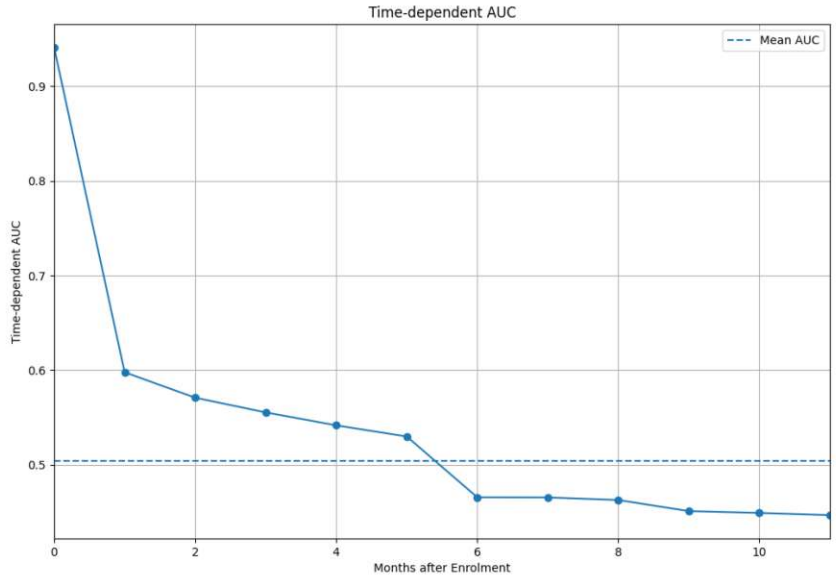

Figure 5.3: Time-Dependent AUC Curve for the Cox Elastic-Net

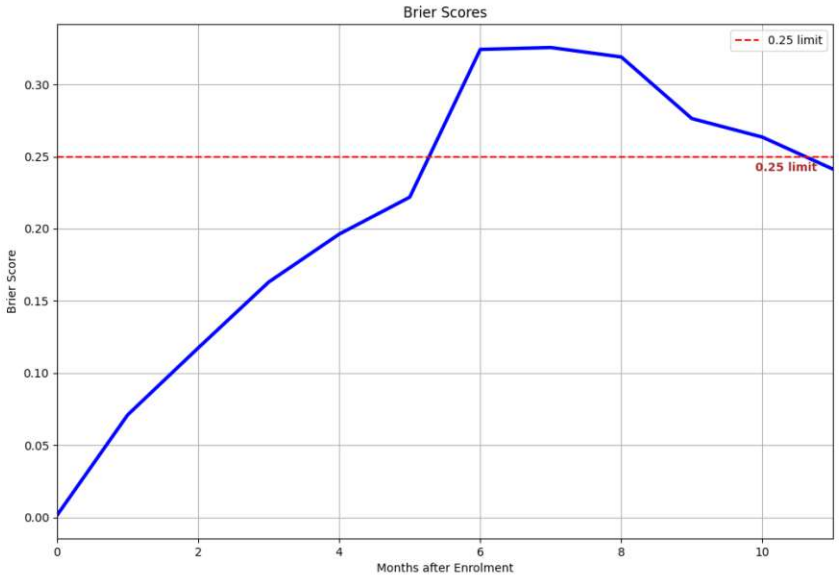

Figure 5.4: Time-Dependent Brier Scores for the Cox Elastic-Net

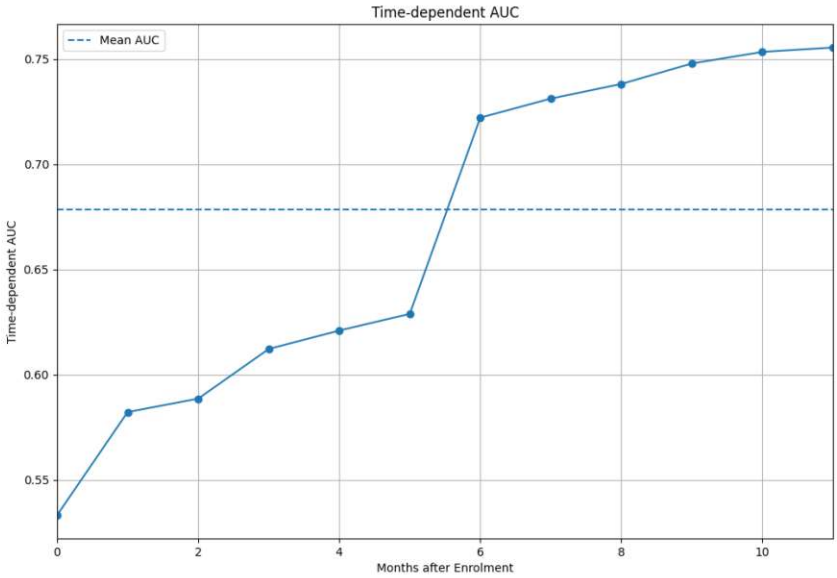

Figure 5.5: Time-Dependent AUC Curve for the Gradient Boost

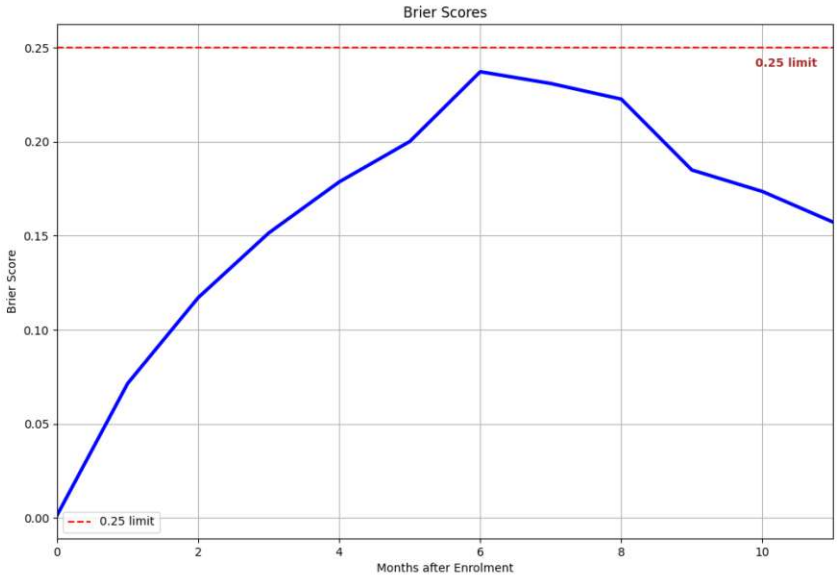

Figure 5.6: Time-Dependent Brier Scores for the Gradient Boost

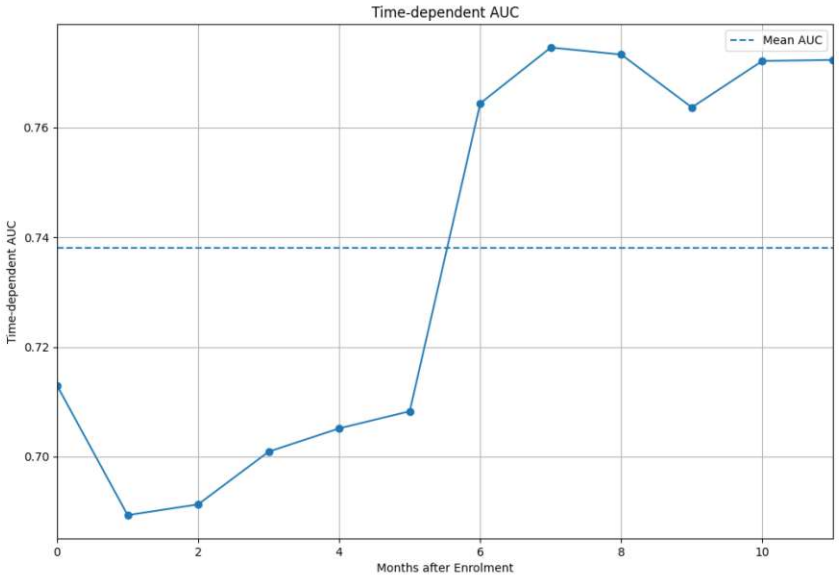

Figure 5.7: Time-Dependent AUC Curve for the Random Forest

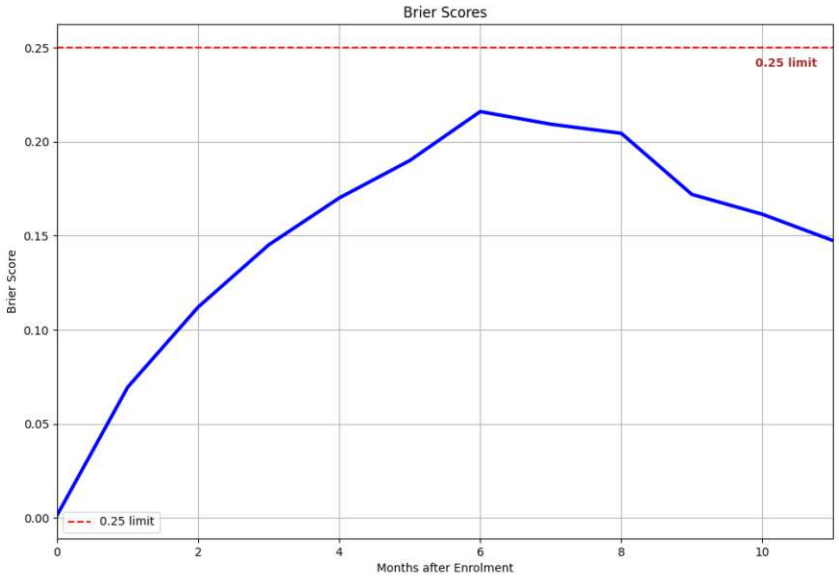

Figure 5.8: Time-Dependent Brier Scores for the Random Forest

5.2 Time-Variant Methods

Similarly to Table 5.1 in Section 5.1, Table 5.2 below summarises the performance of the time-variant methods in this study. The details for the base Kaplan-Meier is also provided to faciliate comparison against a known basis. Figures 5.9 to 5.16 follow and show the individual time-dependent AUC and Brier curves for the time-variant methods.

Model Name	C-Index	Mean AUC	Int. Brier	RMSE
Kaplan-Meier (Base)	0.640	0.651	0.175	0.0967
Extended Cox	0.651	0.914	0.076	0.1192
LTRC Forest (full dataset)	0.677	0.771	0.156	0.0989
LTRC Forest (reduced dataset)	0.680	0.775	0.152	0.0965
LTRC Forest (minimal dataset)	0.661	0.678	0.168	0.1679

Table 5.2: Summary of time-variant algorithms used in comparison to the original Kaplan-Meier base

The Extended Cox results showed substantially more accurate forecasting with better AUC values and Brier scores across the board, although it showed substantially worse RMSE values. By contrast, the LTRC forest showed modest improvements over the Kaplan-Meier as well as the original time-invariant random forest (refer Table 5.1), although RMSE remained relatively high.

The unexpectedly strong results for the extended Cox model suggests an issue in implementation; one possible cause could be the introduction of immortal time bias in the actual prediction implementation, which was originally designed for time-invariant survival analysis. By contrast, the more down-to-earth values of the LTRC forest were made directly from Yao's implementation which explicitly discusses creating survival curves that only take past values into account [84]. This will be discussed further in Chapter 6.

The reduced dataset produced very similar results to the full dataset with values showing that manually paring the dataset does have some advantages, such as theoretically better performance and better generalising ability. On the other hand, the minimal dataset shows performance that is in between that of the base Kaplan-Meier and the other LTRC forest models. This shows that the curation of data would be very useful prior to training any model on Yao's implementation of the time-variant relative risk forest. For all relative risk forest models, the time-dependent Brier and AUC curves exhibited similar shapes as expected.

In Section 5.3, we will further discuss the performance improvements obtained from the reduced and minimal datasets and the clear advantage that comes from working on smaller datasets.

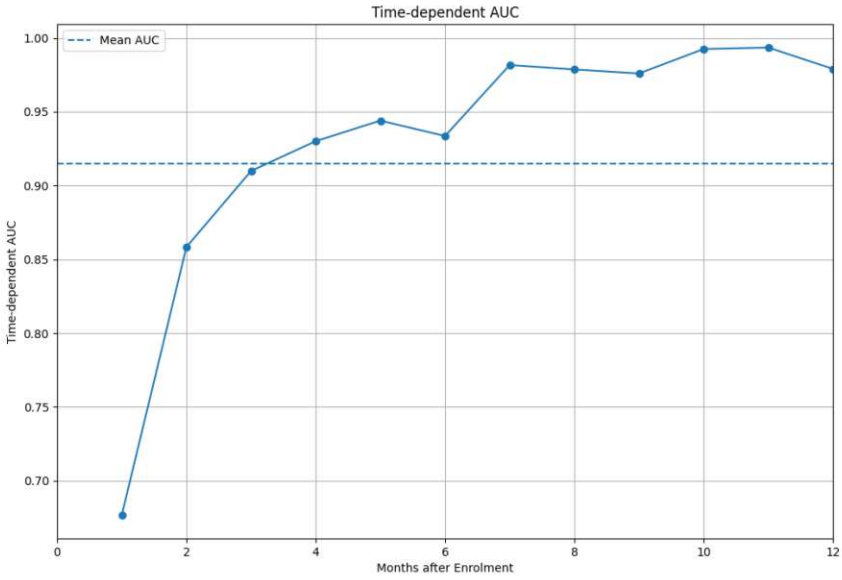

Figure 5.9: Time-Dependent AUC Curve for the Extended Cox Model which incorporates time-variant variables

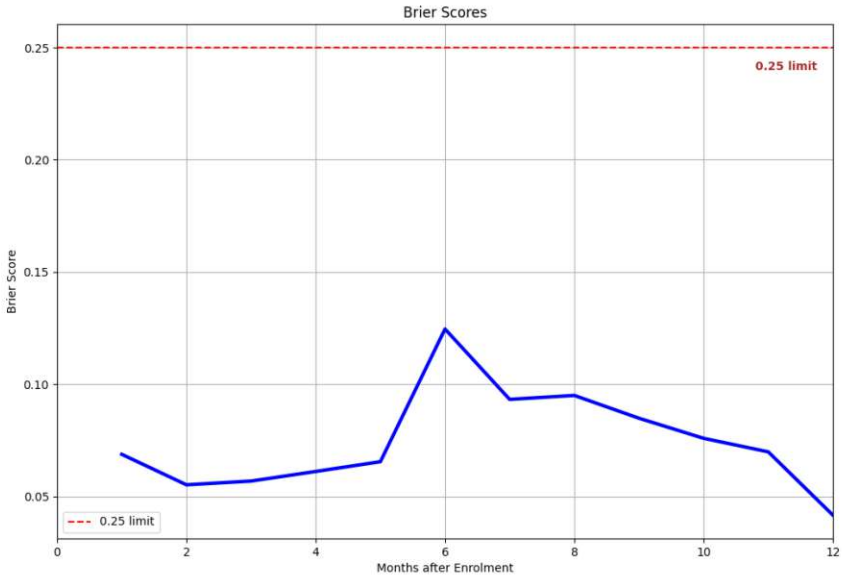

Figure 5.10: Time-Dependent Brier Scores for the Extended Cox Model

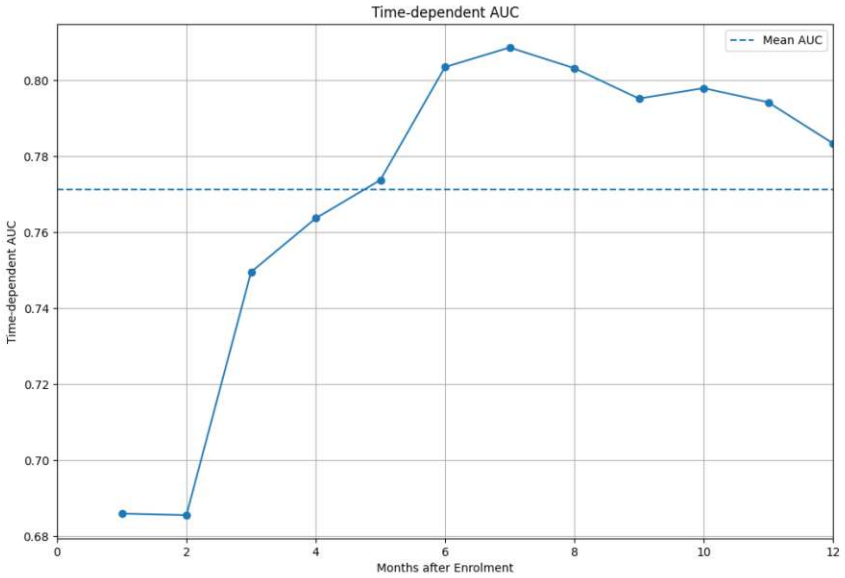

Figure 5.11: Time-Dependent AUC Curve for the LTRC Forest model with the full time-variant dataset

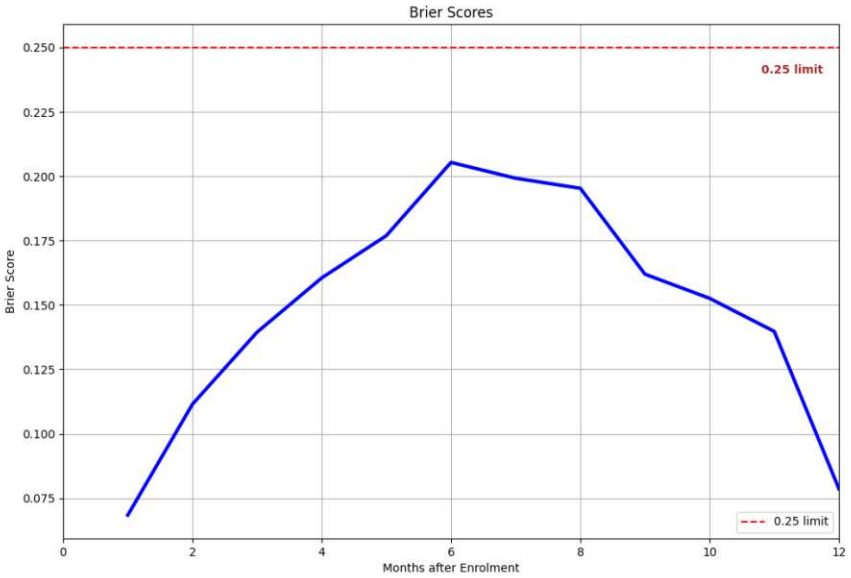

Figure 5.12: Time-Dependent Brier Scores for the LTRC Forest model with the full time-variant dataset

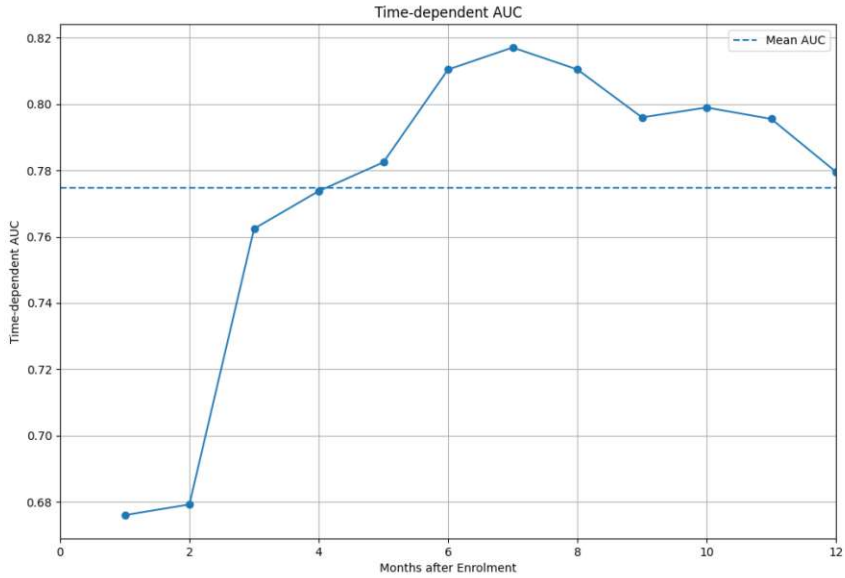

Figure 5.13: Time-Dependent AUC Curve for the LTRC Forest model with the reduced time-variant dataset

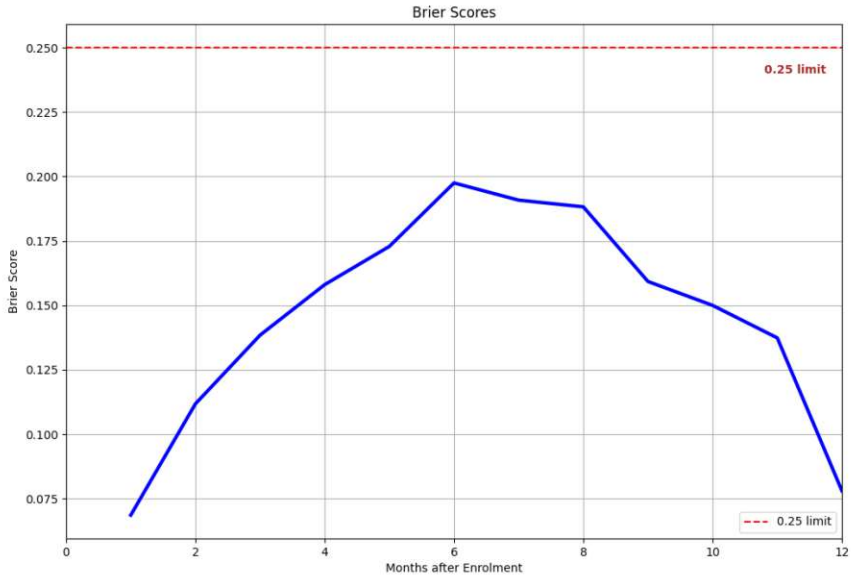

Figure 5.14: Time-Dependent Brier Scores for the LTRC Forest model with the reduced time-variant dataset

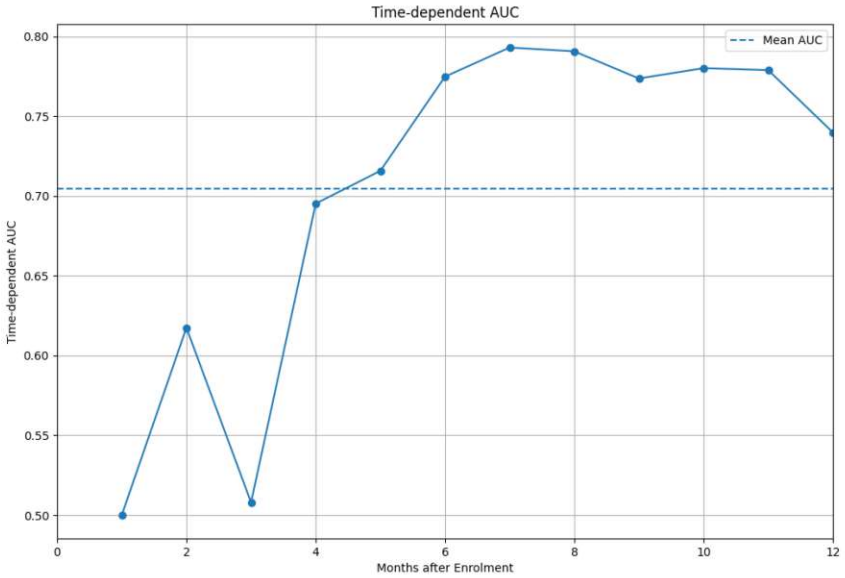

Figure 5.15: Time-Dependent AUC Curve for the LTRC Forest model with the minimal time-variant dataset

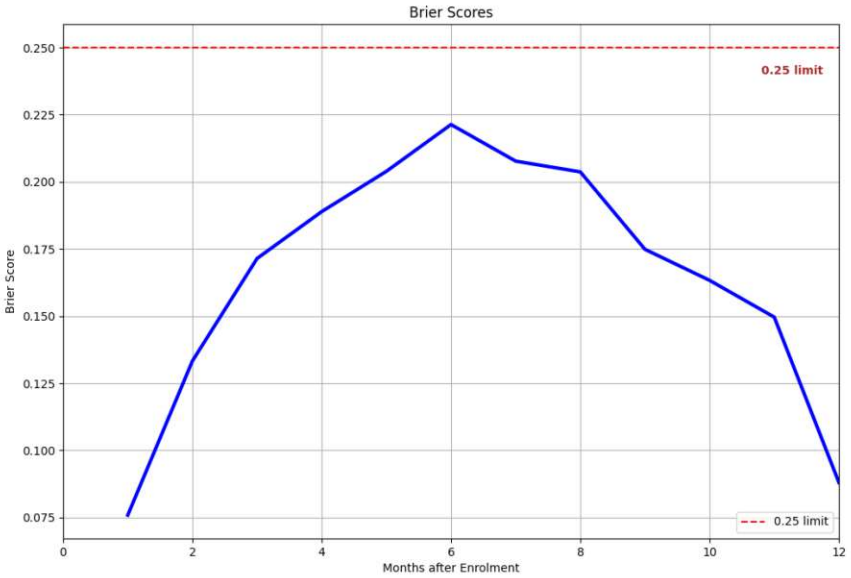

Figure 5.16: Time-Dependent Brier Scores for the LTRC Forest model with the minimal time-variant dataset

5.3 Computational Performance

Table 5.3 shows the time required to run the appropriate hyperparameter tuning and to train the model. For all runs, 50 cores from the Fangorn server described in Section 4.7 were used.

In general, basic methods had acceptable times (except for Gradient Boost) which are practical for retraining to incorporate new variables or test other hypotheses. Noting the size of the dataset, Gradient Boost performed substantially worse from a computational time perspective in comparison with all other methods.

Model Name	Time for Hyperparmeter Tuning
Basic Methods	
Kaplan-Meier (Base)	<1s*
Cox Elastic-Net	31m 20s
Gradient Boost	117h 56m 01s
Random Forest	1h 46m 00s
Time-Dependent Variable Methods	
Extended Cox	3h 36m 27s
LTRC Forest (full)	163h 20m 20s†
LTRC Forest (reduced)	156h 40m 30s
LTRC Forest (minimal)	26h 14m 58s

Table 5.3: Summary of times required to run 50 iterations on a Random Search algorithm with 5-fold cross-validation for hyperparameter tuning by model type, except for LTRC relative risk forests where the time shown is for a single iteration training run

As can be expected, incorporating time-dependent variables substantially blew out the time required to train the models: 7x for the extended Cox model compared to the normal Cox and 92x when comparing a single full dataset run of the relative risk forest against the full 50-iteration 5-fold cross-validation training times for the random forest. Surprisingly, the reduced dataset did not show substantial time improvements over the full dataset although the minimal dataset had an 84% reduction to 26 hours. Nonetheless, the substantial time requirements made it impractical to run full hyperparameter tuning on the time-variant LTRC forest.

*no hyperparameter tuning was done for the Kaplan-Meier

†time for all LTRC forest runs was for a single run without hyperparameter tuning or cross-validation

CHAPTER 6

Discussion

6.1 Overall Model Performance

Compared to the most basic model, namely the Kaplan-Meier model stratified by product type, most models did not perform substantially better at discriminating the order of client churn (as evidenced by the C-index values obtained in Table 5.1). However, the random forest managed to obtain substantially improved performance on AUC and Brier scores, suggesting a substantial increase in accuracy although discrimination still remains comparable. In other words, the random forest was predicting survival times more accurately, although it ranks the customers in the test data similarly to the base model.

The fact that ensemble learners performed substantially better than the Cox equivalent suggests that the proportional hazard assumption is definitely incorrect, but also that the variables not used in stratification of the base model provide the strongest signal and the main driver for survival time. Nonetheless, the ensemble learners, particularly the random forest appeared to be able to find a signal in the mass of relatively weak variables. The possibility also remains that there are other variables during the customer onboarding process which may provide a stronger signal to the model.

However, this performance improvement comes at a substantial computational cost, with model training requiring several hours for the time-invariant versions. Nonetheless, this seems to be a reasonable time should the model prove itself useful in predicting the expected lifetime of a customer.

The pipeline that has been built is capable of taking in any new variables as long as the data format is in the format described in Sections 2.3 and 2.5.2. This means experiments can be run with new variables that may potentially provide some predictive power; however, this also creates the potential risks of data mining and p-hacking.

6.2 The Value of Time-Dependent Variables

In terms of hypothesised effect, there was a higher expectation that the time-variant variables would provide a much stronger effect of improvement. As it were, the random forest implementation provided substantial improvements in accuracy (through the AUC and Brier score metrics) but only modest improvements in discrimination (through the C-index).

However, the Cox implementation which severely underperformed in the basic variant appeared to provide substantial improvements in accuracy with a 0.91 mean AUC value. This lack of agreement with the relative risk forest is a cause for suspicion, particularly because of the weaker performance against outliers as evidenced by its substantially higher RMSE.

While this effect could be true, and other research using similar methodology shows the same effect whereby the extended Cox performs better than the relative risk forest[92], certain aspects of the extended Cox implementation gives some reason to pause. As discussed in Section 4.5.2, the well-received Python `lifelines` library does not provide a native function to generate a survival curve, and the R `survival` package also does not provide an entirely straightforward survival curve generating function. In both cases, the authors have provided documentation warning against the immortal time bias discussed in Section 2.5.1. Although the immortal time bias issue discussed in Section 2.5.1 appears to be avoidable with appropriate study design, the reluctance of the creators of both available libraries to provide an explicit predictor function and their statements in documentation about it suggest that the naïve generation of the survival curve somehow introduces future-timed covariates into the training data. Extensive reading of available literature did not appear to adequately explain this issue and understanding this would be highly recommended as a future piece of work.

However, another distinct possibility of the cause of this is simple data leakage. With the naïve implementation of the Extended Cox and its survival curve generation, future data could easily be provided to the algorithm which then results in substantially better, albeit nonsensical predictions.

On the other hand, Yao's implementation of LTRCForests explicitly discusses the usage of the counting algorithm in her implementation to avoid this bias. In other words, when looking at the results of this study, that of the LTRCforest algorithms would be more reliable.

> "To predict, we would need to know the covariates values beyond the observed times, but if we knew that, we would also know if the subject was still alive or not!"
>
> (Cam Davidson-Pilon, `lifelines` author[19])

6.3 Considerations on Using Time-Dependent Models

Notwithstanding the issues discussed above in Section 6.2, another major consideration of using time-dependent variables in survival analysis appears to be the cutting-edge nature of such problems. Unlike most standard machine learning applications where highly customisable libraries for every implementation are readily available, even for deep learning applications, libraries for time-varying survival analysis are not as common. Furthermore, the applications used are clearly designed for low-dimensional and smaller datasets compared to the ones used in this work.

This work has effectively been a proof of concept to show that it is possible to use these libraries in a customer churn application; however, in a corporate context which is usually built towards the Python language, there is a complete dearth of time-variant libraries for survival analysis.

In addition to the issues discussed in Section 6.2, it would be highly valuable to dedicate further work to either discovering more efficient algorithms or programming more efficient implementations which work better with large datasets.

CHAPTER 7

Conclusion and Future Work

This thesis investigated the application of survival analysis methods to a customer churn context in the edtech sector. Taking a dataset containing several hundred thousand clients, we tested and compared various classic statistical approaches as well as machine learning approaches and compared them to the naïve approach of using simple segmentation and Kaplan-Meier survival curves. In addition, we also investigated the inclusion of time-variant variables to determine if the model performed better with their inclusion or otherwise.

In general, we only obtained a modest improvement with the usage of machine learning methods, particularly the random forest model. Most of this improvement was in accuracy (namely, estimation of survival time) as opposed to discrimination (i.e. determining which subjects will fall away first). As the corporate goal of this project was to improve estimation, the accuracy improvement is deemed to be satisfactory.

While including time-variant variables appeared to further improve the performance of our models, it also exposed some potential issues in production-level usage:

Firstly, the lack of available libraries in Python makes it a challenge to integrate into normal production pipelines. The libraries that are currently available, while academically robust, struggle when dealing with large datasets even for what in a commercial context would be considered a modestly-sized dataset.

Secondly, there was a lack of clarity in terms of applying basic survival curve generation functions to simpler extended Cox functions which created doubt in the results; particularly as the results suggested simpler Cox functions provided substantially better performance compared to other methodologies. Investigating this effect would be a good focus for future work building on this thesis.

List of Figures

List of Tables

Acronyms

AFT accelerated failure time. 16

AUC area under the curve. 41, 42, 47, 53, 54, 61

C-index Concordance Index. 2, 17, 18, 41, 42, 53, 54, 61

CAC customer acquisition costs. 3–5

COGS cost of goods sold. 5

CPIU counting process information unit. 30–32, 59

GBM gradient boosting model. 28

IBS Integrated Brier Score. 2, 18, 42

IPCW inverse probability of censoring weight. 17, 18

KPI key performance indicator. 3, 4

LTV lifetime value. 1, 4–6

ML machine learning. 23, 25, 30, 42

MRR monthly recurring revenue. 3, 5

PCR principal component regression. 35

RF-SLAM Random Forest for Survival, Longitudinal, and Multivariate data. 30, 31, 59

SVM support vector machine. 24

Bibliography

[1] P. Pfeifer, "The optimal ratio of acquisition and retention costs," *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 13, pp. 179–188, 01 2005.

[2] F. Reichheld and T. Teal, *The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value*. Audio business book summaries from Harvard Business School Press, Harvard Business School Press, 1996.

[3] H. Kvamme, O. Borgan, and I. Scheel, "Time-to-event prediction with neural networks and cox regression," 2019.

[4] A. Manzoor, M. Atif Qureshi, E. Kidney, and L. Longo, "A review on machine learning methods for customer churn prediction and recommendations for business practitioners," *IEEE Access*, vol. 12, pp. 70434–70463, 2024.

[5] s. Ameri, M. Jahanbani Fard, R. B. Chinnam, and C. Reddy, "Survival analysis based framework for early prediction of student dropouts," 10 2016.

[6] J. A. Martínez-Carrascal, M. Hlosta, and T. Sancho-Vinuesa, "Using survival analysis to identify populations of learners at risk of withdrawal: Conceptualization and impact of demographics," *The International Review of Research in Open and Distributed Learning*, vol. 24, p. 1–21, Feb. 2023.

[7] "The beginnings of subscription publication in the seventeenth century," *Modern Philology*, vol. 29, no. 2, pp. 199–224, 1931.

[8] M. Ritter and H. Schanz, "The sharing economy: A comprehensive business model framework," *Journal of Cleaner Production*, vol. 213, pp. 320–331, 2019.

[9] F. Bandulet, "Software-as-a-service as disruptive innovation in the enterprise application market: An empirical analysis of revenue growth and profitability among saas providers (2005 – 2015).," 09 2017.

[10] Y. Ge, S. He, J. Xiong, and D. E. Brown, "Customer churn analysis for a software-as-a-service company," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 106–111, 2017.

[11] C. W. J. Lindström, B. Maleki Vishkaei, and P. De Giovanni, "Subscription-based business models in the context of tech firms: theory and applications," *International Journal of Industrial Engineering and Operations Management*, vol. 6, p. 256–274, Aug. 2023.

[12] P. E. Pfeifer, M. E. Haskins, and R. M. Conroy, "Customer lifetime value, customer profitability, and the treatment of acquisition spending," *Journal of Managerial Issues*, vol. 17, no. 1, pp. 11–25, 2005.

[13] V. Kumar, R. Venkatesan, T. Bohling, and D. Beckmann, "Practice prize report—the power of clv: Managing customer lifetime value at ibm," *Marketing Science*, vol. 27, no. 4, pp. 585–599, 2008.

[14] C. Heitz, M. Dettling, and A. Ruckstuhl, "Modelling customer lifetime value in contractual settings," *International Journal of Services Technology and Management*, vol. 16, no. 2, pp. 172–190, 2011.

[15] D. Kleinbaum and M. Klein, *Survival Analysis: A Self-Learning Text, Third Edition*. Statistics for Biology and Health, Springer New York, 2016.

[16] T. Stainer, "The notion of Censoring in Survival Analysis — jigso.com." `https://jigso.com/the-notion-of-censoring-in-survival-analysis/`, 12-02-2021. [Accessed 24-11-2024].

[17] M. via Wikimedia Commons, "Survival function 1," 06.10.2016.

[18] T. G. Clark, M. J. Bradburn, S. B. Love, and D. G. Altman, "Survival analysis part i: Basic concepts and first analyses," *British Journal of Cancer*, vol. 89, p. 232–238, July 2003.

[19] C. Davidson-Pilon, "lifelines: survival analysis in python," *Journal of Open Source Software*, vol. 4, no. 40, p. 1317, 2019.

[20] K.-M. Leung, R. M. Elashoff, and A. A. Afifi, "Censoring issues in survival analysis," *Annual Review of Public Health*, vol. 18, p. 83–104, May 1997.

[21] S. W. Lagakos, "General right censoring and its impact on the analysis of survival data," *Biometrics*, vol. 35, no. 1, pp. 139–156, 1979.

[22] R. Oller, G. Gómez, and M. L. Calle, "Interval censoring: Identifiability and the constant-sum property," *Biometrika*, vol. 94, no. 1, pp. 61–70, 2007.

[23] S. W. Lagakos and J. S. Williams, "Models for censored survival analysis: A cone class of variable-sum models," *Biometrika*, vol. 65, no. 1, pp. 181–189, 1978.

[24] M. L. Moeschberger, "Life tests under dependent competing causes of failure," *Technometrics*, vol. 16, no. 1, pp. 39–47, 1974.

[25] M. Zhou, *Empirical Likelihood Method in Survival Analysis*. Chapman & Hall/CRC Biostatistics Series, CRC Press, 2015.

[26] R. C. Paul Meier, Theodore Karrison and H. Xie, "The price of kaplan–meier," *Journal of the American Statistical Association*, vol. 99, no. 467, pp. 890–896, 2004.

[27] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958.

[28] F. E. Harrell, *Cox Proportional Hazards Regression Model*, pp. 475–519. Cham: Springer International Publishing, 2015.

[29] S. Pölsterl, "scikit-survival: A library for time-to-event analysis built on top of scikit-learn," *Journal of Machine Learning Research*, vol. 21, no. 212, pp. 1–6, 2020.

[30] T. M. Therneau, *A Package for Survival Analysis in R*, 2020.

[31] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for cox's proportional hazards model via coordinate descent," *Journal of Statistical Software*, vol. 39, no. 5, 2011.

[32] M. Schemper, "Cox analysis of survival data with non-proportional hazard functions," *The Statistician*, vol. 41, no. 4, p. 455, 1992.

[33] E. T. Lee and J. W. Wang, "Identification of prognostic factors related to survival time: Cox proportional hazards model," Apr. 2003.

[34] M. J. Stensrud and M. A. Hernán, "Why test for proportional hazards?," *JAMA*, vol. 323, p. 1401, Apr. 2020.

[35] A. Sjölander and P. W. Dickman, "Why test for proportional hazards—or any other model assumptions?," *American Journal of Epidemiology*, vol. 193, pp. 926–927, 02 2024.

[36] N. E. Breslow, "Analysis of survival data under the proportional hazards model," *International Statistical Review / Revue Internationale de Statistique*, vol. 43, no. 1, pp. 45–57, 1975.

[37] G. BEIS, A. ILIOPOULOS, and I. PAPASOTIRIOU, "An overview of introductory and advanced survival analysis methods in clinical applications: Where have we come so far?," *Anticancer Research*, vol. 44, no. 2, pp. 471–487, 2024.

[38] W. R. Swindell, "Accelerated failure time models provide a useful statistical framework for aging research," *Experimental Gerontology*, vol. 44, no. 3, pp. 190–200, 2009.

[39] N. Stroustrup, W. E. Anthony, Z. M. Nash, V. Gowda, A. Gomez, I. F. López-Moyado, J. Apfeld, and W. Fontana, "The temporal scaling of caenorhabditis elegans ageing," *Nature*, vol. 530, p. 103–107, Jan. 2016.

[40] A. L. Koch, "The logarithm in biology 1. mechanisms generating the log-normal distribution exactly," *Journal of Theoretical Biology*, vol. 12, no. 2, pp. 276–290, 1966.

[41] J. W. Gamel and R. L. Vogel, "Comparison of parametric and non-parametric survival methods using simulated clinical data," *Statistics in Medicine*, vol. 16, p. 1629–1643, July 1997.

[42] J. M. Bland and D. G. Altman, "The logrank test," *BMJ*, vol. 328, p. 1073, Apr. 2004.

[43] E. Longato, M. Vettoretti, and B. Di Camillo, "A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models," *Journal of Biomedical Informatics*, vol. 108, p. 103496, 2020.

[44] F. E. Harrell, "Evaluating the yield of medical tests," *JAMA: The Journal of the American Medical Association*, vol. 247, p. 2543, May 1982.

[45] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in Medicine*, vol. 30, no. 10, pp. 1105–1117, 2011.

[46] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in Medicine*, vol. 18, p. 2529–2545, Sept. 1999.

[47] D. Collett, *Modelling Survival Data in Medical Research*. Chapman & Hall/CRC Texts in Statistical Science, CRC Press, 2015.

[48] S. Suissa, "Immortal time bias in pharmacoepidemiology," *American Journal of Epidemiology*, vol. 167, p. 492–499, Jan. 2008.

[49] M. H. GAIL, "Does cardiac transplantation prolong life?: A reassessment," *Annals of Internal Medicine*, vol. 76, p. 815, May 1972.

[50] H. Gurney, D. Dodwell, N. Thatcher, and M. Tattersall, "Escalating drug delivery in cancer chemotherapy: A review of concepts and practice – part 1," *Annals of Oncology*, vol. 4, no. 1, pp. 23–34, 1993.

[51] E. A. Terry Therneau, Cynthia Crowson, "Using Time Dependent Covariates and Time Dependent Coeffcients in the Cox Model." `https://cran.r-project.org/web/packages/survival/vignettes/timedep.pdf`, 2024. [Accessed 07-12-2024].

[52] W. Luo, D. Phung, T. Tran, S. Gupta, S. Rana, C. Karmakar, A. Shilton, J. Yearwood, N. Dimitrova, T. B. Ho, S. Venkatesh, and M. Berk, "Guidelines for developing and reporting machine learning predictive models in biomedical research: A multidisciplinary view," *J Med Internet Res*, vol. 18, p. e323, Dec 2016.

[53] P. Doupe, J. Faghmous, and S. Basu, "Machine learning for health services researchers," *Value in Health*, vol. 22, pp. 808–815, Jul 2019.

[54] Y. Huang, J. Li, M. Li, and R. R. Aparasu, "Application of machine learning in predicting survival outcomes involving real-world data: a scoping review," *BMC Medical Research Methodology*, vol. 23, Nov. 2023.

[55] P. Filzmoser, "Advanced methods for regression and classification: Lecture notes," October 2021.

[56] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[57] P. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets," pp. 655–660, 11 2007.

[58] F. M. Khan and V. B. Zubek, "Support vector regression for censored data (svrc): A novel tool for survival analysis," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 863–868, 2008.

[59] V. Van Belle, K. Pelckmans, J. A. Suykens, and S. Van Huffel, "Support vector machines for survival analysis," in *Proceedings of the third international conference on computational intelligence in medicine and healthcare (cimed2007)*, pp. 1–8, 2007.

[60] L. Evers and C.-M. Messow, "Sparse kernel methods for high-dimensional survival data," *Bioinformatics*, vol. 24, pp. 1632–1638, 05 2008.

[61] V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. Suykens, "Support vector methods for survival analysis: a comparison between ranking and regression approaches," *Artificial Intelligence in Medicine*, vol. 53, no. 2, pp. 107–118, 2011.

[62] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[63] T. K. Chun, "Random forest explain," 2021.

[64] N. Raghavan, A. M. I. M. De Bondt, W. Talloen, D. Moechars, H. W. H. Göhlmann, and D. Amaratunga, "The high-level similarity of some disparate gene expression measures," *Bioinformatics*, vol. 23, pp. 3032–3038, 09 2007.

[65] D. Ghosh and J. Cabrera, "Enriched random forest for high dimensional genomic data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 5, pp. 2817–2828, 2022.

[66] P. F. Smith, S. Ganesh, and P. Liu, "A comparison of random forest regression and multiple linear regression for prediction in neuroscience," *Journal of Neuroscience Methods*, vol. 220, no. 1, pp. 85–91, 2013.

[67] A. M. Molinaro, S. Dudoit, and M. J. van der Laan, "Tree-based multivariate regression and density estimation with right-censored data," *Journal of Multivariate Analysis*, vol. 90, pp. 154–177, July 2004.

[68] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M. J. Van Der Laan, "Survival ensembles," *Biostatistics*, vol. 7, pp. 355–373, 12 2005.

[69] H. Ishwaran, E. Blackstone, C. Pothier, and M. Lauer, "Relative risk forests for exercise heart rate recovery as a predictor of mortality," *Journal of the American Statistical Association*, vol. 99, pp. 591–600, 02 2004.

[70] H. Ishwaran, U. Kogalur, E. Blackstone, and M. Lauer, "Random survival forests," *The Annals of Applied Statistics*, vol. 2, 12 2008.

[71] H. Ishwaran, M. Lu, and U. B. Kogalur, "randomForestSRC: variable importance (VIMP) with subsampling inference vignette." `http://randomforestsrc.org/articles/vimp.html`, 2021. accessed date: 09.12.2024.

[72] E. Morris, K. He, Y. Li, Y. Li, and J. Kang, "SurvBoost: An R Package for High-Dimensional Variable Selection in the Stratified Proportional Hazards Model via Gradient Boosting," *The R Journal*, vol. 12, no. 1, pp. 105–117, 2020.

[73] Y. Chen, Z. Jia, D. Mercola, and X. Xie, "A gradient boosting algorithm for survival analysis via direct optimization of concordance index," *Computational and Mathematical Methods in Medicine*, vol. 2013, no. 1, p. 873595, 2013.

[74] S. Wiegrebe, P. Kopper, R. Sonabend, B. Bischl, and A. Bender, "Deep learning for survival analysis: a review," *Artif. Intell. Rev.*, vol. 57, Feb. 2024.

[75] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, Feb. 2018.

[76] D. Faraggi and R. Simon, "A neural network model for survival data," *Statistics in Medicine*, vol. 14, no. 1, pp. 73–82, 1995.

[77] K. L. Pickett, K. Suresh, K. R. Campbell, S. Davis, and E. Juarez-Colunga, "Random survival forests for dynamic predictions of a time-to-event outcome using a longitudinal biomarker," *BMC Medical Research Methodology*, vol. 21, p. 216, Oct 2021.

[78] H. Moradian, W. Yao, D. Larocque, J. S. Simonoff, and H. Frydman, "Dynamic estimation with random forests for discrete-time survival data," 2021.

[79] S. Cygu, H. Seow, J. Dushoff, and B. M. Bolker, "Comparing machine learning approaches to incorporate time-varying covariates in predicting cancer survival time," *Scientific Reports*, vol. 13, p. 1370, Jan 2023.

[80] H. T. Nguyen, H. D. Vasconcellos, K. Keck, J. P. Reis, C. E. Lewis, S. Sidney, D. M. Lloyd-Jones, P. J. Schreiner, E. Guallar, C. O. Wu, J. A. Lima, and B. Ambale-Venkatesh, "Multivariate longitudinal data for survival analysis of cardiovascular event prediction in young adults: insights from a comparative explainable study," *BMC Medical Research Methodology*, vol. 23, p. 23, Jan 2023.

[81] C. Morgan, "Landmark analysis: A primer," *Journal of Nuclear Cardiology*, vol. 26, 02 2019.

[82] J. G. Ibrahim, H. Chu, and L. M. Chen, "Basic concepts and methods for joint models of longitudinal and survival data," *Journal of Clinical Oncology*, vol. 28, p. 2796–2801, June 2010.

[83] S. Wongvibulsin, K. C. Wu, and S. L. Zeger, "Clinical risk prediction with random forests for survival, longitudinal, and multivariate (rf-slam) data analysis," *BMC Medical Research Methodology*, vol. 20, Dec. 2019.

[84] W. Yao, H. Frydman, D. Larocque, and J. S. Simonoff, "Ensemble methods for survival function estimation with time-varying covariates," *Statistical Methods in Medical Research*, vol. 31, no. 11, pp. 2217–2236, 2022. PMID: 35895510.

[85] W. Yao, H. Frydman, D. Larocque, and J. S. Simonoff, "Ltrcforests: Ensemble methods for survival data with time-varying covariates," July 2020.

[86] G. Novack, "Building a One Hot Encoding Layer with TensorFlow." `https://towardsdatascience.com/building-a-one-hot-encoding-layer-with-tensorflow-f907d686bf39`, 2020. [Accessed 27-12-2024].

[87] Y. J. Shen and S. G. Huang, "Improve survival prediction using principal components of gene expression data," *Genomics Proteomics Bioinformatics*, vol. 4, pp. 110–119, May 2006.

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

[89] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.

[90] A. Zheng, N. Shelby, and E. Volckhausen, "Evaluating machine learning models," *Machine Learning in the AWS Cloud*, 2019.

[91] T. M. Therneau, *A Package for Survival Analysis in R*, 2024. R package version 3.8-3.

[92] L. Badolato, A. G. Decter-Frain, N. Irons, M. L. Miranda, E. Walk, E. Zhalieva, M. Alexander, U. Basellini, and E. Zagheni, "The limits of predicting individual-level longevity," Feb. 2023.