Universitat Oberta de Catalunya

Generación automática de un informe de resultados orientado a análisis de supervivencia

Paula Fernández Martínez

Área 2 – Subárea 2: Análisis de datos

Máster en Bioinformática y Bioestadística

Nuria Pérez Álvarez

(Carles Ventura Royo)

06/2022

FICHA DEL TRABAJO FINAL

Título:	Generación automática de un informe de resultados orientado a análisis de supervivencia
Nombre autor/a:	Paula Fernández Martínez
Nombre PDC:	Nuria Pérez Álvarez
Nombre PRA:	Carles Ventura Royo
Fecha de entrega:	06/2022
Titulación:	Máster en Bioinformática y Bioestadística
Área:	Área 2 – Subárea 2: Análisis de datos
Idioma:	Castellano
Núm. de créditos:	15
Palabras clave:	Análisis de Supervivencia, R, Machine Learning, VIH, Antirretroviral

Resumen

Los análisis de supervivencia se utilizan para analizar el tiempo que transcurre hasta que se produce un evento, que debe ser binario (éxito/fracaso). Estos estudios tienen un incoveniente y es que, como conllevan seguimiento, es usual la aparición de censuras: se produce una censura cuando el tiempo está incompleto ya sea porque el evento no se llega a producir durante el estudio o porque se pierde el seguimiento del paciente. Por tanto, para su análisis, se deben utilizar modelos y algoritmos específicamente diseñados para tratar con estas censuras. Si bien entre los métodos tradicionales encontramos el modelo de regresión de Cox y las curvas de Kaplan-Meier, a lo largo de este trabajo se planteó el uso de tres algoritmos de Machine Learning: los árboles de supervivencia, el algoritmo Naive Bayes y el algoritmo Support Vector Machine (SVM). Además, se diseñó un informe dinámico en R que permite automatizar los análisis a partir de diferentes conjuntos de datos con pequeñas modificaciones. Se analizaron 112 pacientes infectados de VIH, 91 hombres y 15 mujeres, con una edad media de 38.1 años, divididos en dos tratamientos. Tras aplicar los algoritmos, se obtuvieron los mejores resultados con el SVM de kernel lineal, mientras que el peor resultado lo ofrecieron los árboles de supervivencia. Respecto al algoritmo Naive Bayes, se encuentra a mitad de camino entre los otros dos, lo que invita a seguir explorándolo para intentar mejorar su implementación. Por tanto, este trabajo aporta nueva información sobre la aplicación de las técnicas de Machine Learning orientadas a análisis de supervivencia del campo de la biomedicina, cuya implementación aún no se encuentra completamente consolidada.

Abstract

Survival analyses are used to investigate the time to event occurrence, which should be binary (success/failure). These studies have a disadvantage due to censoring, which occurs when the time is incomplete either because the event does not occur during the study or because the patient is lost to follow-up. Therefore, models and algorithms specifically designed to deal with these censorships must be used. While traditional methods include the Cox regression model and Kaplan-Meier curves, this study proposes the use of three Machine Learning algorithms: survival trees, Naive Bayes and Support Vector Machine (SVM). In addition, a dynamic report was designed in R to automate the analyses from different datasets with minor modifications. A total of 112 HIV-infected patients, 91 men and 15 women, with a mean age of 38.1 years, divided into two treatments, were analysed. After applying the algorithms, the best results were obtained with the linear kernel SVM, while the worst result was obtained with the survival trees. Regarding the Naive Bayes algorithm, it is halfway between the other two, which invites further exploration to try to improve its implementation. Therefore, this work provides new information on the application of Machine Learning techniques oriented to survival analysis in the biomedical field, whose implementation is not yet fully consolidated.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Contexto y justificación

Los análisis de supervivencia se utilizan para analizar el tiempo que transcurre hasta que se produce el evento de interés, que debe ser binario (éxito/fracaso). Es decir, la particularidad de estos análisis es que lo que nos interesa saber es el tiempo que pasa hasta el evento, no simplemente con qué frecuencia ocurre [1], y para ello es necesario llevar a cabo un estudio longitudinal.

Por tanto, la variable respuesta es el tiempo hasta el evento, que suele denominarse tiempo de supervivencia. Esta variable suele ser continua y conlleva una problemática: el tiempo puede estar incompleto en algunos sujetos, y es aquí donde entra el concepto de "censura". Se considera una censura cuando el evento de interés no se observa debido a que el estudio finalizó antes de que ocurriese o a que se perdió el seguimiento del paciente [2].

Por ejemplo, si estamos analizando el tiempo de supervivencia en pacientes con cáncer de pulmón durante 10 años y uno de los individuos no fallece en ese tiempo, se considera una censura. De igual modo, si el paciente deja de asistir a las revisiones, también hablaremos de censura.

Este tipo de análisis son esenciales en los estudios biomédicos, ya que permiten evaluar los tiempos de supervivencia a las enfermedades en función de variables como el sexo o los trata-

mientos. Su importancia se ve reflejada en los múltiples estudios que los utilizan actualmente, como el de Ahmad et al. [3], Hsu et al. [4], Solomon et al. [5], o Chi et al.[6]

Si bien los análisis de supervivencia han mostrado ser muy útiles en los estudios de investigación, su realización conlleva algunos problemas y puede ser laboriosa. Es por ello por lo que se plantea la necesidad de hacer estos análisis más accesibles y poder explotar a fondo su potencial.

Entre los métodos tradicionales se encuentra la regresión de Cox o las curvas de Kaplan-Meier, entre otros, pero ya se están empezando a aplicar técnicas basadas en Machine Learning en el desarrollo de estos análisis, como son el algoritmo Support Vector Machine (SVM) [7], los árboles de supervivencia o las redes neuronales [2], e incluso existen paquetes de R diseñados para ello, como *mlr3proba* [8].

Sin embargo, a la hora de analizar los datos de un estudio, surge la pregunta: ¿qué método debo utilizar? ¿Hay alguno que ofrezca mejores resultados? Y eso sin contar la dificultad que puede conllevar el diseñar los análisis. Por tanto, intentaremos dar solución a esta problemática a lo largo del TFM.

1.2. Objetivos

Objetivo general

- Diseñar un informe dinámico que permita automatizar los análisis de supervivencia a partir de diferentes conjuntos de datos utilizando como muestra una serie de pacientes infectados por VIH que se encuentran bajo tratamiento antirretroviral.

1.2.1. Objetivos específicos

- Conocer y comprender los métodos de supervivencia clásicos y de Machine Learning.
- Determinar las ventajas y desventajas de los métodos de supervivencia clásicos y de Machine Learning.

- Seleccionar bibliográficamente cuáles son los métodos más adecuados para estudiar el tiempo hasta fracaso al tratamiento antirretroviral en una cohorte de pacientes infectados por VIH.
- Aplicar la metodología seleccionada a una cohorte de pacientes infectados por VIH.
- Comparar los métodos de aprendizaje automático seleccionados mediante el índice C de Harrell.
- Generar un informe dinámico y publicarlo en GitHub.

1.3. Enfoque y método

Para llevar a cabo este estudio se realizará una búsqueda bibliográfica exhaustiva de las metodologías actuales aplicadas a los análisis de supervivencia, con el objetivo de elegir la más adecuada para su aplicación a una cohorte de pacientes infectados por VIH.

Los datos utilizados en el TFM proceden del estudio Lake [9]. Se trata de un estudio prospectivo aleatorizado y multicéntrico donde se utilizan dos antirretrovirales diferentes. Se realizará una exploración de los datos para seleccionar las variables relevantes para el análisis y para poder enmarcar adecuadamente los resultados que se obtengan.

A continuación, se aplicarán las metodologías seleccionadas y se compararán mediante el índice C de Harrell.

Finalmente, se elaborará un informe dinámico para facilitar la aplicación del análisis en cualquier base de datos. Para ello se utilizará RStudio [10], donde el código se puede diseñar en un Compile Report o en un documento Markdown. En este caso, se utilizará un Compile Report.

Si bien se podría intentar adaptar algún informe existente, en este caso se optará por construir uno de cero, ya que un paso previo es elegir el mejor método para el análisis en función de los datos existentes y un informe pre-hecho podría no ajustarse a las condiciones observadas. Además, para la realización de la memoria se utilizará R en combinación con LaTeX.

1.4. Planificación

La planificación de las tareas que se muestra a continuación se ha llevado a cabo con el programa *GanttProject*. En la lista se han incluido las fechas de entrega de las diferentes PEC programadas por la UOC, que se utilizaron como guía para la planificación general.

Tarea

Nombre	Fecha de inicio	Fecha de fin
PEC 0 - Definición de los contenidos del trabajo	16/2/22	23/2/22
Revisar la documentación proporcionada por la tutora	16/2/22	18/2/22
Revisar la guía del estudiante	16/2/22	18/2/22
Redactar la propuesta de TFM	19/2/22	23/2/22
PEC 1 - Plan de trabajo	24/2/22	7/3/22
Realizar las correcciones propuestas por la tutora	24/2/22	26/2/22
Diseñar el plan de trabajo	27/2/22	2/3/22
Comenzar búsqueda bibliográfica	28/2/22	7/3/22
PEC 2 - Desarrollo del trabajo - Fase 2	8/3/22	11/4/22
Realizar las correcciones propuestas por la tutora	8/3/22	12/3/22
Continuar ampliando los contenidos teóricos	8/3/22	1/4/22
Realizar una exploración inicial de los datos	4/4/22	11/4/22
PEC 3 - Desarrollo del trabajo - Fase 2	12/4/22	16/5/22
Realizar las correcciones propuestas por la tutora	12/4/22	16/4/22
Aplicar los diferentes métodos de análisis	17/4/22	28/4/22
Árbol de supervivencia	17/4/22	22/4/22
Naive Bayes	20/4/22	25/4/22
Support Vector Machine	23/4/22	28/4/22
Comparar metodologías	29/4/22	7/5/22
Diseñar informe dinámico	8/5/22	16/5/22
PEC 4 - Cierre de la memoria	17/5/22	2/6/22
Realizar las correcciones propuestas por la tutora	17/5/22	21/5/22
Finalizar la redacción de la memoria final	21/5/22	2/6/22
PEC 5a - Elaboración de la presentación	30/5/22	6/6/22
Elaboración de la presentación	30/5/22	2/6/22
Realizar la exposición del TFM	3/6/22	6/6/22
PEC 5b - Defensa pública	13/6/22	23/6/22

Figura 1.1: Lista de tareas (*GanttProject*)

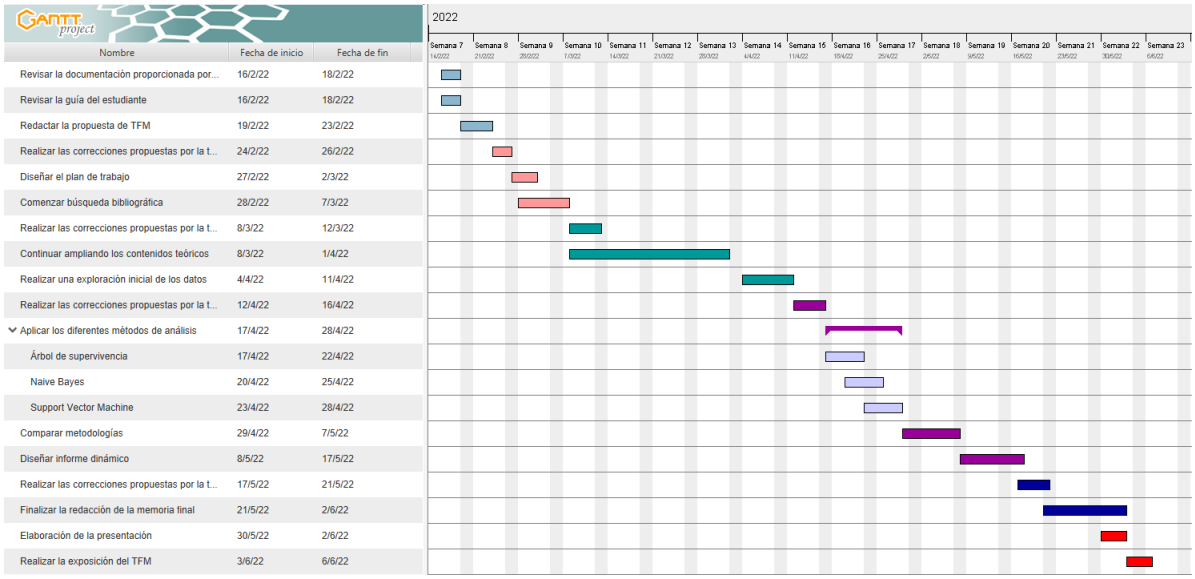

Figura 1.2: Diagrama de Gant de tareas (*GanttProject*)

PEC 0 - Definición de los contenidos

1. Revisar la documentación inicial proporcionada por la tutora sobre el TFM.
2. Revisar la guía del estudiante facilitada por la UOC.
3. Redactar la propuesta de TFM

PEC 1 - Plan de trabajo

1. Realizar las correcciones propuestas por la tutora.
2. Diseñar el plan de trabajo.
3. Comenzar una búsqueda bibliográfica exhaustiva para conocer los diferentes métodos existentes para la realización de análisis de supervivencia.

PEC 2 - Desarrollo - Fase 1

1. Realizar las correcciones propuestas por la tutora.

2. Continuar ampliando los contenidos teóricos.

3. Realizar una exploración inicial de los datos.

PEC 3 - Desarrollo - Fase 2

1. Realizar las correcciones propuestas por la tutora.

2. Aplicar los diferentes métodos encontrados sobre una base de datos real.

   - Árbol de supervivencia.
   - Naive Bayes.
   - Support Vector Machine.

3. Comparar los resultados obtenidos con los diferentes métodos y establecer cuál de ellos es más recomendable utilizar.

4. Diseñar un informe dinámico que aplique el análisis de forma automática sobre diferentes bases de datos sin necesidad de llevar a cabo modificaciones.

PEC 4 - Cierre de la memoria

1. Realizar las correcciones propuestas por la tutora.

2. Finalizar la redacción de la memoria final del TFM.

PEC 5a - Elaboración de la presentación

1. Elaborar la presentación del TFM.

2. Realizar la exposición del TFM.

PEC 5b - Defensa pública

1. Defender el TFM.

2. Responder en el acto público a las preguntas planteadas por el tribunal.

1.5. Breve resumen de los capítulos

El trabajo de distribuye en 5 capítulos, cada uno de los cuales se centra en explicar una parte importante del mismo y cuya información termina confluyendo en el quinto y último capítulo.

El trabajo comienza en el capítulo 1, donde se pone contexto al TFM, se plantean los objetivos y se expone la planificación diseñada para finalizarlo con éxito.

A continuación, en el capítulo 2 se recoge el estado actual del tema, y se explican los diferentes modelos que se van a utilizar para poder comprender correctamente los resultados.

Una vez explicados todos los conceptos necesarios para situar el trabajo, pasamos al capítulo 3, donde se explica detalladamente qué datos se van a utilizar y cuál va a ser la metodología a seguir para la realización de los análisis.

El siguiente paso se encuentra en el capítulo 4, donde se recogen los resultados obtenidos con los diferentes algoritmos y se dicute cuál es el más óptimo.

Finalmente, en el capítulo 5 se resumen los resultados obtenidos y se obtienen las conclusiones pertinentes. También se mira al futuro y se plantean diferentes puntos que podrían valorarse más adelante para complementar el presente trabajo.

Capítulo 2

Estado del arte

El objetivo principal de un análisis de supervivencia es conocer el tiempo que pasa hasta que ocurre el evento de interés. Este evento es binario (éxito/fracaso) y puede ser desde el desarrollo de una condición patológica hasta la renuncia a un puesto de trabajo. Para realizar el análisis, se toma una muestra de individuos en los que se miden una serie de covariables que aportarán información para la estimación de la variable respuesta, que es el tiempo de supervivencia.

La particularidad de estos análisis es que podemos trabajar con individuos en los que, por una causa u otra, no se observa el evento de interés. Estas "bajas"se denominan censuras, y es debido a ellas que los análisis que se pueden hacer de este tipo de datos se vuelven muy específicos. Esto es así hasta el punto de que cualquier algoritmo que quiera realizarse sobre ellos debe ser adaptado previamente al tratamiento de censuras.

Los análisis de supervivencia se pueden abordar desde dos puntos de vista: la aproximación estadística tradicional o los métodos de *Machine Learning*. Ambas metodologías tienen el mismo objetivo: estimar la probabilidad y el tiempo de supervivencia de los pacientes. Sin embargo, los métodos estadísticos tradicionales se centran en la estimación de curvas de supervivencia para obtener la distribución de los tiempos del evento y las propiedades de los parámetros, mientras que las técnicas de *Machine Learning* combinan los métodos tradicionales con el aprendizaje automático para centrarse en la predicción del evento en un momento concreto de tiempo [2].

A lo largo de este capítulo abordaremos tanto los análisis de supervivencia en general como algunos de los métodos aplicados en la actualidad para llevarlos a cabo.

2.1. Análisis de supervivencia

Siendo $T \geq 0$ una variable aleatoria que representa el tiempo de supervivencia, podemos definir principalmente dos funciones [11]:

Por un lado, la función de supervivencia $S(t)$ representa la probabilidad de que un individuo sobreviva desde el inicio del estudio hasta un tiempo t dado. Por tanto, en el momento inicial, cuando t = 0, el valor de la función es 1, ya que ningún individuo ha sufrido el evento. Una vez vaya pasando el tiempo, el valor irá disminuyendo. Esta función va a describir la supervivencia global de la muestra [12].

$$S(t) = P(T \geq t) \tag{2.1}$$

Por otro lado, la función de riesgo (o *hazard*) $h(t)$ es la probabilidad de que a un individuo le suceda el evento en el tiempo t, es decir, representa la tasa instantánea de sucesos para un individuo que ha sobrevivido hasta el momento t [13]. Esta función nos daría la probabilidad de que, por ejemplo, se desarrolle una recidiva en un paciente operado de cáncer de colon a los 2 años de la operación.

A continuación se muestra la fórmula de la función de riesgo, donde F(t) es la función de distribución acumulada de eventos y f(t) es la función de densidad del evento. Como se puede observar, la expresión matemática se puede reducir a la relación entre la función de densidad del evento y la función de supervivencia.

$$h(t) = \lim_{\Delta t \to \infty} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = \lim_{\Delta t \to \infty} \frac{F(t + \Delta t) - F(t)}{\Delta t * S(t)} = \frac{f(t)}{S(t)} \tag{2.2}$$

Es importante diferenciar las probabilidades que obtenemos a partir de ambas funciones: la de supervivencia se centra en la no ocurrencia del evento y la de riesgo en la ocurrencia. Además, a partir de la función de riesgo podemos calcular alrededor de qué valores se sitúa el pico máximo de eventos [12].

El objetivo último de un análisis de supervivencia es calcular estas dos funciones para poder obtener conclusiones relativas al tiempo de supervivencia y a las diferentes covariables pero, como se comentaba anteriormente, estos análisis presentan una problemática muy concreta: las

censuras. Esto hace que los tiempos de supervivencia sean desconocidos para un subconjunto del grupo de estudio [13].

Las censuras pueden ser de tres tipos: censura derecha, censura izquierda y censura de intervalo.

Hablaremos de *censura por la derecha* si el evento no se produce o se produce después de finalizar el estudio (se localiza a la derecha del tiempo de censura). En este caso, el individuo no ha experimentado (todavía) el evento en el momento de cierre del estudio. Este tipo de censura es el más frecuente en estudios biomédicos. Sería el caso del individuo B en la figura 2.1.

Se denomina *censura por la izquierda* cuando el evento ocurre antes de comenzar la observación. Por ejemplo, si queremos analizar la recidiva de cáncer de pecho a los 3 meses de la operación y cuando las pacientes que acuden a la revisión ya tienen recidiva. Sería el caso del individuo C en la figura 2.1.

La *censura de intervalo* es aquella en la que no se sabe exactamente en qué momento ha ocurrido el evento, solo se maneja un intervalo de tiempo. Esto puede ocurrir cuando la evaluación del seguimiento se realiza de manera periódica y el evento tiene lugar entre dos observaciones.

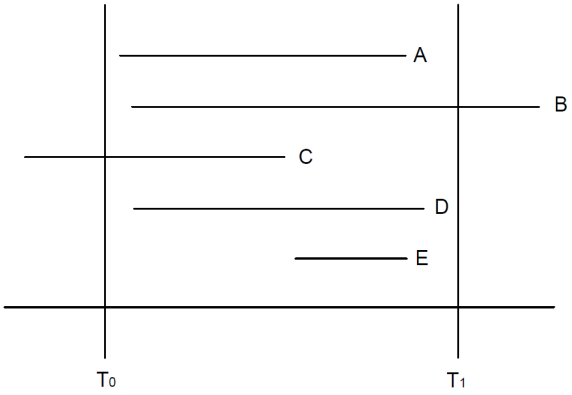

Figura 2.1: Ejemplo de censuras por la derecha y por la izquierda (elaboración propia).

Como veremos a continuación, tanto los métodos tradicionales como los adaptados de Machine Learning están diseñados especialmente para tratar con estas censuras.

2.2. Métodos tradicionales

Los métodos estadísticos tradicionales pueden dividirse en tres categorías [2]:

- Modelos no paramétricos. Los métodos no paramétricos son más eficientes cuando no existe una distribución subyacente para el tiempo del evento o no se cumple el supuesto del riesgo proporcional. En este grupo se encuentran las curvas de Kaplan-Meier.
- Modelos semiparamétricos. El método más utilizado es el modelo de regresión de Cox, que se basa en el supuesto de riesgos proporcionales y emplea la probabilidad parcial para la estimación de los parámetros. Es semiparamétrico porque la distribución subyacente sigue siendo desconocida pero se basa en un modelo de regresión, que es paramétrico.
- Modelos paramétricos. Son los más precisos en la estimación cuando el tiempo hasta el evento sigue una distribución específica. Un ejemplo, en este caso, es la regresión lineal.

Debido a la limitación existente para la extensión del TFM, trataremos en profundidad únicamente los dos métodos más utilizados: las curvas de Kaplan-Meier y el modelo de regresión de Cox.

2.2.1. Curvas de Kaplan-Meier

Esta técnica calcula la probabilidad de supervivencia dividiendo el periodo de seguimiento en segmentos de duración variable, de manera que cada segmento es el intervalo entre dos eventos terminales no simultáneos. En cada segmento se calcula la probabilidad de supervivencia como el producto de dicha probabilidad al inicio y al final del intervalo, siempre que el sujeto esté vivo al inicio. Es decir, es una probabilidad condicionada de muerte en el intervalo [14].

El método de Kaplan-Meier es adecuado para explorar la supervivencia de las poblaciones investigadas y para comparar las diferencias en la supervivencia bruta entre los grupos de exposición. Además, permite la presentación de curvas de supervivencia. Sin embargo, tiene una limitación, y es que no proporciona una estimación del efecto y un intervalo de confianza

para comparar la supervivencia en diferentes grupos de pacientes [15]. Además, no permite modelar cuando la covariable es numérica (por ejemplo, la edad).

En la Figura 2.2 se muestra una curva de Kaplan-Meier de ejemplo llevada a cabo con los datos de *lung* del paquete *survival* [16] de R, donde podemos ver las curvas de supervivencia de hombres (rojo) y mujeres (azul). este resultado es, además, estadísticamente significativo ($p - valor = 0{,}0013$).

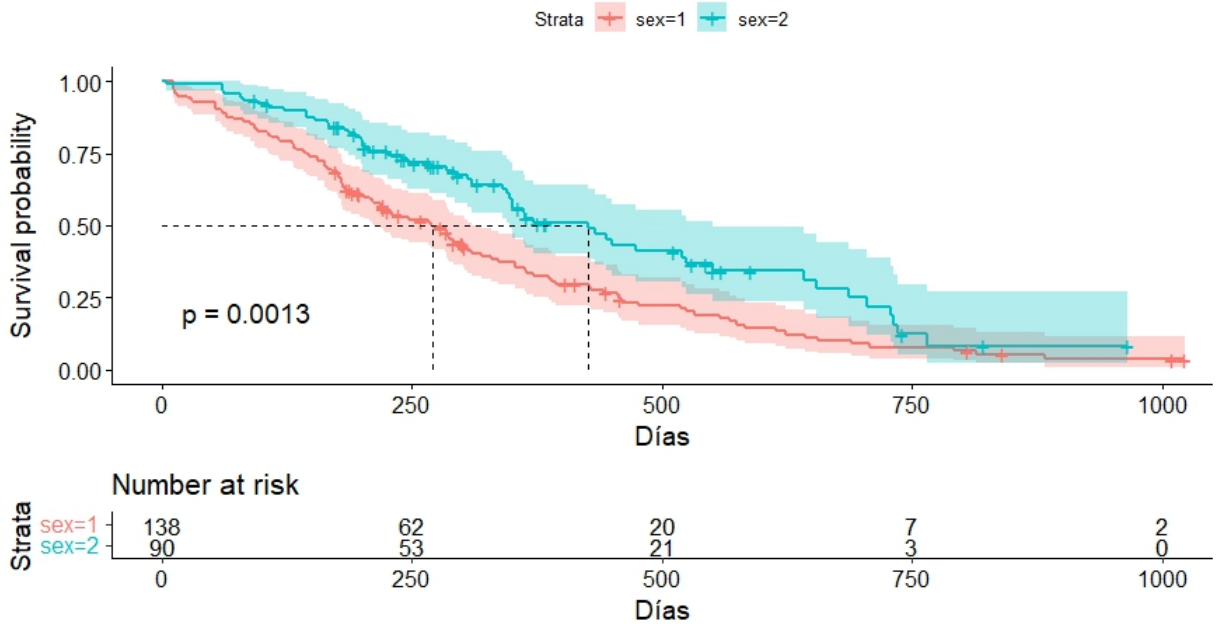

Figura 2.2: Curva de Kaplan-Meier llevada a cabo con los datos de *lung* (elaboración propia).

2.2.2. Regresión de Cox

El análisis de regresión de riesgos proporcionales de Cox es una de las herramientas estadísticas más utilizadas para los análisis de supervivencia. Este modelo define una tasa instantánea de mortalidad (o del evento que se esté analizando) denominada función de riesgo (*hazard function*) y estima la diferencia proporcional en la función de riesgo ya sea entre los grupos de tratamiento o debida a los cambios en las variables de exposición [17], por lo que puede ajustar los efectos de confusión de otras variables, que pueden ser numéricas [15].

Se denomina modelo de riesgos proporcionales porque asume que el exceso de riesgo (la distancia que separa el logaritmo de la tasa de incidencia de ambos grupos) es constante a lo largo del tiempo de seguimiento. Debido a esta asunción, la no proporcionalidad de las curvas de supervivencia puede ser un problema a la hora de interpretar los resultados [18]. Una forma sencilla de comprobar la proporcionalidad en R es utilizando la función *coxph* del paquete *survival* [16]

En el cuadro 2.1 y el gráfico 2.3 se muestra un resumen estadístico de un modelo de Cox y un forest plot de los resultados, ambos obtenidos en Rstudio a partir de los datos de *lung* del paquete *survival* [16]. El modelo se construyó utilizando como covariables la edad y el sexo, que está categorizada en 1 = hombres, 2 = mujeres. La interpretación es similar a la de cualquier modelo de regresión: en este caso vemos que la supervivencia es mayor en las mujeres respecto a lo hombres (categoría de referencia), siendo el resultado estadísticamente significativo ($\beta = -0{,}51$, $e^{-0{,}51} = 0{,}60$; $p - valor = 0{,}002$), mientras que la edad parece no tener influencia en la supervivencia . La conclusión relativa al sexo es la misma que se obtuvo en la Figura 2.2.

	Coeficientes	exp(coeficientes)	LI95*	LS95*	se(coeficientes)	p-valor
sexo	-0.513	0.599	0.431	0.831	0.167	0.002
edad	0.017	1.017	0.999	1.036	0.009	0.065

**LI95/LS95: límites inferior y superior del intervalo al 95 %*

Cuadro 2.1: Modelo de regresión de Cox ajustado por sexo y edad con los datos de*lung* (elaboración propia).

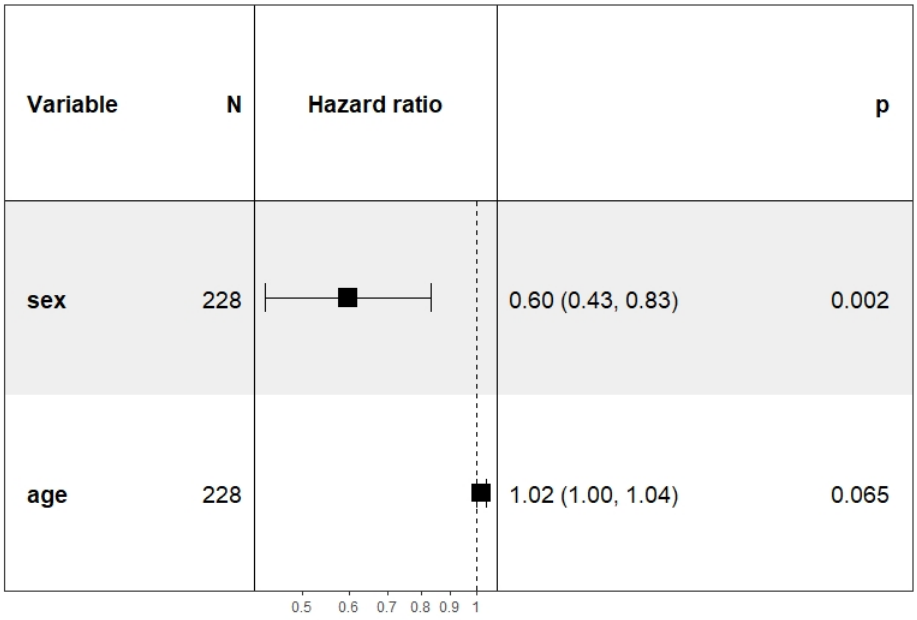

Figura 2.3: Forest plot del modelo de regresión de Cox (elaboración propia).

2.3. Métodos de Machine Learning

Dentro de la rama de Machine Learning nos podemos encontrar múltiples algoritmos, pero en este TFM vamos a tratar algunos de los más utilizados en el campo de la biomedicina [2]:

- Árboles de supervivencia (Survival Trees). Son un tipo de árboles de clasificación y regresión adaptados para manejar censuras. La idea es dividir recursivamente los datos basándose en un criterio de división particular (el evento de interés), manteniendo los objetos similares entre sí en el mismo nodo.

- Métodos bayesianos. Se basan en el teorema de Bayes donde se trabaja con una probabilidad a priori y una probabilidad a posteriori, de manera que se pueden ver los cambios en los valores de probabilidad antes y después de tener en cuenta un determinado acontecimiento. Dentro de este grupo tenemos el algoritmo Naive Bayes. Sin embargo, este algoritmo tiene un inconveniente, y es que supone la independencia entre todas las características, lo cual no siempre se cumple para los análisis de supervivencia.

- Support Vector Machines. Se trata de un método de aprendizaje supervisado que se utiliza principalmente para la clasificación pero también puede adaptarse para resolver problemas de regresión y supervivencia.

2.3.1. Árboles de supervivencia

Los árboles de supervivencia son modelos no paramétricos que permiten detectar automáticamente las interacciones que puedan existir entre las variables sin necesidad de especificarlas de antemano, lo que supone una ventaja respecto a los modelos paramétricos y semi-paramétricos. De manera general, los métodos basados en árboles se basan en ir formando grupos con sujetos similares entre sí en función del resultado de interés [19]. Además, estos árboles orientados a regresión se pueden emplear para predecir el tiempo de supervivencia [20].

Cuando se aplican a supervivencia, los árboles dividen el espacio de las covariables en regiones cada vez más pequeñas, denominadas nodos, que contienen observaciones con resultados de supervivencia homogéneos [21]. Un mismo árbol puede agrupar a los sujetos según su comportamiento de supervivencia en función de sus covariables, permitiendo establecer grupos de pronóstico [19]. La distribución de la supervivencia en las particiones finales (hojas) puede analizarse mediante diversas técnicas estadísticas, como las estimaciones de la curva de Kaplan-Meier [21].

Para la aplicación de este algoritmo deben incorporarse reglas de división y de poda, para seleccionar las particiones que se añaden al árbol y controlar en qué punto deben dejar de añadirse. Las reglas de división en los árboles de supervivencia [22] se basan generalmente en:

- Medidas de distancia entre nodos que buscan maximizar la diferencia entre las observaciones en nodos separados.
- Medidas de pureza que buscan agrupar observaciones similares en un solo nodo.

La mayoría de algoritmos para la construcción de árboles de supervivencia utilizan la poda de complejidad de costes para determinar el tamaño correcto del árbol. Esta metodología consiste en seleccionar un árbol que minimice una combinación ponderada del error total del árbol y

la complejidad de éste, con pesos relativos determinados por validación cruzada, aunque hay otros métodos como el uso del Criterio de Información de Akaike (AIC) o el uso del valor p para detener el árbol cuando las divisiones dejan de ser estadísticamente significativas [21].

2.3.2. Naive Bayes

El algoritmo Naive Bayes describe un método sencillo para aplicar el teorema de Bayes a problemas de clasificación. Es el más común dentro de los métodos de Machine Learning que utilizan métodos bayesianos, especialmente en la clasificación de textos. Este algoritmo asume que todas las características del conjunto de datos analizado son igual de importantes e independientes, algo que no se suele cumplir en el mundo real, y de ahí nace su nombre (naive = ingenuo) [23]. Es uno de los métodos más sencillos para proporcionar modelos predictivos para el diagnóstico, el pronóstico y la planificación del tratamiento a partir de datos retrospectivos [24].

Como este algoritmo asume la independencia condicionada por clase, es decir, considera que los eventos son independientes siempre que estén condicionados por el mismo valor de clase, si alguno de los eventos presenta una probabilidad de 0 enmascararía el resto de la evidencia. Por tanto, se suele aplicar el estimador de Laplace, que añade un valor a cada contador en la tabla de frecuencia evitando así el 0 [23].

El algoritmo sigue el siguiente proceso:

1. Calcula las probabilidades a priori para las clases indicadas.
2. Calcula las probabilidades condicionales para cada atributo de cada clase.
3. Multiplica las probabilidades del punto 1 y 2.
4. Realiza la clasificación en función del resultado del punto 3.

Este algoritmo se ha utilizado con éxito en análisis de supervivencia en contextos clínicos [25] [26], y es por ello que se ha incluido en este trabajo.

2.3.3. Support Vector Machines (SVM)

El objetivo del algoritmo de SVM es crear un límite llamado hiperplano, el cual divide los datos en grupos con valores similares entre sí. Si los grupos se pueden separar completamente por una línea recta, se dice que son linealmente separables, aunque también puede ocurrir que no lo sean. [23]. Este método tiene múltiples aplicaciones, entre las cuales se encuentran la clasificación de los datos de expresión génica obtenidos en microarrays para identificar enfermedades genéticas, la categorización de texto para identificar el idioma de un documento o clasificar documentos por un criterio o la detección de eventos "raros"pero importantes, como fallos en motores de combustión, fallos de seguridad o terremotos. Sin embargo, el uso más generalizado de este algoritmo es para clasificación binaria.

Cuando trabajamos en dos dimensiones, el objetivo del algoritmo es identificar la línea que separa las dos clases. Para ello se busca el *Maximum Margin Hyperplane* (MMH), que es el que crea la mayor separación entre las dos clases. Este MMH mejora la posibilidad de que los grupos permanezcan en el lado correcto de la separación al utilizar nuevos datos.

Cuando las clases son linealmente separables, el MMH está lo más lejos posible de los límites exteriores de los dos grupos de datos. Estos límites exteriores se conocen como *convex hull*.

Por otro lado, si las clases no son linealmente separables se crea una variable *slack*, que permite que algunos puntos caigan en el lado incorrecto de la línea. A estos puntos se les aplica un valor de coste (denominado C) y en este caso el algoritmo, en lugar de buscar el MMH, busca el mínimo coste total.

Otra opción para afrontar el problema de no linealidad es utilizar un *kernel trick*, que transforma los datos de forma que una relación no lineal se asemeje a una lineal.

Las funciones *kernel* más utilizadas son:

- Kernel lineal: se expresa como el producto de las características.
- Kernel polinómico: añade una transformación no lineal simple de los datos.
- Kernel sigmoidal: utiliza un función de activación sigmoidea, de manera parecida a las redes neuronales.

- Kernel gaussiano RBF: es similar a una red neuronal RBF(función de base radial).

Sin embargo, el SVM tiene una limitación, y es que no incluye ningún parámetro para incluir la información incompleta, por lo que se suelen omitir los datos censurados. Este enfoque conlleva una pérdida de información a la hora de realizar un análisis de supervivencia, por lo que se han desarrollado diferentes enfoques para incorporar la censura al SVM [27]:

- Enfoque de clasificación: el modelo aprende a asignar a las muestras con tiempos de supervivencia más cortos un rango inferior considerando todos los pares de muestras posibles en los datos de entrenamiento.
- Enfoque de regresión: el modelo aprende a predecir directamente el tiempo de supervivencia.

Capítulo 3

Metodología

Los datos analizados en este TFM proceden del estudio Lake [9]. Se trata de un estudio prospectivo aleatorizado y multicéntrico llevado a cabo en 19 centros españoles y 1 italiano. Se hizo un seguimiento de 48 semanas a 126 pacientes infectados por VIH-1 de 18 años o más, los cuales fueron repartidos aleatoriamente en una ratio 1:1 en dos grupos con diferente tratamiento: unos recibieron tratamiento oral con efavirenz + abacavir/lamivudina (EFV + Kivexa®) una vez al día y otros con lopinavir/ritonavir dos veces al día además de Kivexa® una vez al día (Kaletra + Kivexa). Se analizó la respuesta viral e inmunológica la semana 4 y, a partir de ahí, cada 3 meses. El éxito del tratamiento se definió para una carga viral plasmática de menos de 50 copias/mL (carga viral indetectable), y el fracaso para una carga mayor de 50 copias/mL en la semana 24 o si un paciente que ya había visto reducida su carga viral volvía a pasar el umbral.

Antes de llevar a cabo el análisis, se exploraron los datos con el objetivo de detectar fallos en la recogida de datos, que se sustituyeron por NAs, y seleccionar las variables con menor número de valores perdidos.

Las variables a analizar son:

- Sexo.
- Edad.
- Tipo de tratamiento.

  - EFV + Kivexa.
  - Kaletra + Kivexa.

- Tiempo de infección (tpo_vih_meses).
- Estado inmunológico del paciente, que se midió al inicio y a las semanas 12, 24, 26 y 48. Este estado se valoró mediante:
  - Carga viral de VIH (CV_número de semana).
  - Contaje absoluto de lifoncitos CD4 (CD4A_número de semana).

Para realizar los análisis de supervivencia, se necesitan principalmente dos variables: la variable tiempo y la variable dicotómica (que denominaremos status) que especifica el éxito o fracaso. Para la primera, se tomó el tiempo máximo de seguimiento, y la variable *status* se estableció de la siguiente manera: 0 = carga viral $\leq$ 50 copias/mL (curado), 1 = carga viral $>$ 50 copias/mL (no curado). Además, se contabilizaron como fracasos (1) aquellos registros que solo contaban con el análisis de la carga viral ($>$ 50 copias/mL) en el momento 0 del estudio.

El análisis de supervivencia se llevó a cabo utilizando los siguientes métodos:

- Árboles de supervivencia. Para analizar los datos con árboles de supervivencia se utilizaron los paquetes *randomForestSRC*, *ggRandomForests*.
- Naive Bayes. Para la aplicación de este algoritmo se utilizó el paquete *e1071*.
- Algoritmo Support Vector Machine (SVM). Se utilizó el paquete *survivalsvm*.

Para la elección de la metodología más adecuada, se realizó una comparación de los resultados utilizando el índice de concordancia de Harrell. Este valor se encuentra entre 0 y 1 y mide lo bien que el predictor clasifica a los individuos en términos de supervivencia [28]. Si el valor está próximo a 0, el rendimiendo del modelo será bajo, si el valor es próximo a 0.5 el rendimiento será medio y los valores cercanos a 1 reflejan un buen rendimiento. Se midió utilizando la función *estC* del paquete *compareC*.

Respecto al diseño del informe, se utilizó el formato *rmdformats::readthedown* del paquete *rmdformats*. Se generó un script en RStudio que permite obtener un Compile Report en formato html de manera automática. El informe contiene tanto el tratamiento previo de los datos como los diferentes resultados, acompañados de comentarios que se actualizan en función de lo que se esté analizando. Este diseño está planteado para que sea lo más automático posible y permita utilizar el código con algunas modificaciones menores. Tanto el código como un informe de prueba se encuentran disponibles en el siguiente repositorio público: https://github.com/paufermart/TFM-Bioinformatics.

Para todo el desarrollo del código se utilizó RStudio con la versión de R 4.2.0 [10]. El TFM se escribió en LaTeX utilizando el editor de texto Texmaker 5.1.2.

Capítulo 4

Resultados y Discusión

4.1. Preparación de los datos

La base de datos original contaba con 169 variables y 116 registros. Se encontró que el 42.21 % eran valores perdidos (Figura 4.1), por lo que, como paso previo a los análisis, se intentó reducir ese porcentaje seleccionando las variables que tuviesen menos del 20 % de NAs.

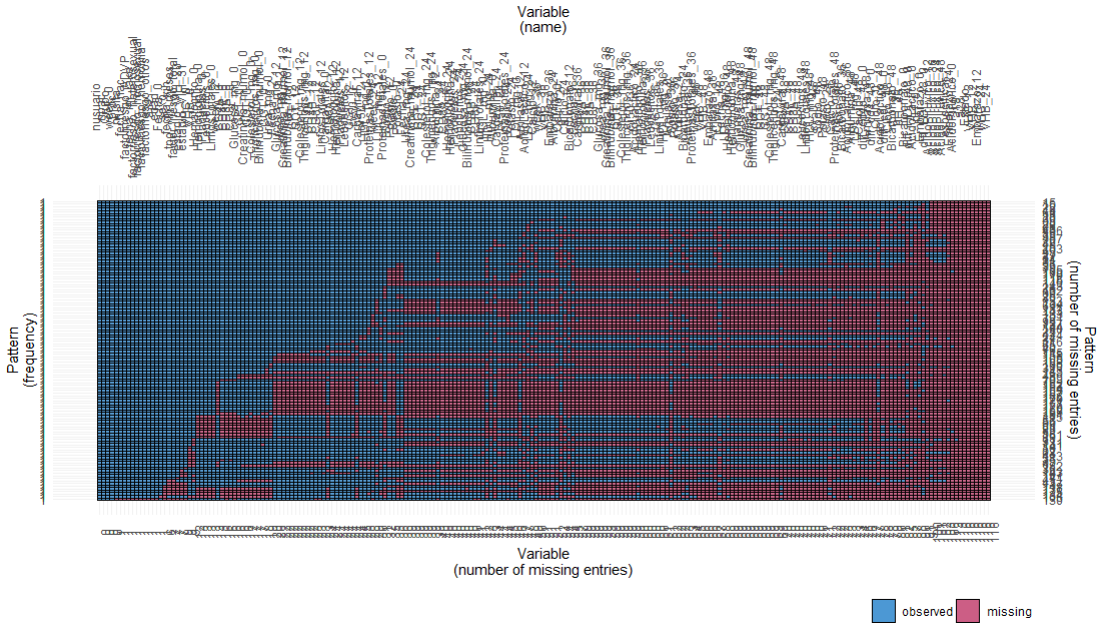

Figura 4.1: Distribución inicial de valores perdidos (NAs).

También se eliminaron aquellas variables sin relevancia clínica, como el nombre del paciente o el número de visita, así como aquellas que se encontraban duplicadas, como la fecha de nacimiento (existe otra variable que recoge la edad) o las variables que se refieren a los factores de riesgo de manera individual, ya que tenemos una variable que los incluye a todos.

Además, se detectaron erratas en el tiempo de infección, donde aparecían registrados tiempos negativos e iguales a 0. Estos valores se sustituyeron por NAs.

Tras este primer cribaje, el número de variables se redujo a 37 y el porcentaje de NAs general al 14.14 % (Figura 4.2).

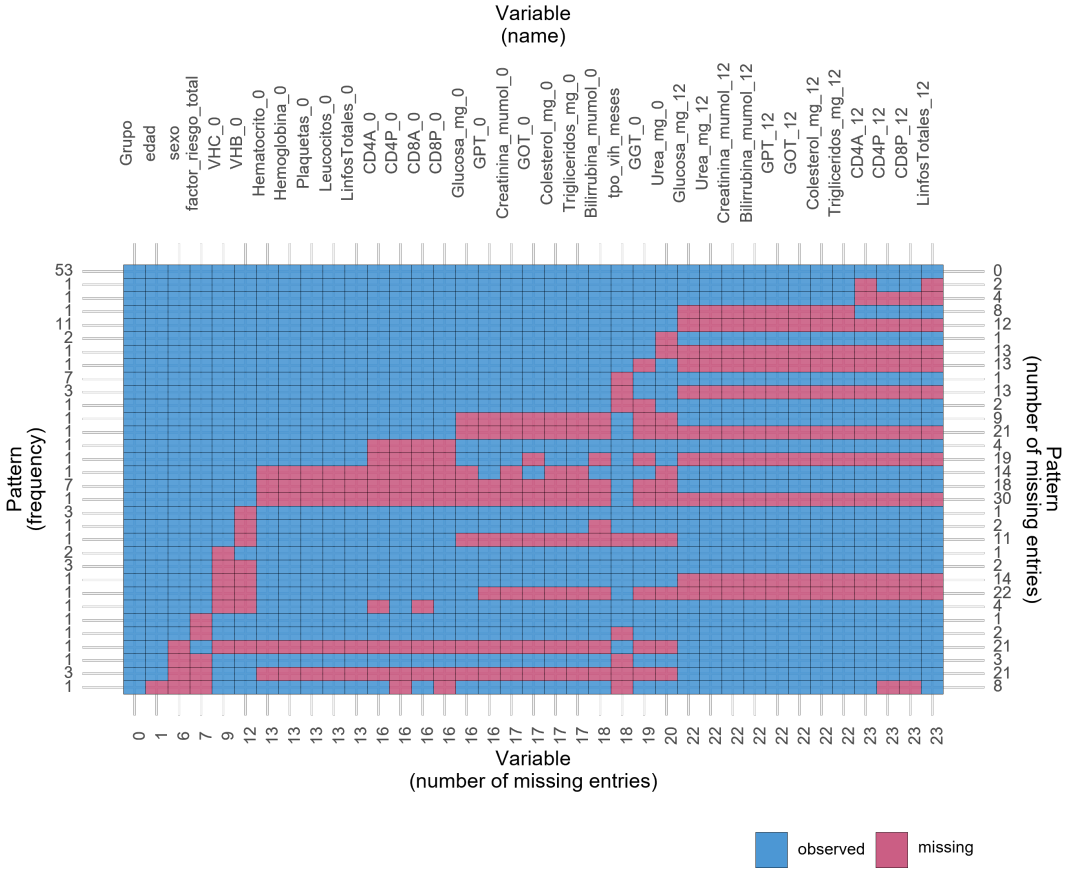

Figura 4.2: Distribución de valores perdidos (NAs) tras la limpieza de la base de datos.

Después se incorporaron las variables *tiempo*, que recoge el tiempo de seguimiento de los pacientes, y *status*, que recoge el éxito/fracaso (0,1). En este punto se eliminaron 4 pacientes para los que el seguimiento fue inexistente.

Como se puede observar en el cuadro 4.1, tenemos 78 éxitos y 34 fracasos. La mayoría de los pacientes (42 %) completaron el seguimiento de 48 semanas. Sin embargo, también tenemos un 11.6 % de pacientes que no superaron la primera medida (0 semanas). Tanto el seguimiento como el desenlace es similar en ambos grupos de tratamiento.

	EFV + Kivexa (N=56)	**Kaletra + Kivexa (N=56)**	**Total (N=112)**
Tiempo			
0	8 (14.3 %)	5 (8.9 %)	13 (11.6 %)
12	11 (19.6 %)	9 (16.1 %)	20 (17.9 %)
24	9 (16.1 %)	12 (21.4 %)	21 (18.8 %)
36	4 (7.1 %)	7 (12.5 %)	11 (9.8 %)
48	24 (42.9 %)	23 (41.1 %)	47 (42.0 %)
status			
0	41 (73.2 %)	37 (66.1 %)	78 (69.6 %)
1	15 (26.8 %)	19 (33.9 %)	34 (30.4 %)

Cuadro 4.1: Resumen de frecuencias de las variables de supervivencia

4.2. Exploración inicial de los datos

De los 126 participantes en el estudio original, se ha podido acceder a los datos de 112, 56 en cada grupo de tratamiento. Para el análisis descriptivo se tuvieron en cuenta las variables sexo, edad, tiempo de infección y estado inmunológico del paciente.

Se analizaron 91 hombres y 15 mujeres, con una edad media general de 38.1 años (Cuadro 4.2). El tiempo de infección por VIH varía entre 0.233 y 285 meses, con una mediana de 6.05.

Respecto a la carga viral, podemos observar cómo va disminuyendo a lo largo del tratamiento con antirretrovirales, como era de esperar (Figura 4.3).

	EFV + Kivexa (N=56)	**Kaletra + Kivexa (N=56)**	**Total (N=112)**
Sexo			
Hombre	46 (82.1 %)	45 (80.4 %)	91 (81.3 %)
Mujer	8 (14.3 %)	7 (12.5 %)	15 (13.4 %)
NAs	2 (3.6 %)	4 (7.1 %)	6 (5.4 %)
Edad			
Media (SD)	38.5 (8.55)	37.6 (8.13)	38.1 (8.32)
Mediana [Min, Max]	38.0 [21.0, 59.0]	37.0 [20.0,58.0]	37.0 [20.0, 59.0]
NAs	1 (1.8 %)	1 (1.8 %)	2 (1.8 %)
Tiempo de infección			
Media (SD)	36.5 (62.1)	20.5 (44.3)	28.0 (53.7)
Mediana [Min, Max]	13.9 [0.233, 285]	4.32 [0.467, 214]	6.05 [0.233, 285]
NAs	12 (21.4 %)	6 (10.7 %)	18 (16.1 %)

Cuadro 4.2: Estadísticos descriptivos de sexo, edad y tiempo de infección por VIH.

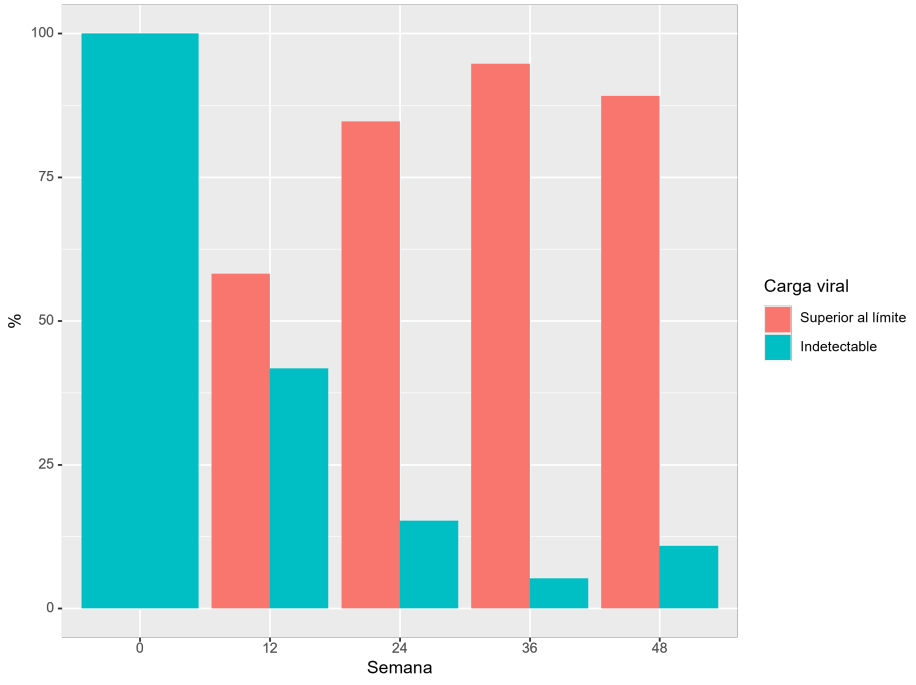

Figura 4.3: % de individuos en cada categoría de carga viral a lo largo del estudio.

Además, como parte de la exploración inicial se ha realizado un análisis de supervivencia tradicional, mediante la elaboración de una curva de Kaplan-Meier.

Al realizar la curva de Kaplan-Meier con todas las covariables que no fueron descartadas en el preprocesamiento de datos, podemos observar que la probabilidad de curación disminuye con el tiempo, situándose alrededor del 75 % al final del estudio (Figura 4.4). El descenso que tenemos en el momento 0 hace referencia a aquellos individuos cuyo seguimiento se perdió tras la primera medida.

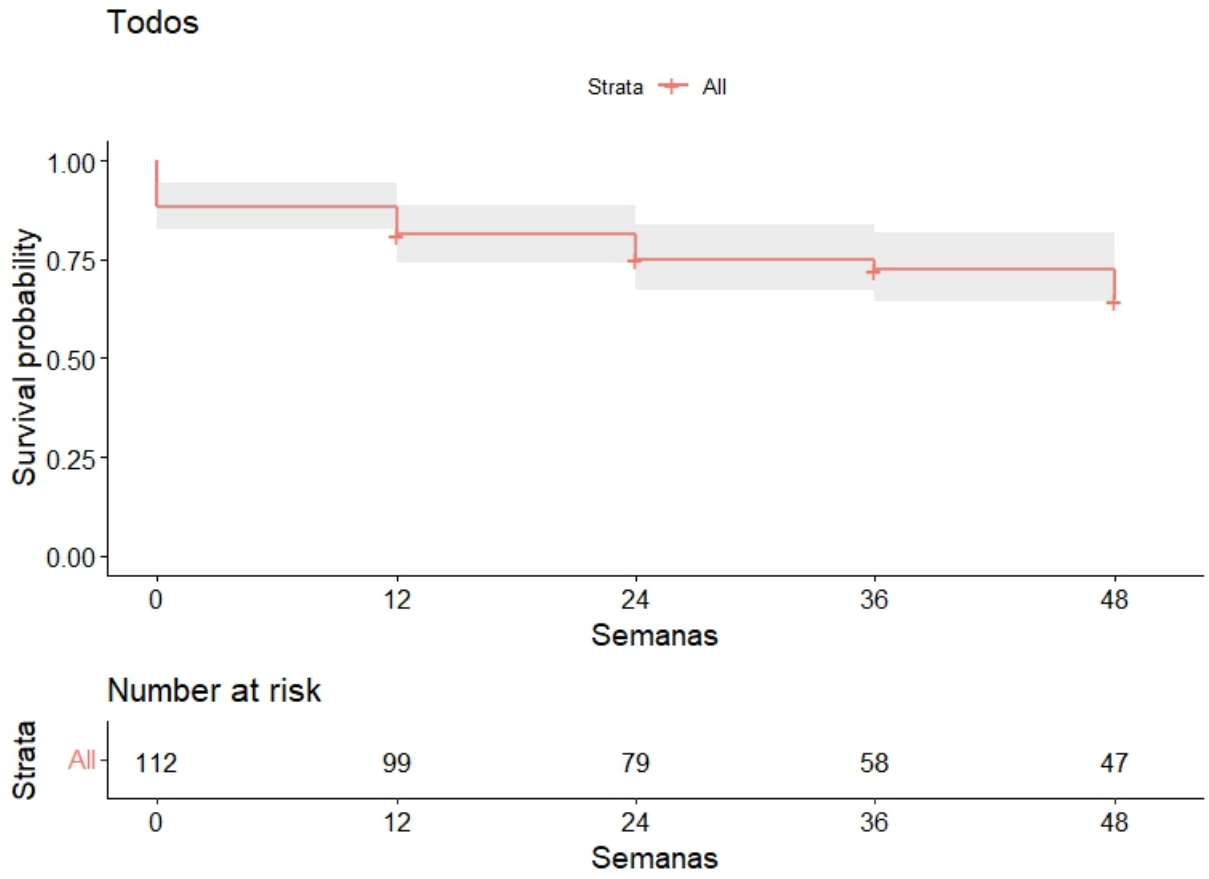

Figura 4.4: Curva de KM para el modelo general

Si agrupamos por tratamientos, vemos que no hay diferencias estadísticamente significativas entre ellos (p-valor = 0.54), aunque la probabilidad de curación del grupo con EFV + Kivexa es ligeramente más elevada al final del estudio (Figura 4.5).

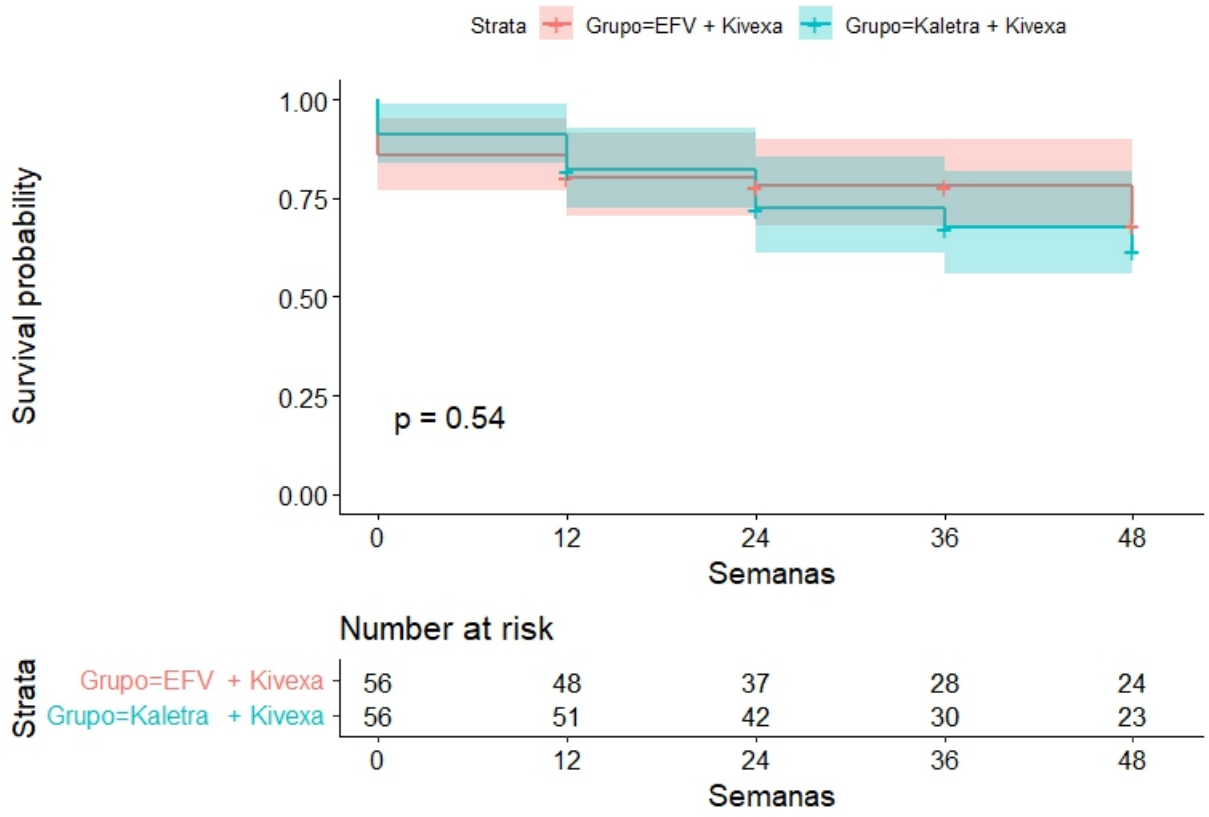

Figura 4.5: Curva de KM para el modelo por grupos

4.3. Aplicación de los algoritmos

Una vez completada la exploración inicial de los datos, se prepararon para la aplicación de los diferentes algoritmos.

Se dividieron en dos subconjuntos de datos: uno de entrenamiento, con el 67 % de los datos, y otro de prueba, con el 37 % restante. Además, la categoría de tiempo 0 se transformó en 0.0 para evitar una mala interpretación de dicha categoría, ya que en algunos casos los algoritmos pueden entender el 0 como ausencia de tiempo.

Cualquier otra transformación llevada a cabo para el análisis, se comentará en el apartado del algoritmo donde se haya llevado a cabo.

4.3.1. Árboles de supervivencia

Para la generación de los árboles de supervivencia, se utilizó el paquete *randomForestSRC*. Entre los motivos por los que se seleccionó este algoritmo y, en especial, este paquete de R, es la posibilidad de llevar a cabo una imputación de los NAs que, dada la base de datos con la que se está trabajando en este estudio, resulta muy conveniente.

Se comprobó que todas las variables estaban en formato numérico, ya que si no el algoritmo da problemas. Se midieron 37 covariables y se indicó que se llevase a cabo una imputación de los valores perdidos, que es una opción incluida en la función *rfscr*.

Se optó por hacer dos modelos:

- Por un lado, se diseñó un bosque con 100 árboles a partir del conjunto de datos de entrenamiento, donde se obtuvo un error de 0.6032 . Al realizar las predicciones, el error disminuyó a 0.3605, obteniéndose una C de 0.3558.
- Por otro lado, se diseñó un bosque con 300 árboles, donde se obtuvo un error de 0.7170. Al realizar las predicciones, el error disminuyó a 0.3533, obteniéndose una C de 0.3483.

Estos resultados son bastante similares, aunque quizás algo mejores en el modelo con 100 árboles, por lo que habría que quedarse con este último (Cuadro 4.3). Sin embargo, los valores de C son bastante pequeños y empeoran al aumentar el número de árboles.

Número de árboles	Error Entrenamiento	Error Test	C de Harrell
100	0.6032	0.3605	0.3558
300	0.7170	0.3533	0.3483

Cuadro 4.3: Resultados para los árboles de supervivencia

Además, se analizó qué covariables tenían mayor influencia sobre las variables dependientes en los modelos generados (*tiempo* y *status*). Como se puede observar en la Figura 4.6, vemos que la mayoría de variables tienden a reducir el error, mientras que hay 12 aumentan dicho error. Habría que cribar estas variables y ver por qué influyen negativamente sobre el error.

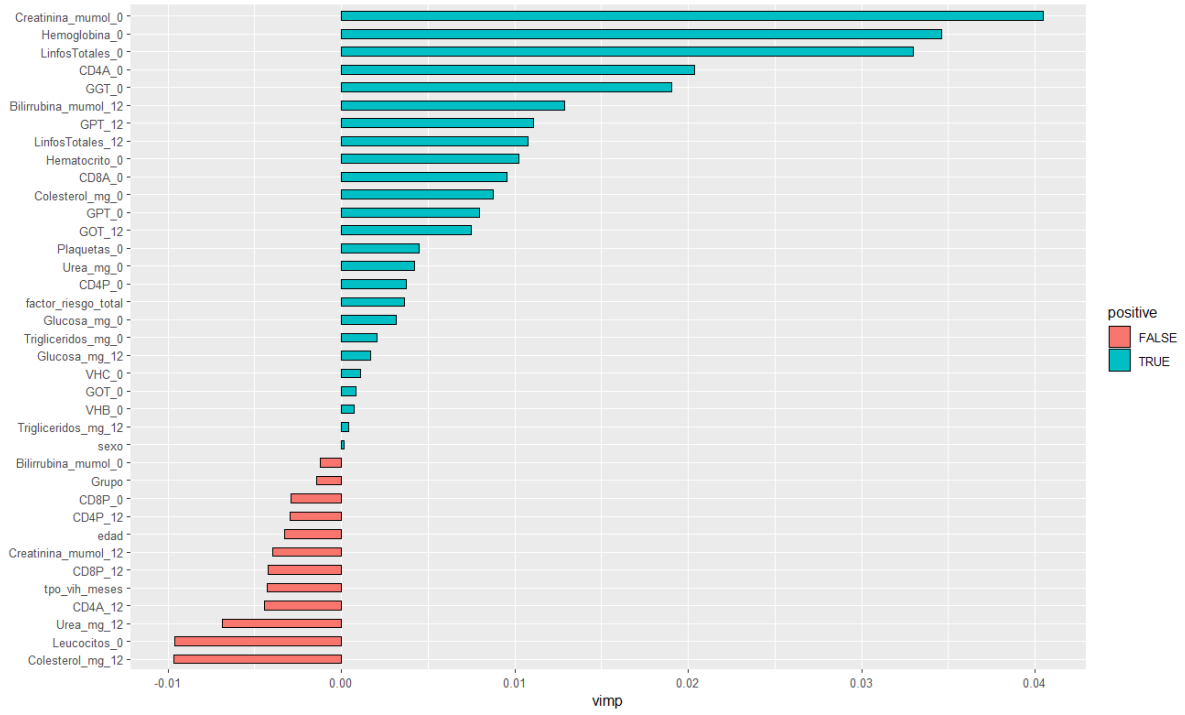

Figura 4.6: Error en la predicción con 300 árboles

4.3.2. Naive Bayes

En el caso de este algoritmo, los datos de train y test se pudieron aplicar sin ninguna modificación extra, por lo que se tuvieron en cuenta los valores perdidos de las covariables.

Se entrenó el modelo sobre 75 individuos y luego se hicieron las predicciones utilizando 37 individuos. En ambos casos, se analizaron 37 covariables y se obtuvo el valor de C mediante la función *est*, del paquete *compareC*.

Se diseñaron dos modelos: uno sin la corrección de Laplace y otro con ella. En ambos casos, el valor para la C es el mismo: 0.6067 (Cuadro 4.4), por lo que la corrección de Laplace parece no ser necesaria.

Este resultado mejora el obtenido con los árboles de supervivencia.

Modelo	C de Harrell
Laplace = 0	0.6067
Laplace = 1	0.6067

Cuadro 4.4: C de Harris para los modelos de Naive Bayes

4.3.3. Support Vector Machine

Para poder llevar a cabo este algoritmo, se eliminaron los valores perdidos de los datos, puesto que con ellos el código daba error. Por tanto, se construyó el algoritmo en base a 52 observaciones, en lugar de las 112 utilizadas en el resto. Esto supone una disminución importante, por lo que los resultados obtenidos pueden estar sesgados.

Se diseñó un modelo de regresión y se utilizaron tres tipos de kernel:

- Lineal.
- Radial (RBF).
- Aditivo.

El modelo con mayor C fue el modelo lineal (Cuadro 4.5), con un valor de 0.6176, por lo que es el que mejor predice los resultados, tanto en este grupo, como en general.

Si bien este valor es el más alto de todos los algoritmos planteados, también es donde menos individuos han podido ser analizados y, por tanto, debemos ser cuidadosos con la interpretación.

También se ha obtenido el peor resultado de todo el estudio con este algoritmo, concretamente con el kernel aditivo (C = 0.1765).

Tipo de kernel	C de Harris
Lineal	0.6176
Radial (RBF)	0.5000
Aditivo	0.1765

Cuadro 4.5: C de Harris para los modelos de SVM

4.3.4. Elección del mejor modelo

En base a los resultados obtenidos con los diferentes modelos, el mejor índice C se obtuvo con el algoritmo Support Vector Machine con kernel lineal (C = 0.6176), por lo que parecer ser que esta metodología es la más adecuada para tratar datos de este tipo. Sin embargo, no se pudo analizar el mismo número de datos que en el resto de algoritmos, por lo que deberían hacerse más pruebas antes de sacar una conclusión definitiva. Sin embargo, los otros dos kernel que se probaron, el RBF y el aditivo, arrojaron malos resultados, especialmente el último, cuyo valor de C fue el más bajo de todo el estudio.

Por otro lado, los peores resultados los arrojaron los árboles de supervivencia, por lo que en principio habría que descartarlos como opción. Además, a mayor número de árboles, el error aumenta, lo cual resulta llamativo. También se debe tener en cuenta que en este algoritmo se utilizaron métodos de imputación de valores perdidos que, en una base de datos con un % de NAs tan elevado, puede marcar la diferencia. Tal vez, en otro contexto, el resultado sería mejor.

En cuanto al algoritmo Naive Bayes, ronda una C de 0.6, a medio camino entre los otros dos. Este algoritmo no siempre es la mejor opción, ya que asume la independencia de las variables, algo que no siempre se cumple, pero en este caso esta propiedad no parece haberle penalizado mucho.

En base a estos resultados, para los datos del estudio Lake [9] deberíamos quedarnos con el modelo de predicción de SVM con kernel lineal.

4.4. Generación del informe automático

Todo el código R utilizado en la generación del informe se implementó en RStudio y se organizó con el objetivo de generar un informe lo más automatizado posible.

Se generó una cabecera, donde cualquier usuario puede indicar su nombre y el título que quiere incluir en el informe, así como una función para que la fecha se actualice automáticamene con la compilación.

Además, se implementó la opción de que el código sea visible o no, en función de lo que se prefiera, y se generó un índice interactivo.

El informe se distribuye de la siguiente manera:

- Limpieza de datos. El primer paso consta de una observación general de los datos y de una limpieza de valores perdidos.
- Análisis descriptivo. En este apartado se transforman los datos para describir a los individuos de la muestra.
    - Transformación de los resultados
- Algoritmos de Machine Learning. En este punto, se preparan los datos para la aplicación de los algoritmos y se implementan los diferentes modelos.
    - Preparación de los datos
    - Árbol de supervivencia
    - Naive Bayes
    - Support Vector Machine
- Resumen de los resultados. Finalmente, se muestran los resultados de los algoritmos y se selecciona el que presenta mayor valor C.

Todo el informe va acompañado de explicaciones. También se incluyeron en el texto los valores más relevantes, que se actualizan automáticamente en función de los datos que se utilicen. El formato de salida es HTML.

Capítulo 5

Conclusiones

5.1. Conclusiones

A lo largo de este trabajo se han explorado varias opciones para analizar en términos de supervivencia unos datos de pacientes de VIH sometidos a tratamiento antirretroviral.

Se obtuvo que, de las 169 variables y 116 registros iniciales, el 42.21 % eran valores perdidos, por lo que hubo que realizar una limpieza de los datos. Tras el procesado inicial, se pudieron analizar 112 individuos, 78 de los cuales tuvieron éxito con el tratamiento y 34 no. El seguimiento de los pacientes fue variable, llegando la mayoría a completar el estudio.

De los pacientes con sexo registrado, 91 eran hombres y 15 mujeres, con una edad media de 38.1 años. Los grupos de tratamiento estaban balanceados, encontrándose 56 individuos en cada uno de ellos.

Para el análisis de supervivencia, se estudiaron en profundidad los análisis clásicos (las curvas de Kaplan Meier y el modelo de regresión de Cox), así como varios algoritmos de Machine Learning orientados a este tipo de problema: los árboles de supervivencia, el algoritmo Naive Bayes y las Máquinas de Soporte Vectorial (SVM).

Si bien tenía ciertas nociones relativas a estos algoritmos gracias a la asignatura de Machine Learning del Máster, no había llegado a profundizar tanto en su aplicación. Además, nunca los había aplicado a supervivencia, lo que ha supuesto un reto.

Los resultados de cada algoritmo fueron comparados mediante el índice C de Harrell. La mayor puntuación se obtuvo con el SVM con kernel linal, mientras que los peores se obtuvieron con los árboles de supervivencia, siendo peor el modelo con 300 árboles frente al modelo con 100. Respecto al algoritmo Naive Bayes, ambos modelos arrojaron valores más próximos a los obtenidos en el SVM lineal, lo que invita a seguir explorándolos en busca de una mejora en la implementación.

Respecto a la consecución de los objetivos planteados, se ha conseguido responder a todas las preguntas, aunque se han llevado a cabo cambios en el proceso. Una vez implementados los algoritmos, que inicialmente iban a compararse mediante sensibilidad, especificidad y AUC, se observó que iba a ser más práctico llevar la comparación con el índice C de Harrell. También se incorporaron otros cambios necesarios a la hora de aplicar los algoritmos y en los que no se había reparado anteriormente, como generar más de un conjunto de árboles de supervivencia o realizar una limpieza de variables con exceso de valores perdidos.

Tanto la plantilla de R como el informe generado de prueba en formato html se encuentran en el siguiente repositorio: https://github.com/paufermart/TFM-Bioinformatics.

Para finalizar, me gustaría añadir que, a la vista de los resultados, queda patente la utilidad de los algoritmos de Machine Learning para analizar y modelar grandes volúmenes de datos, y la relevancia clínica que esto puede tener en la predicción de tiempos de supervivencia.

5.2. Trabajo futuro

Como trabajo futuro sería interesante explorar la aplicación de una metodología de imputación de valores perdidos, ya que la base de datos utilizada tenía una gran cantidad de ellos que dificultaron el análisis, algo que suele ocurrir cuando hablamos de estudios enfocados en la salud pública. Los estimadores obtenidos a partir de bases de datos con gran cantidad de valores perdidos pueden ser ineficientes y estar sesgados, por lo que deben ser abordados para evitar la pérdida de casos, ya que en el análisis solo se tendrán en cuenta los individuos con información completa [29]. Entre los mecanismos para abordar este problema se encuentran MCAR (*Missing Completely At Random*), que asume que la ausencia de datos es completa-

mente al azar y MAR (*Missing At Random*, donde se asume una dependencia respecto al resto de variables.

Además, otra posible línea de trabajo sería tanto realizar los análisis con otros algoritmos, por ejemplo mediante redes neuronales, como repetir los análisis llevados a cabo en este estudio utilizando otros paquetes diferentes, comparando su rendimiento con los resultados obtenidos en este trabajo. En este caso se optó por trabajar con árboles de supervivencia con los paquetes *randomForestSRC* y *ggRandomForests*, con el algoritmo Naive Bayes utilizando el paquete *e1071* y con diferentes variantes del algoritmo Support Vector Machine mediante el paquete *survivalsvm*, mientras que para el índice C se utilizó el paquete *compareC*. Estas elecciones estuvieron determinadas por el tipo de datos con los que se iba a trabajar y por la capacidad de comparación que ofrecían, aún siendo metodologías tan diferentes en su base. Sin embargo, otras alternativas pueden ser los paquetes *rpart* y *randomForest* para los árboles o *naivebayes* para el algoritmo del mismo nombre. También el índice C se puede obtener a partir de otros paquetes, como son *dynpred* con la función *cindex* o *Hmisc* con la función *rcorrcens*.

Asimismo, como ya se ha comentado a lo largo del TFM, queda pendiente ahondar en el uso del algoritmo Naive Bayes, pues se trata de un algoritmo de fácil aplicación y cuyos resultados son prometedores.

Capítulo 6

Glosario

C-Index: índice C de Harrell

KM: Kaplan-Meier

NB: Naive Bayes

SVM: Support Vector Machine

Bibliografía

[1] F. E. Harrell, "Introduction to Survival Analysis," in *Regression Modeling Strategies*, ch. 17, pp. 399–422, Springer, 2015.

[2] P. Wang, Y. Li, and C. K. Reddy, "Machine learning for survival analysis: A survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–39, 2019.

[3] T. Ahmad, A. Munir, S. H. Bhatti, M. Aftab, and M. A. Raza, "Survival analysis of heart failure patients: A case study," *PLoS ONE*, vol. 12, no. 7, pp. 1–8, 2017.

[4] J. Y. Hsu, J. A. Roy, D. Xie, W. Yang, H. Shou, A. H. Anderson, J. R. Landis, C. Jepson, M. Wolf, T. Isakova, M. Rahman, and H. I. Feldman, "Statistical Methods for Cohort Studies of CKD: Survival Analysis in the Setting of Competing Risks," *Clinical Journal of the American Society of Nephrology*, vol. 12, pp. 1181–1189, jul 2017.

[5] B. J. Solomon, D. W. Kim, Y. L. Wu, K. Nakagawa, T. Mekhail, E. Felip, F. Cappuzzo, J. Paolini, T. Usari, Y. Tang, K. D. Wilner, F. Blackhall, and T. S. Mok, "Final overall survival analysis from a study comparing first-line crizotinib versus chemotherapy in alk-mutation-positive non–small-cell lung cancer," *Journal of Clinical Oncology*, vol. 36, no. 22, pp. 2251–2258, 2018.

[6] K. N. Chi, S. Chowdhury, A. Bjartell, B. H. Chung, A. J. Pereira de Santana Gomes, R. Given, A. Juárez, A. S. Merseburger, M. Özgüroğlu, H. Uemura, D. Ye, S. Brookman-May, S. D. Mundle, S. A. McCarthy, J. S. Larsen, W. Sun, K. B. Bevans, K. Zhang, N. Bandyopadhyay, and N. Agarwal, "Apalutamide in Patients With Metastatic Castration-Sensitive

Prostate Cancer: Final Survival Analysis of the Randomized, Double-Blind, Phase III TITAN Study," *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, vol. 39, no. 20, pp. 2294–2303, 2021.

[7] M. Nemati, J. Ansary, and N. Nemati, "Machine-Learning Approaches in COVID-19 Survival Analysis and Discharge-Time Likelihood Prediction Using Clinical Data," *Patterns*, vol. 1, no. 5, p. 100074, 2020.

[8] R. Sonabend, F. J. Király, A. Bender, B. Bischl, and M. Lang, "mlr3proba: an R package for machine learning in survival analysis," *Bioinformatics*, vol. 37, pp. 2789–2791, sep 2021.

[9] P. Echeverria, E. Negredo, G. Carosi, J. Gálvez, J. Gómez, A. Ocampo, J. Portilla, A. Prieto, J. López, R. Rubio, *et al.*, "Similar antiviral efficacy and tolerability between efavirenz and lopinavir/ritonavir, administered with abacavir/lamivudine (kivexa®), in antiretroviral-naïve patients: A 48-week, multicentre, randomized study (lake study)," *Antiviral research*, vol. 85, no. 2, pp. 403–408, 2010.

[10] RStudio Team, *RStudio: Integrated Development Environment for R*. RStudio, PBC, Boston, MA, 2022.

[11] C. Kartsonaki, "Survival analysis," *Diagnostic Histopathology*, vol. 22, pp. 263–270, jul 2016.

[12] P. Rebasa, "Conceptos básicos del análisis de supervivencia," *Cirugía Española*, vol. 78, pp. 222–230, oct 2005.

[13] 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, pp. 232–238, jul 2003.

[14] N. Benitez-Parejo, M. Rodriguez del Aguila, and S. Perez-Vicente, "Survival analysis and Cox regression," *Allergologia et Immunopathologia*, vol. 39, pp. 362–373, nov 2011.

[15] V. S. Stel, F. W. Dekker, G. Tripepi, C. Zoccali, and K. J. Jager, "Survival Analysis II: Cox Regression," *Nephron Clinical Practice*, vol. 119, no. 3, pp. c255–c260, 2011.

[16] T. M. Therneau, *A Package for Survival Analysis in R*, 2022. R package version 3.3-1.

[17] D. McGregor, J. Palarea-Albaladejo, P. Dall, K. Hron, and S. Chastin, "Cox regression survival analysis with compositional covariates: Application to modelling mortality risk from 24-h physical activity patterns," *Statistical Methods in Medical Research*, vol. 29, pp. 1447–1465, may 2020.

[18] P. C. van Dijk, K. J. Jager, A. H. Zwinderman, C. Zoccali, and F. W. Dekker, "The analysis of survival data in nephrology: basic concepts and methods of Cox regression," *Kidney International*, vol. 74, pp. 705–709, sep 2008.

[19] I. Bou-Hamad, D. Larocque, and H. Ben-Ameur, "A review of survival trees," *Statistics Surveys*, vol. 5, pp. 44–71, jan 2011.

[20] L. Stipanek, F. Habarta, I. Mala, L. Marek, and F. Pazdirek, "A Machine-learning Approach to Survival Time-event Predicting: Initial Analyses using Stomach Cancer Data," in *2020 International Conference on e-Health and Bioengineering (EHB)*, pp. 1–4, IEEE, oct 2020.

[21] D. Bertsimas, J. Dunn, E. Gibson, and A. Orfanoudaki, "Optimal survival trees," *Machine Learning*, apr 2022.

[22] Y. Zhou and J. J. McArdle, "Rationale and Applications of Survival Tree and Survival Ensemble Methods," *Psychometrika*, vol. 80, pp. 811–833, sep 2015.

[23] B. Lantz, *Machine Learning with R: Expert techniques for predictive modeling to solve all your data analysis problems*. Packt Publishing Ltd, 2015.

[24] W. Kim, K. S. Kim, and R. W. Park, "Nomogram of Naive Bayesian Model for Recurrence Prediction of Breast Cancer," *Healthcare Informatics Research*, vol. 22, no. 2, p. 89, 2016.

[25] J. Wolfson, S. Bandyopadhyay, M. Elidrisi, G. Vazquez-Benitez, D. M. Vock, D. Musgrove, G. Adomavicius, P. E. Johnson, and P. J. O'Connor, "A Naive Bayes machine learning approach to risk prediction using censored, time-to-event data," *Statistics in Medicine*, vol. 34, pp. 2941–2957, sep 2015.

[26] E. van der Heide, R. Veerkamp, M. van Pelt, C. Kamphuis, I. Athanasiadis, and B. Ducro, "Comparing regression, naive Bayes, and random forest methods in the prediction of individual survival to second lactation in Holstein cattle," *Journal of Dairy Science*, vol. 102, pp. 9409–9421, oct 2019.

[27] J. Fouodo, Cesaire, K., R. Konig, Inke, C. Weihs, A. Ziegler, and N. Wright, Marvin, "Support Vector Machines for Survival Analysis with R," *The R Journal*, vol. 10, no. 1, p. 412, 2018.

[28] H. Ishwaran, M. S. Lauer, E. H. Blackstone, M. Lu, and U. B. Kogalur, "randomForestSRC: random survival forests vignette," 2021.

[29] G. Hernández, D. Moriña, and A. Navarro, "Imputación de valores ausentes en salud pública: conceptos generales y aplicación en variables dicotómicas," *Gaceta Sanitaria*, vol. 31, pp. 342–345, jul 2017.