Trabajo Fin de Máster

Desarrollo de modelos predictivos y control de calidad en la industria de toldos

Luis Alamancos Rodríguez

Máster en Técnicas Estadísticas

Curso 2025-2026

Propuesta de Trabajo Fin de Máster

Título en galego: Desenvolvemento de modelos preditivos e control de calidade na industria de toldos
Título en español: Desarrollo de modelos predictivos y control de calidad en la industria de toldos
English title: Predictive modelling development and quality control in the awning industry
Modalidad: Modalidad B
Autor/a: Luis Alamancos Rodríguez, Universidade da Coruña
Director/a: Salvador Naya Fernández, Universidade da Coruña; Javier Tarrío Saavedra, Universidade da Coruña
Tutor/a: Dolores Balboa Illodo, Toldos Gómez S.L.
Breve resumen del trabajo: El Trabajo de Fin de Máster se centrará en el análisis y modelización estadística de procesos operativos con componente temporal, empleando datos reales procedentes de la actividad de la empresa Toldos Gómez S.L. Desde una perspectiva metodológica, se explorarán técnicas de modelización predictiva basadas en Aprendizaje Automático con el objetivo de estimar los tiempos de ocurrencia de eventos relevantes (por ejemplo, la facturación de pedidos) y comprender la dinámica subyacente de dichos procesos. El desarrollo del presente TFM incluirá la exploración y preparación de los datos, la formulación y ajuste de modelos, la validación de su capacidad predictiva y la interpretación de los resultados, poniendo el foco en su aplicabilidad para la mejora de la planificación, la eficiencia y el control de los procesos operativos en el contexto empresarial.
Recomendaciones:

Otras observaciones:

Don/doña Salvador Naya Fernández, [categoría académica] de la Universidade da Coruña, don/doña Javier Tarrío Saavedra, [categoría académica] de la Universidade da Coruña, y don/doña Dolores Balboa Illodo, [cargo] de Toldos Gómez S.L., informan que el Trabajo Fin de Máster titulado

Desarrollo de modelos predictivos y control de calidad en la industria de toldos

fue realizado bajo su dirección por don/doña Luis Alamancos Rodríguez para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal. Además, don/doña Salvador Naya Fernández, don/doña Javier Tarrío Saavedra y don/doña Luis Alamancos Rodríguez

√ sí ☐ no

autorizan a la publicación de la memoria en el repositorio de acceso público asociado al Máster en Técnicas Estadísticas.

En A Coruña, a 17 de julio de 2026.

El/la director/a:
Don/doña Salvador Naya Fernández

Firmado por NAYA FERNANDEZ SALVADOR - ***5373** el día 17/07/2026 con un certificado

El/la director/a:
Don/doña Javier Tarrío Saavedra

TARRIO SAAVEDRA JAVIER - DNI 53164914T

Firmado digitalmente por TARRIO SAAVEDRA JAVIER - DNI 53164914T
Fecha: 2026.07.17 22:05:00 +02'00'

El/la tutor/a:
Don/doña Dolores Balboa Illodo

El/la autor/a:
Don/doña Luis Alamancos Rodríguez

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, Disposición 2978 del BOE núm. 48 de 2022), **el/la autor/a declara** que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas, etc.).
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, una sección, una demostración, etc., constituyen una adaptación casi literal de alguna fuente existente.

Y acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Agradecimientos

Quiero agradecer, en primer lugar, a mis directores la paciencia y la disponibilidad mostradas a lo largo de todo el trabajo, así como la facilidad con la que han resuelto cada inconveniente surgido.

Gracias también a mi tutora en la empresa, y a Toldos Gómez S.L. en su conjunto, por darme la oportunidad de realizar este trabajo con ellos y por el buen trato recibido durante todo el proceso.

Y gracias a las personas que han estado presentes durante este tiempo, y también a las que ya no lo están, porque de una forma u otra han formado parte.

Índice general

Resumen

Resumen en español

Este trabajo aborda un problema de Toldos Gómez S.L., empresa gallega dedicada a la fabricación e instalación de toldos: saber, para cada pedido que entra, cuánto va a tardar en convertirse en una factura. La variable de interés es el número de días entre el registro del pedido en el ERP de la empresa y la emisión de la factura correspondiente. Al extraer los datos, una parte de los pedidos todavía no se ha facturado: de ellos solo se sabe que llevan, al menos, un determinado número de días en curso. Tratar esa información parcial sin descartarla y sin dar por conocido un desenlace que todavía no se ha producido es el motivo por el que se recurre al análisis de supervivencia en lugar de a una regresión convencional. A partir de los datos del ERP (sistema de planificación de recursos empresariales) se construye una cadena de preparación, modelado y evaluación. Se ajustan tres familias de modelos —el modelo de riesgos proporcionales de Cox, el Random Survival Forest y el Gradient Boosting con pérdida de Cox— y se compara su capacidad para predecir la probabilidad de facturación en varios horizontes, usando métricas pensadas para datos con observaciones censuradas. Como el ERP exporta los datos a nivel de línea de artículo y la empresa factura el pedido completo, los datos se agregan por pedido antes de cualquier análisis. El documento finaliza con un análisis de los tiempos de cada etapa del proceso productivo, apoyado en una segunda fuente de datos, y con una valoración de las limitaciones del planteamiento seguido.

Palabras clave: análisis de supervivencia; modelo de Cox; Random Survival Forest; Gradient Boosting; tiempo hasta facturación; censura; C-índice; Brier Score.

English abstract

This work addresses a problem faced by Toldos Gómez S.L., a Galician company that manufactures and installs awnings: for every order that comes in, how long will it take to become an invoice? The variable of interest is the number of days between the registration of an order in the company's ERP system and the issuing of the corresponding invoice. When the data are extracted, some orders have not yet been invoiced: all that is known about them is that they have been open for at least a certain number of days. Handling that partial information without discarding it, and without treating an outcome that has not yet occurred as if it were known, is why survival analysis is used here rather than a conventional regression model. Starting from the ERP (Enterprise Resource Planning) data, a chain of preparation, modelling and assessment is built. Three families of models are fitted —the Cox proportional hazards model, the Random Survival Forest and Gradient Boosting with Cox loss— and their ability to predict the probability of invoicing over several horizons is compared, using metrics designed for data with censored observations. Since the ERP exports records at the article-line level and the company invoices whole orders, the data are aggregated by order before any analysis. The document closes with an analysis of the duration of each stage of the production process, drawing on a second data source, and with an assessment of the limitations of the approach followed.

Keywords: survival analysis; Cox proportional hazards model; Random Survival Forest; Gradient Boosting; time to invoicing; censoring; C-statistic; Brier Score.

Capítulo 1

Introducción

1.1. Motivación y contexto

Toldos Gómez S.L. es una empresa gallega con sede en A Coruña dedicada a la fabricación, venta e instalación de toldos y productos afines. Su actividad cubre, entre otros, el segmento residencial, el industrial y el sector de la rotulación o el transporte, con ciclos de producción que pueden ser muy distintos entre sí: instalar un toldo doméstico, fabricar una lona industrial a medida o preparar un encargo de rotulación no siguen necesariamente los mismos plazos ni las mismas etapas. La empresa se constituyó en 1990 a partir de un taller de guarnicionería fundado en 1909 en Arzúa y reorientado hacia la confección de toldos y lonas a finales de los años sesenta. Su sede central está en Arzúa, con centros de trabajo en Santiago de Compostela y Bergondo. Los pedidos se reparten entre producto relativamente estandarizado, que se monta con rapidez, y encargos a medida que requieren diseño previo, compra de material a terceros y, en muchos casos, instalación en casa del cliente. Esa convivencia de dos formas de trabajar dentro del mismo taller forma parte del contexto de la variabilidad de tiempos que este trabajo intenta describir.

Este trabajo nace de una colaboración con la empresa. Toldos Gómez estaba desarrollando internamente, en Power BI, una herramienta para anticipar cuándo se facturaría cada pedido. Esa herramienta se apoyaba en promedios históricos por tipo de pedido y devolvía una fecha estimada única, sin ninguna medida de la incertidumbre. La necesidad era real: saber con antelación cuándo va a entrar el dinero de cada pedido permite planificar la tesorería y seguir mejor el proceso productivo. Aquí se aborda el mismo problema desde una perspectiva estadística, con una diferencia: en lugar de una fecha estimada, se ofrece una distribución de probabilidad sobre el tiempo hasta la facturación. Eso permite decir cuándo y con qué grado de incertidumbre. Además, un promedio calculado solo sobre pedidos ya facturados arrastra el sesgo que se describe en la sección 1.3: deja fuera precisamente los pedidos que más tardan.

El sistema de información central de la empresa es un ERP (*Enterprise Resource Planning*), una plataforma que centraliza pedidos, fabricación, albaranes, facturas, clientes y proveedores. Cada transacción se anota con su fecha y su estado, lo que permite reconstruir, pedido a pedido, su ciclo de vida.

La unidad de análisis es el pedido completo, no la línea de pedido. Los registros del ERP están a nivel de línea: cada artículo dentro de un pedido genera su propia fila. Aquí se agregan por pedido porque esa es la unidad sobre la que la empresa factura. Como consecuencia, el número de líneas de cada pedido (`n_lineas`) deja de ser una cuenta administrativa y pasa a usarse como predictor: cuantas más líneas tiene un pedido, más complejo es previsiblemente el encargo.

El ciclo de vida de un pedido sigue, a grandes rasgos, la secuencia

$$\text{Pedido} \longrightarrow \text{Fabricación} \longrightarrow \text{Albarán} \longrightarrow \text{Factura}.$$

Cada etapa puede demorarse por motivos distintos y bastante independientes entre sí. La fabricación depende de si hay materiales disponibles y de la carga del taller. El albarán —el documento que certifica

Figura 1.1: Zona de corte y confección de lona en la nave de Toldos Gómez S.L.

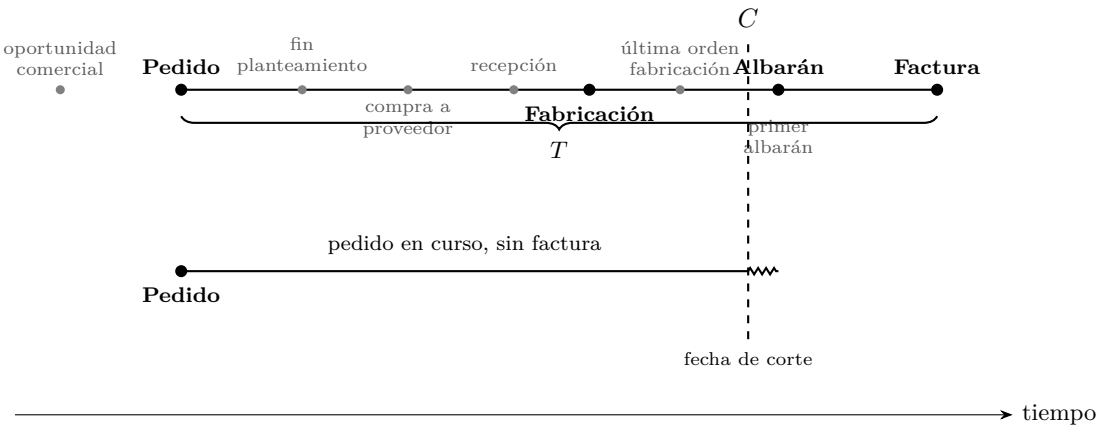

Figura 1.2: Ciclo de vida de un pedido, con los hitos intermedios y la fecha de corte. La fila superior representa un pedido ya facturado, con el intervalo T que modela este trabajo e «hitos productivos» del capítulo 5 superpuestos. La fila inferior representa un pedido censurado: entra en el sistema pero no llega a Factura antes de la fecha de corte C.

que la mercancía se ha entregado o instalado— puede bloquearse por incidencias en la instalación o en el transporte. Y la facturación, que en teoría es el paso administrativo final, puede retrasarse por cuestiones burocráticas o contractuales que nada tienen que ver con la fabricación. Muchas de estas demoras no son visibles hasta que ya han acumulado varios días sin avance.

El impacto económico es directo: cada día entre la entrega y la factura es, contablemente, un día sin ingreso registrado. Una estimación probabilística del tiempo hasta la facturación permite anticipar la posición de tesorería a corto plazo, identificar qué pedidos acumulan retrasos que no se explican por sus características, y priorizar la atención administrativa sobre los casos problemáticos.

El fichero que la empresa exporta desde su ERP recoge 51 028 líneas de pedido registradas entre 2023 y 2026. Tras la agregación por pedido y los criterios de exclusión del capítulo 3, el conjunto de trabajo queda en 27 689 pedidos, de los cuales el 96,8 % han sido facturados y el 3,2 % restante permanece abierto en el momento de la extracción. Para cada pedido se dispone de 26 variables en el conjunto empleado por el modelo de Cox —y hasta 32 en el reservado al Random Survival Forest— que describen sus características, su estado administrativo y productivo, el tipo de cliente y la distancia logística hasta el punto de instalación, entre otras.

1.2. Planteamiento del problema

La pregunta es: ¿cuándo se va a facturar este pedido? Traducirla a un problema definido exige dos decisiones.

La primera es aceptar que la respuesta no puede ser un número exacto, sino una distribución de probabilidad. Los tiempos varían mucho incluso entre pedidos parecidos: mismo tipo de artículo, importe similar, mismo tipo de cliente. Esa variabilidad se debe a factores que no quedan registrados —la disponibilidad de un instalador un día concreto, una incidencia de transporte, un retraso administrativo— y que, por no observarse, no pueden incorporarse como predictores. Dar un único día exacto supondría una precisión que los datos no respaldan.

La segunda es elegir cómo se expresa esa probabilidad. Aquí se estima en tres horizontes fijos:

$$\hat{P}_k(i) \;=\; \hat{\mathbb{P}}\big(T_i \le k \mid \mathbf{X}_i\big), \qquad k \in \{7, 14, 30\} \text{ días},$$

donde T_i es el número de días desde que el pedido i entra en el sistema hasta que se emite su factura, y $\mathbf{X}_i$ son sus características observadas. Los horizontes de 7, 14 y 30 días se corresponden, respectivamente, con la planificación semanal, quincenal y mensual de la empresa.

Si para un pedido $\hat{P}_{14}(i)$ vale 0,80, el modelo dice que hay un 80 % de posibilidades de que esté facturado dentro de dos semanas. La misma cifra permitiría construir un sistema de avisos fijando un umbral por debajo del cual el pedido se señala. La sección 4.6 muestra que ese umbral no puede leerse de forma literal, porque es en las probabilidades bajas donde el modelo resulta más optimista de lo que los datos respaldan.

1.3. Por qué análisis de supervivencia

El análisis de supervivencia no es la única alternativa, pero responde correctamente al problema tal y como se ha planteado. Antes de justificarlo, conviene repasar tres alternativas más directas y por qué no funcionan.

La primera sería calcular, para cada pedido que ya tiene factura, la diferencia en días entre la fecha de la factura y la del pedido, y ajustar sobre esa diferencia una regresión lineal. Esto obliga a descartar los pedidos todavía no facturados (un 3,2 % de la muestra), porque su tiempo hasta la factura no se ha observado. Al eliminarlos, el modelo se entrena solo con los pedidos que ya completaron el proceso, que son sistemáticamente los más rápidos, y el resultado es una predicción optimista que subestima los tiempos de los pedidos problemáticos, que son justamente los que más interesa identificar. Además, la distribución de esos tiempos no cumple las condiciones de una regresión lineal: el histograma tiene una

moda muy temprana y una cola larga de varios meses, la varianza crece con el tiempo esperado y los residuos no son normales, lo que invalida los errores estándar y los contrastes. Y, en último término, la regresión lineal contesta a otra pregunta: da una estimación puntual de días, cuando lo que se busca es una probabilidad en un horizonte fijo.

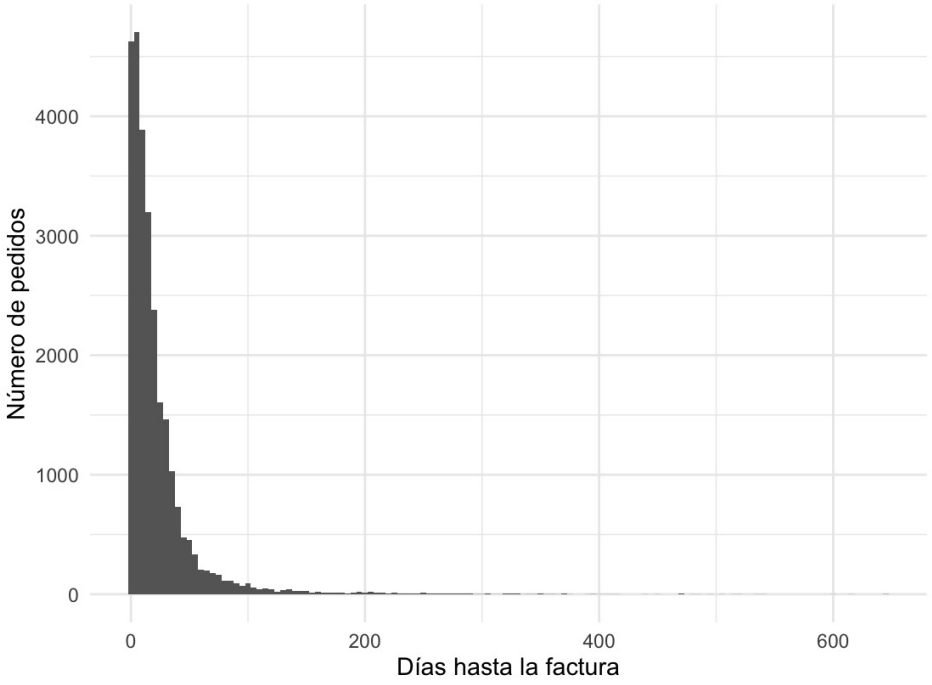

Figura 1.3: Distribución del tiempo hasta la factura en los pedidos facturados.

Otra opción sería convertir el problema en una clasificación binaria en cada horizonte: facturado o no a los k días. La dificultad vuelve a estar en los pedidos abiertos: uno que lleva 3 días sin facturarse no tiene una etiqueta válida para el horizonte de 7 días, porque no se sabe si facturará antes o después. Etiquetarlo por defecto como «no facturado a los 7 días» penalizaría a los pedidos recientes, que simplemente no han tenido tiempo de llegar al horizonte, y descartarlo reproduce la selección sesgada de la alternativa anterior.

La tercera sería incluir todos los pedidos y sustituir el tiempo desconocido de los censurados por un valor fijado de antemano, por ejemplo el máximo observado. Parece resolver el problema de la primera alternativa, pero introduce un sesgo cuya magnitud dependen de los tiempos reales de los pedidos censurados, que por definición se desconocen. El estimador resultante sería inconsistente: aunque hubiera muchos más datos, no convergería al valor verdadero.

El análisis de supervivencia trata de forma distinta, pero igualmente rigurosa, a los pedidos facturados y a los censurados. Para cada pedido i se observa el par $(T_{\text{obs},i}, \delta_i)$:

$$T_{\text{obs},i} = \min(T_i^*, C), \qquad \delta_i = \mathbf{1}(T_i^* \leq C).$$

T_i^* es el tiempo real hasta la factura, el que existiría si pudiéramos esperar indefinidamente; no se observa para los pedidos aún abiertos. C es la fecha de corte, el momento en que se extrajeron los datos. $T_{\text{obs},i}$ es el menor de los dos, y δ_i vale 1 si el pedido ya facturó y 0 si no. Los pedidos con $\delta_i = 0$ solo dicen que llevan, como mínimo, $T_{\text{obs},i}$ días sin facturarse.

La contribución de cada pedido a la verosimilitud es

$$L_i(\theta) = f(T_{\text{obs},i}; \theta)^{\delta_i} \, S(T_{\text{obs},i}; \theta)^{1-\delta_i},$$

donde f es la densidad del tiempo hasta la factura y S la función de supervivencia, que se define en el capítulo 2. Si $\delta_i = 1$, la expresión se reduce a $f(T_{\text{obs},i};\theta)$: el pedido aporta la densidad de haber facturado en el instante en que lo hizo. Si $\delta_i = 0$, se reduce a $S(T_{\text{obs},i};\theta)$: aporta la probabilidad de no haber facturado hasta el momento de la censura. La verosimilitud conjunta es el producto de las contribuciones:

$$L(\theta) = \prod_{\delta_i=1} f(T_{\text{obs},i}) \cdot \prod_{\delta_i=0} S(T_{\text{obs},i}).$$

Ninguna observación se descarta: los facturados informan de cuándo ocurrió el evento y los censurados de que no ha ocurrido a la fecha de corte.

1.4. Revisión de literatura

El análisis de supervivencia viene de la bioestadística —el tiempo hasta la muerte o la recaída de un paciente— y de la ingeniería de fiabilidad —el tiempo hasta el fallo de un componente—. Su aplicación a tiempos de ciclo en procesos empresariales es menos habitual en la literatura revisada para este trabajo, aunque no inexistente: existe una línea consolidada, el *predictive process monitoring*, dedicada a predecir el tiempo restante de un caso a partir de los registros de eventos de un sistema de información (van der Aalst 2016; Verenich et al. 2019; Teinemaa et al. 2019). Lo que sí es menos frecuente en esa línea es el tratamiento explícito de la censura: buena parte de los métodos comparados en Verenich et al. (2019) se ajustan sobre casos ya completados, que es la alternativa descartada en la sección 1.3.

1.4.1. Fundamentos y modelo de Cox

El punto de partida es el estimador no paramétrico de la función de supervivencia de Kaplan y Meier (1958), que estima $S(t)$ sin asumir ninguna forma concreta para la distribución de los tiempos. El mismo principio sostiene el estimador del riesgo acumulado de Aalen (1978), que usan los árboles del Random Survival Forest en cada hoja terminal. Para la implementación en R la referencia es Therneau y Grambsch (2000), que cubre la estimación del modelo de Cox y los residuos de Schoenfeld. Para una estrategia general de construcción, validación y presentación de modelos predictivos de supervivencia, véase también Harrell (2015). De forma más general, incluidos los modelos paramétricos de tiempo de fallo acelerado, puede consultarse Klein y Moeschberger (2003).

Cox (1972) propuso el modelo de riesgos proporcionales. La verosimilitud parcial que introduce permite estimar el efecto de cada predictor sin especificar la función de riesgo basal ni ninguna distribución para los tiempos. La extensión a efectos no lineales mediante splines penalizados se debe a Gray (1992), y el contraste de proporcionalidad basado en los residuos de Schoenfeld lo formalizaron Grambsch y Therneau (1994). El fenómeno de separación perfecta, que puede impedir la convergencia del ajuste cuando una variable predice el desenlace de forma determinista, lo documentan Heinze y Schemper (2001).

1.4.2. Random Survival Forest y Gradient Boosting

Ishwaran et al. (2008) adaptaron el Random Forest de Breiman (2001) a datos de supervivencia: cada partición se decide con el test log-rank, cada hoja estima su riesgo acumulado con Nelson-Aalen (Aalen 1978), y demuestran que la predicción fuera de bolsa (*out-of-bag*, OOB) —la que sale de promediar solo los árboles que no vieron esa observación durante su entrenamiento— es un estimador casi insesgado del error de generalización. Sobre la sensibilidad del método al número de variables candidatas por nodo cuando la señal se concentra en muy pocas variables —una situación que se dará en estos datos—, la referencia es Probst et al. (2019). El Gradient Boosting viene de Friedman (2001); la versión usada aquí, implementada en `mboost`, la documentan Hothorn et al. (2010). Una formulación conjunta de ensembles y boosting para respuestas censuradas está en Hothorn et al. (2006), y una adaptación de métodos de aprendizaje automático al análisis de supervivencia, en Wang et al. (2019).

1.4.3. El trabajo comparable más cercano

El antecedente más cercano es la tesis de Smirnov (2016), que aplica análisis de supervivencia al tiempo hasta el pago de facturas entre empresas y compara también el Cox con el Random Survival Forest. Allí el RSF supera con claridad al Cox, y la autora lo atribuye a que cuenta con el historial de pago de cada cliente, una variable que combina el comportamiento pasado del cliente con el resto de predictores de una forma que un modelo aditivo no puede capturar. En Toldos Gómez no existe una variable equivalente. Esta diferencia se retoma en la sección 6.3.

Las métricas empleadas siguen a Harrell et al. (1996) para el C-índice y la pendiente de calibración, a Graf et al. (1999) para el Brier Score con pesos por probabilidad inversa de censura, y a Uno et al. (2011) para una versión del C-índice más robusta frente a cambios en la distribución de la censura. Para la comparación entre probabilidades predichas y proporciones observadas de la sección 4.6 se siguen Royston y Altman (2013) y Steyerberg (2019).

1.5. Objetivos y estructura del documento

El objetivo principal es construir un procedimiento para predecir el tiempo hasta la facturación en Toldos Gómez y valorar qué familia de modelos resulta más adecuada para este proceso. Para ello se comparan tres familias con capacidades distintas: el modelo de Cox, aditivo e interpretable; el Random Survival Forest, capaz de capturar interacciones sin necesidad de especificarlas de antemano; y el Gradient Boosting con pérdida de Cox, que estima el mismo tipo de modelo que el primero por una vía distinta y con selección automática de variables. Como objetivo secundario, y a partir de una segunda fuente de datos, se describe cuánto tiempo consume cada etapa intermedia del proceso productivo. La validación y la eventual actualización del modelo se plantean siguiendo los principios generales de Steyerberg (2019).

El capítulo 2 desarrolla el marco teórico de los tres métodos y de las métricas de evaluación. El capítulo 3 describe los datos y su preparación. El capítulo 4 presenta el modelado, la comparación conjunta y la evaluación del modelo seleccionado. El capítulo 5 cambia de fuente para describir los tiempos por etapa. El capítulo 6 discute los resultados y recoge las limitaciones, y el capítulo 7 reúne las conclusiones y las líneas de continuación.

Capítulo 2

Marco teórico

Este capítulo recoge cómo funcionan los tres métodos que se usan en el capítulo 4 y las métricas con las que se evalúan.

2.1. Las tres funciones fundamentales

El análisis de supervivencia estudia la distribución de un tiempo aleatorio $T > 0$ hasta que ocurre un evento. Aquí, T es el número de días desde que un pedido entra en el ERP hasta que se emite la factura. La distribución de T puede describirse mediante tres funciones relacionadas entre sí.

La función de supervivencia se define como

$$S(t) = \mathbb{P}(T > t), \qquad S(0) = 1, \qquad \lim_{t\to\infty} S(t) = 0,$$

esto último bajo el supuesto de que todos los pedidos facturables terminan facturándose. $S(t)$ es la probabilidad de que la facturación todavía no haya ocurrido cuando han pasado t días. Un valor como $S(14) = 0{,}40$ quiere decir que el 40 % de los pedidos no se habrá facturado a los 14 días. Es el complementario de la probabilidad definida en el capítulo 1: si $\hat{P}_{14}(i) = 1 - \hat{S}(14 \mid \mathbf{X}_i)$, conocer S es conocer también las probabilidades que interesan a la empresa.

La función de riesgo instantánea (*hazard*), también llamada tasa de riesgo, es

$$h(t) = \lim_{\varepsilon\to 0} \frac{\mathbb{P}(t < T \le t + \varepsilon \mid T > t)}{\varepsilon} = \frac{f(t)}{S(t)},$$

donde $f(t)$ es la densidad de T. El numerador es la probabilidad de que el pedido facture en el intervalo corto que va de t a $t + \varepsilon$, sabiendo que no había facturado en t, de modo que $h(t)$ puede leerse como la velocidad instantánea de facturación en el momento t entre los pedidos que siguen sin facturar. Un riesgo decreciente indicaría que los pedidos que llevan más tiempo en el sistema facturan cada vez más despacio; uno creciente, lo contrario.

La función de riesgo acumulada se obtiene integrando la instantánea:

$$H(t) = \int_0^t h(u)\,du, \qquad S(t) = \exp\big(-H(t)\big).$$

La segunda igualdad conecta las tres funciones: conociendo $H(t)$ se recupera $S(t)$ sin estimar la densidad ni asumir ninguna forma paramétrica. Es lo que permite que el Random Survival Forest —que en cada hoja estima un riesgo acumulado— convierta esa estimación en una curva de supervivencia.

En los capítulos 1 y 2 se emplea la notación matemática de la tabla 2.1; a partir del capítulo 3, los nombres de las columnas del conjunto de datos en tipografía monoespaciada. La sección 3.3 establece la correspondencia.

Tabla 2.1: Notación empleada a lo largo del documento.

Símbolo	Significado
T	Tiempo hasta la facturación (variable aleatoria)
T_i^*	Tiempo real hasta la factura del pedido i; no siempre observable
C	Fecha de corte de la extracción de datos
$T_{\text{obs},i}$	Tiempo observado del pedido i: mín(T_i^*, C)
δ_i	Indicador de evento: 1 si el pedido facturó, 0 si está censurado
$\mathbf{X}_i$	Vector de características observadas del pedido i
$S(t)$	Función de supervivencia: $\mathbb{P}(T > t)$
$h(t)$	Función de riesgo instantánea
$h_0(t)$	Función de riesgo basal del modelo de Cox
$H(t)$	Función de riesgo acumulada
$f(t)$	Función de densidad de T
β	Vector de coeficientes del modelo de Cox
$\hat{P}_k(i)$	Probabilidad estimada de que el pedido i facture antes del día k

2.2. Observaciones censuradas

La censura de este trabajo es una censura administrativa a derecha por fecha de cierre (Therneau y Grambsch 2000): la fecha de corte C es fija y conocida de antemano —el día en que se extrajeron los datos— y no depende de las características de cada pedido. Esto contrasta con la censura aleatoria, en la que cada individuo abandona el estudio en un momento distinto y por razones propias. Una exposición sistemática de los distintos mecanismos de censura y truncamiento puede consultarse en Klein y Moeschberger (2003).

El supuesto que se necesita para que el estimador de supervivencia sea válido es que la censura sea no informativa: que la probabilidad de que un pedido esté censurado sea independiente de su tiempo real hasta el evento, una vez tenidas en cuenta sus características observadas. Ese supuesto no se puede contrastar con los datos disponibles. Los pedidos que siguen abiertos en la fecha de corte tienen, en efecto, tiempos observados más largos que los facturados, pero eso es una consecuencia directa del propio mecanismo de censura —un pedido solo puede seguir abierto si lleva tiempo abierto— y no es evidencia de que la censura sea informativa. Cabe la posibilidad, no contrastable con los datos disponibles, de que una parte de los pedidos abiertos lo estén por arrastrar alguna incidencia. Si existiera una dependencia no recogida entre la censura y el tiempo hasta la facturación, podría afectar a los coeficientes, a las probabilidades de supervivencia, a la calibración y a las métricas predictivas. Comprobarlo requeriría una validación temporal sobre una cohorte con seguimiento suficiente, que se propone en la sección 7.2. Con un 3,2 % de censura, la magnitud del problema está acotada. Esta cuestión se retoma en la sección 6.4.1.

2.3. Estimador de Kaplan-Meier

Antes de ajustar ningún modelo con predictores conviene ver cómo se comporta la facturación en conjunto, sin asumir ninguna forma para la distribución de los tiempos. Para eso se usa el estimador de Kaplan-Meier.

Sean $t_1 < t_2 < \cdots < t_D$ los distintos días en los que se observó al menos una facturación. Para cada t_j, sea n_j el número de pedidos que seguían en riesgo justo antes de ese día —ni habían facturado ni habían sido censurados— y d_j el número de facturas emitidas en t_j. El estimador de Kaplan y Meier

(1958) es

$$\hat{S}(t) = \prod_{t_j \le t} \left(1 - \frac{d_j}{n_j}\right).$$

El cociente d_j/n_j es la proporción de pedidos en riesgo que facturaron ese día, y $1-d_j/n_j$ la probabilidad de no facturarlo, condicionada a seguir en riesgo. El productorio multiplica esas probabilidades a lo largo de todos los días de facturación hasta t, y da la probabilidad acumulada de seguir sin facturar. Los pedidos censurados no producen ningún salto en la curva, pero dejan de contar en el denominador n_j a partir del momento en que se censuran.

Cuando interesa comparar la curva entre dos grupos se recurre al test log-rank (Therneau y Grambsch 2000): en cada día con al menos una factura se compara cuántas ocurrieron realmente en cada grupo frente a cuántas se esperarían si ambos tuvieran la misma curva. Sumando esas diferencias se obtiene un estadístico cuya distribución es conocida bajo la hipótesis de curvas idénticas. Ese mismo criterio, aplicado para elegir el mejor punto de corte de una variable en cada nodo de un árbol, es el que emplea el Random Survival Forest.

2.4. Modelo de riesgos proporcionales de Cox

2.4.1. Formulación

Kaplan-Meier describe la facturación en conjunto o por grupos, pero no permite preguntar cuánto más rápido factura un pedido de una línea de negocio que uno de otra, teniendo en cuenta también su importe y su complejidad. Para eso hace falta un modelo con varios predictores.

Una posibilidad serían los modelos paramétricos de tiempo de fallo acelerado, que asumen una distribución concreta para T (Weibull, log-normal, log-logística) y modelan el logaritmo del tiempo como función lineal de los predictores (Klein y Moeschberger 2003). No se emplean aquí por dos motivos: exigen acertar con la forma de la distribución, mientras que el enfoque que sigue no impone ninguna, y su ventaja principal, la lectura de los coeficientes como factores multiplicativos sobre el tiempo, no se aprovecharía en un trabajo cuyo objetivo es predecir probabilidades en horizontes fijos. Ajustar uno y contrastarlo formalmente contra el modelo elegido queda apuntado en la sección 7.2.

El modelo más usado en supervivencia es el de riesgos proporcionales de Cox (1972):

$$h(t \mid \mathbf{X}_i) = h_0(t) \cdot \exp\!\left(\mathbf{X}_i^{\top} \beta\right).$$

Hay un riesgo base, $h_0(t)$, que sería el mismo para todos los pedidos si no tuvieran características propias, y cada pedido lo escala según sus variables. Ese escalado no puede ser negativo, y por eso se escribe como una exponencial.

Al comparar dos pedidos, el riesgo base se cancela y queda un cociente entre los dos riesgos que no cambia con el tiempo. A eso se le llama supuesto de riesgos proporcionales, y hay que comprobarlo para poder interpretar los coeficientes como efectos constantes. Como se verá en el capítulo 4, no se cumple en estos datos.

Cox (1972) encontró una forma de estimar los coeficientes sin decir nada sobre el riesgo base. Cada día que se emite al menos una factura se reparte una probabilidad entre todos los pedidos que ese día podían haber facturado, en proporción al riesgo de cada uno; el que realmente facturó se lleva la parte que le corresponde según su riesgo relativo. Multiplicando esa probabilidad a lo largo de todos los días se obtiene la verosimilitud parcial, y los coeficientes son los que la hacen máxima.

2.4.2. Splines penalizados

El Cox tal y como se ha descrito asume que el efecto de cada variable continua es una línea recta. Los splines penalizados (Gray 1992) dejan que cada variable continua tenga una curva, estimada a partir de los propios datos, y penalizan la curvatura para que no se ajuste al ruido. Si la relación real es lineal, la curva estimada acaba pareciéndose a una recta.

Los splines permiten que el efecto de cada variable sea no lineal, pero siguen sumando el efecto de unas variables al de otras: un Cox con splines sigue siendo un modelo aditivo. Lo que no puede recoger es que el efecto de una variable dependa del valor que tome otra.

2.4.3. Contraste de riesgos proporcionales

Grambsch y Therneau (1994) propusieron un contraste que, para cada variable, compara en cada momento de facturación el valor que tenía esa variable en el pedido que facturó frente al valor medio de los pedidos que en ese momento seguían compitiendo por facturar. Si el efecto es constante en el tiempo, esas diferencias no deberían mostrar tendencia; si crecen o decrecen, el efecto cambia con el tiempo y el Cox básico se queda corto para esa variable. Un valor p por debajo de 0,05 se interpreta como evidencia de ese cambio. En R se calcula con `cox.zph()` (Therneau y Grambsch 2000).

2.5. Random Survival Forest

2.5.1. Algoritmo

Cuando el proceso real contiene combinaciones entre variables, un modelo aditivo se queda corto. El Random Survival Forest (Ishwaran et al. 2008) es la versión para supervivencia del Random Forest de Breiman (2001), y puede capturar esas combinaciones sin que haya que decirle de antemano cuáles son.

Se construyen mil árboles, cada uno entrenado sobre una muestra distinta de pedidos, extraída al azar y con reemplazamiento. En cada división se prueban varias variables y varios puntos de corte al azar, y se elige la combinación que mejor separa a los pedidos rápidos de los lentos, con el criterio del test log-rank. El árbol sigue dividiéndose hasta que cada rama final tiene un número mínimo de pedidos. En cada rama final se estima el riesgo acumulado con el estimador de Nelson-Aalen (Aalen 1978). Para un pedido nuevo, cada árbol lo hace caer hasta una rama final que le asigna un riesgo, y la predicción es el promedio de las mil estimaciones.

2.5.2. Hiperparámetros

De todo lo anterior hay dos cosas que en realidad son decisiones que hay que tomar: cuántas variables se sortean en cada nodo antes de elegir el mejor corte (`mtry`) y cuántos eventos, como mínimo, tiene que tener cada rama final (`nodesize`). Por defecto, `mtry` toma un valor del orden de $\sqrt{p}$, siendo p el número de predictores.

Ese valor por defecto tiene sentido cuando la información útil está repartida entre muchas variables: al obligar a cada árbol a mirar variables distintas, se consigue la diversidad que hace que promediarlos merezca la pena. El problema es que deja de tener sentido cuando la señal se concentra en muy pocas variables. Si de las p variables solo una o dos aportan algo, sortear $\sqrt{p}$ candidatas en cada nodo implica que, la mayoría de las veces, la variable que de verdad importa ni siquiera entra en el sorteo, y el árbol termina partiendo por variables que no dicen nada (Probst et al. 2019). Así, el bosque puede acabar rindiendo peor de lo que el problema permitiría, y eso no tendría nada que ver con la estructura del proceso, sino con esta elección del hiperparámetro. Como se verá en la sección 4.3, la importancia de variables del bosque, en este caso, se concentra prácticamente en una sola variable.

2.5.3. Validación fuera de bolsa e importancia de variables

Cada árbol se entrena con una muestra que deja fuera, en promedio, a un tercio de los pedidos, así que se puede evaluar cada pedido usando solo los árboles que no lo vieron durante su entrenamiento. Eso equivale a probar el modelo con datos que nunca ha visto, sin reservarlos aparte, y es lo que se conoce como validación fuera de bolsa (OOB). La estimación OOB proporciona una validación interna útil; no obstante, puede resultar optimista cuando se utiliza repetidamente para seleccionar hiperparámetros o

comparar configuraciones, porque las mismas observaciones que sirven para estimar el error intervienen también en la elección del modelo.

Para saber qué variables usa el bosque se barajan al azar los valores de cada una entre los pedidos que se evalúan fuera de bolsa, rompiendo cualquier relación con el tiempo hasta la factura. Si al barajarla el modelo empeora, la variable aportaba información; si no cambia nada, no aportaba gran cosa. Es lo que se conoce como VIMP.

2.6. Gradient Boosting con pérdida de Cox

El tercer método busca minimizar lo mismo que el Cox, pero en vez de resolver el ajuste de una sola vez, lo hace acumulando muchas correcciones pequeñas:

$$\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \nu \cdot h_m(x),$$

donde $h_m(x)$ se ajusta para corregir el error que ha quedado de la predicción anterior, y ν limita cuánto puede corregir cada iteración (Friedman 2001).

La pieza h_m, el llamado aprendiz base, es la que decide qué tipo de modelo se está ajustando en realidad. En `mboost` (Hothorn et al. 2006, 2010) hay varias opciones para elegir:

- Si se usan aprendices lineales (`glmboost`), la suma de todas las correcciones sigue siendo una función lineal y aditiva de los predictores. Es un boosting por componentes: en cada iteración se elige una única variable y se actualiza su coeficiente. Lo que se obtiene, en el fondo, es un modelo de Cox ajustado por descenso de gradiente, con la ventaja de que selecciona variables automáticamente, pero sigue sin ser capaz de descubrir interacciones.
- Si se usan aprendices de tipo árbol (`blackboost`), cada corrección ya puede combinar varias variables a la vez, así que la suma final deja de ser aditiva y sí puede captar interacciones.
- Si se usan aprendices de tipo spline (`gamboost`), cada corrección es no lineal en una variable, pero se siguen sumando entre sí: el resultado es aditivo, aunque ya no lineal.

Aquí se emplea `glmboost`, es decir, boosting por componentes con aprendices lineales, que en la configuración usada resulta prácticamente aditivo. Su ajuste se detalla en la sección 4.4.

2.7. Métricas de evaluación

El C-índice mide si el modelo ordena bien los pedidos por velocidad de facturación, pero no dice si las probabilidades concretas tienen sentido. Por eso se añade una medida de si esas probabilidades encajan con lo que pasa, una medida de cómo se comporta el error a lo largo del tiempo, y una versión del C-índice que no se vea afectada si la proporción de censura cambia. La evaluación se organiza, por tanto, en discriminación, calibración y error predictivo, siguiendo el marco de Steyerberg et al. (2010).

2.7.1. C-índice de Harrell y validación cruzada

La definición de Harrell et al. (1996) es

$$C = \frac{\#\{\text{pares concordantes}\} + \frac{1}{2}\,\#\{\text{empates}\}}{\#\{\text{pares comparables}\}}.$$

Se cogen pares de pedidos donde se sepa con certeza cuál facturó antes y se comprueba si el modelo también cree que ese fue el más rápido. Un valor de 0,5 equivale a ordenar al azar y uno de 1 a acertar siempre.

Calcularlo sobre los mismos datos con los que se entrenó el modelo da una cifra optimista. Para tener una idea más realista se usa validación cruzada: se parte el conjunto en 5 trozos, se entrena con 4 y se calcula el C-índice sobre el que quedó fuera, repitiendo 5 veces y promediando. Cuanto más se parezcan las dos cifras, menos está sobreajustando el modelo.

Comparar C-índices calculados con procedimientos de validación distintos —uno en entrenamiento, otro en validación cruzada, otro fuera de bolsa— no es legítimo. La tabla 4.2 anota el procedimiento de cada cifra, y la sección 4.5 solo compara las que son comparables.

2.7.2. Brier Score con ponderación por censura

El Brier Score mira si las probabilidades en sí mismas son razonables, comparando lo que predice el modelo con lo que pasó. Con pedidos censurados no siempre se sabe qué pasó, así que Graf et al. (1999) propusieron ponderar cada pedido según lo probable que sea que siga siendo observable en ese momento. La estimación consistente del Brier Score bajo censura a derecha se desarrolla con más generalidad en Gerds y Schumacher (2006).

Un modelo que dijera siempre que nadie va a facturar tendría un Brier Score bajo al principio sin ser bueno, así que hace falta un punto de comparación: el modelo nulo, que asigna la misma probabilidad a todos los pedidos sin mirar sus características.

2.7.3. C-índice de Uno

El C-índice de Harrell depende de cuánta censura hay en los datos con los que se calcula. Si la empresa facturase de forma sistemáticamente más rápida o más lenta, cambiaría la censura y el C-índice podría no ser comparable, sin que el modelo hubiera cambiado. Uno et al. (2011) propusieron una versión que corrige esto dando más peso a los pedidos con más riesgo de perderse por la censura. En R se calcula con `concordance(..., timewt = "n/G2", tau)`. En este trabajo se comparan ambas estimaciones de manera descriptiva, sin adoptar un umbral universal.

2.7.4. Pendiente de calibración

Se coge la predicción que da el Cox completo y se usa como única variable en un Cox nuevo. El coeficiente que resulta —la pendiente de calibración— valdría 1 si el modelo original estuviera perfectamente calibrado. Por debajo de 1 indicaría que el modelo es más extremo de lo que los datos justifican; por encima, que es más conservador de lo necesario.

Cuando esta comprobación se hace sobre los mismos datos con los que se entrenó el Cox, el resultado sale cercano a 1 casi por construcción y no dice nada sobre cómo se comportaría el modelo con pedidos nuevos (Van Houwelingen 2000). Por eso en el capítulo 4 se denomina pendiente aparente de calibración y no se presenta como una validación independiente. En general, la calibración debe evaluarse gráficamente y mediante parámetros de calibración, no solo con promedios agregados (Van Calster et al. 2019).

2.7.5. Comparación con una muestra independiente

Las métricas anteriores se calculan sobre el mismo conjunto con el que se ajustó el modelo, o sobre particiones de ese conjunto. Una comprobación distinta consiste en aplicar el modelo ya ajustado a una muestra que no intervino en el ajuste y comparar lo que predice con lo que ocurrió (Royston y Altman 2013). La forma más simple es comparar, en cada horizonte k, la probabilidad media predicha con la proporción real:

$$\bar{Y}_k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(t_i \le k), \qquad \bar{\hat{P}}_k = \frac{1}{n}\sum_{i=1}^{n} \hat{P}_k(i).$$

Esa comparación resume la calibración en un único número por horizonte y puede ocultar que el modelo calibre bien en unos rangos de probabilidad y mal en otros, así que se acompaña de una comparación por deciles de la probabilidad predicha (Steyerberg 2019), que admite una lectura gráfica directa sobre la diagonal de referencia (Austin et al. 2020).

La muestra independiente disponible en este trabajo está formada solo por pedidos que terminaron facturándose, de modo que no reproduce la población sobre la que se ajustó el modelo, que incluye

pedidos censurados. Por eso, en la sección 4.6 el procedimiento se presenta como una comparación descriptiva en una muestra de pedidos finalmente facturados, y no como una validación externa en sentido estricto. Una validación externa exigiría una cohorte que incluyese todos los pedidos elegibles, y se propone en la sección 7.2.

Capítulo 3

Datos y preparación

3.1. Fuentes de datos

Para este trabajo se parte de dos ficheros que salen del ERP de Toldos Gómez.

El primero es el conjunto principal (`Datos_usar.xlsx`), que recoge todos los pedidos que hay registrados en el ERP, tengan factura o no. Es el único de los dos que incluye pedidos censurados, así que es el que se usa para entrenar y evaluar los modelos del capítulo 4. Aquí hay que tener en cuenta una cosa: el ERP no exporta un pedido por fila, sino una línea por artículo, de modo que un pedido con varios artículos aparece repetido en varias filas, todas con la misma fecha de pedido, la misma fecha de factura y los mismos datos de cliente. De cómo se resuelve esto se encarga la sección 3.2.

El segundo es el conjunto de hitos (`Datos_usar2.xlsx`), que solo tiene pedidos ya facturados, pero a cambio aporta las fechas de los hitos intermedios: cuándo se detectó la oportunidad comercial, cuándo se cerró el planteamiento técnico, cuándo se compró a los proveedores y cuándo llegó el material, y la fecha del primer albarán. Como no tiene pedidos censurados, no sirve para entrenar un modelo de supervivencia (se quedarían fuera justo los pedidos más problemáticos), pero es muy útil para comparar las probabilidades del Cox con lo que pasó de verdad y para ver cuánto tarda cada etapa del proceso.

Lo ideal habría sido tener un único fichero que juntara, para todos los pedidos, tanto el desenlace como los hitos intermedios, porque entonces esos hitos podrían usarse directamente como predictores. Pero ese fichero no existe: los hitos solo quedan registrados cuando el pedido se cierra administrativamente, así que para los pedidos que siguen abiertos simplemente no están disponibles todavía. Por eso se opta por usar el conjunto principal para el modelado y dejar el de hitos para las duraciones y la comparación de probabilidades, sin mezclar las dos fuentes en un mismo ajuste. Hacerlo de otra forma metería un sesgo de selección, porque usar los hitos como predictor solo tendría sentido para los pedidos ya facturados, y no habría manera de comprobar si ese efecto se sostiene también en los censurados. Se vuelve sobre esta limitación en la sección 6.4.1.

3.1.1. Cruce entre las dos fuentes

El fichero de hitos tiene 14 577 líneas que corresponden a 7 997 pedidos distintos, todos facturados. Esas líneas se agregan por pedido (quedándose con la fecha más temprana de cada hito y la categoría más frecuente en las variables cualitativas) y se cruzan con `dat_cox` por la columna `Pedido`. El resultado, `dat_enriquecido`, se queda en 7 900 pedidos: algo menos que los 7 997 de partida, porque solo entran los que aparecen en las dos fuentes a la vez y que además pasan los filtros del Cox. Como las dos fuentes ya vienen agregadas por pedido, el cruce es sencillo, uno a uno.

Hay un detalle curioso: un 0,1 % de los pedidos de `dat_enriquecido` figuraba todavía como abierto en el fichero principal en la fecha de corte, pero en el fichero de hitos ya aparecía facturado. Lo más probable es que las dos extracciones no se hicieran exactamente el mismo día.

3.2. Unidad de análisis: el pedido

El fichero principal tiene algo más de 51 000 filas, pero corresponden a unos 31 500 pedidos distintos. Esa diferencia no es un fallo del ERP, es simplemente cómo exporta los datos: cada fila es un artículo dentro de un pedido. Si el pedido es de un solo toldo, ocupa una fila; si lleva tres toldos, una lona y una instalación, ocupa cinco. Hay pedidos que llegan a tener más de 80 líneas.

Por eso la unidad de análisis tiene que ser el pedido, no la línea. Y no es una cuestión de gusto metodológico, es que el propio suceso que se quiere modelar funciona así: la empresa no factura líneas sueltas, factura el pedido entero. Todas las líneas de un mismo pedido comparten fecha de factura porque, en realidad, son la misma factura. Si se modelara a nivel de línea, se estarían tratando como sucesos independientes cosas que en realidad son el mismo suceso repetido tantas veces como artículos tenga el pedido.

Y no agregar tendría consecuencias serias. El modelo se creería que tiene 51 000 observaciones independientes cuando en realidad tiene unas 31 500, lo que hace que los errores estándar salgan artificialmente más pequeños de lo que deberían. Además, cada pedido pesaría según su número de líneas: uno con 80 líneas contaría como 80 pedidos, mientras que uno con una sola línea contaría como uno, y los pedidos con más líneas no son precisamente un caso cualquiera, sino los más complejos. Y para rematarlo, al hacer la validación cruzada, las líneas de un mismo pedido podrían acabar repartidas entre entrenamiento y prueba, así que el modelo estaría siendo evaluado en pedidos que, en la práctica, ya ha visto, y el C-índice dejaría de medir generalización de verdad.

3.2.1. Reglas de agregación

Tabla 3.1: Cómo se pasa de línea a pedido, según el tipo de variable.

Tipo de variable	**Qué se hace**	**Por qué**
Constantes dentro del pedido	Se coge el primer valor	Cosas como la fecha de pedido, el municipio, el tipo de cliente o el CNAE son iguales en todas las líneas del pedido, así que da igual cuál se coja.
Variables que se suman	Se suman	El importe de cada línea sale de cantidad × precio × (1 − descuento), así que el importe del pedido es, lógicamente, la suma de sus líneas. Con la cantidad pasa lo mismo.
Estructura del pedido	Se cuenta	El número de líneas se guarda como variable propia (`n_lineas`). En el fichero original esa columna en realidad numeraba las líneas, no las contaba.
Categorías descriptivas	Se queda con la de mayor importe	Un pedido de 900 € de toldo y 100 € de rotulación es, para la empresa, un pedido de toldo. Fiarse del importe tiene más sentido que fiarse de cuál categoría se repite más, que en un pedido con muchas líneas baratas de un tipo y una cara de otro elegiría la categoría equivocada.
Estado del proceso	Se queda con el más atrasado	Si una sola línea tiene la orden de fabricación bloqueada, el pedido entero está bloqueado: no avanza hasta que avanza la línea más rezagada.
Fecha de factura	La última fecha, y solo si todas las líneas están facturadas	Un pedido no se considera facturado mientras le quede una línea suelta sin facturar. Coger la primera fecha haría que los pedidos grandes, que son los que más tardan, parecieran más rápidos de lo que son.

3.2.2. El orden de los filtros

Aquí hay algo importante que tener en cuenta: el mismo filtro no da el mismo resultado según se aplique antes o después de agregar las líneas en pedidos.

El filtro por tipo de albarán se aplica primero, a nivel de línea. Las líneas de tipo «No Facturable», «Solo para transporte» y similares no son pedidos que nunca vayan a facturar, sino más bien logística interna dentro de un pedido que sí va a facturar. Por ejemplo, un pedido con una línea de reparación y otra línea de cambio de tela marcada como no facturable factura la reparación con total normalidad. Quitando esa línea antes de agregar, el pedido queda intacto; si en cambio se excluyera el pedido entero, se estaría perdiendo información válida. Con este filtro el fichero pasa de 51 028 a 45 051 líneas, y de 31 554 a 27 917 pedidos (los que desaparecen son los que tenían todas sus líneas marcadas como no facturables). Y de propina, buena parte de los casos que a primera vista parecían facturación parcial se resuelven solos en cuanto se retira la línea que nunca iba a facturar.

El filtro por importe negativo, en cambio, se aplica después de agregar, ya sobre el importe sumado. Hay pedidos que mezclan líneas positivas con una línea de abono negativa que compensa parte del importe: si se filtrara por línea, se descartaría el abono pero se conservaría el cargo, y el pedido quedaría con un importe inflado. Por el mismo motivo se añade también la condición de que la cantidad neta no sea negativa, y además porque una cantidad negativa impediría hacer la transformación logarítmica de la sección 3.4.

Después de agregar y aplicar estos filtros, el conjunto se queda en 27 735 pedidos. Quitando además los pedidos con tiempo observado negativo (tres casos que son, sencillamente, errores de registro), quedan 27 732, que son los que forman `dat_rsf_completo`.

3.2.3. Exclusión de variables

Se eliminan `ImporteFactura`, `AnhoFactura`, `MesFactura` y `SemanaFactura` porque solo existen una vez que la factura ya se ha emitido, y usarlas sería, en la práctica, predecir el evento con el propio evento. También se elimina `ComentarioLineaPedido`, por ser texto libre con un valor prácticamente distinto en cada línea. `Precio` se quita por ser redundante con `ImportePedido`, y `CodCliente` se descarta como identificador que no aporta señal por sí mismo. `AnhoPedido` y `MesPedido` son redundantes con `FechaPedido`. Por último, `TipoPedidoSinCargo` se elimina porque está ausente en la inmensa mayoría de las líneas.

3.2.4. Exclusión de filas

Los dos últimos criterios solo aplican al conjunto del Cox. El Random Survival Forest no tiene esos problemas de convergencia, así que puede seguir usando esos pedidos sin problema, y es precisamente uno de los motivos por los que se acaban construyendo varios conjuntos distintos.

3.2.5. Separación perfecta y variables excluidas del Cox

La separación perfecta pasa cuando, para un valor concreto de una variable, todos los pedidos con ese valor tienen el mismo desenlace: por ejemplo, todos los pedidos con el albarán bloqueado están censurados sin excepción. Con eso, el Cox intenta subir el coeficiente de esa variable sin ningún límite, porque cuanto más lo sube mejor explica lo que está viendo, y el ajuste simplemente no converge; es lo que Heinze y Schemper (2001) llaman verosimilitud monótona. Con la colinealidad perfecta pasa algo parecido: impide resolver las ecuaciones de estimación.

Al Random Survival Forest ninguno de los dos problemas le afecta, porque no estima coeficientes ni resuelve ninguna ecuación: en cada división simplemente comprueba si un corte separa bien a los pedidos rápidos de los lentos. Eso sí, la colinealidad puede repartir la importancia entre variables correlacionadas y complicar la lectura de la VIMP: si dos variables llevan la misma información, barajar solo una apenas empeora el modelo porque la otra sigue estando disponible, así que las dos pueden acabar con una importancia baja aun siendo informativas. La tabla 3.3 recoge las variables afectadas.

Tabla 3.2: Qué filas se excluyen del conjunto principal y por qué.

Se excluyen	Nivel	Motivo
Líneas con tipo de albarán «No Facturable», «Recoge en nave no facturable», «Solo para transporte» o «Alquiler»	Línea	Son logística interna, no ventas: nunca generan factura propia. Se quita la línea, no el pedido; el pedido solo desaparece si todas sus líneas son de este tipo.
Pedidos con importe o cantidad neta negativa	Pedido	Son abonos contables. Se comprueba sobre el importe ya sumado, porque filtrando por línea el importe de los pedidos que mezclan cargo y abono saldría inflado.
Pedidos con tiempo observado negativo	Pedido	Son errores de registro: la fecha de factura queda antes que la fecha de pedido.
Pedidos con `SituacionOF` = «Bloqueada» (agrupa también «creada» y «detenida»); solo en el conjunto del Cox	Pedido	Están todos censurados sin excepción, lo que provoca separación perfecta y hace que el Cox no converja (sección 3.2.5).
Pedidos con `FormaEntrega` = «Ya instalado – nota de abono»; solo en el conjunto del Cox	Pedido	Dan una razón de riesgos disparatadamente alta que desestabiliza el resto de coeficientes.

Tabla 3.3: Variables que se quedan fuera del Cox pero que el Random Survival Forest sí puede usar.

Variable	Qué problema da en el Cox	Por qué el RSF sí puede usarla
`AlbaranBloqueado`	Separación perfecta: los pedidos con este indicador activo están todos censurados.	El RSF no estima coeficientes; solo evalúa si el corte mejora el test log-rank.
`Cumplimentado`	Cuasi-separación: entre los pedidos «pendiente», prácticamente ninguno ha facturado.	El corte sigue siendo informativo aunque el desequilibrio sea muy grande.
`Fabricado`, `PrioridadOF`, `tiene_OF`, `Albaran`	Colinealidad perfecta con `SituacionOF`.	El RSF no necesita invertir ninguna matriz de diseño.

3.3. Variables de supervivencia

La fecha de corte se fija como $C = \text{máx}(\texttt{FechaPedido})$. A partir de ahí:

$$\delta_i \;=\; \mathbf{1}\{\texttt{FechaFactura}_i \text{ observada}\}, \qquad \texttt{tiempo\_obs}_i = \begin{cases} \texttt{FechaFactura}_i - \texttt{FechaPedido}_i & \text{si } \delta_i = 1, \\ C - \texttt{FechaPedido}_i & \text{si } \delta_i = 0. \end{cases}$$

En R esto se calcula simplemente con `!is.na(FechaFactura)`. Con una tasa de censura del 3,2 % (879 pedidos de 27 689), el modelo cuenta con el desenlace real de más de 19 de cada 20 pedidos, así que la información que se pierde por censura es, en conjunto, bastante limitada.

De aquí en adelante, `tiempo_obs` y `evento` son la forma concreta en la que se implementan las cantidades $T_{\text{obs},i}$ y δ_i de la tabla 2.1.

3.4. Predictores y transformaciones

En total se dispone de 32 variables, repartidas en seis bloques.

Temporales. Semana del año (`SemanaPedido`), día de la semana, si cae en verano o en navidad, y una codificación cíclica del mes con seno y coseno (`sen12` $= \sin(2\pi \cdot \text{mes}/12)$, `cos12` $= \cos(2\pi \cdot \text{mes}/12)$). La razón de esta última es sencilla: con un número del 1 al 12, diciembre y enero quedarían tan lejos entre sí como enero y julio, cuando en realidad están pegados; con esta codificación quedan cerca, como corresponde en el calendario.

Plazo solicitado por el cliente. `AnticipacionSolicitada` recoge cuántos días pide el cliente entre que se registra el pedido y la fecha en la que quiere recibirlo, calculada como `FechaSolicitada` − `FechaPedido`. Como acaba siendo la variable con más peso en todos los modelos, merece la pena aclarar dos cosas sobre ella. Primero, su valor queda fijado en el momento en que el pedido entra en el sistema, así que está disponible justo cuando el modelo necesita predecir, y no tiene el problema que sí tenían las variables eliminadas en la sección 3.2.3. Y segundo, mide una petición del cliente, no un compromiso de la empresa ni una fecha de entrega real, así que no es una forma disfrazada del propio desenlace.

Volumen y variables económicas. Número de líneas del pedido, como aproximación a lo complejo que es (`n_lineas`), cantidad pedida, importe, y si hubo descuento (variable binaria, porque la mayoría de los pedidos no lleva descuento).

Proceso y estado administrativo. Si está cumplimentado, si está fabricado, estado de la orden de fabricación, forma de entrega, tipo de albarán, si tiene anticipo, si el albarán está bloqueado, prioridad administrativa y de fabricación, y si dispone de hoja de almacén.

Cliente y negocio. Línea de negocio, tipo de cliente, actividad económica (CNAE, agrupando en «Otros» los sectores con menos de 50 observaciones), campaña de marketing, tipo de factura, si incluye rotulación, tipo de contabilidad, y una categorización del artículo en 14 tipos, obtenida aplicando reglas de texto sobre los códigos de artículo.

Logística. Tiempo de conducción en minutos desde la base logística hasta el municipio del cliente (`tiempo_min`), calculado sobre rutas reales con OSRM.

Cinco variables tienen una distribución bastante asimétrica, así que se transforman con $x' = \log(1+x)$, que admite valores cero y no cambia el orden entre observaciones: `ImportePedido`, `n_lineas`, `CantidadPedido`, `tiempo_min` y `AnticipacionSolicitada`. Los valores ausentes de `AnticipacionSolicitada` y de `tiempo_min` se imputan por la mediana. Se llegó a construir un indicador binario de imputación para cada una, aunque al final no se incorporó al conjunto de predictores definitivo (sección 6.4.1).

Antes de ajustar, algunos niveles se agrupan para evitar categorías con muy pocas observaciones: los estados «creada» y «detenida» de la orden de fabricación se tratan junto con «bloqueada», y «lanzada» junto con «con imputaciones», porque son fases contiguas del mismo proceso; en `Prioridad`, `TipoFactura`, `Anticipo` y la campaña de marketing, los niveles con muy pocas observaciones se agrupan en una categoría residual. En `FechaSolicitada` se descartan las fechas que caen fuera del intervalo 2020–2027, que son huecos del formato de exportación de Excel, y también las anticipaciones negativas, que indicarían una fecha solicitada anterior al propio pedido; cuando un pedido tiene varias líneas con fecha solicitada distinta, se toma la más tardía, por ser la que de verdad cierra el encargo completo.

3.5. Los conjuntos de datos finales

Con un solo conjunto no habría sido suficiente para comparar el Cox y el RSF como es debido. El Cox no puede incorporar seis de las variables disponibles, y además tiene que excluir algunos pedidos que le impiden converger. Si el RSF rindiera de forma distinta, esa diferencia podría deberse a la estructura del proceso, pero también, sencillamente, a que dispone de más variables o de más pedidos, y con un único conjunto no habría manera de distinguir una cosa de la otra. `dat_rsf_fair` iguala al

Tabla 3.4: Los tres conjuntos que salen del conjunto principal.

Conjunto	Variables	Pedidos	Para qué sirve
`dat_cox`	26	27 689	El modelo de Cox y todas sus variantes
`dat_rsf_fair`	26	27 689	Comparar en igualdad de condiciones con el Cox: mismas variables, mismos pedidos
`dat_rsf_completo`	32	27 732	El Random Survival Forest sin restricciones: todas las variables, todos los pedidos

Cox en todo, así que es la comparación limpia. `dat_rsf_completo`, en cambio, le da al bosque todas las ventajas posibles, sin ninguna de las restricciones que el Cox necesita. Si ni así el bosque mejora al Cox, ya no hace falta un cuarto conjunto para saber que la desventaja no venía ni de las variables ni de los pedidos.

3.6. Entorno computacional y reproducibilidad

Todos los análisis se han hecho en R [versión]. Para el modelo de Cox, sus variantes con splines y los contrastes de proporcionalidad se usa `survival`; para el Random Survival Forest, `randomForestSRC` (Ishwaran y Kogalur 2023); para el Gradient Boosting, `mboost`; para el Brier Score con ponderación por censura, `pec`; y para los tiempos de conducción sobre rutas reales, `osrm`. El tratamiento de los datos y los gráficos se apoyan en `tidyverse`, `lubridate`, `readxl`, `survminer` y `broom`.

Todo lo que tiene un componente aleatorio se ha fijado con semilla: el remuestreo de los árboles del bosque, cómo se reparten las observaciones entre los pliegues de la validación cruzada, y la extracción de la submuestra del Gradient Boosting. El apéndice A recoge los fragmentos de código que corresponden a cada una de las decisiones de este capítulo.

Capítulo 4

Modelado, resultados y evaluación

Este capítulo recoge el ajuste y la comparación de los modelos. Se sigue este orden: primero Kaplan-Meier sin covariables, luego el Cox en sus distintas variantes, después las configuraciones del Random Survival Forest y el Gradient Boosting. Al final se juntan todos los resultados en una comparación conjunta, y al modelo que sale elegido se le aplican las métricas de la sección 2.7, primero sobre los propios datos de ajuste y después sobre una segunda fuente.

Cada modelo tiene aquí un papel distinto. El Cox es el modelo de referencia: interpretable y de uso consolidado. El Random Survival Forest aporta justo lo que al Cox le falta por diseño, capturar interacciones sin tener que especificarlas a mano, y aquí funciona sobre todo como diagnóstico: si no mejora al Cox, es una señal de que esas interacciones no hacían falta. Y el Gradient Boosting minimiza la misma función de pérdida que el Cox pero por una vía de estimación distinta, con selección automática de variables, lo que permite comprobar si el buen rendimiento del Cox depende del procedimiento de estimación o no.

4.1. Kaplan-Meier

Se estima $\hat{S}(t)$ sobre todos los pedidos de `dat_cox`. La curva cae con fuerza en los primeros 15 días, que es donde se concentra la mayor parte de la facturación, y después deja una cola larga y bastante plana de varios meses. En los tres horizontes de interés:

$$\hat{S}(7) = 0{,}662, \qquad \hat{S}(14) = 0{,}464, \qquad \hat{S}(30) = 0{,}216,$$

con errores estándar en torno a 0,003 en los tres casos. Es decir: sin mirar ninguna característica del pedido, un 33,8 % ya está facturado a los 7 días, un 53,6 % a los 14 y un 78,4 % a los 30. Estas cifras son la referencia frente a la que hay que juzgar cualquier modelo que sí use predictores.

Comparando por línea de negocio con el test log-rank sale un estadístico de 5 062 con 7 grados de libertad y un valor p por debajo de 2×10^{-16}: las curvas difieren claramente entre líneas. La diferencia se concentra sobre todo en tres grupos: «Sin línea de negocio», «MOHEVA» y «Lonas para el transporte» facturan más rápido de lo esperable si todas las curvas fueran iguales, mientras que «Protección solar» y «Carpas y estructuras» facturan más despacio. Esto ya deja intuir el peso que la línea de negocio va a tener en el modelo de Cox.

`tiempo_obs` tiene medianas más altas entre los pedidos censurados que entre los facturados. Como ya se explicó en la sección 2.2, esto es justo lo que cabe esperar de una censura administrativa, y no es evidencia de que la censura sea informativa.

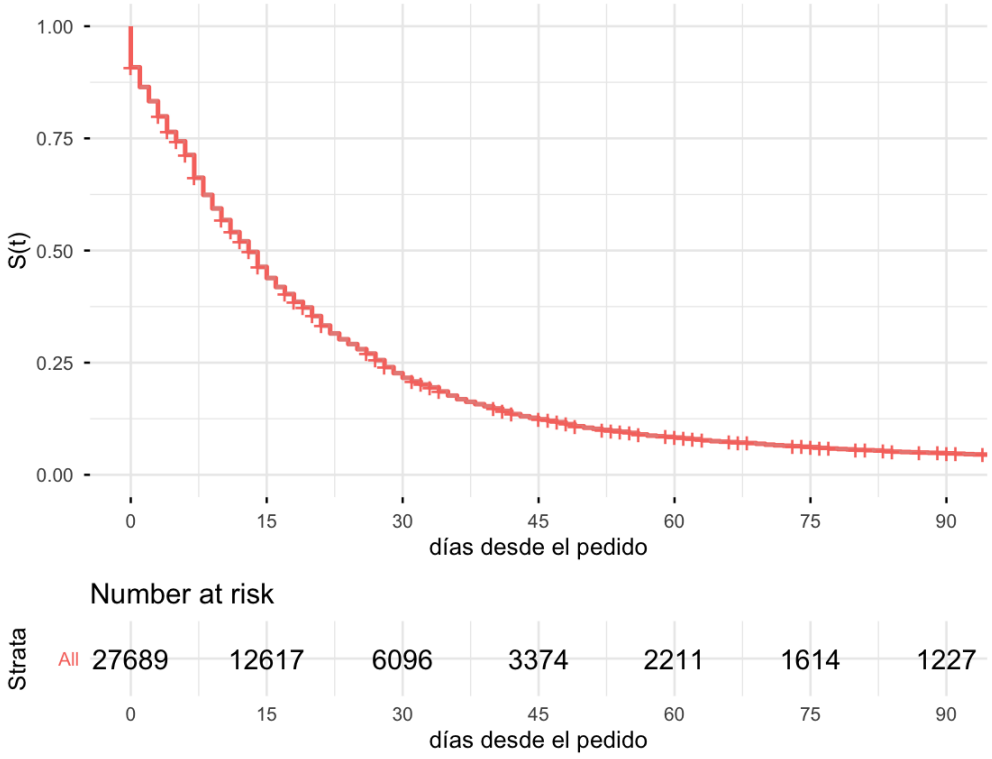

Figura 4.1: Curva de Kaplan-Meier global del tiempo hasta la facturación.

4.2. Modelo de Cox

4.2.1. Cox lineal

Se ajusta un Cox estándar sobre las 26 variables de `dat_cox`, con `ties = .efron"`. Con 27 689 pedidos agregados por jornada, es normal que varios pedidos facturen el mismo día, y la verosimilitud parcial necesita alguna regla para repartir esos empates. Aquí se tratan con la aproximación de Efron (1977), que encaja mejor que la de Breslow cuando los empates son tan frecuentes como en este caso.

La tabla 4.1 recoge los diez coeficientes con menor valor p, expresados como razones de riesgos ($\exp(\hat{\beta})$). La tabla completa, con los 87 términos del modelo, está en el apéndice B.

`AnticipacionSolicitada` es el término con menor valor p de todos, por delante incluso del nivel «MOHEVA» de `LineaNegocio`, que factura unas once veces más rápido que la categoría de referencia, algo coherente con que sean productos más estandarizados. `AnticipacionSolicitada` tiene un efecto decreciente y muy preciso: cuanto más margen pide el cliente, más despacio tiende a facturar el pedido. Completan la lista `TipoContabilidad`, `Anticipo`, una categoría de `FormaEntrega` y dos niveles de `SituacionOF`, lo cual encaja con el peso que ya se intuía de la orden de fabricación.

Conviene leer estos resultados con cierta cautela: como se explica en la sección siguiente, el supuesto de riesgos proporcionales no se sostiene en estos datos, así que cada razón de riesgos hay que entenderla como un efecto promedio a lo largo del seguimiento, no como un efecto constante en el tiempo.

El C-índice es 0,8150 sobre los datos de entrenamiento y 0,8136 en validación cruzada de 5 particiones (desviación típica 0,0038). La diferencia entre ambos, 0,0014, es pequeña. Y como cada pedido ocupa una sola fila, cada pedido cae en un único pliegue, con lo que la cifra de validación cruzada realmente está midiendo generalización y no memorización.

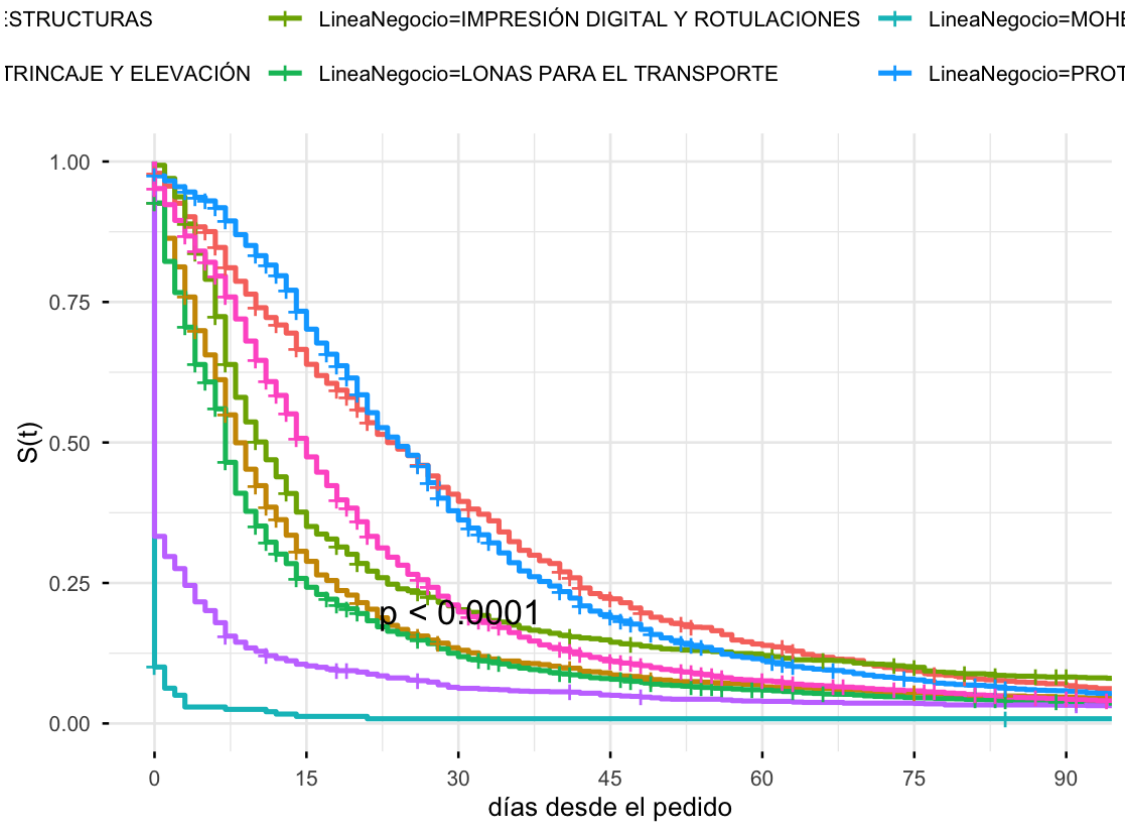

Figura 4.2: Curvas de Kaplan-Meier por línea de negocio.

Tabla 4.1: Los diez coeficientes con menor valor p del modelo de Cox lineal.

Término	**HR** $= \exp(\hat{\beta})$	**E. estándar**	**Estadístico**	**Valor** p
`AnticipacionSolicitada`	0,570	0,0081	−69,2	$< 0{,}0001$
`LineaNegocio`: MOHEVA	10,9	0,0814	29,3	$< 0{,}0001$
`TipoContabilidad`: 8	0,512	0,0301	−22,2	$< 0{,}0001$
`FormaEntrega`: instalación sin compra ni fabricación previa	3,21	0,0560	20,9	$< 0{,}0001$
`SituacionOF`: sin orden de fabricación	3,64	0,0695	18,6	$< 0{,}0001$
`SituacionOF`: finalizada	2,57	0,0628	15,1	$< 0{,}0001$
`Anticipo`: 1	0,667	0,0293	−13,8	$< 0{,}0001$
`LineaNegocio`: lonas para el transporte	1,56	0,0325	13,7	$< 0{,}0001$
`FormaEntrega`: instalar en nave TGM	0,692	0,0359	−10,3	$< 0{,}0001$
`Prioridad`: 3	1,23	0,0204	10,1	$< 0{,}0001$

Los nombres de los niveles se muestran completos y no como los abrevia la consola de R. La categoría de referencia de `LineaNegocio` es «Carpas y estructuras»; la de `SituacionOF` es «Con imputaciones», nivel que agrupa también las órdenes lanzadas. Una razón de riesgos mayor que 1 indica facturación más rápida que la categoría de referencia.

4.2.2. Contraste de proporcionalidad

El contraste de Grambsch y Therneau (1994) da $\chi^2 = 13\,617{,}4$ con 86 grados de libertad y un valor p prácticamente nulo: el supuesto de proporcionalidad se viola con claridad. Si se ordenan las variables por el estadístico dividido entre sus grados de libertad, las desviaciones más marcadas son las de `AnticipacionSolicitada` ($\chi^2 = 4\,962{,}3$, 1 gl), `ImportePedido` ($\chi^2 = 4\,953{,}1$, 1 gl), `TipoContabilidad` ($\chi^2 = 2\,207{,}1$, 1 gl) y `SituacionOF` ($\chi^2 = 3\,870{,}9$, 2 gl); `FormaEntrega` es la que tiene el estadístico total más alto ($\chi^2 = 5\,870{,}6$, 5 gl). El hecho de que la variable con más peso en el modelo, `AnticipacionSolicitada`, sea también de las que más se alejan de la proporcionalidad obliga a leer su razón de riesgos como un promedio a lo largo del seguimiento, no como algo fijo.

Que este supuesto no se cumpla no es raro en un contexto empresarial: el efecto del estado de la orden de fabricación, por ejemplo, puede pesar más al principio del seguimiento y diluirse después. Esto no invalida el modelo para el objetivo predictivo, porque el C-índice sigue siendo válido aunque las razones de riesgos no sean constantes, pero sí limita leer los coeficientes como efectos fijos en el tiempo. Con 27 689 observaciones, además, el contraste detecta como significativas desviaciones que pueden ser irrelevantes en magnitud: un valor p nulo con este tamaño de muestra dice que la desviación existe, no que sea grande. Por eso conviene mirar también la figura 4.3. Esta limitación se retoma en la sección 6.4.2.

Dada esta evidencia de no proporcionalidad, una alternativa razonable serían los modelos paramétricos flexibles de Royston y Parmar (2002), que permiten que la forma del riesgo cambie con el tiempo en vez de suponerla constante. Ajustarlos queda fuera del alcance de este trabajo, y se recoge como línea futura en la sección 7.2.

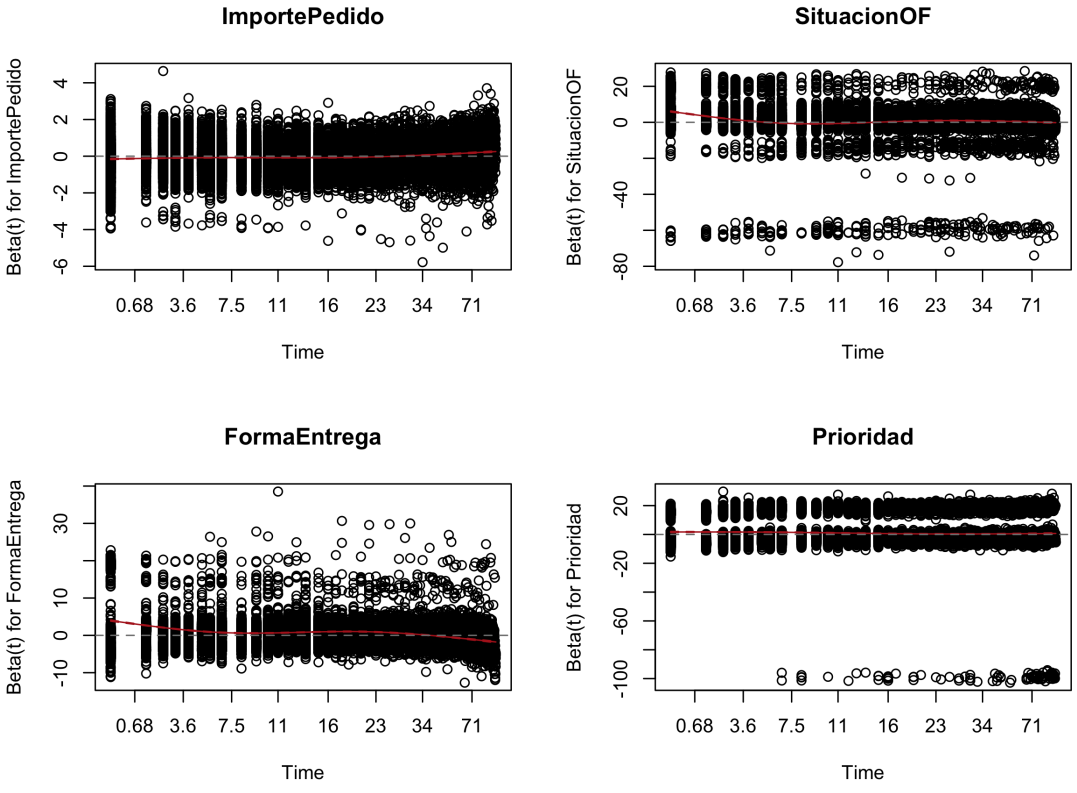

Figura 4.3: Residuos de Schoenfeld escalados frente al tiempo para cuatro de las variables con mayor evidencia de no proporcionalidad: `ImportePedido`, `SituacionOF`, `FormaEntrega` y `Prioridad`.

4.2.3. Cox con splines penalizados

Se ajusta una segunda variante, sustituyendo el efecto lineal de las variables continuas por splines penalizados. Los efectos parciales sugieren algunas no linealidades: `ImportePedido` decae más de lo que un efecto lineal predeciría, `AnticipacionSolicitada` cae de forma bastante pronunciada, y `tiempo_min` no muestra ninguna tendencia clara. El C-índice sube a 0,8199, una mejora de 0,0049 sobre el Cox lineal: las no linealidades existen, pero su efecto sobre la capacidad predictiva es más bien modesto. Estas dos cifras están calculadas sobre los datos de entrenamiento y solo son comparables entre sí.

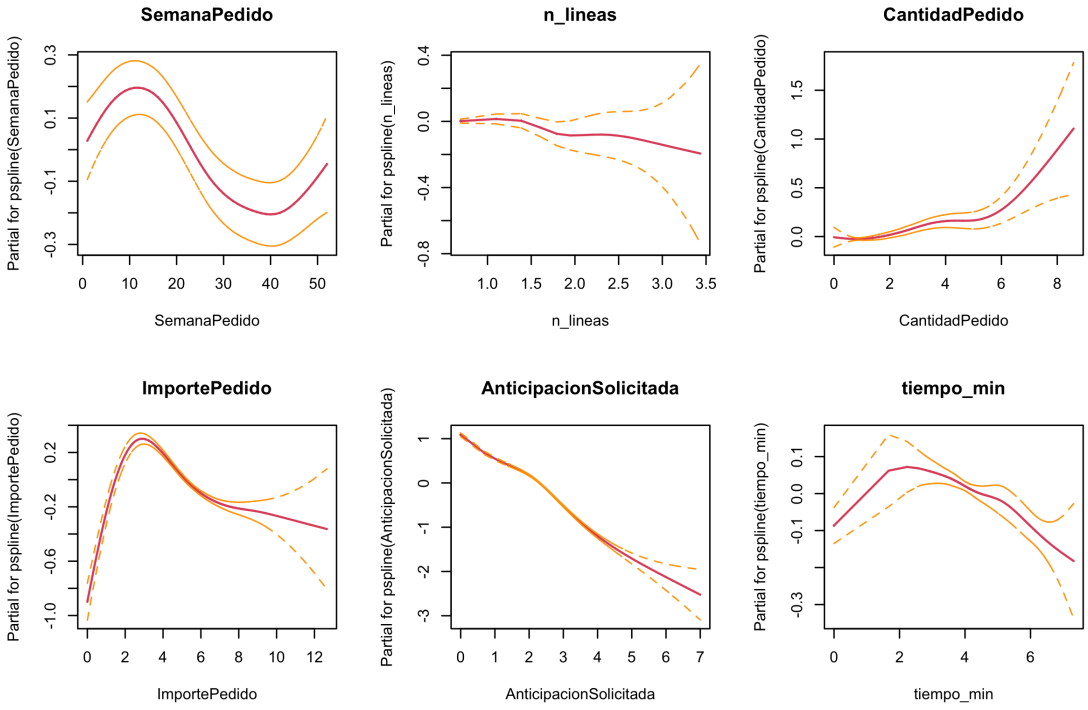

Figura 4.4: Efectos parciales de las variables continuas en el Cox con splines penalizados.

Esa mejora de 0,0049 viene a decir que relajar la linealidad, manteniendo la aditividad, apenas aporta nada. No dice nada, en cambio, sobre la aditividad en sí misma, de la que ya se ocupan las secciones 4.3 y 4.4.

4.3. Random Survival Forest

Se ajustan las dos variantes del capítulo 3, con los mismos hiperparámetros: 1 000 árboles, mínimo 30 observaciones por hoja, 10 puntos de corte aleatorios por variable en cada nodo, y `mtry` en su valor por defecto, que aquí resulta ser 6.

- RSF «fair» (`dat_rsf_fair`, 26 variables, los 27 689 pedidos del Cox): C-índice OOB 0,5783.
- RSF «completo» (`dat_rsf_completo`, 32 variables, los 27 732 pedidos): C-índice OOB 0,5751.

Las dos cifras salen casi iguales y las dos quedan muy por debajo del Cox. Darle al bosque todas las variables y todos los pedidos no le sirve de nada: el C-índice no sube, de hecho baja tres milésimas. Con esto quedan descartadas las dos causas que se apuntaban como candidatas en el capítulo 3: ni el número de variables ni el conjunto de pedidos explican la diferencia.

4.3.1. Diagnóstico del bajo rendimiento del bosque

La diferencia con el Cox es demasiado grande como para atribuirla sin más a la estructura del proceso. Un bosque suele aproximar bastante bien una función aditiva: peor que el modelo que la asume desde el principio, pero por un margen pequeño, no por casi veinticuatro puntos de C-índice. Una brecha de este tamaño apunta más bien a la configuración del método o a cómo se representan los datos, no tanto al proceso en sí.

La explicación que no se ha podido descartar tiene que ver con cómo llegan las variables categóricas al bosque. Por coste computacional se codifican como números enteros antes de ajustar el modelo: `LineaNegocio` pasa a ser un número del 1 al 8, `DescripcionCNAE` del 1 al 25, y así sucesivamente, siguiendo el orden alfabético de sus niveles. El problema es que un árbol solo puede partir una variable numérica con un umbral, del tipo «¿es menor que c?». Con `LineaNegocio` codificada del 1 al 8, el árbol puede separar $\{1, 2, 3\}$ de $\{4, \ldots, 8\}$, pero no puede aislar juntas, por ejemplo, las categorías 2 y 7 frente al resto, aunque esas dos se comporten de forma parecida en la realidad. El orden alfabético no tiene nada que ver con lo rápido o lento que factura cada categoría, así que en la práctica se está obligando al árbol a cortar por un criterio que no significa nada. Y esto no es anecdótico: dieciocho de las treinta y dos variables llegan así al bosque.

Lo correcto habría sido entregarlas como factores nominales, dejando que el algoritmo probara particiones arbitrarias de sus niveles. Se intentó y no fue posible: cada intento provocaba la caída completa de la sesión de R, sin ningún mensaje de error recuperable, y el fallo se reprodujo incluso reduciendo el número de árboles, desactivando la paralelización, agrupando los niveles menos frecuentes y probando con solo una fracción de los pedidos y unas pocas variables.

Este trabajo, por tanto, no puede determinar si el Random Survival Forest rendiría mejor con la codificación correcta, ni descartar del todo que la cifra fuera de bolsa esté afectada por alguna otra causa que no se ha llegado a identificar. La cifra que se reporta es la que sale con codificación entera, y hay que leerla con esa salvedad; la sección 6.4.2 lo recoge como limitación. Lo que sí deja claro el ajuste es que el bosque reconoce cuál es la variable dominante (`AnticipacionSolicitada` encabeza también su importancia de variables, igual que encabeza los coeficientes del Cox), pero no consigue combinarla bien con el resto de predictores. En este problema, con esta codificación, sumar linealmente varias señales moderadas funciona mejor que ir partiendo el espacio de forma recursiva; es una observación sobre cómo rinden estas dos familias de modelos en este caso concreto, no una conclusión sobre cómo es el proceso en el fondo.

4.3.2. Importancia de variables

El VIMP, calculado con 100 árboles sobre `dat_rsf_fair`, deja un resultado bastante claro: la variable `AnticipacionSolicitada` domina con 0,0560, más de seis veces la siguiente (`Anticipo`, con 0,0089). El resto ya está por debajo de 0,005: `n_lineas` con 0,0043, `TipoRotulacion` con 0,0033, `TipoAlbaran` con 0,0029, `tiempo_min` con 0,0023 y `HojaAlmacen` con 0,0019 cierran el grupo de las que todavía aportan algo. Esto coincide con el Cox: `AnticipacionSolicitada` es la única variable con un peso claramente dominante en los dos rankings a la vez, calculados de dos formas completamente distintas.

El resto del ranking es difícil de leer como una jerarquía con sentido. De las 26 variables, más de la mitad tienen un VIMP indistinguible de cero, o incluso negativo, y ahí se cuelan algunas que el Cox sí identifica como relevantes: `ImportePedido`, `SituacionOF` y `Prioridad` quedan fuera de las quince primeras, a pesar de estar entre los términos claramente significativos del Cox. Que el VIMP aplane casi todas las variables encaja con el diagnóstico de la sección anterior. Y conviene recordar también la cautela de la sección 3.2.5: cuando varias variables comparten información, barajar solo una apenas empeora el modelo y la VIMP tiende a repartirse entre ellas, así que un valor bajo no siempre significa que la variable no aporte nada.

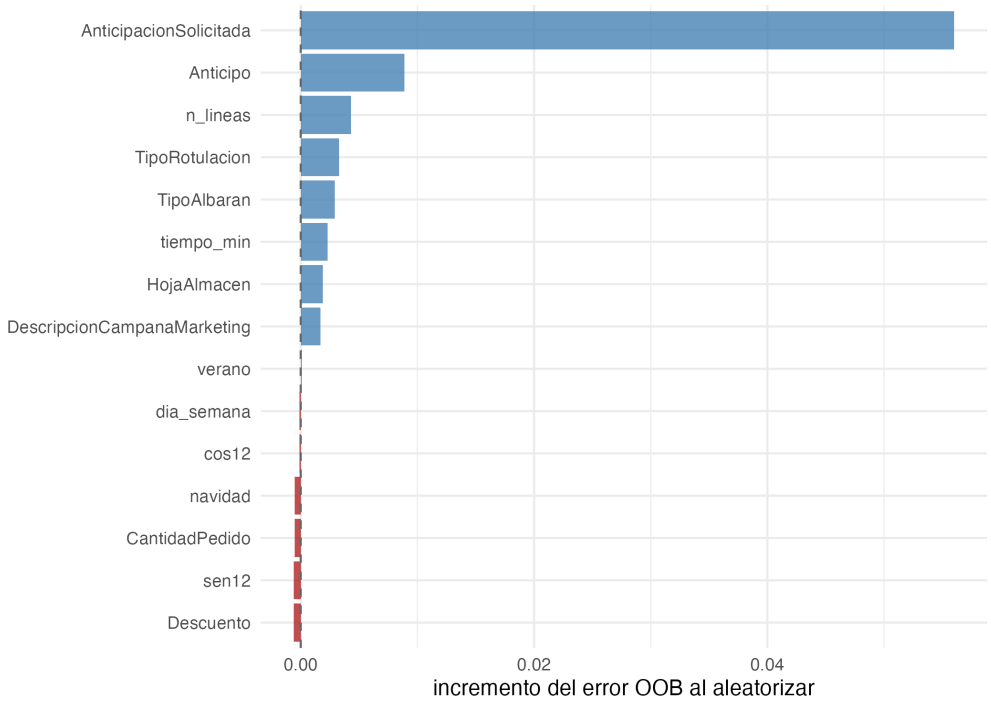

Figura 4.5: Importancia de variables por permutación (VIMP) en el RSF «fair».

4.4. Gradient Boosting

El modelo se ajusta con `glmboost` sobre una muestra de 5 000 pedidos de `dat_cox`, porque `glmboost` escala mal a medida que crece el tamaño de la muestra. La matriz de diseño se construye una sola vez sobre todo `dat_cox` y luego se reparte por filas, de modo que entrenamiento y predicción usan siempre el mismo número de columnas. El número de iteraciones se elige con `cvrisk()` y validación cruzada de 5 particiones. El C-índice, evaluado sobre los 22 689 pedidos que se quedaron fuera de la muestra de ajuste, es 0,8151. Como la unidad de análisis es el pedido, los de entrenamiento y el resto son conjuntos disjuntos, así que esta cifra está validada de verdad fuera de muestra y se puede comparar con la del Cox en validación cruzada (0,8136).

Que esta variante salga tan parecida al Cox es lo esperable. Como se explicó en la sección 2.6, `glmboost` es un boosting por componentes con aprendices lineales: en el fondo ajusta un modelo lineal y aditivo, el mismo tipo de modelo que el Cox pero por una vía de estimación distinta. Su interés aquí es doble. Por un lado confirma que el buen resultado del Cox no depende de cómo se haya estimado, y que la selección automática de variables llega a un modelo equivalente al del Cox con todas las variables dentro. Por otro, no aporta ninguna evidencia sobre si hay interacciones, porque por construcción no puede detectarlas.

Hay un segundo motivo por el que esta coincidencia es informativa. El boosting con aprendices lineales llega a 0,8151 sobre los mismos pedidos y con las mismas 26 variables con las que el bosque se queda en 0,5783. Los dos son procedimientos de aprendizaje automático con selección implícita de variables, así que la diferencia entre ellos no está en lo sofisticado del algoritmo, sino en cómo combinan los predictores. Esto respalda la lectura de la sección 4.3.1.

La importancia por frecuencia de selección vuelve a poner a `AnticipacionSolicitada` en primer lugar (0,198), diez veces por encima de la siguiente. El resto es una cola larga de contribuciones pequeñas: un nivel de `DescripcionCNAE` (0,021), el nivel «sin orden de fabricación» de `SituacionOF` (0,018), un nivel de `FormaEntrega` (0,014) y `TipoContabilidad` (0,012). Es la tercera vez que `AnticipacionSolicitada`

sale como la señal más fuerte, y cada vez calculada de una forma distinta. La sección 6.2 se detiene en qué significa esto.

4.5. Comparación conjunta

Tabla 4.2: Comparación de todos los modelos ajustados, con el tipo de validación de cada cifra.

Modelo	C-índice	Validación	Configuración
Cox con splines	0,8199	Entrenamiento	Splines penalizados en las 6 variables continuas
Gradient Boosting, aprendices lineales	0,8151	Fuera de muestra	`glmboost`, $n = 5\,000$, $\nu = 0{,}1$, `mstop` por `cvrisk`
Cox lineal	0,8150	Entrenamiento	26 variables, `ties = .ᵉᶠron"`
Cox lineal	0,8136	Val. cruzada (5)	Ídem
Random Survival Forest «fair»	0,5783	Fuera de bolsa	26 variables, 27 689 pedidos, 1 000 árboles, `mtry = 6`
Random Survival Forest «completo»	0,5751	Fuera de bolsa	32 variables, 27 732 pedidos, ídem

Solo son comparables entre sí las cifras obtenidas con el mismo procedimiento de validación. La comparación que de verdad importa es la de las dos validadas fuera de los datos de ajuste: el Cox en validación cruzada (0,8136) y el boosting fuera de muestra (0,8151).

Los resultados se agrupan en dos bloques bastante claros. El Cox en sus dos variantes y el Gradient Boosting se quedan en torno a 0,815, con apenas seis milésimas de diferencia entre el mejor y el peor del grupo. Las dos variantes del Random Survival Forest, en cambio, se quedan en torno a 0,577, a casi veinticuatro puntos de distancia.

El Cox resuelve el problema de la empresa con una capacidad discriminante razonable y estable: 0,8136 en validación cruzada frente a 0,8150 en entrenamiento. El Gradient Boosting llega, por una vía de estimación completamente distinta y con selección automática de variables, prácticamente al mismo sitio; que dos procedimientos tan distintos coincidan sugiere que el buen resultado del Cox no es un artefacto de cómo se haya estimado, y que tampoco hay mucho margen que ganar seleccionando variables sobre el conjunto disponible.

Y queda lo que no se puede afirmar. El razonamiento de «los modelos que no asumen aditividad no mejoran al Cox, luego el proceso es aditivo» no se sostiene, por dos motivos independientes. El único método sin restricción de aditividad que se ha ajustado, el Random Survival Forest, rinde tan por debajo que su resultado no puede leerse como evidencia sobre cómo es el proceso. Y el Gradient Boosting usa aprendices lineales, así que asume aditividad desde el principio, y que coincida con el Cox no aporta nada sobre esta cuestión.

Con lo que hay, el modelo de Cox ofrece el mejor equilibrio entre rendimiento e interpretabilidad: capacidad discriminante alta y estable, razones de riesgos que la empresa puede interpretar, y ningún método alternativo de los probados lo mejora. Que sea realmente superior a las alternativas habría que confirmarlo con una validación homogénea, que aplique el mismo procedimiento de partición y las mismas métricas a todas las familias comparadas. Que el proceso sea además aditivo es una hipótesis razonable y compatible con lo que se ha visto, pero este trabajo no lo demuestra.

4.6. Evaluación del modelo seleccionado

4.6.1. Evaluación interna sobre la muestra de ajuste

Las tres métricas que siguen se calculan sobre los mismos datos con los que se entrenó el modelo. Así que miden rendimiento aparente, y no permiten saber cómo se comportaría el modelo con pedidos

que todavía no ha visto.

Tabla 4.3: Rendimiento aparente: Brier Score en los horizontes de interés e Integrated Brier Score sobre la muestra de ajuste.

Horizonte	**Error (modelo nulo)**	**Error (Cox)**	$\mathbf{R^2_{Brier}}$
$t = 7$ días	0,2237	0,1206	0,4609
$t = 14$ días	0,2487	0,1421	0,4286
$t = 30$ días	0,1695	0,1122	0,3381
IBS (hasta el percentil 90)	0,1768	0,1104	0,3754

El Cox reduce el error frente al modelo nulo en torno a un 43–46 % a 7 y 14 días, y un 34 % a 30 días. El Integrated Brier Score, calculado hasta el percentil 90 de `tiempo_obs`, da una reducción del 37,5 %. Una posible explicación de por qué el error crece con el horizonte, aunque no se puede contrastar con estos datos, es que cuanto más lejos se proyecta la predicción más pesan factores que las 26 variables no recogen. El cálculo con `pec` necesita ajustar el modelo con la matriz de diseño explícita, de la que se excluye `TipoAlbaran` por dar lugar a un término linealmente dependiente del resto (apéndice B); el C-índice de ese ajuste coincide con el del modelo completo, así que esta exclusión no cambia la lectura de la tabla.

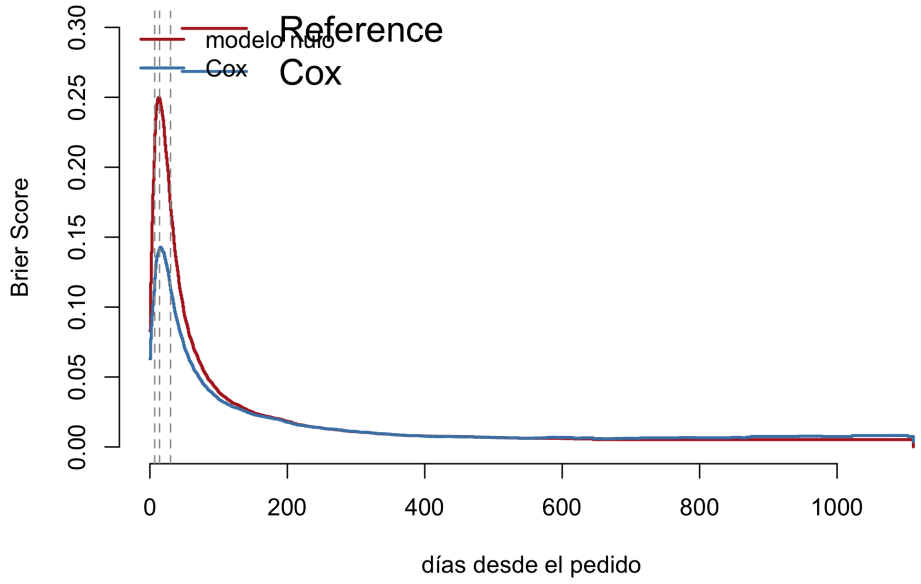

Figura 4.6: Brier Score del modelo de Cox y del modelo nulo a lo largo del horizonte.

Con $\tau = 30$ días, el C-índice de Uno es $C_\tau = 0{,}8131$, frente a 0,8150 de Harrell sobre la muestra de ajuste: una diferencia de 0,0019. Estas dos cifras se comparan de forma descriptiva: que estén tan cerca sugiere que la censura no está distorsionando de forma apreciable la cifra de Harrell, sin que eso implique adoptar ningún umbral concreto.

La pendiente aparente de calibración, ajustada sobre el predictor lineal del Cox, sale 1,0000. Este resultado es un poco redundante y no debería presentarse como una validación independiente: el

predictor lineal y el modelo de calibración usan el mismo `dat_cox` con el que ya se entrenó el Cox, así que el valor sale cerca de 1 casi por construcción (Van Houwelingen 2000).

4.6.2. Comparación con la muestra de pedidos ya facturados

El conjunto de hitos no intervino en ningún momento del ajuste, y además todos sus pedidos están facturados. Eso permite coger el Cox ya ajustado y aplicarlo directamente a esos pedidos, sin reestimar nada, y comparar lo que predice con lo que realmente pasó (Royston y Altman 2013).

Aquí hay que tener cuidado con una cosa: esta muestra solo tiene pedidos que terminaron facturándose, mientras que el modelo se ajustó sobre una población que incluye un 3,2 % de pedidos censurados, que previsiblemente son los más lentos. Como las dos poblaciones no son idénticas, lo que sigue es una comparación descriptiva y no una validación externa en sentido estricto: parte de cualquier diferencia que aparezca puede deberse a esa distinta composición de las muestras y no a un fallo del modelo.

Sobre `dat_enriquecido` se calculan, con `survfit(cox_basico, newdata = ...)`, las probabilidades predichas para cada uno de los 7 900 pedidos que aparecen en las dos fuentes a la vez.

Tabla 4.4: Proporción real frente a probabilidad media predicha en la muestra de pedidos finalmente facturados.

Horizonte	Proporción real $\bar{Y}_k$	Probabilidad media predicha $\bar{\hat{P}}_k$	Diferencia
7 días	0,336	0,372	+0,036
14 días	0,527	0,581	+0,054
30 días	0,779	0,808	+0,029

La diferencia sale positiva y de un tamaño parecido en los tres horizontes: el modelo tiende a sobreestimar la probabilidad de facturación, entre 3 y 6 puntos porcentuales por encima de lo que pasa en realidad. No es un sesgo enorme, pero sí es consistente en la misma dirección las tres veces, lo que apunta más a un sesgo real que a ruido de muestreo.

Esta comparación resume la calibración en un único número por horizonte, y eso puede esconder que el modelo calibre mejor en unos rangos de probabilidad que en otros. Para comprobarlo se ordenan los pedidos en diez deciles según $\hat{P}_{30}$ y, dentro de cada decil, se compara la probabilidad media predicha con la proporción real de pedidos que facturaron antes de 30 días.

Tabla 4.5: Comparación por deciles de $\hat{P}_{30}$ en la muestra de pedidos finalmente facturados.

Decil	Predicho	Real	n
1	0,396	0,199	790
2	0,585	0,416	790
3	0,688	0,654	790
4	0,765	0,800	790
5	0,833	0,870	790
6	0,891	0,948	790
7	0,944	0,957	790
8	0,980	0,970	790
9	0,998	0,984	790
10	1,000	0,990	790

Esta tabla matiza un poco la conclusión anterior. En los deciles bajos (1 y 2) el modelo sobreestima con bastante claridad: en el decil 1 predice de media un 0,396 cuando la realidad es 0,199, prácticamente

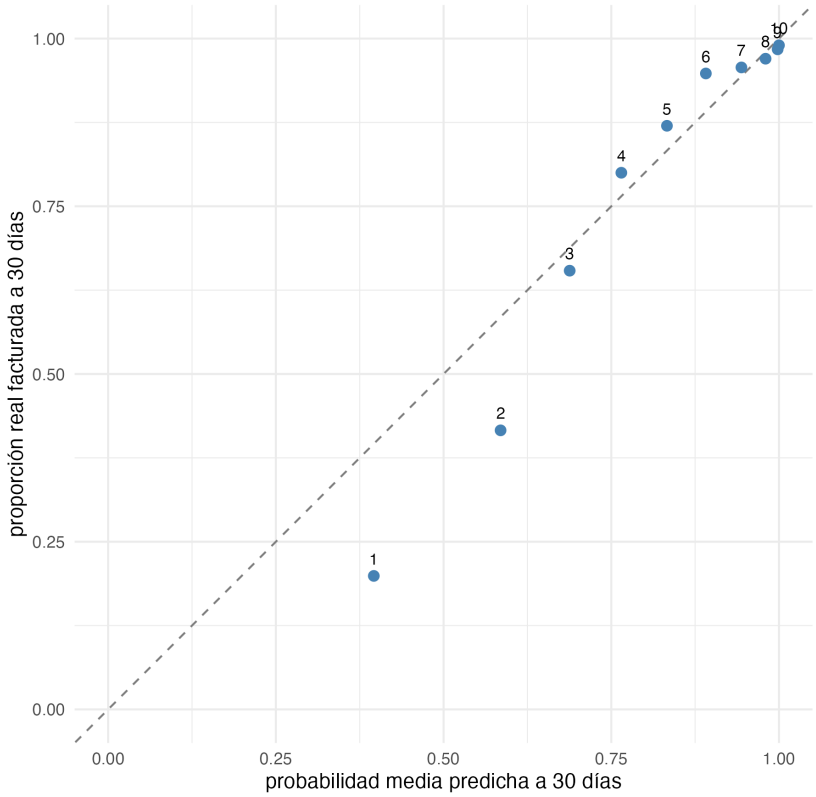

Figura 4.7: Comparación entre probabilidad predicha y proporción observada, por deciles de $\hat{P}_{30}$.

el doble. En los deciles intermedios y altos pasa justo lo contrario, aunque de forma más moderada, hasta que las dos cantidades acaban convergiendo casi del todo en los deciles 8 a 10. Así que el sesgo positivo de la tabla 4.4 parece venir sobre todo de los pedidos a los que el modelo, de entrada, les asigna una probabilidad baja de facturar rápido.

Conviene no tomarse estos deciles como algo definitivo todavía: tienen un carácter más bien exploratorio y no deberían convertirse ya en umbrales operativos de alerta. Están calculados sobre una muestra que, por construcción, excluye los pedidos censurados, que son justamente los que un sistema de avisos tendría que detectar. Para fijar un umbral de verdad habría que repetir el análisis sobre una cohorte que incluya todos los pedidos elegibles, como se propone en la sección 7.2.

Esta observación conecta con la discusión sobre la censura de la sección 2.2: el decil 1 es justo donde se concentrarían los pedidos parecidos a los censurados, y resulta ser el peor calibrado de todos. Es compatible con la sospecha de que la censura no sea del todo no informativa, aunque esa sospecha, con los datos disponibles, no se puede llegar a contrastar.

Capítulo 5

Análisis de los hitos del proceso productivo

Este capítulo cambia de fuente para responder a algo que el conjunto principal no puede contestar: cuánto tiempo se lleva cada etapa intermedia del proceso, desde que se detecta la oportunidad comercial hasta que el pedido queda entregado y facturado.

5.1. Propósito y alcance

El fichero de hitos no tiene ningún pedido censurado, y por eso no sirve para entrenar un modelo de supervivencia: le faltan justo los pedidos más problemáticos. Aquí se usa para otra cosa: describir cuánto tarda cada etapa, sin meter ningún modelo de por medio. La otra utilidad de este fichero, comparar las probabilidades del Cox con lo que pasó de verdad, ya se trató en la sección 4.6.2.

Las duraciones que vienen a continuación se calculan solo con pedidos ya facturados, así que cuentan cómo va el proceso cuando llega a buen puerto. No dicen nada de los pedidos que se quedan por el camino, que son precisamente los que motivan el modelo de los capítulos anteriores. Con esa salvedad, lo que aporta este capítulo es algo que la empresa no tenía: una foto de cómo se reparte el tiempo entre las distintas etapas.

5.2. Duraciones por etapa

Se definen seis duraciones, cada una como la diferencia entre las fechas de dos hitos seguidos:

- **Comercial:** desde que se detecta la oportunidad hasta que se registra el pedido.
- **Planteamiento:** desde el pedido hasta que se cierra el diseño y la configuración del encargo.
- **Aprovisionamiento:** desde el primer pedido de compra a proveedores hasta que llega el último.
- **Intervalo entre la primera y la última orden de fabricación.**
- **Entrega:** desde el pedido hasta el primer albarán.
- **Administrativa:** desde el primer albarán hasta que se emite la factura.

Son las que aparecen marcadas sobre el ciclo de vida del pedido en la figura 1.2.

El percentil 90 es la columna que deja ver la cola larga de la etapa administrativa: su mediana es de solo 2 días, pero un 10 % de los pedidos tarda 14 días o más.

Antes de ponerse a interpretar las medianas conviene explicar por qué faltan tantos datos en algunas etapas. Todo apunta a que buena parte de esos valores ausentes no son fallos de registro, sino que sencillamente esa fase no existió para ese pedido: un pedido estandarizado, sin diseño a medida, no pasa por planteamiento, y uno que no necesita comprar nada a terceros no pasa por aprovisionamiento. Con esa lectura, que solo un 12,3 % de los pedidos necesite aprovisionamiento externo y un 40,9 %

Tabla 5.1: Duraciones por etapa del proceso productivo, en días.

Etapa	n válidos	% ausente	Mediana	P_{25}	P_{75}	P_{90}
Comercial	2 390	69,7	22	9	49	109
Planteamiento	3 232	59,1	5	3	8	18
Aprovisionamiento	969	87,7	8	4	16	31
Intervalo entre órdenes de fabricación	2 682	66,1	0	0	0	4
Entrega	7 897	0,0	8	2	21	38
Administrativa	7 900	0,0	2	1	5	14

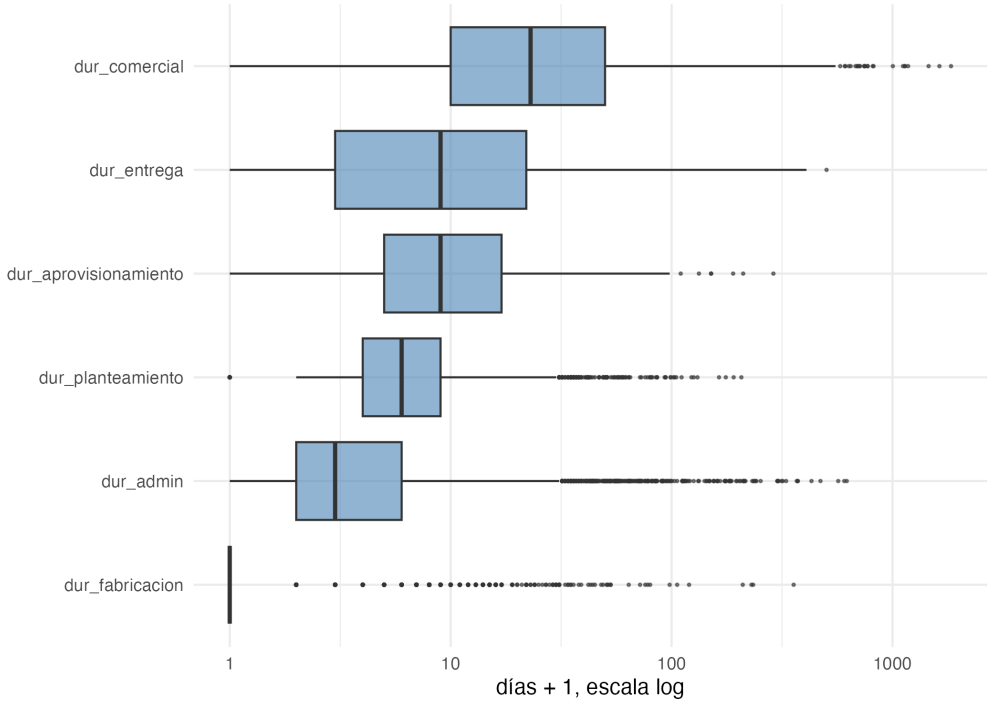

Figura 5.1: Distribución de las duraciones por etapa, en escala logarítmica.

pase por una fase de planteamiento explícita encaja con que buena parte de los productos son bastante estandarizados. Ahora bien, esa lectura no vale para todas las etapas por igual: como se verá en el apartado siguiente, en la etapa comercial la ausencia depende sobre todo de la línea de negocio, es decir, es una cuestión de cómo se registra, no de cómo es el pedido.

Hay dos etapas que merecen un comentario aparte. La primera es el intervalo entre la primera y la última orden de fabricación: la mediana y los dos cuartiles salen exactamente 0, con un 66,1 % de valores ausentes. Eso no quiere decir que fabricar no lleve tiempo, sino que, para la mayoría de los pedidos con al menos una orden registrada, la primera y la última caen el mismo día, lo más probable porque son pedidos con una sola orden, y ahí la diferencia `FechaUltimaOM` − `FechaPrimeraOM` sale cero por pura construcción. En el fondo, esta variable mide más si un pedido tuvo una o varias órdenes repartidas en el tiempo que la duración real de fabricar, y por eso se le ha cambiado el nombre en la tabla.

Para medir bien esta etapa harían falta dos marcas de tiempo que el ERP hoy no guarda: cuándo entra una orden de fabricación en el taller y cuándo se cierra. Es un cambio pequeño en el sistema de

información, solo dos campos por orden, y probablemente el que más información aportaría, porque fabricación es la única etapa del ciclo que este trabajo no puede observar. Se retoma como línea futura en la sección 7.2.

La segunda etapa que merece comentario es la administrativa, la única que, junto con la de entrega, está disponible para prácticamente todos los pedidos. Su mediana es de 2 días, con un P_{75} de 5, bastante más corta que la comercial (22 días de mediana) o que la de aprovisionamiento (8 días). A primera vista, parece que el cuello de botella administrativo no sería, para la mayoría de los pedidos, la etapa más lenta del proceso. Pero la mediana esconde la cola larga que sí deja ver el P_{90}, y de eso trata la siguiente sección.

5.2.1. Sobre la etapa comercial

La etapa comercial es, con diferencia, la más larga de todo el ciclo: 22 días de mediana, con un P_{75} de 49 y un P_{90} de 109. Y transcurre entera antes de que arranque el reloj del modelo: cuando el pedido se registra en el ERP, esos 22 días de mediana ya han pasado. El modelo de supervivencia no la cubre, ni falta que le hace, porque su punto de partida es `FechaPedido`. Tampoco es una etapa comparable con las demás en otro sentido: mide cuánto tarda una oportunidad en convertirse en pedido, algo que depende sobre todo de la decisión del cliente y no del proceso productivo de la empresa.

El 69,7 % de ausencia en esta etapa hay que leerlo de forma distinta al resto. No se reparte por igual entre líneas de negocio: Protección Solar registra la oportunidad comercial en la mayoría de sus pedidos, mientras que en el resto de líneas eso pasa solo en una minoría, y en MOHEVA casi nunca. Es decir, la ausencia no parece cosa del azar, sino de que cada línea de negocio gestiona y registra la venta a su manera. Así que los 22 días de mediana reflejan, sobre todo, el comportamiento de Protección Solar, más que el del conjunto de pedidos en general.

Con esa cautela por delante, si la empresa quisiera acortar el tiempo total desde que se detecta una oportunidad hasta que se cobra, el dato solo permite decir con cierta confianza que ese margen de mejora existe en Protección Solar. Extenderlo al resto de líneas exigiría primero mejorar cómo registran ellas mismas la oportunidad comercial.

Capítulo 6

Discusión y limitaciones

6.1. Interpretación del rendimiento comparado de los modelos

El resultado más llamativo del capítulo 4 es que el modelo más simple de los tres, el Cox, rinde tan bien como los otros dos. Antes de sacar conclusiones apresuradas conviene pararse a pensar qué significa esto realmente.

Una lectura fácil sería decir que el aprendizaje automático está sobrevalorado y que el modelo clásico gana sin más. Pero no es eso lo que dicen los datos: como se vio en la sección 4.3.1, el bosque tuvo un problema técnico con la codificación de las variables que lo dejó en clara desventaja. La comparación, sencillamente, no fue justa.

Otra lectura fácil sería pensar que el mal resultado del bosque confirma que el proceso es aditivo. Pero un bosque puede rendir mal por mil motivos que no tienen nada que ver con eso, y dar por buena esa conclusión sería, en el fondo, dar por probado justo lo que se quería comprobar. En la sección 4.3 ya se descartó que la diferencia viniera del número de variables o de pedidos; lo que no se ha podido descartar es el problema de la codificación.

Así que, sobre si el proceso es aditivo o no, este trabajo no puede pronunciarse. Lo que sí puede decir es que el Cox resuelve bien el problema de la empresa, que ningún otro método probado lo mejora, y que el Gradient Boosting llega prácticamente al mismo sitio por un camino distinto. Que no existan interacciones aprovechables no se puede afirmar, porque el único método capaz de detectarlas no compitió en igualdad de condiciones. En resumen: el Cox funciona y nada de lo probado lo supera, y con eso basta para justificar la herramienta que se propone en el capítulo 7.

Y esto no es raro, además. En predicción clínica, Christodoulou et al. (2019) revisan muchas comparaciones publicadas entre regresión logística y métodos de aprendizaje automático y no encuentran que estos últimos ganen de forma sistemática. La explicación que dan encaja con lo que ha pasado aquí: que un método flexible gane no depende del método en sí, sino de si el problema tiene realmente estructura que un modelo simple no pueda capturar. Y avisan de algo que también vale para este trabajo: muchas de esas comparaciones están sesgadas porque no se ajustan las dos familias en igualdad de condiciones. Por eso en la sección 7.2 se propone, como primer paso pendiente, hacer una validación homogénea de verdad.

6.2. El plazo solicitado por el cliente

`AnticipacionSolicitada` sale la primera en los tres rankings de importancia: en el Cox por valor p, en el VIMP del bosque y en la frecuencia de selección del boosting. Los tres se calculan de formas totalmente distintas, así que es difícil pensar que sea casualidad o un efecto raro de un solo método. Hasta el bosque, con lo mal que rinde en general, la coloca en primer lugar.

Merece la pena pararse a pensar si esto es una señal real o si en realidad se está haciendo trampa de alguna forma. La variable recoge el margen que pide el cliente entre que hace el pedido y la fecha en la que quiere recibirlo. Es un dato que ya se conoce en el momento en que el pedido entra al sistema, y no depende de nada que pase después, así que no tiene el problema que sí tenían las variables que se quitaron en la sección 3.2.3. Tampoco es una fecha de entrega real ni un compromiso de la empresa, es simplemente lo que pide el cliente, así que se puede usar como predictor sin miedo a estar haciendo trampa.

Lo que viene a decir todo esto es que la señal más fuerte del proceso no sale del taller ni de administración, sino de lo que pide el propio cliente: los tiempos de facturación se ajustan bastante bien al margen que el cliente da al hacer el pedido.

Si el plazo que pide el cliente predice mejor que cualquier otra cosa, entonces la palanca para mejorar los tiempos de facturación igual no está tanto en el taller como en cómo se negocia y se registra ese plazo en el momento de la venta. Esto es una hipótesis, no algo que este trabajo pueda demostrar (haría falta poder intervenir, no solo observar), y hay que decirlo con cuidado: también podría ser al revés, que los clientes pidan plazos largos precisamente porque ya intuyen que su encargo va a ser lento. Lo único que se puede afirmar con los datos disponibles es que las dos cosas van muy de la mano, no cuál es la causa de cuál.

6.3. Comparación con el trabajo de referencia

En la sección 1.4.3 se presentó el trabajo de Smirnov (2016) como el antecedente más parecido a este: mismo tipo de problema, misma comparación entre Cox y Random Survival Forest, pero con un resultado justo al revés, con el bosque ganando claramente al Cox.

Los dos resultados se pueden entender juntos, sin embargo. Smirnov cuenta con el historial de pago de cada cliente, una variable que resume su comportamiento pasado y que, por su naturaleza, interactúa con el resto: lo que significa un importe alto o un plazo largo depende de qué cliente sea. Un modelo aditivo no puede recoger eso, y ahí es donde el bosque saca ventaja. En Toldos Gómez no hay nada parecido: los pedidos se describen solo por sus propias características, sin ninguna variable que resuma cómo se ha comportado el cliente antes.

Visto así, los dos trabajos no se contradicen: el bosque gana cuando hay una variable relacional que interactúa con las demás, y no gana cuando no la hay. Esta idea es más útil que cualquiera de los dos resultados por separado, porque permite, ante un problema nuevo, hacerse una idea de si merece la pena la complejidad extra con solo mirar qué variables hay disponibles. Aun así, hay que añadir la misma cautela de antes: aquí el bosque no compitió en igualdad de condiciones, así que la comparación con Smirnov es sugerente pero no una prueba.

Todo esto apunta a que la mejora con más recorrido no sería usar un modelo más sofisticado, sino conseguir una variable más: si el ERP permitiera construir un historial de cliente, la comparación entre modelos podría salir distinta, y de paso el propio Cox mejoraría.

6.4. Limitaciones

6.4.1. Limitaciones de los datos

La censura podría no ser del todo inocente. Tanto Kaplan-Meier como el Cox necesitan asumir que la censura no aporta información, y eso no se puede comprobar del todo con lo que hay: haría falta ver justo lo que no se ve. En la sección 4.6.2 sale un indicio que va en esa línea: el decil de probabilidad más baja, que agrupa a los pedidos más parecidos a los censurados, es también el peor calibrado. Es solo un indicio, pero apunta en la dirección que cabría esperar si la censura sí influyera. Con un 3,2 % de censura tampoco es un problema que pueda ser muy grande. La forma de comprobarlo de verdad sería una validación temporal, que se propone en la sección 7.2.

Los hitos intermedios no se pueden usar como predictores. El ERP solo los guarda cuando el pedido ya se ha cerrado, así que no están disponibles cuando hace falta predecir. Usarlos solo para los pedidos ya facturados metería el mismo sesgo que ya se descartó en la sección 1.3. Es una pena, porque probablemente serían informativos: saber que un pedido lleva tres semanas esperando material diría bastante sobre cuándo va a facturarse.

No hay historial del cliente. Cada pedido se describe solo por sus propias características, y no hay ninguna variable que resuma cómo se ha comportado ese cliente antes. Esto pesa en el resultado principal del trabajo: la comparación entre modelos se hace sobre un conjunto que, por cómo está construido, favorece a los modelos aditivos.

El indicador de imputación se quedó fuera. Los valores ausentes de `AnticipacionSolicitada` y de `tiempo_min` se rellenan con la mediana (sección 3.4). Se llegó a crear un indicador de si un valor estaba imputado o no, pero al final no entró en el modelo. Como `AnticipacionSolicitada` es la variable con más peso en los tres modelos, no tener ese indicador impide saber si lo que importa es el plazo en sí o el hecho de que no se registrara. Queda pendiente.

La duración de fabricación no está bien medida. Lo que hay es el intervalo entre la primera y la última orden de fabricación, que no es lo mismo que el tiempo real que se tarda en fabricar. De hecho, es la única etapa de todo el ciclo sobre la que este trabajo no puede decir nada fiable.

Al agregar por pedido se pierde información de línea. Al reducir cada pedido a una sola fila se pierde detalle que solo existe a nivel de línea. Un pedido con cinco artículos distintos y otro con cinco unidades del mismo artículo acaban con el mismo valor en `n_lineas`, y la categoría dominante por importe no siempre describe bien un pedido variado. Parte de esto se podría recuperar sin perder la unidad de análisis correcta, y se recoge como línea futura en la sección 7.2.

Aparte merece mención `CantidadPedido`: al sumar, mezcla unidades de cosas distintas (cinco toldos más dos complementos más una instalación dan «ocho», un número que en realidad no significa nada). Se ha mantenido por seguir con el planteamiento original, pero su interpretación es floja, y de hecho su importancia en los tres modelos sale, como cabía esperar, muy baja.

6.4.2. Limitaciones metodológicas

El supuesto de riesgos proporcionales no se cumple. El contraste de la sección 4.2.2 lo rechaza con claridad. Esto no invalida el modelo para predecir: el C-índice y las probabilidades siguen funcionando aunque los cocientes de riesgos cambien con el tiempo. Lo que sí impide es leer los coeficientes de la tabla 4.1 como algo fijo: cuando se dice que la línea MOHEVA factura casi once veces más rápido, esa cifra es un promedio a lo largo del tiempo, y el efecto real podría ser mayor al principio y menor después, o al revés. Se podría tratar estratificando, con interacciones con el tiempo o con los modelos flexibles de Royston y Parmar (2002), pero no se ha hecho aquí porque el objetivo es predecir y el coste en interpretabilidad sería alto. Es una decisión que se puede discutir.

El Gradient Boosting se ajusta con menos datos. Por coste computacional se entrena solo con 5 000 de los 27 689 pedidos. El C-índice sí se calcula fuera de esa muestra, así que la cifra es válida y comparable con la validación cruzada del Cox, pero sigue siendo una limitación entrenar con menos de la quinta parte de los datos disponibles.

El RSF no se pudo ajustar con la codificación correcta. Es la limitación más seria del trabajo, y la que más pesa en la conclusión principal. Las variables categóricas llegan al bosque codificadas como números, siguiendo el orden alfabético de sus niveles, lo que limita mucho los cortes que el árbol puede hacer. Se intentó arreglarlo y no hubo manera: todos los intentos hacían que R se cerrara sin dar

ningún error que ayudara a entender por qué. El resultado es que la comparación de la tabla 4.2 no está hecha en igualdad de condiciones, y lo que rinde el bosque no se puede tomar como una prueba sobre cómo es realmente el proceso.

No se han optimizado bien los hiperparámetros del bosque. No se ha buscado sistemáticamente el mejor `mtry`, `nodesize` ni número de árboles: el problema de la codificación es anterior y mucho más grave, así que no tenía sentido afinar hiperparámetros sobre una representación de los datos que ya sabíamos que era mala. Además, usar la estimación fuera de bolsa una y otra vez para elegir configuraciones tiende a dar resultados demasiado optimistas, así que si se hiciera esa búsqueda algún día, habría que apoyarla en una partición aparte.

La evaluación interna no es una validación de verdad. El Brier Score, el Integrated Brier Score y el C-índice de Uno están calculados sobre la misma muestra de ajuste, así que miden solo rendimiento aparente. La pendiente de calibración sale prácticamente 1 casi por definición (Van Houwelingen 2000). Habría sido mejor repetir todo esto con validación cruzada, y queda pendiente.

No se ha contrastado directamente si hay interacciones. El único método sin restricción de aditividad que se ha probado es el Random Survival Forest, y ya se ha explicado por qué su resultado no es fiable para esto. Faltan por hacer, y se recogen en la sección 7.2, un boosting con árboles como aprendiz base y un contraste de interacciones directamente dentro del Cox.

6.4.3. Limitaciones de validez externa

Todo lo que se concluye aquí vale para una empresa, un ERP y una ventana de tiempo concretos. No hay ninguna garantía de que el resultado se sostenga en otra empresa del sector, ni siquiera con datos parecidos. Como se argumentó en la sección 6.3, el resultado depende de qué variables hay disponibles, y eso es cosa del sistema de información de esta empresa en concreto, no del sector en general.

La ventana de datos va de enero de 2023 a abril de 2026, ya después de las fases más disruptivas de la pandemia sobre las cadenas de suministro, así que eso no debería ser aquí una fuente de heterogeneidad. La mediana de `tiempo_obs` por trimestre no cambia mucho a lo largo del periodo, se mueve entre 10 y 17 días sin ninguna tendencia clara. La única excepción es el último trimestre, 2026-T2, con una mediana de solo 7 días y bastantes menos pedidos que el resto, pero esto tiene una explicación sencilla: al cortar la extracción a mitad de ese trimestre, lo que queda registrado son sobre todo los pedidos más recientes y los que facturaron más rápido, mientras que los lentos de ese mismo tramo todavía no habían dado tiempo a aparecer como facturados. El resto del periodo no muestra nada raro.

Además, hay una señal de que el modelo no funciona igual de bien ni siquiera dentro de la propia empresa: en la sección 4.6.2 aparece una diferencia de entre 3 y 6 puntos al aplicarlo a la segunda fuente de datos. Parte de eso puede deberse a que las dos muestras no son exactamente iguales, pero de todas formas es un aviso de lo que cabría esperar al aplicar el modelo a datos nuevos.

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

El objetivo de este trabajo era construir un procedimiento que anticipara el tiempo hasta la facturación de los pedidos de Toldos Gómez S.L., sin tener que descartar los pedidos que todavía seguían abiertos en el momento de sacar los datos. Plantearlo como un problema de supervivencia permitió juntar en un mismo análisis los pedidos ya facturados y los censurados, y traducir el resultado en probabilidades de facturación a 7, 14 y 30 días, que tienen un significado directo para la gestión.

La decisión que más condiciona todo lo demás fue usar el pedido, y no la línea de artículo, como unidad de análisis: es lo que la empresa factura, y además es la única forma de que la validación cruzada mida lo que de verdad tiene que medir. El análisis no paramétrico mostró que buena parte de la facturación se concentra en las primeras semanas (un tercio de los pedidos a los 7 días, más de la mitad a los 14), pero también que queda una cola de pedidos que tarda bastante más.

De todas las variables disponibles, `AnticipacionSolicitada` salió una y otra vez como la más relevante, tanto en el Cox como en los dos métodos de aprendizaje estadístico, calculada cada vez de una forma distinta. Esto sugiere que el margen que pide el cliente resume buena parte de lo complejo que es el pedido y de las condiciones en las que entra a producción, y que la principal fuente de variabilidad en los tiempos está más en lo que pide el cliente que en el propio taller.

De los modelos probados, el Cox fue el que mejor equilibrio dio entre capacidad predictiva, estabilidad e interpretabilidad: un C-índice de 0,8136 en validación cruzada, una reducción del error frente a un modelo sin covariables de entre el 34 % y el 46 %, y apenas 0,0014 de diferencia entre entrenamiento y validación cruzada. Sus probabilidades se pueden usar directamente en una herramienta de seguimiento, y sus coeficientes ayudan a entender qué hace que un pedido facture más rápido o más lento. Aun así, hay que decir esto con cautela: que el Cox sea mejor que las alternativas habría que confirmarlo con una validación en igualdad de condiciones, porque el Random Survival Forest no llegó a competir de forma justa y el Gradient Boosting usado también asume aditividad.

El segundo conjunto de datos sirvió para describir las duraciones de las distintas etapas y mostró que la fase administrativa suele ser corta (2 días de mediana), mientras que la comercial, la de aprovisionamiento y la de entrega son más variables. Al comparar las probabilidades del modelo con lo que pasó de verdad en esa segunda muestra, se vio un sesgo hacia el optimismo, moderado pero constante, sobre todo en los pedidos a los que el modelo, de entrada, ya les daba poca probabilidad de facturar rápido.

En conjunto, el análisis de supervivencia parece una herramienta útil para convertir los datos del ERP en probabilidades con sentido para la gestión. Antes de usarlo en el día a día, quedan por resolver la validación temporal, la recalibración y la forma de incorporar información del proceso a medida que avanza.

7.2. Líneas futuras

7.2.1. Validación temporal externa

Lo primero que habría que hacer es una validación temporal de verdad: entrenar con pedidos de un periodo y evaluar con pedidos de meses posteriores, siempre que hayan tenido tiempo suficiente para cada horizonte. Esa muestra debería incluir todos los pedidos elegibles, no solo los que acabaron facturando, que es justo la limitación de la comparación de la sección 4.6.2. Ahí se podrían calcular, con el mismo procedimiento para los tres modelos, el C-índice, el Brier Score y las curvas de calibración. Si se viera una desviación sistemática, el modelo se podría recalibrar sin tener que reconstruirlo entero. De paso, esto también permitiría vigilar si el proceso cambia con el tiempo o si hay estacionalidad, y sería la manera de comprobar la sospecha sobre la censura informativa que queda abierta en la sección 6.4.1.

Y de paso resuelve también el problema metodológico que atraviesa todo el capítulo 4: si se usa la misma partición y las mismas métricas para los tres modelos, la comparación deja de depender de cómo se validó cada uno por separado y pasa a ser una comparación justa de verdad.

7.2.2. Predicción dinámica con los hitos productivos

La segunda línea sería convertir el modelo, que ahora es estático y solo mira el momento de entrada del pedido, en un sistema que se actualice sobre la marcha. Para eso habría que empezar a registrar en los pedidos abiertos los hitos que ahora solo se guardan al cerrar el pedido: inicio y fin de fabricación, compras, incidencias, carga del taller, disponibilidad de instaladores, estado del albarán. Con eso se podrían construir modelos que actualicen la probabilidad de facturación cada vez que cambia algo del pedido. De paso, convendría comparar esto con el Cox con coeficientes que cambian en el tiempo, con modelos paramétricos flexibles (Royston y Parmar 2002) y con métodos de aprendizaje automático, usando siempre las mismas particiones. Todo esto podría acabar integrado en Power BI como una probabilidad que se actualiza sola, con un nivel de alerta y las variables que explican el riesgo de retraso.

7.2.3. Registrar lo que falta

Las limitaciones de datos de la sección 6.4.1 se arreglan en el sistema de información, no en el modelo:

- **Historial de cliente.** Guardar, para cada cliente, un resumen de sus pedidos anteriores (cuántos, cuánto tardaron, con qué variabilidad), siempre usando solo información previa al pedido actual para no colar información del futuro. Según lo que se vio en la sección 6.3, es la mejora que más partido daría, y la que de verdad permitiría poner a prueba la idea de la aditividad.
- **Marcas de tiempo de fabricación.** Con solo dos campos por orden (cuándo entra al taller y cuándo se cierra) la etapa productiva pasaría a poder medirse.
- **Registrar los hitos en tiempo real**, no solo cuando se cierra el pedido, que es lo que hace falta para poder predecir de forma dinámica.

7.2.4. Poner a prueba la aditividad

El capítulo 4 deja sin resolver si hay interacciones que se puedan aprovechar. Quedan tres cosas por probar:

- Conseguir ajustar el Random Survival Forest con las variables categóricas como factores de verdad, no como números. El problema fue del entorno de cómputo, no del método, así que lo razonable sería intentarlo en otra máquina o con otra implementación: `ranger` permite el mismo tipo de modelo y trata los factores de otra manera. Solo entonces tendría sentido afinar los hiperparámetros del bosque.
- Probar `blackboost`, un boosting con árboles que mantiene la pérdida de Cox pero sin asumir aditividad. Es la comprobación más directa, porque no depende de que el bosque esté bien configurado.

- Meter interacciones directamente en el Cox: probar por separado las que tienen sentido a priori (`AnticipacionSolicitada` × `LineaNegocio`, `AnticipacionSolicitada` × `SituacionOF` e `ImportePedido` × `LineaNegocio`) y compararlas con el Cox lineal. Con $n = 27\,689$ el valor p casi seguro saldrá significativo aunque la interacción no importe mucho, así que lo que habría que mirar de verdad es si sube el C-índice en validación cruzada.

7.2.5. Otras ideas

También quedan pendientes: ajustar un modelo paramétrico de tiempo de fallo acelerado y compararlo con el Cox (sección 2.4); tratar el incumplimiento de proporcionalidad con estratificación, interacciones con el tiempo o modelos flexibles; y enriquecer cómo se representa el pedido con variables que ahora se pierden al agregar, como el número de tipos de artículo distintos o si el encargo incluye instalación o rotulación. También se podría aplicar control estadístico de procesos a las etapas con registro completo, sobre todo la administrativa y la de entrega, con cartas de control (Montgomery 2013) como las que ya implementan paquetes de R descritos en Flores et al. (2021), y hacer un análisis de capacidad frente a los plazos que la empresa se compromete a cumplir, algo que aquí no se ha podido abordar por no tener esos compromisos en forma de dato.

7.2.6. Cómo se llevaría esto a la empresa

El Cox ajustado se resume, en el fondo, en dos cosas: el vector de coeficientes $\hat{\beta}$ y la función de riesgo acumulada $\hat{H}_0(t)$ en los tres horizontes. Con eso, la probabilidad de un pedido nuevo se calcula con la fórmula de la sección 2.1, que no es más que una suma y una exponencial, así que se podría integrar sin mucho esfuerzo en la herramienta de Power BI que la empresa ya está desarrollando: un proceso que cada noche recoja los pedidos abiertos del ERP y devuelva las tres probabilidades. El modelo se podría reentrenar cada trimestre o cada semestre, comprobando en cada revisión el C-índice sobre los pedidos facturados desde el último ajuste. Eso sí, para un sistema de avisos hay que tener en cuenta lo que ya se dijo en la sección 4.6.2: los deciles calculados ahí son exploratorios, el modelo es optimista precisamente en las probabilidades bajas, y para fijar un umbral real haría falta antes la validación temporal descrita más arriba.

Bibliografía

Aalen O (1978) Nonparametric inference for a family of counting processes. The Annals of Statistics 6:701-726. DOI: 10.1214/aos/1176344247.

Austin PC, Harrell FE, van Klaveren D (2020) Graphical calibration curves and the integrated calibration index (ICI) for survival models. Statistics in Medicine 39:2714-2742. DOI: 10.1002/sim.8570.

Breiman L (2001) Random forests. Machine Learning 45:5-32. DOI: 10.1023/A:1010933404324.

Christodoulou E, Ma J, Collins GS, Steyerberg EW, Verbakel JY, Van Calster B (2019) A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. Journal of Clinical Epidemiology 110:12-22. DOI: 10.1016/j.jclinepi.2019.02.004.

Cox DR (1972) Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) 34:187-220. DOI: 10.1111/j.2517-6161.1972.tb00899.x.

Efron B (1977) The efficiency of Cox's likelihood function for censored data. Journal of the American Statistical Association 72:557-565. DOI: 10.1080/01621459.1977.10480613.

Flores M, Fernández-Casal R, Naya S, Tarrío-Saavedra J (2021) Statistical quality control with the qcr package. The R Journal 13:194-217. DOI: 10.32614/RJ-2021-034.

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. The Annals of Statistics 29:1189-1232. DOI: 10.1214/aos/1013203451.

Gerds TA, Schumacher M (2006) Consistent estimation of the expected Brier score in general survival models with right-censored event times. Biometrical Journal 48:1029-1040. DOI: 10.1002/bimj.200610301.

Graf E, Schmoor C, Sauerbrei W, Schumacher M (1999) Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine 18:2529-2545.

Grambsch PM, Therneau TM (1994) Proportional hazards tests and diagnostics based on weighted residuals. Biometrika 81:515-526. DOI: 10.1093/biomet/81.3.515.

Gray RJ (1992) Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis. Journal of the American Statistical Association 87:942-951. DOI: 10.1080/01621459.1992.10476248.

Harrell FE, Lee KL, Mark DB (1996) Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Statistics in Medicine 15:361-387.

Harrell FE (2015) Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. 2ª ed. Springer, Cham. DOI: 10.1007/978-3-319-19425-7.

Heinze G, Schemper M (2001) A solution to the problem of monotone likelihood in Cox regression. Biometrics 57:114-119. DOI: 10.1111/j.0006-341X.2001.00114.x.

Hothorn T, Bühlmann P, Dudoit S, Molinaro A, van der Laan MJ (2006) Survival ensembles. Biostatistics 7:355-373. DOI: 10.1093/biostatistics/kxj011.

Hothorn T, Bühlmann P, Kneib T, Schmid M, Hofner B (2010) Model-based boosting 2.0. Journal of Machine Learning Research 11:2109-2113.

Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008) Random survival forests. The Annals of Applied Statistics 2:841-860. DOI: 10.1214/08-AOAS169.

Ishwaran H, Kogalur UB (2023) randomForestSRC: Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC). R package version [versión]. https://cran.r-project.org/package=randomForestSRC. Accedido [xx] de [mes] de 2026.

Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. Journal of the American Statistical Association 53:457-481. DOI: 10.1080/01621459.1958.10501452.

Klein JP, Moeschberger ML (2003) Survival Analysis: Techniques for Censored and Truncated Data. 2ª ed. Springer, Nueva York. DOI: 10.1007/b97377.

Montgomery DC (2013) Introduction to Statistical Quality Control. 7ª ed. Wiley, Hoboken.

Probst P, Wright MN, Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. WIREs Data Mining and Knowledge Discovery 9:e1301. DOI: 10.1002/widm.1301.

Royston P, Parmar MKB (2002) Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21:2175-2197. DOI: 10.1002/sim.1203.

Royston P, Altman DG (2013) External validation of a Cox prognostic model: principles and methods. BMC Medical Research Methodology 13:33. DOI: 10.1186/1471-2288-13-33.

Smirnov J (2016) Modelling late invoice payment times using survival analysis and random forests techniques. Tesis, University of Tartu.

Steyerberg EW, Vickers AJ, Cook NR, Gerds T, Gonen M, Obuchowski N, Pencina MJ, Kattan MW (2010) Assessing the performance of prediction models: a framework for traditional and novel measures. Epidemiology 21:128-138. DOI: 10.1097/EDE.0b013e3181c30fb2.

Steyerberg EW (2019) Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating. 2ª ed. Springer, Cham. DOI: 10.1007/978-3-030-16399-0.

Teinemaa I, Dumas M, La Rosa M, Maggi FM (2019) Outcome-oriented predictive process monitoring: review and benchmark. ACM Transactions on Knowledge Discovery from Data 13:1-57. DOI: 10.1145/3301300.

Therneau TM, Grambsch PM (2000) Modeling Survival Data: Extending the Cox Model. Springer, Nueva York. DOI: 10.1007/978-1-4757-3294-8.

Uno H, Cai T, Pencina MJ, D'Agostino RB, Wei LJ (2011) On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Statistics in Medicine 30:1105-1117. DOI: 10.1002/sim.4154.

van der Aalst WMP (2016) Process Mining: Data Science in Action. 2ª ed. Springer, Berlín. DOI: 10.1007/978-3-662-49851-4.

Van Calster B, McLernon DJ, van Smeden M, Wynants L, Steyerberg EW (2019) Calibration: the Achilles heel of predictive analytics. BMC Medicine 17:230. DOI: 10.1186/s12916-019-1466-7.

Van Houwelingen HC (2000) Validation, calibration, revision and combination of prognostic survival models. Statistics in Medicine 19:3401-3415.

Verenich I, Dumas M, La Rosa M, Maggi FM, Teinemaa I (2019) Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Transactions on Intelligent Systems and Technology 10:1-34. DOI: 10.1145/3331449.

Wang P, Li Y, Reddy CK (2019) Machine learning for survival analysis: a survey. ACM Computing Surveys 51:1-36. DOI: 10.1145/3214306.

Apéndice A

Código en R

Este apéndice recoge los fragmentos de código correspondientes a las decisiones descritas en el cuerpo del trabajo. No se reproduce el código completo, sino las partes que permiten verificar cómo se tomó cada decisión.

A.1. Variables de supervivencia

Corresponde a la sección 3.3.

```
fecha_corte <- max(dat$FechaPedido, na.rm = TRUE)

dat <- dat %>%
  mutate(
    evento = as.integer(!is.na(FechaFactura)),
    tiempo_obs = ifelse(
      evento == 1,
      as.numeric(difftime(FechaFactura, FechaPedido, units = "days")),
      as.numeric(difftime(fecha_corte, FechaPedido, units = "days"))
    )
  ) %>%
  filter(tiempo_obs >= 0)
```

A.2. Agregación de línea a pedido

Corresponde a las secciones 3.2.1 y 3.2.2. El filtro por tipo de albarán se aplica antes de agregar; el de importe negativo, después.

```
excluir_albaran <- c("No Facturable", "Recoge en nave no facturable",
                     "Solo para transporte", "ALQUILER")

dat_lin <- dat_raw %>%
  filter(!is.na(TipoAlbaran), !TipoAlbaran %in% excluir_albaran)

# factores ordenados de peor a mejor: min() da el estado mas atrasado
dat_lin <- dat_lin %>%
  mutate(
    SituacionOF = factor(SituacionOF,
                         levels = c("BLOQUEADA", "LANZADA", "CON IMPUTACIONES",
                                    "FINALIZADA", "SIN_OF"), ordered = TRUE)
  )

dat <- dat_lin %>%
  group_by(Pedido) %>%
```

```
  summarise(
    n_lineas = n(),
    FechaPedido = first(FechaPedido),
    ImportePedido  = sum(importe_linea,  na.rm = TRUE),
    CantidadPedido = sum(cantidad_linea, na.rm = TRUE),
    FechaFactura = if (all(!is.na(FechaFactura))) max(FechaFactura) else as.Date(NA),
    LineaNegocio = LineaNegocio[which.max(importe_linea)],
    FormaEntrega = FormaEntrega[which.max(importe_linea)],
    SituacionOF  = as.character(min(SituacionOF)),
    .groups = "drop"
  )

# el filtro de abonos va DESPUES de agregar, sobre las sumas ya hechas
dat <- dat %>% filter(ImportePedido >= 0, CantidadPedido >= 0)
```

A.3. Conjuntos de datos

Corresponde a la sección 3.5.

```
vars_fuera_cox <- c("Fabricado", "PrioridadOF", "tiene_OF",
                    "AlbaranBloqueado", "Albaran", "Cumplimentado")
vars_26 <- setdiff(vars_32, vars_fuera_cox)

dat_cox <- dat %>%
  filter(SituacionOF != "BLOQUEADA",
         FormaEntrega != "YA INSTALADO - NOTA ABONO (Normal)") %>%
  select(Pedido, tiempo_obs, evento, all_of(vars_26))

dat_rsf_fair <- dat %>%
  filter(Pedido %in% dat_cox$Pedido) %>%
  select(tiempo_obs, evento, all_of(vars_26)) %>%
  codificar_rsf(vars_factor_rsf)

dat_rsf_completo <- dat %>%
  select(tiempo_obs, evento, all_of(vars_32)) %>%
  codificar_rsf(vars_factor_rsf)
```

A.4. Ajuste de los modelos

Corresponde a las secciones 4.2, 4.3 y 4.4.

```
cox_basico <- coxph(Surv(tiempo_obs, evento) ~ . - Pedido,
                    data = dat_cox, ties = "efron")
ph_test <- cox.zph(cox_basico)

cox_sp <- coxph(
  Surv(tiempo_obs, evento) ~
    pspline(SemanaPedido) + pspline(n_lineas) + pspline(CantidadPedido) +
    pspline(ImportePedido) + pspline(AnticipacionSolicitada) + pspline(tiempo_min) +
    sen12 + cos12 + verano + navidad + dia_semana + Descuento + SituacionOF +
    FormaEntrega + TipoAlbaran + Anticipo + Prioridad + HojaAlmacen +
    LineaNegocio + TipoCliente + DescripcionCNAE +
    DescripcionCampanaMarketing + TipoFactura + TipoRotulacion +
    TipoContabilidad + tipo_articulo,
  data = dat_cox, ties = "efron"
)

set.seed(60)
rsf_fair_obj <- rfsrc(Surv(tiempo_obs, evento) ~ ., data = dat_rsf_fair,
                      ntree = 1000, nodesize = 30, nsplit = 10,
```

```
                      block.size = 10, importance = "none")

# glmboost escala mal: se entrena con una muestra y se evalua en el resto.
# la matriz de diseno se construye UNA sola vez sobre todo dat_cox y luego se
# parte por filas, asi entrenamiento y prediccion usan las mismas columnas
X_completa <- model.matrix(~ . - 1, data = dat_cox %>%
                             select(-Pedido, -tiempo_obs, -evento))
y_completa <- Surv(dat_cox$tiempo_obs, dat_cox$evento)

set.seed(60)
idx_muestra <- sample(nrow(dat_cox), 5000)

modelo_boost <- glmboost(x = X_completa[idx_muestra, ], y = y_completa[idx_muestra],
                         family = CoxPH(),
                         control = boost_control(mstop = 100, nu = 0.1))
set.seed(60)
cv_boost <- cvrisk(modelo_boost,
                   folds = cv(model.weights(modelo_boost), type = "kfold", B = 5))
modelo_boost[mstop(cv_boost)]
```

A.5. Métricas de evaluación

Corresponde a las secciones 4.2.1 y 4.6.1.

```
# validacion cruzada: con una fila por pedido, cada pedido cae en un pliegue
set.seed(60)
folds <- sample(rep(1:5, length.out = nrow(dat_cox)))
c_cv  <- numeric(5)

for (k in 1:5) {
  train_k <- dat_cox[folds != k, ] %>% mutate(across(where(is.factor), droplevels))
  test_k  <- dat_cox[folds == k, ]
  fit_k   <- coxph(formula_cv, data = train_k, ties = "efron")
  lp_k    <- predict(fit_k, newdata = test_k, type = "lp")
  c_cv[k] <- concordance(Surv(tiempo_obs, evento) ~ I(-lp_k),
                         data = test_k)$concordance
}

# brier score con ponderacion por censura
t_max   <- as.numeric(quantile(dat_cox_pec$tiempo_obs, 0.90))
pec_res <- pec(object = list("Cox" = cox_x),
               formula = Surv(tiempo_obs, evento) ~ 1,
               data = dat_cox_pec, times = seq(1, t_max, by = 1),
               cens.model = "marginal", splitMethod = "none")
ibs <- crps(pec_res, times = t_max)

# c-indice de uno
lp <- predict(cox_basico, newdata = dat_cox, type = "lp")
c_uno <- concordance(Surv(tiempo_obs, evento) ~ I(-lp), data = dat_cox,
                     timewt = "n/G2", tau = 30)$concordance

# pendiente aparente de calibracion (sobre la propia muestra de ajuste)
cal_modelo <- coxph(Surv(tiempo_obs, evento) ~ lp, data = dat_cox)
```

A.6. Hitos y comparación de probabilidades

Corresponde a las secciones **??**, 4.6.2 y 5.2.

```
hitos <- dat2 %>%
  group_by(Pedido) %>%
```

```
  summarise(
    across(starts_with("Fecha"),
           ~ if (all(is.na(.x))) as.Date(NA) else min(.x, na.rm = TRUE)),
    Fabricacion  = moda(Fabricacion),
    FormaEntrega = moda(FormaEntrega),
    LineaNegocio = moda(LineaNegocio),
    .groups = "drop"
  )

# dat_cox ya esta a nivel de pedido y conserva Pedido: el cruce es 1 a 1
dat_enriquecido <- dat_cox %>%
  inner_join(hitos, by = "Pedido", suffix = c("", "_h2")) %>%
  select(-ends_with("_h2"))
stopifnot(nrow(dat_enriquecido) == n_distinct(dat_enriquecido$Pedido))

sf_val   <- survfit(cox_basico, newdata = dat_enriquecido)
sf_res   <- summary(sf_val, times = c(7, 14, 30))
prob_mat <- t(1 - sf_res$surv)

deciles <- dat_val %>%
  filter(!is.na(P_pred_30d)) %>%
  mutate(decil = ntile(P_pred_30d, 10)) %>%
  group_by(decil) %>%
  summarise(pred = round(mean(P_pred_30d), 3),
            real = round(mean(real_30d), 3), n = n(), .groups = "drop")
```

Apéndice B

Coeficientes completos del modelo de Cox

La tabla 4.1 recoge los diez coeficientes con menor valor p. La tabla siguiente recoge los 87 términos que constituyen el modelo, de los cuales 86 son estimables.

Término	**HR**	**Error estándar**	**Estadístico**	**Valor p**
AnticipacionSolicitada	0.570	0.0081	-69.22	0.0000
LineaNegocioMOHEVA	10.879	0.0814	29.33	0.0000
TipoContabilidad8	0.512	0.0301	-22.23	0.0000
FormaEntregaINSTALACIÓN SIN COMPRA NI FABRICACIÓN PREVIA	3.214	0.0560	20.86	0.0000
SituacionOFSIN_OF	3.645	0.0695	18.59	0.0000
SituacionOFFINALIZADA	2.573	0.0628	15.06	0.0000
Anticipo1	0.667	0.0293	-13.82	0.0000
LineaNegocioLONAS PARA EL TRANSPORTE	1.561	0.0325	13.69	0.0000
FormaEntregaINSTALAR EN NAVE TGM	0.692	0.0359	-10.25	0.0000
Prioridad3	1.230	0.0204	10.15	0.0000
DescripcionCNAEFabricación de otros artículos confeccionados con textiles, excepto prendas de vestir	0.008	0.5119	-9.52	0.0000
TipoFactura3	3.556	0.1391	9.12	0.0000
FormaEntregaINSTALAR	0.770	0.0317	-8.26	0.0000
PrioridadOtros	0.416	0.1095	-8.01	0.0000
DescripcionCNAEFabricación de artículos de marroquinería y viaje, artículos de guarnicionería y talabartería	0.350	0.1418	-7.40	0.0000
DescripcionCNAEVenta de vehículos automóviles	0.369	0.1396	-7.14	0.0000
DescripcionCNAEFabricación de cerveza	0.482	0.1045	-6.99	0.0000
ImportePedido	0.962	0.0057	-6.71	0.0000
LineaNegocioSIN LÍNEA DE NEGOCIO	1.364	0.0470	6.61	0.0000
HojaAlmacen	0.809	0.0327	-6.47	0.0000
DescripcionCNAEFabricación de otro material de transporte	0.458	0.1265	-6.18	0.0000
DescripcionCNAEPublicidad	2.522	0.1558	5.94	0.0000

Término	HR	Error estándar	Estadístico	Valor p
DescripcionCNAEFabricación de carrocerías para vehículos de motor, de remolques y semirremolques	0.583	0.1114	-4.84	0.0000
n_lineas	0.900	0.0252	-4.20	0.0000
DescripcionCNAEOtras actividades de impresión	1.701	0.1299	4.09	0.0000
DescripcionCNAEConstrucción de autopistas, carreteras,campos de aterrizaje, vías férreas y centros deportivos	0.563	0.1427	-4.02	0.0001
DescripcionCNAEIntermediarios del comercio de muebles, artículos para el hogar y ferretería	1.561	0.1157	3.85	0.0001
LineaNegocioPROTECCIÓN SOLAR	1.152	0.0368	3.84	0.0001
FormaEntregaRECOGE EN NAVE	0.905	0.0267	-3.74	0.0002
DescripcionCNAEInstalaciones eléctricas	1.667	0.1389	3.68	0.0002
DescripcionCNAEFabricación de otra maquinaria agraria	1.708	0.1518	3.52	0.0004
DescripcionCampanaMarketingSIN	1.172	0.0470	3.37	0.0007
tipo_articuloCliente_Catalogo	1.464	0.1167	3.26	0.0011
CantidadPedido	1.028	0.0092	3.00	0.0027
DescripcionCNAEMudanzas	1.565	0.1552	2.89	0.0039
Descuento	0.935	0.0238	-2.82	0.0048
DescripcionCNAEFabricación de carpintería metálica	1.367	0.1119	2.79	0.0052
TipoClienteOtros	0.591	0.1990	-2.64	0.0083
DescripcionCampanaMarketingOTRAS	1.673	0.1952	2.64	0.0084
TipoRotulacionROTULADO	1.113	0.0432	2.48	0.0132
TipoClientePARTICULAR	1.062	0.0246	2.45	0.0141
tipo_articuloFunda	0.775	0.1108	-2.30	0.0212
dia_semanaMon	0.955	0.0205	-2.25	0.0245
verano	0.947	0.0243	-2.24	0.0250
Anticipo2+	0.383	0.4480	-2.14	0.0323
tipo_articuloTecho	0.791	0.1174	-1.99	0.0462
cos12	0.967	0.0169	-1.99	0.0468
LineaNegocioTEXTILES TÉCNICOS	1.067	0.0328	1.98	0.0482
dia_semanaTue	0.962	0.0208	-1.85	0.0641
tipo_articuloOtros	0.831	0.1055	-1.75	0.0797
DescripcionCNAEFabricación de vehículos de motor	1.242	0.1263	1.71	0.0868
sen12	0.960	0.0237	-1.70	0.0886
DescripcionCNAESIN	1.177	0.0956	1.70	0.0890
tipo_articuloLogistica	0.719	0.1950	-1.69	0.0902
FormaEntregaSÓLO ENTREGAR	0.953	0.0311	-1.53	0.1252
DescripcionCNAETransporte de otras mercancías por carretera	1.204	0.1228	1.51	0.1298
dia_semanaThu	0.970	0.0209	-1.47	0.1412
tipo_articuloCinta	0.844	0.1175	-1.44	0.1488
DescripcionCNAEFabricación de productos para la alimentación de animales de granja	1.238	0.1481	1.44	0.1493

Término	**HR**	**Error estándar**	**Estadístico**	**Valor p**
`DescripcionCNAEOtras actividades empresariales`	1.258	0.1672	1.37	0.1699
`SemanaPedido`	0.998	0.0013	-1.25	0.2095
`tipo_articuloPuerta`	0.864	0.1237	-1.18	0.2381
`tipo_articuloServicio`	0.891	0.1070	-1.08	0.2816
`dia_semanaWed`	0.980	0.0206	-1.01	0.3145
`LineaNegocioCINTAS DE TRINCAJE Y ELEVACIÓN`	1.053	0.0561	0.92	0.3590
`DescripcionCNAEMantenimiento y reparación de vehículos de motor`	1.146	0.1491	0.91	0.3613
`DescripcionCNAEProducción y primera transformación de aluminio`	1.113	0.1200	0.89	0.3712
`tipo_articuloComplemento`	0.906	0.1118	-0.89	0.3758
`LineaNegocioIMPRESIÓN DIGITAL Y ROTULACIONES`	1.041	0.0486	0.83	0.4076
`DescripcionCNAETransporte de mercancías por carretera`	1.097	0.1131	0.82	0.4138
`tipo_articuloToldo`	0.919	0.1092	-0.77	0.4410
`tipo_articuloCortina`	0.924	0.1163	-0.68	0.4946
`DescripcionCNAEComercio al por mayor de madera`	0.916	0.1534	-0.58	0.5652
`Anticipo2`	0.927	0.1364	-0.55	0.5811
`tipo_articuloLona`	0.944	0.1046	-0.55	0.5817
`tiempo_min`	0.997	0.0051	-0.52	0.6033
`navidad`	1.021	0.0408	0.51	0.6067
`TipoRotulacionSIN ROTULACIÓN`	0.979	0.0421	-0.50	0.6199
`tipo_articuloVinilo_Rotulacion`	0.949	0.1169	-0.45	0.6557
`Prioridad2`	1.008	0.0186	0.41	0.6805
`TipoClienteEMPRESA`	0.990	0.0240	-0.41	0.6847
`DescripcionCNAEOtros`	1.033	0.0983	0.33	0.7409
`DescripcionCampanaMarketingMAHOU`	1.008	0.0572	0.14	0.8921
`TipoClienteSECTOR PÚBLICO`	0.988	0.0932	-0.13	0.8995
`TipoFactura1`	1.008	0.1396	0.06	0.9554
`TipoRotulacionROTULADO + IMPRESIÓN DIGITAL`	0.995	0.1012	-0.05	0.9582
`TipoAlbaranStandard`	NA	0.0000	NA	NA

El nivel `TipoAlbaranStandard` resulta linealmente dependiente del resto de la matriz de diseño una vez aplicados los filtros del conjunto, de modo que `coxph()` no lo estima y lo devuelve como `NA`. El modelo tiene por tanto 86 coeficientes efectivos, que coinciden con los 86 grados de libertad del contraste global de proporcionalidad de la sección 4.2.2.

Índice de tablas

Índice de figuras