Traballo Fin de Máster

Personalización do horario de contacto en banca dixital mediante metodoloxías de aprendizaxe estatística

Santiago Alonso Pardal

Máster en Técnicas Estatísticas

Curso 2025-2026

II

Proposta de Traballo Fin de Máster

Título en galego: Personalización do horario de contacto en banca dixital mediante metodoloxías de aprendizaxe estatística
Título en español: Personalización del horario de contacto en banca digital mediante metodologías de aprendizaje estadístico
English title: Personalization of contact schedule in digital banking through statistical learning methodologies
Modalidade: Modalidade B
Autor/a: Santiago Alonso Pardal, Universidade de Santiago de Compostela
Director/a: Rubén Fernández Casal, Universidade da Coruña; Manuel Oviedo de la Fuente, Universidade da Coruña
Titor/a: María Oliveira Pérez, ABANCA
Breve resumo do traballo: O obxectivo é determinar a hora ideal para o envío de emails a cada cliente e así maximizar a súa interacción. Mediante metodoloxías de Aprendizaxe Estatística, analizamos a probabilidade de lecturas segundo o horario co fin de deseñar unha política de envíos personalizada e eficiente.

Don Rubén Fernández Casal, profesor contratado doutor da Universidade da Coruña, don Manuel Oviedo de la Fuente, profesor axudante doutor da Universidade da Coruña e dona María Oliveira Pérez, Product Service Leader de ABANCA informan que o Traballo Fin de Máster titulado

Personalización do horario de contacto en banca dixital mediante metodoloxías de aprendizaxe estatística

foi realizado baixo a súa dirección por don Santiago Alonso Pardal para o Máster en Técnicas Estatísticas. Estimando que o traballo está terminado, dan a súa conformidade para a súa presentación e defensa ante un tribunal. Ademais, Don Rubén Fernández Casal, don Manuel Oviedo de la Fuente e don Santiago Alonso Pardal

√ sí ☐ no

autorizan a publicación da memoria no repositorio de acceso público asociado ao Máster en Técnicas Estatísticas.

En Santiago de Compostela, a 3 de xuño de 2026.

O/a director/a:
Don Rubén Fernández Casal

Firmado digitalmente por FERNANDEZ CASAL RUBEN - 33317724Q
Fecha: 2026.06.02 18:20:22 +02'00'

O/a director/a:
Don Manuel Oviedo de la Fuente

Manuel Oviedo de la Fuente
Firmado digitalmente por Manuel Oviedo de la Fuente
Fecha: 2026.06.02 16:41:35 +02'00'

O/a titor/a:
Doña María Oliveira Pérez

OLIVEIRA PEREZ MARIA - 44837878E
Firmado digitalmente por OLIVEIRA PEREZ MARIA - 44837878E
Fecha: 2026.06.02 22:06:52 +02'00'

O/a autor/a:
Don Santiago Alonso Pardal

ALONSO PARDAL SANTIAGO - 45959407W
Firmado digitalmente por ALONSO PARDAL SANTIAGO - 45959407W
Fecha: 2026.06.02 22:28:59 +02'00'

Declaración responsable. Para dar cumprimento á Ley 3/2022, de 24 de febreiro, de convivencia universitaria, referente ao plaxio no Traballo Fin de Mestrado (Artigo 11, Disposición 2978 del BOE núm. 48 de 2022), **o/a autor/a declara** que o Traballo Fin de Mestrado presentado é un documento orixinal no que se tiveron en conta as seguintes consideracións relativas ao uso de material de apoio elaborado por outros/as autores/as:

- Todas as fontes usadas para a elaboración deste traballo foron citadas convenientemente (libros, artigos, apuntes de profesorado, páxinas web, programas,...)
- Calquera contido copiado ou traducido textualmente foi posto entre comiñas, citando a súa procedencia.
- Fíxose constar explicitamente cando un capítulo, sección, demostración,... sexa unha adaptación case literal dalgunha fonte existente.

E, acepta que, se se demostrara o contrario, se lle apliquen as medidas disciplinarias que correspondan.

Agradecementos

Gustaríame agradecer a todas aquelas persoas que fixeron posible a elaboración da presente memoria.

En primeiro lugar aos meus titores María Oliveira Pérez, Rubén Fernández Casal e Manuel Oviedo de la Fuente polo seu apoio e consello ao longo do traballo. Seguidamente, gustaríame agradecer a todos os membros do departamento de *Intelixencia de Clientes* pola súa axuda e indicacións durante o proceso de integración na empresa. En particular, pola súa cálida acollida, guía e paciencia, quero agradecer ao equipo de *Modelos de Analítica Avanzada*: María, Rubén, Jose e Manuel. Tamén querería agradecer a *ABANCA Corporación Bancaria S.A.* por darme a oportunidade de coñecer e traballar no contexto bancario, nunha temática tan interesante e profundamente relacionada co ámbito da estatística.

Grazas tamén á miña familia. Á de sangue: meus pais e miña irmá, avoas, tíos, primos e, especialmente, ás miñas "compis de piso" Ñi e Andrea. Tamén a esa familia escollida que, malia non compartir lazos biolóxicos, sinto como a anterior: Emma, cuxa paciencia e fortaleza axudáronme nos momentos máis complicados do proceso, a miña madriña e a Andrés, Inesita, Andrés fillo, Raquel e Guillermo. Así mesmo quero agradecer, por un lado, a aqueles profesionais da ensinanza que confiaron en min e me motivaron a continuar traballando e mellorando: Antonio, Federico e Jere. E, por outro, a aquelas persoas que, sen adicarse a ensinar, tamén souberon facelo: meus avós Paulino e Cesáreo, meu tío José e David. Por último, pero non menos importante, aos meus amigos, por ser cómplices durante os momentos de ledicia e levantarme nos de dificultade: aos meus amigos da carreira, Raquel, Leti e ao *"Basket Praíña"*.

Estas son só algunhas das persoas que fixeron posible o desenvolvemento deste traballo, emporiso, hai moitas outras que, por unha evidente falta de espazo, non aparecen recollidas nestas liñas. A todos e todas eles, de corazón, grazas.

Índice

Resumo

Resumo en galego

A principal meta deste traballo é propoñer, para cada cliente, unha hora do día idónea para o envío dun *email*; a fin de obter a mellor interacción posible. Este proceso inseriríase dentro das estratexias coñecidas como *Email Marketing Personalization*, as cales buscan mellorar a experiencia da interacción do cliente co *email* personalizando todos os aspectos relativos ao lanzamento de campañas vía correo electrónico. En particular, a individualización da hora de envío para cada usuario. Por iso, resulta clave para calquera empresa contar con algún protocolo destas características. Consecuentemente, neste proxecto abordaremos esta problemática dende o punto de vista da Aprendizaxe Estatística; axustando modelos de *Machine Learning* para avaliar a probabilidade de que un dos nosos clientes interactúe cun destes correos en función da hora de envío. Posteriormente, buscaremos interpretar estes modelos e, a partir deles, desenvolver unha política de envíos eficiente para a súa xestión.

English abstract

The primary objective of this work is to propose, for each customer ,the ideal time of day to send an email; in order to achieve the best possible engagement. This process falls within the strategies known as Email Marketing Personalization, which aim to enhance the customer's interactive experience by customizing all aspects related to the launch of email campaigns. Specifically, tailoring the delivery time for each individual user. Therefore, implementing a protocol of this nature is crucial for any company. Consequently, this project addresses this issue from a Statistical Learning perspective, fitting Machine Learning models to evaluate the probability of a customer interacting with an email based on its send time. Subsequently, we aim to interpret these models and leverage them to develop an efficient delivery policy for campaign management.

Capítulo 1

Introdución

1.1. Motivación

ABANCA Corporación Bancaria S.A. é unha entidade que oferta diversos servizos financeiros. Esta abrangue distintas áreas de negocio con diferentes atribucións e funcións. Unha delas é a conformada polo departamento de *Intelixencia de Clientes* (*I.C.*), dedicada á disciplina coñecida como *Business Intelligence*. Seu cometido é recoller e analizar toda a información da empresa relativa aos clientes co fin de xerar coñecementos útiles no proceso de toma de decisións da empresa. A idea disto é axustar a oferta dos produtos financeiros da compañía ás necesidades e características do cliente. Dito departamento está conformado á súa vez por dúas seccións: *Modelos de Analítica Avanzada* e *Customer Relationship Management (C.R.M.) Omnicanal e Marketing Automation*. Os obxectivos máis específicos destas áreas son os seguintes:

- Dar soporte ás distintas vías de negocio, resolvendo problemas de uso. En particular, desenvolvendo modelos de propensión, detección temperá de abandono ou outros modelos avanzados. Para isto empregan técnicas de Aprendizaxe Estatística e *Machine Learning*.
- Contrastar a utilidade e vixencia destes modelos e informes desenvoltos, baseándose en diferentes métricas e indicadores de rendemento (denominados **KPIs**). O **indicador clave de rendemento** ou **KPI** é calquera métrica que permite avaliar o desempeño dunha empresa, sector ou sistema económico en relación con obxectivos estratéxicos específicos (Empresa, 2026).
- Estudar a base de clientes da entidade mediante técnicas de analítica avanzada para definir o público obxectivo máis axeitado para cada tipoloxía de acción comercial, así como as canles e estratexias de contactos comerciais *omnicanal*. Controlando a adecuación, colisión, saturación, conveniencia e oportunidade destes.
- Seleccionar o segmento de contactos, canles e intres axeitados para optimizar a relación custo-beneficio deste tipo de iniciativas. Nesta función é onde se adscribe o proxecto desenvolto na presente memoria.

En particular e, relacionado con este último punto, o departamento de *I.C.* é o encargado da xestión e orquestración de campañas comerciais *omnicanal*. É dicir, aquelas baseadas no contacto co cliente a través de múltiples canles de comunicación; sendo unha delas o contacto vía *email* que, non só ten carácter puramente administrativo, senón que busca tamén facer publicidade dos seus servizos. Esta clase de acción, en particular, é o que se coñece como ***email marketing***.

O *email marketing* é unha práctica comercial que consiste no envío de correos electrónicos co fin de achegar os produtos dunha empresa á súa clientela. Esta é unha ferramenta básica do coñecido como ***digital marketing***. Segundo a revista Forbes (Haan & Watts, 2026), existen arredor de 4.48

mil millóns de usuarios de correo electrónico no mundo, que reciben unha media de 361.6 mil millóns de *emails* diarios en total. Polo que o público obxectivo desta práctica é extremadamente amplo. Adicionalmente esta clase de procedementos resultan moi proveitosos, xa que unicamente en 2024 estimáronse uns beneficios globais de 12.33 mil millóns de dólares derivados dos mesmos e o seu **ROI** aproximado estaría comprendido entre 32 e 45 dólares (Haan & Watts, 2026). O *Return on Investment* ou **ROI** mide a rendibilidade dunha inversión en base ao seu costo, é dicir, indica a ganancia obtida por cada unidade monetaria investida nun activo dado.

Nos últimos anos, co desenvolvemento do *Big Data* e a incorporación de técnicas analíticas máis modernas, propúxose enfrontar esta casuística dende o punto de vista da *Intelixencia de Clientes*. Nace así o concepto de ***email marketing*** **personalizado**, que consiste en individualizar todo o proceso de envío (dende a hora de contacto, ata o contido do propio correo) para cada cliente. Este proceso ten probada a súa eficacia, xa que se estima un 80 % máis de propensión dos usuarios a contratar servizos presentados a partir dun *email* "personalizado" con respecto a un que non o estea (Haan & Watts, 2026). En particular, recentemente, estase considerando a posibilidade de abordar este tema dende un punto de vista da predición (Araújo e col., 2022). Máis concretamente, dende a óptica do *Machine Learning* (Saleh Abbas & Al-Jailawi, 2024).

Unha das cuestións críticas deste proceso é a hora de envío da comunicación (Ellering, 2025). Dada a enorme cantidade de *emails* que pode chegar a recibir un cliente, moitos deles corren o risco de ser ignorados ou sinalados como *spam* polo servizo de correo electrónico. Esta serie de problemáticas aumentan ao producirse o envío en horas nas que o cliente non está dispoñible. Así mesmo, se o correo é ignorado durante un período de tempo amplo, pode perderse unha ventá de negocio importante ao non ser capaces de contactar con el de xeito instantáneo.

Tendo todo isto presente, o departamento de *I.C.* da corporación *ABANCA* impulsa diversas accións dirixidas a mellorar esta experiencia de usuario. Non obstante, actualmente non existe ningún protocolo de envíos. É por iso que se propuxo empregar modelos de Aprendizaxe Estatística para abordar esta problemática coa maior rigorosidade posible.

1.2. Antecedentes

Nos últimos tempos propuxéronse diversas alternativas para determinar, non necesariamente dende o punto de vista da predición, a hora de envío ideal dunha comunicación vía mensaxería electrónica. Antes de presentar algunhas destas propostas, comezaremos presentando a metodoloxía proposta inicialmente pola entidade.

Para proporcionar unha hora de lectura a un cliente dado agrupan todas as interaccións con el, recollidas no prazo dun ano, e efectúan un reconto das horas nas que o cliente accedeu a eses *emails*; asignándolle aquela hora que se repita un maior número de veces. Este enfoque, se ben é sinxelo de implementar e recolle información sobre o histórico do noso cliente, presenta dúas carencias fundamentais. A primeira é que non considera información relativa á situación do cliente fóra das interaccións mantidas co mesmo e, en segundo lugar, non toma en consideración o tempo transcorrido entre o envío da comunicación e a súa lectura. Polo que este período pode abarcar, dende unha hora, ata meses. Efectuouse un **Test A/B**[1] para avaliar a capacidade de impacto sobre os clientes, chegando á conclusión de que o modelo non se probaba eficaz. É por iso que, dende o departamento de *I.C.* de *ABANCA*, propúxose construír un protocolo máis axeitado mediante técnicas de Aprendizaxe Estatística.

Unha vez proposta a tarefa; procedeuse cunha busca exhaustiva de metodoloxías que permitisen resolver o problema, atopándose catro enfoques fundamentais (Araújo e col., 2022):

[1] O test A/B é unha clase de método de experimentación estatística que consiste en comparar dúas versións dun mesmo elemento para determinar cal é o máis efectivo. Neste caso, o elemento A sería un modelo de envío aleatorios e, o B, o protocolo de envíos da entidade.

- **Metodoloxías de Regresión:** Deligiannis, Argyriou e Kourtesis (2020a) tratan de axustar distintos modelos de regresión para predicir o *Click-Through rate*, que non é máis que a porcentaxe de persoas que acceden á ligazón que adoitan levar adxuntos os correos electrónicos. En particular fan uso de modelos loxísticos, redes neuronais (**RNN**) e regresión **XGBoost**. A idea é predicir esta proporción de *clicks* coa fin de coñecer o impacto que ten unha campaña dada sobre os distintos segmentos de clientes dunha entidade. Posteriormente, os mesmos autores, amplían este traballo buscando predicir a data e instante óptimos para enviar mensaxes personalizados a compradores habituais (Deligiannis, Argyriou & Kourtesis, 2020b).

- **Metodoloxías de Clasificación:** Paralič, Kaszoni e Mačina (2020) propoñen empregar métodos **ML** de clasificación como son as *Árbores de Decisión*, *Random Forest* e *Naive-Bayes* para tratar de predicir a hora de envío axustando tres modelos: un para indicar se o *email* foi ou non aberto, outro para predicir a hora de apertura da comunicación e, finalmente, un modelo que permita predicir o día en que é máis probable que teña lugar.

- **Metodoloxías mixtas de Regresión e Clasificación:** Tamén existen enfoques onde consideran ambas metodoloxías por separado, como por exemplo Saleh Abbas e Al-Jailawi (2024), ou que mesturan ambas; como Sinha, Vinay e Singh (2018). Neste derradeiro traballo efectúase unha predición dende o punto de vista da regresión aplicada á *Análise de Supervivencia*, empregando algoritmos de regresión como o *Cox Proportional Hazard Regression*.

- **Metodoloxías non supervisadas:** Outra alternativa é a proposta por Pal, Bansal, Singh, Hiran e Yadav (2022), onde se efectúa unha *Análise Clúster* para segmentar o conxunto de clientes en base aos seus hábitos e características comúns e, posteriormente, proporcionan unha hora potencial para efectuar o envío mediante un *Algoritmo Bandit*. Deste xeito poden, non só propoñer unha hora ideal, senón tomar en consideración a incerteza da resposta.

Atendendo aos estudos anteriores, propoñemos levar a cabo unha metodoloxía similar adaptada ás necesidades de *ABANCA Corporación Bancaria S.A.*

1.3. Obxectivo e planificación do traballo

Obxectivo: O principal obxectivo deste traballo será recomendar unha hora de envío axeitada para aquelas campañas de *emails* dirixidas ás persoas físicas relacionadas coa nosa entidade mediante metodoloxías de Aprendizaxe Estatística.

Metodoloxía: Para acadar o noso obxectivo propoñemos efectuar predicións da probabilidade de interacción satisfactoria cun usuario condicionada a distintas variables explicativas relacionadas ca hora de envío da comunicación. Para achegalas utilizaremos metodoloxías de *Aprendizaxe estatística* combinadas con procesos de calibrado para refinar estas predicións. En base a estas probabilidades calibradas, asignarémoslle a cada cliente a hora de envío que a maximice.

Planificación: Seguidamente, vamos expor minuciosamente a estrutura que seguiremos nesta memoria para introducir a implementación destes modelos sobre o noso problema:

- **Contexto teórico:** Neste capítulo construiremos a base teórica relativa aos conceptos que trataremos na presente memoria (Capítulo 2).

- **Tratamento dos datos:** Aquí, afondaremos nas bases de datos da entidade para seleccionar aquelas relacións de datos que poidan ter algún interese á hora de efectuar predicións. A partir delas, xeraremos táboas que nos permitan acceder facilmente aos mesmos (Capítulo 3).

- **Modelos:** Neste capítulo (Capítulo 4) levaremos a cabo os seguintes procesos:

- *Partición dos datos:* Nesta sección explicaremos como segmentaremos o noso *dataset*.
- *Selección de variables explicativas:* Exporemos como efectuamos a selección de variables dos nosos modelos.
- *Modelos de lecturas:* Unha vez realizada a selección de hiperparámetros en base a un certo criterio de optimización, axustaremos os diferentes tipos de modelos propostos na memoria.
- *Valoración dos modelos:* Finalmente, efectuaremos unha valoración baseada nas métricas das mostras de validación para facernos unha idea de cal sería o modelo con, a priori, un maior rendemento.
- *Calibración das probabilidades condicionadas:* Unha vez teñamos seleccionado os modelos máis axeitados, procederemos ca calibración das probabilidades condicionadas estimadas por cada un deles.

- **Posta en produción:** Neste capítulo mostraremos unha proposta sobre como implementar as predicións da probabilidades achegadas polo modelo sobre unha base de datos, así como a construción da mesma (Capítulo 5).
- **Liñas de mellora e estudo:** Para rematar coa nosa memoria, presentaremos algunhas das limitacións e problemáticas que enfrontamos durante a elaboración do proxecto e que non foi posible resolver. Ademais, engadiremos futuras liñas de investigación que poderían resultar interesantes (Capítulo 6).

Capítulo 2

Contexto teórico

Neste capítulo contextualizaremos os conceptos teóricos relativos aos modelos que se empregarán nesta memoria. Primeiramente, mostraremos a notación que utilizaremos ao longo do traballo.

A **Aprendizaxe Estatística** é unha disciplina que abrangue distintas ferramentas e técnicas para entender e predicir certos fenómenos en base a observacións dos seus datos. En función da súa natureza, o obxectivo da nosa análise e o tipo de modelo matemático utilizado para a mesma distinguimos diversas clases de metodoloxías, as cales podemos dividir en: **modelos supervisados** e **modelos non supervisados**. Os primeiros son aqueles cuxo algoritmo é adestrado mediante conxuntos de datos etiquetados. É dicir, nestas metodoloxías contamos cunha serie de observacións independentes con certas características (**variables explicativas**) que buscamos ser capaces de relacionar mediante algunha función cunhas respostas (**variable resposta**) asociadas. Notemos que o obxectivo destes modelos pode ser tanto a predición, como entender as relacións entre os datos. Por outra banda, nos modelos non supervisados contaremos unicamente con diversas observacións independentes, sen etiquetar, das variables e o seu obxectivo consiste en identificar patróns, estruturas ou relacións subxacentes aos datos proporcionados.

Dentro dos modelos supervisados atopamos os **modelos de predición**, os cales son unha clase de modelos supervisados que teñen como propósito tratar de, en base ao conxunto de variables explicativas, propoñer valores para a resposta[1]. Un exemplo disto sería coñecer a altura da descendencia, variable resposta, dunha parella baseándonos na dos proxenitores, variables explicativas. En particular, distinguimos entre dúas clases de modelos de predición: **modelos de clasificación** e **de regresión**. Os primeiros son modelos que, dada unha variable resposta categórica, tentan predicir a clase que presentará dita variable en base ás súas variables explicativas. Mentres que, nos segundos, a variable resposta sería de tipo continuo ou de valor real. Non obstante, nesta memoria traballaremos unicamente con metodoloxías de aprendizaxe supervisada.

Deseguido, presentaremos a forma xeral dun modelo de predición supervisado. Sexa $\mathbf{Y}$ a variable resposta aleatoria, continua ou discreta, que, por simplicidade, consideraremos univariante. Por outra banda, $\mathbf{X}$ será un vector aleatorio de lonxitude p que contén ás nosas variables explicativas. Consecuentemente, será da forma seguinte: $\mathbf{X} = (X_1, \ldots, X_p)$ con $p \in \mathbb{Z}^+$. O noso obxectivo será construír modelos, tanto de regresión como clasificación, para predicir a variable resposta mediante as explicativas. Para levar a cabo o noso cometido, contaremos cunha serie de observacións que virán recollidas

[1] Tamén reciben outras nomenclaturas que empregaremos nesta memoria: explicativas e explicada, regresores e obxectivo, etc

no vector $\mathbf{y} = (y_1, \ldots, y_n)^\top$ para a resposta e na matriz:

$$\mathbf{x} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix},$$

para as explicativas. Notemos que a columna de uns inicial servirá para axustar o **intercepto** dos modelos.

Suporemos, ademais, que a variable aleatoria $\mathbf{Y}$ garda algún tipo de relación coa resposta. Por exemplo, no caso da regresión, consideramos a expresión xeral seguinte:

$$\mathbf{Y} = m(\mathbf{X}) + \boldsymbol{\varepsilon}. \tag{2.1}$$

Sendo $m(x)$ a coñecida como **función de regresión** e $\boldsymbol{\varepsilon}$ a **perturbación aleatoria** descoñecida de xeito que: $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$.

2.1. Modelos paramétricos

O enfoque tradicional consiste en asumir que a función de regresión presenta unha forma específica (Hastie, Friedman & Tibshirani, 2001; James, Witten, Hastie & Tibshirani, 2013). Esta clase de modelos abarcan, tanto metodoloxías de clasificación, como de regresión. Comezaremos presentando, a modo de exemplo, unha das primeiras técnicas desenvoltas baixo este paradigma: a **regresión linear**.

2.1.1. Regresión linear

O modelo linear básico é o **Modelo de Regresión Linear Clásico** e ten a estrutura seguinte:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}; \tag{2.2}$$

sendo $\boldsymbol{\beta}$ é un vector de lonxitude $(p+1)$ formado polos coeficientes de regresión (que notaremos por $\beta_0, \beta_1, \ldots, \beta_p$), os cales buscaremos estimar. Notemos que, a expresión anterior, é un caso particular de (2.1) sendo $m(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$ en notación matricial. Esta condición é equivalente a dicir que $m(\mathbf{X}) = \mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$. Ademais, esta clase de modelos asumen certas hipóteses acerca da función de regresión e da perturbación aleatoria. En particular, serían as seguintes:

1. **Hipótese de linearidade:** A variable resposta garda unha relación linear como a descrita en (2.2) coas variables explicativas.
2. **Hipótese de esperanza condicional:** $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$, o que provoca a nulidade da esperanza das perturbacións aleatorias.
3. **Hipótese de homocedasticidade:** $\mathrm{Var}(\varepsilon_t \mid \mathbf{X}) = \mathbb{E}\left[\varepsilon_t^2 \mid \mathbf{X}\right] = \sigma^2 > 0$, $\forall t \in \{1, \ldots, n\}$.
4. **Hipótese de incorrelación:** $\mathrm{Cov}(\varepsilon_t, \varepsilon_s \mid \mathbf{X}) = 0$, $\forall t, s \in \{1, \ldots, n\}$ e $t \neq s$.
5. **Hipótese de Normalidade:** As perturbacións aleatorias, ε_t, seguen unha distribución $\mathcal{N}\left(0, \sigma^2\right)$, $\forall t \in \{1, \ldots, n\}$.
6. **Hipótese de non colinearidade:** Ningunha variable explicativa pode ser expresada como unha combinación lineal exacta das outras variables explicativas.

Notemos que as hipóteses 2, 3 e 4 serían aquelas relacionadas coa perturbación e garanten que esta sexa un **ruído branco**. Unha vez xa temos definido o modelo, estímase o valor dos coeficientes $\boldsymbol{\beta}$ a partir dalgún método de minimización do erro.

Por outra banda; as hipóteses do **Modelo de Regresión Linear Clásico** son, na práctica, moi difíciles de contrastar e mesmo que se verifiquen. Polo tanto, xorden distintas transformacións do modelo anterior que permiten flexibilizar estas condicións. Estes son os coñecidos como **Modelos de Regresión Linear Xeneralizados**; os cales permiten, por exemplo, estender a metodoloxía ao caso de clasificación (Dunn & Smyth, 2018).

2.2. Modelos non paramétricos

Tal e como comentabamos na sección anterior, os modelos paramétricos contan con fortes restrición de forma e distribución. Co fin de axustar modelos que estean libres deste tipo de hipóteses xorden os coñecidos como **modelos non paramétricos**. Nos que tamén atopamos as dúas tipoloxías de métodos: regresión e clasificación.

Os primeiros, trátanse dunha clase especial de regresión na que non se especifica directamente a forma de $m(\mathbf{X})$ (Härdle, Werwatz, Müller & Sperlich, 2004). Deste xeito podemos relaxar enormemente as condicións que impoñemos sobre o noso modelo. Nesta clase de modelos reescribiremos a Ecuación (2.1) do xeito seguinte:

$$\mathbf{Y} = m(\mathbf{X}) + \boldsymbol{\sigma}(\mathbf{X})\,\boldsymbol{\varepsilon}, \tag{2.3}$$

sendo $\boldsymbol{\sigma}(\mathbf{X})$ a varianza condicional de $\mathbf{Y}$ con respecto a $\mathbf{X}$. Notemos que ambos modelos, (2.1) e (2.3), son equivalentes sempre que se reescriban axeitadamente as hipóteses estruturais pertinentes.

Tal e como diciamos, os modelos de regresión (e por extensión, os de clasificación) non paramétricos non requiren das mesmas hipóteses que os modelos lineares xeneralizados. En particular, verifican que:

- Non se especifica unha forma paramétrica para m, unicamente requiriremos certas condicións de regularidade.
- A varianza condicional ($\boldsymbol{\sigma}(\mathbf{X})$) non é necesariamente constante.
- $\boldsymbol{\varepsilon}$ non ten, necesariamente, que ser normal.
- Tomaremos conxuntos de datos independentes, porén, tamén é posible considerar dependencia sobre as observacións.

2.3. Árbores de decisión

As **Árbores de decisión** son unha clase de modelos de predición baseados na segmentación do espacio preditor. A idea principal parte de considerar rexións tan simples que o modelo poda representarse en forma de árbore binaria (Fernández-Casal, Costa & Oviedo de la Fuente, 2024). Existen diversas metodoloxías para o seu cálculo, emporiso, centrarémonos na coñecida como **CART**. Se ben esta é a metodoloxía máis utilizada, tamén existen outros enfoques similares tanto para regresión, como para clasificación (Hothorn, Hornik & Zeileis, 2006; Kass, 1980; Loh, 2009). Para as explicacións e o desenvolvemento desta sección tomaremos como referencia Fernández-Casal e col. (2024).

Entón, buscamos partillar o espacio preditor en J rexións disxuntas $R_1, \ldots, R_J$ e, a cada unha delas, asignarémoslle unha constante: no caso da regresión, a media e, no caso da clasificación, a moda. Estas serán as que empregaremos para efectuar as nosas predicións. Agora ben, debemos establecer algún tipo de criterio, baseado na natureza da resposta, para seleccionar cales serían estas rexións. Distinguiremos entón dúas casuísticas (Fernández-Casal e col., 2024): as **árbores de regresión** e as de **clasificación**.

2.3.1. Árbores de regresión

Se a nosa variable resposta Y é continua, entón construiremos unha versión penalizada da **RSS**. Sexa $\hat{y}_{R_j}$ o valor constante que outorgamos á variable resposta en cada unha das rexións $R_1, \ldots, R_J$, para $j \in \{1, \cdots, J\}$, nas que dividimos o espazo preditor. Entón, buscamos as rexións que minimicen a expresión seguinte (Fernández-Casal e col., 2024; Hastie e col., 2001):

$$\mathbf{RSS}_\alpha = \sum_{j=1}^{t} \sum_{x_i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha t. \tag{2.4}$$

Neste caso α é un hiperparámetro non negativo que definirá a penalización sobre a aprendizaxe do modelo. A idea é que, a maior valor de α, menor tamaño da árbore. De xeito que o noso modelo non se sobreaxuste aos datos cos que foi adestrado, podendo ter máis problemas para efectuar regresións sobre conxuntos de datos distintos dos mesmos.
Non obstante, considerar todas as rexións posibles do noso problema resulta unha tarefa inviable. Polo tanto, propúxose efectuar un modelo iterativo seguindo un criterio *greedy*[2] (Fernández-Casal e col., 2024; Hastie e col., 2001):

1. Dada unha variable explicativa X_j e un punto de corte s, pódense definir os seguintes dous hiperplanos disxuntos:$R_1 = \{X : X_j \leq s\}$ e $R_2 = \{X : X_j > s\}$.

2. Seguidamente, seleccionamos os valores j e s que minimicen:

$$\sum_{x_i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{x_i \in R_2} (y_i - \hat{y}_{R_2})^2.$$

   É dicir, seleccionamos aqueles que minimicen o **RSS** do modelo en cada iteración.

3. Por último, unha vez temos a árbore completa, acometemos a fase de **poda**. Nesta fase empregaremos o criterio dado por (2.4) para eliminar, sucesivamente, nodos internos da nosa árbore. A árbore resultante deste proceso a denominaremos **subárbore**. A razón de levar a cabo esta transformación sobre a árbore completa é evitar que o noso modelo se adapte "demasiado ben" aos datos de adestramento, provocando así que presente graves problemas para efectuar predicións sobre novos conxuntos de datos. É dicir, que o modelo estea sobreaxustado aos datos de adestramento. Idealmente, buscaríamos avaliar cada subárbore posible sobre outra mostra distinta á de adestramento (xeralmente, a de validación) e ver cal delas ofrece mellores resultados. Non obstante, dada a inmensa cantidade de subárbores que poden existir, na práctica isto non resulta realizable. Por outra banda, a Ecuación (2.4) verifica que $\forall \alpha \geq 0$ existe unha única subárbore que a minimiza. Polo tanto, poderíamos efectuar unha selección dalgún tipo para este hiperparámetro α e seleccionar aquela subárbore máis pequena e, consecuentemente, máis simple que minimice a expresión. Para facelo, colápsanse nodos de xeito sucesivo (colapsando aqueles que produzan un menor incremento en (2.4)) dando lugar a unha sucesión finita de subárbores que contén a solución para todos os posibles factores de α.

O hiperparámetro óptimo α pódese obter mediante diversas metodoloxías, porén, a que se propón utilizar é a coñecida como **k-*folds cross validation*** (Fernández-Casal e col., 2024; Hastie e col., 2001). A idea detrás deste método consiste en partillar o conxunto de datos en k conxuntos aleatorios e, para cada un deles, axustar varios modelos outorgándolle distintos valores a este hiperparámetro α. Seguidamente efectúase validación cruzada, é dicir, calcúlase o erro de predición que comete cada un dos modelos sobre os elementos da mostra que non están inseridos no conxunto co que foi adestrado (os $k-1$ *folds* restantes). Deste xeito, podemos obter medidas globais do erro de predición (Fernández-Casal e col., 2024; Hastie e col., 2001). Notemos que este hiperparámetro será común a todos os modelos

[2]Un criterio *greedy* é aquel que non ten en conta como afecta a decisión acadada ao total do proceso, senón que se centra en como o fai nesa interacción en concreto.

que propoñemos nesta memoria e, adicionalmente, todas as librarías permiten efectuar esta selección de xeito automático («CatBoost Documentation», 2026; Fernández-Casal e col., 2024; Liaw, Wiener, Breiman & Cutler, 2022; Shi e col., 2025; Yuan e col., 2026).

Por último, sinalar que, xeralmente, tómase como constante $\hat{y}_{R_j}$ a media das respostas contidas na rexión R_j. É dicir: $\hat{y}_{R_j} = \frac{1}{N_j}\sum_{x_i \in R_j} y_i$, sendo N_j o número total de elementos da mostra contidos na rexión R_j. Non obstante, poderían considerarse outras métricas como, por exemplo, a mediana.

2.3.2. Árbores de clasificación

Seguindo unha idea similar ás de regresión, as árbores de clasificación buscan obter a categoría da resposta construíndo unha árbore binaria. Non obstante, neste caso, a constante asignada en cada nodo final sería a categoría modal dos elementos inseridos no nodo terminal. É dicir; escolleríamos a categoría M_j que verifique:

$$M_j = \arg\max_k \left\{\hat{p}_k^j\right\}, \tag{2.5}$$

sendo $\hat{p}_k^j$ á proporción de observacións inseridas na categoría k e na rexión R_j dada.

O resto de procesos serían totalmente análogos ao caso da regresión, porén, non podemos utilizar $\mathbf{RSS}_\alpha$ como medida do erro. Para solucionar esta problemática, propuxéronse diversas métricas. Para simplificar os cálculos, consideraremos unicamente as proporcións relativas a unha rexión R_j dada. Se supoñemos que existen K categorías, utilizaremos tres medidas do erro diferentes (Fernández-Casal e col., 2024):

- Proporción de erros de clasificación: $1 - \max_k(\hat{p}_k^j)$.
- Índice de Gini: $\sum_{k=1}^K \hat{p}_k^j(1-\hat{p}_k^j)$.
- Entropía (*cross-entropy*): $-\sum_{k=1}^K \hat{p}_k^j \log(\hat{p}_k^j)$.

Xeralmente, cada unha destas métricas emprégase en distintas etapas. A proporción de erros de clasificación utilízase no proceso de poda mentres que, na fase de crecemento, empréganse o índice de Gini ou a entropía.

Variables explicativas categóricas e limitacións

Ata agora, tratamos as casuísticas de contar con variables categóricas ou continuas como resposta. No caso das explicativas, poderíamos tamén atopar ambas as dúas situacións. A segunda delas, é directa; mentres que, para a primeira, propuxéronse dúas liñas de actuación (Fernández-Casal e col., 2024):

- Poderíamos considerar esta clase de variables como variables *dummy*. É dicir, variables indicadoras de cada categoría do preditor.
- Ordenar as categorías da variable explicativa. Idealmente, buscamos considerar todas as ordenacións posibles. Non obstante, isto último non é posible. Polo tanto habería que considerar algunha clase de criterio que permita ofrecer unha solución

Adicionalmente, os modelos *CART* presentan algunhas limitacións. Primeiramente, son modelos moi simples e interpretables pero, precisamente, isto convérteos, xeralmente, en modelos con escaso poder de predición. É por iso que englobamos as árbores de decisión dentro dos coñecidos como **métodos débiles** (Fernández-Casal e col., 2024).

2.4. Metodoloxías *bagging* e *boosting*.

Para tratar de solucionar as limitacións dos métodos débiles xorden as metodoloxías ***bagging*** e ***boosting***. A idea básica desta clase de modelos consiste en combinar distintos métodos simples para reducir a varianza de cada un deles, obtendo así unha mellora na capacidade de predición dos mesmos. En particular, as árbores de regresión, son ideais para este proceso por ser procesos extremadamente sinxelos (Fernández-Casal e col., 2024). Distinguimos dúas clases principais: **bagging** e **boosting**.

2.4.1. *Bagging*

Esta metodoloxía consiste en combinar varios modelos axustados mediante distintas mostras de adestramento e, combinando os resultados achegados por cada un deles, obter unha predición para cada observación. Realizar este procedemento permite, por unha banda, simplificar a solución e, por outra, reducir a variabilidade do mesmo.

A metodoloxía ***bootstrap*** consiste na aproximación na mostraxe dun estatístico, R (Fernández Casal, Cao & Costa, 2023). Xeralmente R vén asociado ou é función de un estimador, $T(\mathbf{X})$, dado coa fin última de obter información sobre certas características do mesmo. A idea principal consiste en aproximar a distribución da $m.a.s$ $\mathbf{X}$ e, en base a dita aproximación, simular mostras de dita variable aleatoria. Seguidamente, obtemos diferentes valores para R de xeito que podamos facernos unha idea de cal sería a súa distribución. O estatístico R pode presentar diversas formas, porén, xeralmente tómanse expresións que sexan estimadores do nesgo ou a varianza de $T(X)$ (Shao & Tu, 1995). Para máis información sobre técnicas de ***bootstrap*** ou remostraxe consultar Fernández Casal e col. (2023) ou Shao e Tu (1995).

Se ben esta clase de técnicas son utilizadas, frecuentemente, para estimar o nesgo e varianza dun estimador (Fernández Casal e col., 2023; Shao & Tu, 1995); neste caso as utilizaremos para obter distintas remostras de adestramento sobre as que axustar nosos modelos débiles. Tal e como comentabamos na Sección 2.3, no caso específico das árbores de decisión, se permitíamos a unha delas adquirir profundidade indefinidamente (mesmo permitíndolle acadar a máxima posible) producíase un efecto de sobreaxuste aos datos de adestramento. Polo tanto; mudar a mostra de adestramento para cada árbore, mesmo lixeiramente, xerará unha gran variabilidade nas predicións do modelo de cada unha das remostras (Fernández-Casal e col., 2024). Consecuentemente, se calculamos a media (para o caso da regresión) ou a moda (para o caso de clasificación) das predicións de cada unha delas permitirá reducir a varianza das mesmas. Este enfoque presenta, ademais, unha vantaxe adicional. Pois permite estimar o erro de predición, de xeito directo, obtendo aquel relativo aos coñecidos como datos ***out-of-bag***. Estes serían aquelas observacións da mostra de adestramento que quedan fóra de cada unha das nosas remostras (Fernández-Casal e col., 2024).

Bosques aleatorios

O tipo de *bagging* que emprega árbores de regresión como modelos base son os coñecidos como **Bosques aleatorios** ou ***Random Forest*** e foi proposto por Breiman (2001). O procedemento xeral para o axuste desta clase de modelos é o descrito no apartado anterior, con todo, formalizaremos este procedemento deseguido. Supoñamos, entón, que contamos cunha nova observación $\overline{\boldsymbol{x}}$ e queremos estimar o valor dunha variable resposta, $\overline{y}$. Por outra banda; consideraremos un *Random Forest* formado por T árbores con J_t follas cada unha $t \in \{1, \ldots, T\}$. Ademais; contamos cunha mostra de adestramento composta por un conxunto de n características, $\mathbf{x}$, xunto coas súas respostas correspondentes, $\mathbf{y}$ para o axuste de dito modelo. Polo tanto; cada árbore t ofrecerá unha predición, que notaremos por $\hat{y}_t$, e será a media das respostas observadas, no caso da regresión, e a categoría modal, no caso da clasificación. É dicir:

- **Regresión:**

$$\hat{y}_t(\overline{\boldsymbol{x}}) = \sum_{j=1}^{J_t} \left(\frac{1}{\sum_{k=1}^{n} \mathbb{1}_{\{\mathbf{x}_{k\cdot} \in R_j^t\}}} \sum_{k=1}^{n} y_k \mathbb{1}_{\{\mathbf{x}_{k\cdot} \in R_j^t\}} \right) \mathbb{1}_{\{\overline{\boldsymbol{x}} \in R_j^t\}}, \quad \forall t \in \{1, \ldots, T\}.$$

- **Clasificación:**

$$\hat{y}_t(\overline{\boldsymbol{x}}) = \sum_{j=1}^{J_t} M_j \mathbb{1}_{\{\overline{\boldsymbol{x}} \in R_j^t\}}, \quad \forall t \in \{1, \ldots, T\}.$$

Sendo R_j^t a rexión j-ésima da árbore t, M_j a categoría modal tal e como queda definida en (2.5), $\mathbf{x}_{k\cdot}$ a fila k-ésima da matriz $\mathbf{x}$ e y_k a k-ésima observación da variable resposta (Fernández-Casal e col., 2024; Hastie e col., 2001). Posteriormente, nos modelos de regresión, promédianse todas as predicións obtidas por cada unha das destas árbores de xeito que:

$$\overline{y} \approx \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t.$$

Se traballamos no contexto da clasificación isto non sería tan directo. Xeralmente escóllese a predición da resposta mediante o criterio do "voto maioritario" (Fernández-Casal e col., 2024; Hastie e col., 2001). Supoñamos que temos unha variable resposta categórica $\mathbf{Y}$ con K categorías, logo escolleríamos como estimación a categoría que se repita máis veces. É dicir, escolleríamos aquela categoría verificando o seguinte:

$$\overline{y} \approx \arg\max_k \left\{ \sum_{t=1}^{T} \mathbb{1}_{\{k=\hat{y}_t\}} : \forall k \in K \right\}. \tag{2.6}$$

Destes razoamentos deducimos que estes modelos contarán cun novo hiperparámetro, o número de remostras ou árbores consideradas. Para o cálculo do seu valor óptimo xéranse remostras e, unha vez non se produzan melloras significativas no erro de predición sobre os datos *out-of-bag*, detemos o proceso; quedándonos con dito número de árbores.

Non obstante, na práctica xorde unha nova dificultade. Malia efectuar unha remostraxe axeitada sobre os datos de adestramento, podemos atoparnos con construcións de árbores moi similares. Este fenómeno é o coñecido como **correlación entre árbores** e pode provocar que a redución da varianza sexa extremadamente pequena (Fernández-Casal e col., 2024). Para evitar que isto teña lugar, Dietterich (2000) propuxo introducir variabilidade na partición de cada nodo, seleccionando o punto de corte de forma aleatoria entre as mellores particións dispoñibles. Xurde, deste xeito, un novo hiperparámetro que permitirá seleccionar un número dado de variables explicativas sobre as que efectuar o corte (Fernández-Casal e col., 2024).

2.4.2. *Boosting*

Se ben o cometido da metodoloxía ***Boosting*** é similar á do *bagging*, si presentan diferenzas estruturais. O primeiro deles busca impulsar (*to boost* en inglés, de ahi o seu nome) modelos sinxelos con pouca capacidade de predición para obter un preditor axeitado. Por exemplo, as árbores de regresión con pouca profundidade son candidatos perfectos para esta clase de metodoloxía (Fernández-Casal e col., 2024). Malia que existen diversas clases de modelos que poden englobarse dentro do *Boosting*, nós centrarémonos naqueles cos que traballaremos na presente memoria. En particular, estes serían os seguintes: ***Extreme Gradient Boosting*** ou **XGBoost**, ***LightGBM*** e ***CatBoost***. Todos eles, son

metodoloxías derivadas da coñecida como: ***Gradient Boosting Machine*** ou **GBM**. Consecuentemente; comezaremos explicando en que consiste ela e, posteriormente, as demais.

A metodoloxía **GBM** foi proposta por J. H. Friedman (2001) e insírese dentro da familia dos **métodos iterativos de descenso de gradiente**[3] (Fernández-Casal e col., 2024). Esta baséase na obtención dun modelo aditivo que minimice unha función de perda, $\mathcal{L}$, dada (Fernández-Casal e col., 2024). Supoñamos que estamos traballando cunha L diferenciable. Entón a función de perdas pódese escribir, en forma aditiva, como:

$$\mathcal{L}(m) = \sum_{i=1}^{n} L\left(y_i, m(x_i)\right), \tag{2.7}$$

sendo $\mathbf{x}_{i\cdot}$ a fila i-ésima da matriz $\mathbf{x}$. Polo que, a dirección de máximo descenso, viría dada por:

$$-\nabla\mathcal{L}(m) = \left(-\frac{\partial L\left(y_i, m(\mathbf{x}_{i\cdot})\right)}{\partial m(\mathbf{x}_{1\cdot})}, \ldots, -\frac{\partial L\left(y_i, m(\mathbf{x}_{i\cdot})\right)}{\partial m(\mathbf{x}_{n\cdot})}\right). \tag{2.8}$$

Para exemplificar o proceso, consideraremos que L é o **RSS** e contamos unha única variable explicativa (é dicir, $\mathbf{x}$ quedaría definido unicamente por un vector de observacións de lonxitude n). Entón, (2.7) reescribiríase como:

$$\mathcal{L}(m) = \sum_{i=1}^{n} L\left(y_i, m(x_i)\right) = \sum_{i=1}^{n} \frac{1}{2}\left(y_i - m\left(x_i\right)\right)^2.$$

Analogamente, reescribiríamos (2.8) sabendo que:

$$-\frac{\partial L\left(y_i, m(x_i)\right)}{\partial m(x_i)} = y_i - m(x_i) = r_i.$$

É dicir, as compoñentes da dirección de máximo descenso serían ás dos residuos de cada iteración (Fernández-Casal e col., 2024). A idea entón é, mediante un proceso iterativo, ir aproximando paulatinamente o valor desta dirección de máximo descenso e quedarnos co modelo resultante de sumar todos os obtidos ao longo das iteracións multiplicados por un parámetro de penalización λ. De aquí en diante notaremos a $-\frac{\partial L(y_i, m(x_i))}{\partial m(x_i)}$ por r_i para facilitar os cálculos.

Se ben m pode vir dado por calquera modelo de **ML**, porén, uns dos máis utilizados son as árbores de decisión (en particular, as de regresión). Este tipo de **GBM** son denominados ***Gradient Boosting Decision Trees*** ou **GBDT** e foron propostos, inicialmente, polo propio J. H. Friedman (2001). A idea consiste en utilizar árbores pequenas e con escaso poder de predición para axustar o modelo en cada iteración. Tendo todo isto en conta, o algoritmo xeral para unha metodoloxía de tipo **GBDT** sería o seguinte (Fernández-Casal e col., 2024; J. H. Friedman, 2001):

1. Seleccionamos un número de iteracións B, un parámetro de regularización λ e o número de cortes d de cada árbore.
2. Establecemos unha predición inicial constante, $\hat{m}^0 = 0$ e fixamos $-\nabla\mathcal{L}(m) = -\mathbf{y}$, é dicir, $r_i = y_i$.
3. Para cada $b = 1, \ldots, B$; repetir:
   a) Axustar unha árbore de regresión, $\hat{m}^b$ con d cortes para $(\mathbf{X}, \mathbf{r})$. É dicir, con variable explicativa $\mathbf{X}$ e resposta $\mathbf{r} = (r_1, \ldots, r_n)^\top$.
   b) Calcular unha versión regularizada da árbore de regresión: $\lambda\hat{m}^b$.
   c) Actualizar as direccións, de xeito que: $r_i \leftarrow r_i - \lambda\hat{m}^b(x_i)$.

[3] Para máis información consultar Nocedal e Wright (2006).

4. Unha vez efectuadas as B iteracións, calculamos o modelo *boosting* do xeito seguinte:

$$\hat{m}(\mathbf{x}) = \sum_{b=1}^{B} \lambda \hat{m}^b(\mathbf{x}).$$

Tal e como observamos, esta metodoloxía dependerá de tres hiperparámetros: B, λ e d. Os cales poden ser seleccionados de xeito óptimo mediante os métodos habituais.

O propio J. Friedman (2002) propuxo unha mellora no seu modelo, que incorpora unha técnica de remostraxe similar á do *bagging*. Esta alternativa é a coñecida como ***Stochastic Gradient Boosting*** ou **SGD**. A diferenza con respecto ao **GBM** é que, en vez de considerar toda a mostra de adestramento para axustar o modelo dado por $(\mathbf{X}, \mathbf{r})$, consideraremos unha remostra aleatoria deles (Fernández-Casal e col., 2024; J. Friedman, 2002). Asi con todo, esta variación incorporaría un novo hiperparámetro que controlaría a fracción dos datos de adestramento empregados no axuste (Fernández-Casal e col., 2024). As principais vantaxes desta nova metodoloxía son que permite reducir a varianza e os tempos de cómputo do proceso iterativo(Fernández-Casal e col., 2024).

Extreme Gradient Boosting

O ***Extreme Gradient Boosting*** ou **XGBoost** trátase dunha variante do **SGB** proposta por Chen e Guestrin (2016). Este procedemento incorpora ao **SGB** diversas melloras: engade na función de perdas unha penalización por complexidade e a regulariza empregando a súa Hessiana[4], o que evita o sobreaxuste, e incorpora hiperparámetros adicionais (Fernández-Casal e col., 2024). En particular, este enfoque baséase na variante do **SGB** que emprega árbores de regresión.

Sexa $m(\mathbf{x}) = \sum_{b=1}^{B} m^b(\mathbf{x})$ un modelo aditivo co que queremos estimar os valores de $\mathbf{Y}$. Chen e Guestrin (2016), propoñen a seguinte función de perdas:

$$\mathcal{L}(m) = \sum_{i=1}^{n} \left[L\left(y_i, m(\mathbf{x}_{i\cdot})\right) + \sum_{b=1}^{B} \Omega(m^b(\mathbf{x}_{i\cdot})) \right], \tag{2.9}$$

sendo $\Omega(m) = \gamma d + \frac{1}{2}\lambda \|w\|^2$, sendo d análogo ao do **GBM**, w son os pesos das follas de cada unha das árbores e λ e γ hiperparámetros de penalización adicionais. Neste caso, γ penaliza o crecemento de cada árbore e λ o peso das follas de cada unha delas. Tamén son coñecidos como parámetros de penalización L_1 e L_2, respectivamente.

A idea do algoritmo iterativo é similar ao caso anterior, non obstante é moito máis complexo e, nalgúns casos, require de computación paralelizada (Chen & Guestrin, 2016). Polo tanto, non nos estenderemos nesta liña.

LightGBM

O ***LightGBM*** é unha variante do **XGBoost** con árbores de decisión, proposta por Ke e col. (2017), cuxo obxectivo central é reducir os tempos de computación do mesmo. Se ben a idea básica é totalmente análoga á do **XGB**, búscase reducir o tempo computacional na construción de árbores mediante as técnicas ***Gradient-based One-Side Sample*** (**GOSS**) e ***Exclusive Feature Bundling*** (**EFB**) (Ke e col., 2017).

Comezaremos presentando o **GOSS**. Unha das maiores vantaxes que ofrecen os modelos ***AdaBoost***[5] é que permite asignar pesos ás observacións mal clasificadas, isto é un bo indicador da importancia das instancias dos nosos datos (Ke e col., 2017). Non obstante; este tipo de axuste non se pode

[4] Esto esixe, necesariamente, que a función sexa, cando menos, dúas veces diferenciable (Fernández-Casal e col., 2024).
[5] Para máis información consultar Fernández-Casal e col. (2024) e Schapire (2013).

implementar, directamente, nos modelos de tipo **GBM** e, en consecuencia, tampouco nos seus derivados. Por iso, Ke e col. (2017) propoñen o que eles denominan modelo **GOSS**. Primeiramente, notan como nos modelos **GBDT** (en particular, nos **XGBoost**) o gradiente de cada unha das observacións pode achegar información sobre o ben clasificada que está a mesma. En particular, notan que se dito gradiente, avaliado nunha instancia dada, é pequeno; entón o erro cometido na clasificación da mesma tamén o é. Dito doutro xeito, noso modelo tenderá a clasificar correctamente a instancia anterior. Polo tanto, Ke e col. (2017) propoñen considerar para o adestramento de cada iteración as observacións con gradientes máis elevados e efectuar unha remostraxe sobre as restantes. Deste xeito, redúcese o conxunto de datos sobre o que se está a traballar facendo que os tempos de execución se reduzan enormemente. Para efectuar esta selección de observacións, ordénanse as instancias en base ao valor absoluto do gradiente e selecciónase o $a \times 100\,\%$ delas e, posteriormente, efectúa unha remostraxe aleatoria sobre o $(1-a) \times 100\,\%$ restante na que se selecciona o $b \times 100\,\%$ delas. Notemos que, de aplicar esta metodoloxía, xorden dous hiperparámetros máis correspondentes con a e b. Para finalizar, multiplican por unha constante $\frac{1-a}{b}$ todos os gradientes das observacións que non foron incluídas na remostra de adestramento da iteración anterior. Deste xeito, evítase que existan instancias "infra-adestradas"[6].

Por outra banda, o **EFB** proposto por Ke e col. (2017) trata de reducir o número de variables explicativas ou *features* nas que se busca dividir cada unha das árbores de decisión. Non obstante, esta redución debe efectuarse de xeito que perxudique o menos posible ás predicións do noso modelo. Primeiro, Ke e col. (2017) notan como, nos modelos onde $\mathbf{X}$ ten unha dimensión moi elevada poden presentar certa "dispersión" ou ***sparsity***[7]. En particular; poden existir variables denominadas *mutuamente excluíntes*, é dicir, que non toman valores distintos de cero simultaneamente. Tendo isto en conta, Ke e col. (2017) propuxeron agrupar en paquetes ou ***bundles*** aquelas variables que fosen *mutuamente excluíntes* para, así, axilizar o proceso de selección de variables no axuste de cada árbore. Para logralo, utilizan técnicas relativas á teoría de grafos[8]. En resumo, reduce enormemente o número de características a considerar (e, consecuentemente, o tempo de computación) sen perder precisión.

En conclusión, o modelo ***LightGBM*** reduce enormemente o tempo de computación do **XGBoost** evitando comprometer, en gran medida, a precisión do mesmo.

CatBoost

O ***CatBoost*** foi proposto por Dorogush e col. (2017) e engade dúas innovacións clave con respecto a outros modelos *boosting*.

Primeiramente incorpora o que denominan ***Ordered Boosting***. Esta metodoloxía busca poñer solución a unha das problemáticas dos modelos *boosting* que eles mesmos denominan ***prediction shift***[9]. Nos modelos de *boosting* usuais, cada nova árbore axústase para predicir o gradiente negativo do modelo na iteración dada. É dicir; o novo modelo combinado da iteración b-ésima, que foi construído a partir dunha observación dada, recalcúlase volvendo a considerar dita instancia. Isto produce no modelo final un nesgo sobre aquelas que xa foron utilizadas. Para solucionalo, Dorogush e col. (2017) comezan xerando unha (malia que poden ser máis) permutación dos datos do conxunto de adestramento. Seguidamente calculan unha serie de modelos que denominan "modelos soporte" de xeito que o i-ésimo deles estea adestrado cos $i-1$ primeiros elementos da permutación. Polo tanto, se a permutación contaba con K elementos, existirán $M_1, \cdots, M_K$ modelos soporte. Seguidamente, para calcular o r_i da iteración dada, utilizaremos o modelo M_{i-1}; de xeito que non se utilice a resposta da observación i-ésima e evitando así o nesgo anteriormente mencionado.

[6] Para máis información, consultar Ke e col. (2017).

[7] Neste contexto, dicimos que unha matriz presenta **dispersión** ou **sparsity** sempre cando a maioría dos seus elementos presentan valores nulos ou insignificantes.

[8] Para máis información consultar Ke e col. (2017)

[9] Para máis información consultar Dorogush e col. (2017).

Outra novidade que introduce é ***Ordered Target Encoding*** que, aplicando unha metodoloxía similar ao *Ordered Boosting*, trata de solucionar as problemáticas derivadas de considerar variables explicativas categóricas. Cando estamos ante este tipo de variables temos varias formas de actuar. A primeira delas consiste en crear variables indicadoras *dummy*[10], este procedemento coñécese como ***one-hot encoding***. Con todo, este enfoque pode producir un gran aumento da dimensionalidade e, en consecuencia, tempos de computación máis elevados. Unha forma de evitar esta problemática é agrupar as diferentes categorías en clústeres e, posteriormente, aplícase *one-hot encoding* sobre eles. En particular, destacan as que se coñecen como ***target statistics***, que consisten en xerar unha nova variable que inclúa as estimacións do valor da variable resposta esperado para cada categoría. Non obstante, isto pode producir a coñecida como **fuga de datos** ou ***data leakage***[11]. Polo que, para solucionar esta problemática, Dorogush e col. (2017) propoñen considerar as permutacións empregadas no *Ordered Boosting* e, dada unha instancia $\mathbf{x}_{i\cdot}$, calcular o valor esperado da variable resposta das observacións correspondentes á permutación $(i-1)$-ésima e que pertenzan á mesma categoría que $\mathbf{x}_{i\cdot}$. Isto permite eliminar a fuga de datos, o que resulta en modelos máis robustos e con mellor xeneralización.

Por último, efectúa algunhas outras mellorías entre as que destacan: o uso de ***Oblivious Decision Trees***, que son árbores de decisión menos propensas ao sobreaxuste e de computación máis eficiente; a ***combinación de características*** de xeito automático, que non é máis que efectuar o *clustering* de categorías que comentabamos anteriormente, e a incorporación de submostraxe con ***Bayesian Bootstrap***, cuxos pormenores poden estudarse en Dorogush e col. (2017).

2.4.3. Librarías para implementar os modelos

Nesta sección exporemos os paquetes do software *R* que empregaremos para axustar os nosos modelos xunto con unha explicación dos seus parámetros principais (0xdata, Inc., 2013; «CatBoost Documentation», 2026; Fernández-Casal e col., 2024; Liaw e col., 2022; Shi e col., 2025; Yuan e col., 2026):

- *Random Forest:* (`h2o`) `h2o.randomForest()`
  - `ntrees`: É o número de árbores do modelo.
  - `mtry`: Refírese ao número de preditores a incluír en cada nodo.
  - `maxnodes`: Este parámetro controla a profundidade máxima admitida para cada árbore.
  - `nodesize`: Sería o número mínimo de observacións permitidas nun nodo terminal.
  - `sampsize`: Recolle a porcentaxe de observacións tomadas para adestrar cada árbore.
- *XGBoost:* (`xgboost`) `xgb.train()`
  - `nrounds`: Controla o número de rondas e dito valor será 1000.
  - `eta`: Taxa de aprendizaxe.
  - `gamma`: Redución mínima na perda necesaria para realizar unha división adicional nun nodo da árbore.
  - `min_child_weight`: Define o mínimo sumatorio de pesos das instancias necesario nun nodo fillo para que permita unha división adicional.
  - `max_delta_step`: Límite do paso máximo permitido na estimación de pesos das follas durante o adestramento.
  - `max_depth`: Controla a profundidade máxima de cada árbore.

[10] Para cada categoría creamos unha nova variable que tome valor 1 se a observación insírese nesa categoría e 0 noutro caso.

[11] O ***data leakage*** ten lugar cando algunha das nosas variables explicativas aporta información a posteriori do evento que describe a nosa variable resposta.

- `subsample`: É a fracción de mostras do conxunto de datos empregadas para adestrar cada árbore.
- `colsample_bytree`: Controla a proporción de variables explicativas a considerar en cada árbore. Un valor baixo permite evitar o sobreaxuste.
- `colsample_bylevel`: Controla a fracción de variables que se consideran en cada nivel de profundidade ao construír unha árbore. Aplícase cada vez que se acada un novo nivel de profundidade na árbore.
- `scale_pos_weight`: Permite outorgar penalizacións aos valores da clase maioritaria nos modelos de clasificación. Pode ser especialmente útil para casos desbalanceados.
- `early.stop.round`: Acciona a interrupción do comando en caso de non producirse melloras significativas durante o número de rondas especificado.

- *LightGBM:* (`lightgbm`) `lgb.train()`
  - `nrounds`: Controla o número de rondas.
  - `learning_rate`: Define a taxa de aprendizaxe.
  - `min_gain_to_split`: Controla a ganancia mínima requirida para efectuar unha división nun nodo dunha árbore.
  - `num_leaves`: Número máximo de follas por árbore.
  - `min_data_in_leaf`: É o número mínimo de datos dunha folla.
  - `lambda_l1`: Parámetro de regularización L_1.
  - `lambda_l2`: Parámetro de regularización L_2.
  - `max_depth`: Indica a profundidade máxima das árbores.
  - `bagging_freq`: Frecuencia do *bagging*. Tomará valor 1 para activarse e 0 noutro caso.
  - `feature_fraction`: Fracción de variables explicativas a utilizar en cada iteración.
  - `scale_pos_weight`: Ponderación para a clase positiva desbalanceada.
  - `early.stop.round`: É análogo ao caso *XGBoost*.
- *CatBoost:* (`catboost`) `catboost.train()`
  - `iterations`: Número de iteracións *boosting*.
  - `learning_rate`: Fixa a taxa de aprendizaxe do modelo.
  - `depth`: Controla a profundidade máxima das árbores de decisión.
  - `l2_leaf_reg`: Parámetro de regularización L_2 aplicado ás follas para evitar sobreaxuste.
  - `rsm`: Indica a porcentaxe de regresores que se utilizarán aleatoriamente en cada división para construír as árbores.
  - `class_weights`: Fixa multiplicadores para os pesos da clase positiva. É similar aos parámetros de pesos dos modelos anteriores.
  - `early_stopping_rounds`: Ten a mesma atribución que `early.stopping.rounds`.

2.5. Xustificación do método

Nesta sección trataremos de xustificar a adecuación do enfoque exposto nesta memoria. Para facelo consideraremos, primeiro, a casuística dos modelos *Random Forest* e, posteriormente, as metodoloxías de *gradient boosting*.

Lembremos que, nas árbores de clasificación *CART*, cada unha das follas ou rexións (R_j) nas que partillabamos o espazo de predición devolvía a categoría modal do nodo terminal. Os *Random Forest* aplicábase o criterio do "voto maioritario", escollendo a categoría dada por (2.6). Este proceso, se ben resulta axeitado para traballos de clasificación propiamente ditos, non serviría como estimador da probabilidade de cada clase por cuestións obvias. Non obstante; pódese propoñer outro tipo de reconto coñecido como ***soft-voting*** o cal consiste en ponderar a suma das proporcións dos nodos terminais de cada unha das árbores (Malley, Kruppa, Dasgupta, Malley & Ziegler, 2012). Deste xeito, se traballamos cunha variable resposta categórica dicotómica con clases "1" e "0", poderíamos obter unha estimación consistente da probabilidade de que $Y_i = 1$ condicionado a que $\mathbf{x}_{i\cdot}$ mediante esta técnica (Malley e col., 2012).

Seguidamente explicaremos en que consiste. Sexa $\mathbf{Y}$ unha variable dicotómica con categorías "1" e "0" que buscamos predicir a partir dunha observación con variables explicativas dadas polo vector $\overline{\boldsymbol{x}}$ e consideremos que o noso *Bosque Aleatorio* está formado por T árbores con J_t follas cada unha $t \in \{1, \ldots, T\}$. Entón, podemos obter a predición da probabilidade da clase "1" do xeito seguinte (Malley e col., 2012):

$$\hat{P}_1^t(\overline{\boldsymbol{x}}) = \sum_{j=1}^{J_t} \hat{p}_1^j = \sum_{j=1}^{J_t} \left(\frac{1}{\sum_{k=1}^{n} \mathbb{1}_{\left\{\mathbf{x}_{k\cdot} \in R_j^t\right\}}} \sum_{k=1}^{n} \mathbb{1}_{\{y_k=1\}} \mathbb{1}_{\left\{\mathbf{x}_{k\cdot} \in R_j^t\right\}} \right) \mathbb{1}_{\left\{\overline{\boldsymbol{x}} \in R_j^t\right\}}, \quad \forall t \in \{1, \ldots, T\}.$$

Neste caso $\hat{p}_1^j$ é a proporción de observacións da clase "1" na folla j, $\mathbf{x}_{k\cdot}$ a fila k-ésima da matriz $\mathbf{x}$ e y_k a observación da variable resposta correspondente a dita fila. Seguidamente, calcularíase o promedio destas probabilidades:

$$\hat{P}_1 = \frac{1}{T} \sum_{t=1}^{T} \hat{P}_1^t. \tag{2.10}$$

Malley e col. (2012) proban no seu artigo que o valor $\hat{P}_1$ é un estimador consistente de $\mathbb{P}(\mathbf{Y} = 1 | \mathbf{X} = \overline{\boldsymbol{x}})$. Ademais, poderíamos calcular tamén a predición da categoría do xeito seguinte:

$$\overline{y} \approx \begin{cases} 1 & \text{se } \hat{P}_1 \geq 0.5, \\ 0 & \text{se } \hat{P}_1 < 0.5. \end{cases}$$

Estas probabilidades para cada unha das observacións pódense calcular considerando o que Malley e col. (2012) denominan un modelo `regRF`. Notemos que, baixo esta notación, podemos calculalo sen máis que manter a variable resposta dicotómica como numérica dentro dun modelo *Random Forest* (Liaw e col., 2022).

No caso dos modelos de *gradient boosting* o procedemento sería lixeiramente distinto. Tanto o *XGBoost*, como o *LightGBM* e o *CatBoost* poden verse como **modelos aditivos**. Definimos un modelo aditivo como un modelo que presenta a forma seguinte (J. Friedman, Hastie & Tibshirani, 2000; Hastie e col., 2001):

$$\mathbb{E}[\mathbf{Y} | \mathbf{X}] = c_0 + \sum_{b=1}^{B} c_b f_b(X), \tag{2.11}$$

sendo f_m funcións suavizadas. En particular, no caso do **GBDT**, podemos estender esta definición considerando $m^b = f_b$ e $\lambda = c_b$ $\forall b \in \{1, \ldots, B\}$. En particular, cando estamos ante modelos de clasificación verifícase o seguinte (J. Friedman e col., 2000):

$$\mathbb{E}\left[\mathbb{1}_{\{\mathbf{Y}=k\}} | \mathbf{X} = \mathbf{x}\right] = \mathbb{P}(\mathbf{Y} = k | \mathbf{X} = \mathbf{x}).$$

Polo tanto, se estamos ante un problema de clasificación con resposta dicotómica, podemos aplicar a transformación *logit* sobre (2.11) combinado co modelo *gradient boosting* de xeito que (J. Friedman

e col., 2000):

$$\log\left(\frac{\mathbb{P}\left(\mathbf{Y}=1|\mathbf{X}=\mathbf{x}\right)}{1-\mathbb{P}\left(\mathbf{Y}=1|\mathbf{X}=\mathbf{x}\right)}\right)=\sum_{b=1}^{B}\lambda m^b(\mathbf{x}).$$

En consecuencia, podemos aproximar a probabilidade da clase positiva como (J. Friedman e col., 2000):

$$\mathbb{P}\left(\mathbf{Y}=1|\mathbf{X}=\mathbf{x}\right)\approx\frac{e^{\sum_{b=1}^{B}\lambda m^b(\mathbf{x})}}{1-e^{\sum_{b=1}^{B}\lambda m^b(\mathbf{x})}}. \tag{2.12}$$

Precisamente estas probabilidades son as que obtemos a partir da función `predict` de *R* co argumento `type="response"` (Yuan e col., 2026).

En resumo, a nosa idea será axustar modelos de clasificación binarios dependentes da hora de envío. Posteriormente, efectuaremos predicións sobre os clientes aos que pretendamos enviarlles o *email* modificando a variable relativa á hora de envío. Finalmente e, para cada usuario, seleccionaremos aquela hora que maximice estas probabilidades.

Capítulo 3

Análise exploratoria dos datos

No presente capítulo trataremos os aspectos relativos á construción do *dataset* e as variables que empregaremos para o desenvolvemento dos modelos, así como a identificación das variables resposta e explicativas.

3.1. Orixe dos datos

Primeiramente, presentaremos as tres táboas principais das que partiremos para construír o noso conxunto de datos de adestramento:

- **Táboa de contactos**: Nesta táboa recóllense todos os datos dispoñibles relativos ás comunicacións da empresa co cliente. Por exemplo: vía axendas comerciais, *banner*, *push*, etc. En particular, estamos interesados naquelas efectuadas vía *email*. As columnas máis relevantes son as seguintes:
  - **CONTACTO_ID**: É o identificador único de cada *email* que é enviado.
  - **PROGRAMA**: Indica o segmento no que se engloba o envío en función da intencionalidade coa que foi emitido. Por exemplo, algúns destes programas son: `SEGUROS_CAPTACION`, `ADMINISTRATIVAS` ou `EXPERIENCIA_CLIENTES`.
  - **CLIENTE_ID**: Identificador único de cada cliente que é contactado.
  - **CANAL**: Indica a través de que servizo foi efectuado o contacto. No noso caso, centrarémonos naqueles realizados vía *email*.
  - **RESPUESTA_EMAIL**: Recolle o estado final da comunicación. No caso dos emails pode tomar tres estados posibles: *Open*, se o *email* foi aberto; *Email Click*, se o cliente accedeu a algunha ligazón adxunta ao correo, e toma valor nulo noutro caso. Ao longo de toda a memoria consideraremos que todo correo *clickado* será tamén lido.
  - **FC_RESPUESTA**: Aparece reflectida a data e a hora do día na que se efectuou a lectura ou *clickado* do *email*. En caso de que non tivese lugar algún deses eventos figuraría como un dato nulo.
  - **ESTADO_CONTACTO**: Indica en que instante da comunicación se atopa a mesma (*failed*, *canceled*, *sent*, ...).
  - **DELIVERY_LABEL**: Contén breves descricións do asunto do envío. Asociada a esta, existen tamén outras columnas: **DELIVERY_ID** e **DELIVERY_INTERNAL_NAME**. A primeira delas identifica de xeito único cada asunto dos envíos e, a segunda, identifica a nomenclatura interna que se outorga ao mesmo.
  - **FC_CONTACTO**: Indica cando se produciu o envío do *email* en formato "DD-MM-YYYY HH:MI:SS.s(6)". É dicir, ven recollida como `timestamp(6)`.

- **CAMPAIGN_LABEL**: Contén a etiqueta da campaña que motiva o envío do *email*. A ela está asociada tamén a columna **CAMPAIGN_DESCRIPTION**, que contén unha breve descrición do motivo da campaña.
- **CATEGORIA**: Esta columna identifica a categoría na que se clasifica o *email*. No caso desta clase de comunicacións distinguimos as seguintes: `Administrativa`, `Comercial-Promocional`, `Comercial-Reactivacion`, `Comercial-Vinculacion`, `Imagen`, `Encuesta`, `Comercial-Captacion`, `Comercial-Retencion` e `Comercial-Vencimiento`.

- **Táboa de datos do cliente**: Nesta táboa agrúpanse datos xerais sobre clientes do banco. Están tomados de xeito que se agrupe información relativa a un cliente nun determinado momento. Isto permite ter unha visión xeral da situación do cliente con respecto á entidade nun instante determinado. Notemos que esta táboa contén 343 variables distintas; polo que, se ben nesta memoria exporemos aquelas que resultan máis axeitadas para o noso estudo, todas elas foron analizadas. As súas columnas principais son:
  - **CLIENTE_ID**: Trátase do identificador único de cada cliente que permitirá conectala coa táboa anterior.
  - **FECHA**: Indica a fecha na que se recollen estas informacións relativas ao cliente.
  - **ESTADO_CLIENTE**: Indica cal é o estado dese usuario con respecto á entidade financeira (cliente activo, inactivo, perdido ou non activado).
  - **SEXO**: Contén o sexo do cliente.
  - **EDAD**: A idade do cliente.
  - **SEGMENTO_ID**: Indica o segmento vital no que a entidade sitúa ao cliente (menor, promotor autónomo, autónomo ou con profesión liberal, novo, adulto, senior, non asignado).
  - **VINCULACION_TRANS_CT**, **VINCULACION_NEG_PT** e **VINCULACION_NEG_CT**: Son distintas métricas da vinculación do cliente coa entidade. A transaccional mídea baseándose nos ingresos, uso de tarxetas e seguros ou domiciliacións do cliente; mentres que a de negocio está baseada nos produtos da entidade que o usuario ten contratados. Os sufixos PT e CT indican se dita información foi calculada unicamente cos datos do primeiro titular do contrato ou para calquera deles.
  - **MESES_ANTIGUEDAD_CLIENTE**: Indica cantos meses leva o usuario sendo cliente da entidade.
  - **IN_DIGITAL**: Clasifica ao usuario en función da cantidade e o uso que fai dos servizos dixitais da empresa. Toma valores de 0 a 3, sendo o 0 un cliente que non ten activo ningún servizo. Notemos que non se trata dunha medida real, senón que é unha estimación baseada nas súas interaccións. Podería suceder, por exemplo, que fose outra persoa a encargada de xestionar eses servizos polo cliente.
  - **NIVEL2**: Esta toma os valores *"FAMI"* ou *"AUTO"*, indicando se o cliente é autónomo ou non.
  - **IMP_NOMINA_SUA** e **IMP_PENSION_SUA**: Miden as medias dos importes de nómina e pensión, respectivamente, nos últimos 4 meses.
  - **SALDO_MEDIO_CLIENTE_CT**: Saldo medio do cliente como calquera titular.

- **Táboa de datos de acceso a banca electrónica e móbil**: Aquí efectúanse distintos cruces entre táboas e cálculos co obxectivo de obter métricas mensuais sobre os accesos a banca electrónica e móbil. Consta das seguintes variables:
  - **CLIENTE_ID**: Trátase do identificador único de cada cliente que actuará como clave primaria.

- **FECHA**: A data do mes na que o usuario accede, tanto a banca móbil, como a banca electrónica. Agruparemos os datos con respecto a esta variable e a anterior para obter as métricas mensuais.
- **HORA_ACCESO**: Recolle o intervalo horario no que se produciu cada un dos accesos. Notemos que se tratarían dos 24 intervalos dunha hora nos que se pode dividir o día. Cada un deles será univocamente identificado mediante o número enteiro que representa o extremo inferior do intervalo $(0, \ldots, 23)$. Por exemplo, ao intervalo de 8:00 a 9:00 corresponderíalle o enteiro 8.
- **CANAL**: Indica a canle pola que se efectuou a comunicación. Tomará os valores 1 e 2, en función de si se accedeu a banca móbil ou electrónica.

A partir destas tres táboas construiremos unha máis complexa, que denominaremos `objeto`. A idea detrás desta nova táboa é almacenar todos os datos dos *emails*, xunto coa información do usuario recollida na táboa de datos do cliente e na de accesos a banca dixital nun período inmediatamente anterior ao instante do envío. De tal xeito que, para cada rexistro, contemos coa información do cliente máis actualizada posible (sen superar, en ningún caso, á data de envío do *email*).

Seguidamente, aplicaremos diversos filtros sobre os datos co fin de acoutar o tamaño dos mesmos e eliminar aqueles que poidan supoñer algunha clase de perda de información ou a aparición de resultados anómalos. Comezaremos, entón, xustificando estes filtros efectuados sobre os nosos datos en función da táboa de orixe. Para a *Táboa de contactos*:

- Tomaremos unicamente aquelas comunicacións que se produzan vía *email*, que fosen enviados correctamente e que non formen parte de ningunha simulación ou proba efectuada polo equipo.
- Ademais, a fin de traballar cun conxunto limitado de datos e que estes sexan representativos, acoutaremos a variable *FC_CONTACTO* nunhas datas específicas. O obxectivo deste filtro é, por unha banda, reducir o número de rexistros empregados para o adestramento dos nosos modelos a un tamaño asumible e, por outra, para evitar contar con datos incompletos dos usuarios. As datas escollidas para acoutar esta variable son: "20-11-2024" e "20-11-2025". O motivo de escoller estas datas é que permiten contar con toda a información recollida ao longo dun ano e con datos recentes sobre as características e o estado do cliente.
- Non teremos en conta *emails* que non fosen debidamente clasificados. É dicir, cuxa *CATEGORIA* tome valor nulo. O motivo é que este feito produciría unha perda de información importante.
- Eliminaremos, tamén, unha serie de correos que, pola natureza do envío dos mesmos, non van achegar información relevante aos nosos modelos. Primeiramente, hay un conxunto de comunicacións que se envían de xeito automático sempre que se interrompe algún trámite de contratación, solicitude dalgún produto ou se comete algún erro nalgunha transferencia online. Polo que, ao ser estes *emails* automáticos, non proporcionarán moita información sobre certos hábitos do noso cliente. Finalmente, non teremos en conta aqueles *emails* relacionados con *ABANCA Empresas*. Se ben podería ser interesante traballar con estes datos, sería necesario axustar un modelo aparte.
- Tamén eliminaremos os *emails* daqueles clientes que non ofrecesen ningunha lectura ou *clickado* durante o período de observación. O motivo disto é que, probablemente, estariamos ante un perfil de cliente que, sistematicamente, ignora calquera comunicación da entidade vía correo electrónico e, polo tanto, non achegará información sobre cal sería a súa hora ideal de contacto.

Prosigamos agora con aqueles filtros que aplicaremos sobre a *táboa de datos do cliente* que deseñamos:

- Centrarémonos, comercialmente, no rango de idades comprendido entre os 18 e 75 anos. Polo que consideraremos, tan só, os *emails* relativos a el.

- Impoñemos que o cliente leve xa un tempo vinculado á empresa (máis de 3 meses), de xeito que podamos contar coa maior cantidade de información posible.
- Lembremos que *ESTADO_CLIENTE* é unha variable que indica cal é a posición do cliente con respecto á entidade. As siglas: "PE", "SA" e "NA"; correspóndense con clientes que, ou ben están en proceso de desligarse da entidade, ou ben non están aínda debidamente incorporados. Consecuentemente, eliminaremos eses rexistros para evitar perdas de información.
- Aqueles clientes dos que, por algún motivo, non coñezamos o seu segmento dixital, *IN_DIGITAL*, serán eliminados.

Unha vez efectuados estes filtros sobre a nosa táboa `objeto`, contaremos cun *dataset* resultante con case 32 millóns de rexistros e, sobre el, traballaremos de aquí en diante.

3.2. Análise descritiva dos datos de contacto

A continuación efectuamos unha análise exploratoria dos nosos datos unha vez aplicados os filtros descritos no apartado anterior. A intención de elaborar esta análise é entender as relacións e estruturas presentes nos datos cos que vamos a traballar ao longo dos seguintes capítulos e definir as variables resposta dos modelos.

Esta sección foi censurada por motivos de confidencialidade. Nela efectuabamos unha análise descritiva inicial dos datos cos que contabamos.

3.3. Variables

O obxectivo desta sección é identificar variables potenciais e efectuar *feature engineering* para definir outras que poidan ser relevantes. Notemos que perseguimos un obxectivo dobre: por unha banda, buscamos construír variables que permitan mellorar a predición dos nosos modelos e, por outra, debemos ter en conta que sexan capaces de recomendar unha hora favorita concreta para o envío do *email*. Consecuentemente, buscaremos ser capaces de equilibrar rendemento e aplicabilidade nos nosos modelos.

3.3.1. Variable resposta

Recordemos que o obxectivo do proxecto é seleccionar, de xeito individualizado para cada cliente, unha hora idónea para o envío de *emails*. Isto ten un propósito dobre: mellorar a experiencia do usuario ao recibir as comunicacións en momentos do día nos que, a priori, goza dunha maior dispoñibilidade, evitando así as lecturas "sistemáticas" nas que o suxeito puido ter marcado como lido ou aberto a mensaxe sen deterse a estudar o seu contido ou que o correo quede "soterrado" na bandexa de entrada, e, por outro lado, aumentar a proporción de interaccións do cliente coas comunicacións da entidade. O enfoque proposto nesta memoria co fin de resolver o problema anterior consiste na creación modelos para predicir a probabilidade de que un consumidor dado interactúe cun *email* enviado nunha hora determinada. Unha vez contemos con dita predición, seleccionaremos aquela hora que maximice a probabilidade anterior.

Tal e como vimos, existen distintas respostas posibles en base ao que consideramos como interacción. Non obstante, indicabamos tamén a dificultade que entrañaba axustar un modelo con variable resposta os *clicks*. Consecuentemente; centrarémonos nas lecturas, engadindo á nosa base de datos a columna *LEIDO*. A cal tomará valores "1" e "0", en función de si se satisfai a condición dada ou non. Por outra banda, tamén comentabamos que sería interesante esixir a condición de que o *email* fose lido nun lapso curto de tempo. Deste xeito, buscaríamos proporcionar a probabilidade de lectura a

curto prazo do cliente. A causa destas reflexións proponse, tamén, a creación dun modelo que trate de predicir a probabilidade de ler o *email* nun período de, como moito, unha hora e media máis tarde do envío. Deste xeito a variable resposta, que será denominada **LEIDO_CP** na base de datos, tomará valor "1" sempre que se verifiquen as condicións previas e "0" noutro caso. É necesario subliñar que este enfoque tampouco sería definitivo; pois a variable resposta resultará naturalmente nesgada polo patrón actual de envíos da entidade, tal e como mostramos na Sección 3.2.

Para a construción da variable anterior é necesario coñecer o período de tempo que pasa dende o instante que se envía o *email* ata que se interactúa con el. Dita información será almacenada en **RETARDO** e estará medida en horas. O *Listing* 3.1 resume o proceso de creación de *RETARDO* a partir de *FC_RESPUESTA* e *FC_CONTACTO*, procedentes ambas da *táboa de contactos*. Primeiro, calcúlase a diferencia en días entre a data de resposta e contacto para, posteriormente, converter dita variable a horas. Seguidamente repetimos o mesmo proceso para as horas, minutos e segundos. Deste xeito teremos as horas transcorridas entre o envío e a lectura. Finalmente, asignamos un valor enormemente elevado (da orde de $9 \cdot 10^9$) a aqueles *emails* que non se len.

```
1         CASE WHEN t.FC_RESPUESTA IS NOT NULL THEN
2         (
3         (CAST( ((t.FC_RESPUESTA-t.FC_CONTACTO) DAY(4)) AS DECIMAL(12,2) ))
              *24.00
4         +((CAST (EXTRACT(HOUR FROM t.FC_RESPUESTA) AS DECIMAL(12,2)))
5         -(CAST (EXTRACT(HOUR FROM t.FC_CONTACTO) AS DECIMAL(12,2))))
6         +((CAST (EXTRACT(MINUTE FROM t.FC_RESPUESTA) AS DECIMAL(12,2)))
7         -(CAST (EXTRACT(MINUTE FROM t.FC_CONTACTO) AS DECIMAL(12,2))))/60
8         +((CAST (EXTRACT(SECOND FROM t.FC_RESPUESTA) AS DECIMAL(12,2)))
9         -(CAST (EXTRACT(SECOND FROM t.FC_CONTACTO) AS DECIMAL(12,2))))/3600
10        )
11        ELSE 9999999999.00 END AS RETARDO
```

Listing 3.1: Consulta para obter a variable *RETARDO*.

3.3.2. Variables explicativas

Nesta sección presentaremos aquelas variables que imos considerar á hora de axustar o nosos modelos e, en caso de que exista, o seu proceso de construción.

A partir da variable *FC_CONTACTO* construiremos: **HORA_ENVIO**, **DIA_CONTACTO_SEMANA** e **MES_CONTACTO**. Para obtelas extraeremos a hora, o día da semana e o mes de *FC_CONTACTO*. A idea é incluílas nos nosos modelos e, así, escoller cal sería a hora, día e mes óptimos para o envío de *emails*. Estas variable son discretas, polo que poderíamos consideralas como variables de tipo factor ou de clasificación.

Seguidamente, vamos a construír variables que reflictan o comportamento histórico de cada cliente. Por exemplo, resultaría interesante coñecer que porcentaxe desas interaccións tivo lugar na hora de envío dos *emails*. Estas estatísticas aparecerán recollidas nas variables **PROP_LEIDOS_HORA** e **PROP_LEIDOS_CP_HORA** mediante unha metodoloxía similar á das proporcións anteriores. A única diferencia é que, neste caso, consideraremos aquelas interaccións que tivesen lugar na mesma hora que *HORA_ENVIO*. Dito doutro xeito, aquelas interaccións acontecidas nos 5 meses previos ao contacto considerado e onde a hora de *FC_RESPUESTA* coincida con *HORA_ENVIO*.

Seguidamente, a partir de *FC_RESPUESTA*, calculamos as variables **HORA_FAV_LEIDO** e **HORA_FAV_LEIDO_CP**. A idea sería extraer a hora de lectura a curto prazo e a longo prazo

máis repetidas do cliente nos 5 meses previos ao envío. Cabe esperar que, a priori, o usuario manterá uns certos hábitos que se replicarán para o instante de contacto. Esta información aparecerá recollida nas variables e, do mesmo xeito que para o resto de intervalos horarios, tomarán valores do 0 ao 23. A súa construción xa foi incorporada no proceso de *TERADATA*, polo que os seus pormenores non serán reflectidos nestas liñas. Non obstante, vamos destacar algúns aspectos relativos á mesma aos que cómpre prestar atención:

- Tecnicamente, o período onde se calculan os datos non é exactamente o que se define anteriormente. Senón que será aquel comprendido entre cinco meses antes e o día previo ao envío. O motivo de isto é evitar que estas variables podan dotar ao modelo de información explícita sobre a resposta.
- Adicionalmente, tomaremos en conta como interacción unicamente aquelas que tiveran lugar no período da observación. O motivo é idéntico ao ítem anterior.
- No caso de que algún cliente non interactuase con ningunha mensaxe ao longo deses 5 meses, o que é equivalente a que a hora favorita tome valor nulo, asignarémoslle o valor 25. Deste xeito, esta variable tamén achegaría información sobre a negativa do cliente a interactuar cos *emails*. Non obstante, isto último efectuarémolo unha vez teñamos calculadas todas as restantes métricas. O motivo é que haberá variables dependentes da mesma e, para efectuar correctamente os cálculos, o ideal sería que os datos nulos mantivesen ese estado.

A partir desta variable, xeraremos outras dúas que nos permitirán aproveitar e interpretar mellor o efecto desta derradeira variable:

- Por unha banda; definiremos dúas variables que identifiquen se a hora de envío coincide coa hora favorita do cliente e as denotaremos como **PERFECCION_LEIDO** e **PERFECCION_LEIDO_CP**, en función do tipo de resposta. Estas variables serían binarias, tomando valor "1" se ambas horas coinciden e "0" noutro caso.
- Tomaremos como posible variable a distancia circular que existe entre a hora de envío e a hora favorita de cada clase de resposta. Recollerémolas entón nas variables: **RETARDO_APROX_LEIDO** e **RETARDO_APROX_LEIDO_CP**. Dada H_e unha hora de envío e H_f a hora favorita para unha iteración dada, definimos dita distancia como:

$$d(H_e, H_f) = \begin{cases} H_f - H_e & \text{se } H_e \leq H_f, \\ 24 - H_e + H_f & \text{se } H_e > H_f. \end{cases} \tag{3.1}$$

  Deste xeito, de ter enviado un *email* nunha hora posterior á favorita, consideraremos que lerá a mensaxe ao día seguinte na súa hora favorita. Isto permitirá que o modelo penalice, sempre que sexa representativo, o feito de realizar un envío nunha hora distinta á favorita. Máis especificamente, o feito de envialo nun instante posterior a esta hora.

Dado que tamén nos interesará ter variables que permitan explicar o *RETARDO* de lectura que un cliente presentará, calculamos a variable **RETARDO_MEDIANO_HORA** como a mediana dos retardos relativos aos *emails* enviados na mesma hora do correo ao longo dos cinco meses previos ao envío.

A partir da *táboa de datos de acceso a banca electrónica e móbil* podemos obter novas variables. Por exemplo, a variable **TELEFONO1** recolle se un usuario da entidade interactuou, ou non, coa banca móbil no último mes. É de supoñer que, si ten acceso a este tipo de aplicación, existen bastantes posibilidades de que tamén poida acceder ao *mail* vía teléfono móbil. Isto permite acurtar os retardos e promover unha maior interacción.

Tamén a partir desta táboa podemos obter as variables **NUM_DIAS_ACCESO** e **PROP_DIAS_ACCESO**. A primeira delas mide o número de días que noso cliente accedeu a servizos de banca

electrónica e móbil na mesma hora de envío do *email* e, a segunda, mide a proporción deses accesos con respecto ao total do cliente nese mes. A idea sería que *NUM_DIAS_ACCESO* permita identificar a capacidade do cliente para acceder ao correo nesa hora, mentres que **PROP_DIAS_ACCESO** pode discernir entre se unha determinada hora é máis adecuada para o usuario ou non.

Para rematar consideraremos unha segmentación efectuada pola entidade mediante *Clustering*, **CLUSTER_BF**. Non afondaremos máis nestas categorías por unha cuestión de confidencialidade.

Ademais de todas as variables consideradas agora, teremos tamén en consideración algunhas das que estivemos tratando na Sección 3.1 xunto con outras que tamén están incluídas nas distintas táboas. Non obstante; ao longo da memoria centrámonos, principalmente, en aquelas que van resultar de maior utilidade para os modelos.

Missing values

Unha vez efectuados todos os nosos filtros e construídas as nosas variables, estudaremos cantos datos nulos presentan as columnas da nosa táboa.

Xa nos capítulos anteriores comentabamos o que sucedía cos valores nulos das seguintes variables:

- *CATEGORIA*: Eliminabamos todos aqueles *emails* cuxa categoría non estivese especificada. O motivo desta decisión era que non nos permitía coñecer cal sería a intencionalidade que presentaba dita comunicación. Polo tanto, non nos permitirían posteriormente efectuar unha clasificación.
- *CLUSTER_FB* e *CLUSTER_ID*: Recordemos que existía un certo número de clientes ao que non se lles tiña asignado un clúster específico. Polo tanto, creabamos unha nova clase para agrupar a todos estes datos.
- *IN_DIGITAL*: Omitíamos os valores nulos desta variable. Esta decisión foi tomada dado que apenas existían unhas centenas destes datos, que nun conxunto de case 32 millóns non resulta moi representativo, e porque, polo xeral, contiñan outras columnas da táboa de datos dos clientes con valores nulos.

Non obstante; tamén atopamos outras variables que conteñen datos faltantes, ou que están almacenados deste xeito pola propia construción das mesmas, que resultaría preciso estudar:

- *RESPUESTA_EMAIL*: Na Sección 3.1 diciamos que, se un *email* foi ignorado, dita columna toma valores nulos. Dado que, a partir de *RESPUESTA_EMAIL* construíamos as variables dicotómicas *LEIDO* e *LEIDO_CP*, ditos valores tomarían valor "0" nestas últimas.
- *FC_RESPUESTA*: Pola mesma razón que a anterior, esta variable conterá diversos datos nulos. Non obstante, dado que non intervén no axuste de ningunha variable explicativa ou resposta, non tomaremos medidas con respecto aos mesmos.
- *RETARDO*: Por construción, tomará valores nulos sempre que a mensaxe non fose lida ou *clickada*. Agora ben, xa comentabamos anteriormente que para eses casos asignabamos o valor 9999999999.00. Esto permitirá que, para a estatística de *RETARDO_MEDIANO*, o modelo poda diferenciar entre clientes que adoitan ler os seus *emails* ou non.
- *PROP_LEIDOS* e *PROP_LEIDOS_CP*: Estas columnas poden presentar valores nulos sempre que un cliente non teña recibido ningún *email* nos 5 meses previos á observación. Polo que deberíamos asignarlles valor 0 a todos eles.
- *HORA_FAV_LEIDO* e *HORA_FAV_LEIDO_CP*: Se un cliente non presentase ningunha lectura (a curto ou calquera prazo) nos 5 meses previos ao envío, estas dúas columnas presentarán nulos. Para estes casos, asignarémoslle o valor 25. Deste xeito o modelo poderá distinguir se un cliente leva 5 meses ou máis inactivo.

- *RETARDO_APROX_LEIDO* e *RETARDO_APROX_LEIDO*: Tomarán, ao igual que as variables anteriores, valores nulos sempre que non exista *HORA_FAV_LEIDO* ou *HORA_FAV_LEIDO_CP*, respectivamente. Nesta casuística, optaremos por asignarlle valor 25 a estes datos. Isto débese a que é maior ao máximo do valor que poderían tomar e servirá, tamén, para distinguir cando un cliente non adoita interactuar coas comunicacións.

Deseguido, efectuaremos unha análise preliminar do efecto destas variables explicativas sobre a resposta.

3.3.3. Análise preliminar das variables explicativas

Esta sección permanece censurada por motivos de confidencialidade. Nela estudabamos o efecto que tiñan as variables da sección anterior sobre as distintas respostas.

Capítulo 4

Modelos

O obxectivo será crear modelos capaces de predicir se un usuario vai, ou non, interactuar cun *email* dado e, baseándonos en dita predición, obter a hora que maximice a probabilidade de éxito en función de cada cliente.

Lembremos que buscabamos axustar modelos con variables resposta *LEIDO* e *LEIDO_CP*. Polo tanto, estamos ante un caso de variable obxectivo dicotómica, podendo tomar dous posibles valores ("1" e "0") cunha probabilidade que buscamos estimar. Dado que ambas as dúas variables son coñecidas, botaremos man de modelos de predición supervisados. En particular, os seguintes: *Random Forest*, *LightGBM*, *CatBoost* e *XGBoost*. A elección destes modelos baséase en dúas premisas: por unha banda, carecen de hipóteses básicas sobre os datos da mostra e, por outra, presentan un alto poder de predición. Ademais, os modelos de tipo *gradient boosting* resultan moi axeitados para manexar grandes cantidades de datos. Non obstante, temos unha importante limitación técnica. A base de datos coa que contamos está formada por ata 32 millóns de rexistros de *emails*. Esta cantidade de información resulta totalmente inabarcable para efectuar o axuste de modelos. Ademais, tal e como comentabamos no Capítulo 3, a pauta de envíos da entidade produce un nesgo neles. Por todo iso, efectuaremos unha metodoloxía de ***subsampling*** sobre a nosa mostra orixinal de xeito que conte, aproximadamente, con 3 millóns de observacións. Para facelo, dividiremos a mostra en dúas partes: unha que agrupe os rexistros das horas máis representativas da mostra (de 9:00 a 22:00) e outra que agrupe as horas restantes. Seguidamente, efectuamos a mostraxe en tres fases:

1. Calculamos a proporción da mostra que representan, por un lado, as observacións recollidas entre as 9:00 e las 22:00 e, por outro, aquelas que teñan lugar nas restantes horas. No caso das horas máis representativas, estas abranguen un 97.16 % da mostra orixinal e, o restante 2.84 %, serían aquelas horas menos representativas. A idea é que, na mostra final de 3 millóns de datos, estas proporcións se manteñan.

2. Para as horas con maior representatividade, impoñemos que todas conten co mesmo número de instancias. Deste xeito, asegurámonos de que se minimice o nesgo producido polo patrón actual de envíos da entidade. Seguidamente, levamos a cabo unha mostraxe estratificada polos campos: *CATEGORIA*, *MES_CONTACTO*, *RESPUESTA_EMAIL* e *HORA_RESPUESTA*. No caso daquelas menos representativas, unicamente efectuamos unha mostraxe estratificada polas variables: *CATEGORIA*, *MES_CONTACTO*, *RESPUESTA_EMAIL*, *HORA_ENVIO* e *HORA_RESPUESTA*.

3. Finalmente, agrupamos todos os datos na mostra final.

Deste xeito, logramos corrixir boa parte do nesgo existente sobre as horas de envío que viña producido pola pauta de envíos actual da entidade. Ademais, a estratificación mantén as proporcións de

certas características clave como son: *CATEGORIA*, *MES_CONTACTO*, *RESPUESTA_EMAIL* e *HORA_RESPUESTA*.

4.1. Partición dos datos

Unha vez efectuada esta remostraxe, é necesario dividir a nova mostra en tres segmentos distintos:

- **Adestramento:** Este conxunto recollerá aqueles datos que empregaremos para axustar os modelos e abarcará o 60 % da mostra orixinal.
- **Validación:** Os datos deste segmento servirán para seleccionar os hiperparámetros óptimos e validar o modelo. Axustaremos modelos para as distintas combinacións de hiperparámetros e seleccionaremos o mellor deles. Estará formado polo 20 % dos datos da mostra orixinal.
- **Test:** Este grupo servirá para comprobar como de ben se axusta o modelo aos datos facilitados pola empresa e será o 20 % restante da mostra.

Estas particións obtéñense efectuando unha mostraxe de xeito que ningunha mostra comparta rexistros dun mesmo cliente. É dicir, se a mostra test contén as comunicacións dun determinado usuario, as restantes dúas mostras non conterán ningunha do mesmo. O motivo de realizar este proceso é evitar o efecto de dependencia.

4.2. Selección de variables explicativas

As variables consideradas para os modelos serán aquelas tratadas ao longo do Capítulo 3. Para a súa selección non foi posible efectuar un procedemento selector de regresores estándar. Isto débese á enorme cantidade de datos coa que contamos e a limitación computacional da que dispón a entidade. Non obstante, propúxose un proceso similar en base a diferentes criterios:

1. Axustamos algúns *Random Forest* para cada resposta considerada e estudamos que variables presentan unha maior importancia sobre o conxunto.
2. Calculamos as correlacións entre cada unha das variables numéricas e impoñemos que exista unha correlación máxima de 0.7. O obxectivo sería evitar dependencias.
3. Finalmente, axustamos modelos considerando as variables resultantes dos procesos anteriores e procedemos coa selección de hiperparámetros.

Este procedemento permite reducir o número de variables a axustar e, consecuentemente, o tempo computacional.

Adicionalmente, fixéronse unha serie consideracións técnicas sobre as variables. Inicialmente, tivemos en conta todos os regresores que tiñamos dispoñibles e que foron discutidos ao longo do Capítulo 3. Non obstante, a medida que transcorría o traballo decatámonos do seguinte:

- Se ben as variables *PROP_LEIDOS_CP* e *PROP_LEIDOS* resultan moi útiles á hora de clasificar se o *email* vai ser (ou non) lido, presentan un importante hándicap. Xeralmente, os modelos que toman en consideración esta variable tenden a outorgarlle moito peso. Isto provoca que, o efecto doutras variables sobre a resposta, quede totalmente oculto. Isto dificulta enormemente a identificación da hora que aumente a probabilidade de lectura. Polo tanto decidiuse excluír estas para evitar o enmascaramento de regresores relacionados ca hora de envío, malia que isto reduza lixeiramente a precisión global do modelo (**AUC**).

- Inicialmente, consideramos introducir as variables *DIA_CONTACTO_SEMANA* e *MES_CONTACTO*. Non obstante, existía un gran desequilibrio entre o número de *emails* enviados por día e mes; o que podería introducir un importante nesgo sobre o modelo. Ademais, preséntanse dificultades de imputación no proceso de posta en produción. Polo que, finalmente, non as incluímos no axuste dos nosos modelos.

- As variables *HORA_FAV_LEIDO* e *HORA_FAV_LEIDO_CP* presentan un efecto similar a *PROP_LEIDOS_CP* e *PROP_LEIDOS* pero, adicionalmente, tenden a empeorar o rendemento dos modelos con respecto a outros. Ademais, ao tratarse de variables compostas por 25 categorías, o custo computacional é extremadamente elevado. En consecuencia, para tratar de capturar o efecto desta hora favorita sen perder precisión, consideraremos no seu lugar as variables: *PERFECCION_LEIDO*, *PERFECCION_LEIDO_CP*, *RETARDO_APROX_LEIDO* e *RETARDO_APROX_LEIDO_CP*.

- Para introducir a hora de envío como variable explicativa dos modelos, consideramos tres formatos diferentes:

  - Variable categórica (*HORA_ENVIO*).
  - Variable circular a tramos con descomposición seno-coseno, tal e como se recolle en Rueda, Fernández, Barragán, Mardia e Peddada (2016).
  - Variable de tipo continuo (*HORA_CIRC_C*).

  Finalmente, escollemos a derradeira das opcións por unha cuestión técnica. O primeiro dos enfoques conta cunha clara desvantaxe que é o custo computacional e, tras efectuar diversos axustes, os resultados que achegaban non diferían significativamente dos que ofrecía a terceira opción. A segunda, tampouco amosaba diferencias con ela. Isto pode ser debido á ausencia de densidade de datos da nosa mostra entre as 23:00 e as 8:00 e que os modelos onde se utilizan *Random Forest* como base, requiriría dunha densidade de datos suficiente para efectuar particións (Hastie e col., 2001; James e col., 2013). Polo tanto, a principal vantaxe de considerar unha variable circular queda anulada polo feito de utilizar este tipo de modelos.

4.3. Selección de hiperparámetros

A selección efectuouse considerando e axustando distintos valores para os hiperparámetros de cada un dos modelos e implementando metodoloxía de **optimización bayesiana** aplicada a metodoloxías de *Machine Learning* (Shahriari, Swersky, Wang, Adams & de Freitas, 2016; Snoek, Larochelle & Adams, 2012). Deseguido explicaremos en que consiste, en particular, a optimización bayesiana e, posteriormente, como poderíamos explicala sobre o noso problema.

A optimización bayesiana busca alcanzar o óptimo dunha determinada función obxectivo que resulta descoñecida ou, na práctica, moi difícil de calcular do xeito máis eficiente posible (Shahriari e col., 2016). Supoñamos un espazo de busca $\mathcal{X}$, entón buscamos o mínimo dunha función $f(x)$, $\forall x \in \mathcal{X}$. É dicir, un x^* tal que:

$$x^* = \arg\min_{x \in \mathcal{X}} \{f(x)\}. \tag{4.1}$$

Dada a descoñecida natureza de $f(x)$, buscaremos construír un modelo que a aproxime dita función e, en base ao mesmo, escoller aqueles x que acheguen unha mellora esperada máis elevada (Shahriari e col., 2016).

Entón, consideremos unha familia de modelos parametrizados a partir dun vector $\mathbf{w}$ descoñecido e $\mathcal{D}$ o conxunto de datos cos que contamos para o seu axuste (Shahriari e col., 2016). Polo tanto, dende

o punto de vista da estatística bayesiana (Chou, 1977), podemos definir a probabilidade a posteriori de $\mathbf{w}$ condicionado a $\mathcal{D}$ como:

$$p\left(\mathbf{w}|\mathcal{D}\right) = \frac{p\left(\mathcal{D}|\mathbf{w}\right)p(\mathbf{w})}{p(\mathcal{D})}.$$

Esta probabilidade a posteriori recolle as nosas crenzas actualizadas do valor de $\mathbf{w}$ en base a un conxunto de datos observados $\mathcal{D}$. No numerador recóllese un modelo de verosimilitude, $p\left(\mathcal{D}|\mathbf{w}\right)$, e no denominador a verosimilitude marxinal $p(\mathcal{D})$. Esta derradeira densidade é, na práctica, inviable computacionalmente (Shahriari e col., 2016), porén, non depende de $\mathbf{w}$ e, polo tanto, sería unha constante normalizadora. É dicir, $p\left(\mathbf{w}|\mathcal{D}\right) \propto p\left(\mathcal{D}|\mathbf{w}\right)p(\mathbf{w})$. Esta distribución a posteriori permite cuantificar a incertidume.

Agora, centrarémonos no enfoque non paramétrico, posto que resulta máis axeitado para a nosa casuística. Sexa $\mathbf{X}$ unha matriz de deseño con n filas $\mathbf{x_i}$ e $\mathbf{y}$ o vector de saídas relativo a dita matriz. Notemos que estas saídas veñen suxeitas a unha certa variabilidade descoñecida. Consecuentemente non ten que verificarse, necesariamente, a seguinte condición: $y_i = f(\mathbf{x}_i)\ \forall i$. Ademais, asumimos que a varianza observacional, σ^2, é fixa e consideremos que $\mathbf{V}_0$ é a matriz de covarianzas a priori dos coeficientes ou parámetros da regresión, $\mathbf{w}$. Entón, suporemos que $p(\mathbf{w}|\mathbf{V}_0) = \mathcal{N}\left(0, \mathbf{V}_0\right)$. Isto permite integrar estes parámetros de xeito que (Shahriari e col., 2016):

$$p(\mathbf{y}|\mathbf{X}, \sigma^2) = \int p\left(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2\right) p(\mathbf{w}|0, \mathbf{V}_0) d\mathbf{w} = \mathcal{N}\left(\mathbf{y}|0, \mathbf{X}\mathbf{V}_0\mathbf{X}^\top + \sigma^2\mathbf{I}\right).$$

Se reescribimos, agora, a ecuación anterior substituíndo a matriz $\mathbf{X}$ por unha matriz de deseño no espazo das características ou *features*, $\Phi\left(\mathbf{X}\right) = \mathbf{\Phi}$, temos que (Shahriari e col., 2016):

$$p(\mathbf{y}|\mathbf{X}, \sigma^2) = \mathcal{N}\left(\mathbf{y}|0, \mathbf{\Phi}\mathbf{V}_0\mathbf{\Phi}^\top + \sigma^2\mathbf{I}\right).$$

Neste contexto, a matriz $\mathbf{\Phi}\mathbf{V}_0\mathbf{\Phi}^\top$ é simétrica e semidefinida positiva. Polo que poderíamos construír unha función tipo núcleo tal que:

$$\mathbf{K}_{i,j} = \mathbf{\Phi}(\mathbf{x}_i)\mathbf{V}_0\mathbf{\Phi}(\mathbf{x}_j)^\top.$$

Este proceso permite, adicionalmente, efectuar predicións sobre as saídas, y_*, relativas a puntos de entrada potenciais descoñecidos, x_* do xeito seguinte (Shahriari e col., 2016):

$$p\left(y_*|\mathbf{X}, \mathbf{X}_*, y, \sigma^2\right) = \frac{p\left(y_*, y|\mathbf{X}, \mathbf{X}_*, \sigma^2\right)}{p\left(y|\mathbf{X}, \sigma^2\right)}.$$

Este tipo construción é o que se coñece como **proceso gaussiano**, os cales quedan definidos na súa totalidade mediante a súa función de medias a priori $\mu_0 : \mathcal{X} \to \mathbb{R}$ e unha función *kernel* semidefinida positiva $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Este tipo de procesos asume, adicionalmente, os seguintes dous puntos (os cales poden seguirse dos razoamentos previos):

- $\mathbf{f}|\mathbf{X} \sim \mathcal{N}\left(\left(\mu_0(\mathbf{x}_1), \cdots, \mu_0(\mathbf{x}_n)\right), \mathbf{K}\right)$,
- e $\mathbf{y}|\mathbf{f}, \sigma^2 \sim \mathcal{N}\left(\mathbf{f}, \sigma^2\mathbf{I}\right)$.

Sendo $x_1, \cdots, x_n$ os n puntos de observación (formados por unha ou máis variables), $\mathbf{y}$ o seu vector de saídas, $\mathbf{f} = (f_1, \ldots, f_n)$ con $f_i = f(\mathbf{x}_i)\ \forall i \in \{1, \ldots, n\}$ e $\mathbf{K}$ unha matriz con compoñentes $k_{i,j}$ dados por unha función tipo *kernel* da forma: $k(\mathbf{x}_i, \mathbf{x}_j)$ (Shahriari e col., 2016).

Entón, podemos efectuar predicións sobre como se comportaría a función f para valores descoñecidos de $\mathbf{x}$. Pois, baixo os supostos anteriores e dados os pares de observacións $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ considerados previamente, podemos afirmar que a variable aleatoria $f(\mathbf{x})$ segue unha normal de media $\mu_n(\mathbf{x})$ e varianza $\sigma_n^2(\mathbf{x})$ tales que (Shahriari e col., 2016):

- $\mu_n(\mathbf{x}) = \mu_0(\mathbf{x}) + \mathbf{k}(\mathbf{x})^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(y - (\mu_0(x_1), \ldots, \mu_0(x_n)))$.
- $\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x})^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x})$.

Sendo $\mathbf{k}(\mathbf{x})$ un vector de covarianzas entre $\mathbf{x}$ e $\mathbf{x}_1 \ldots, \mathbf{x}_n$. Os *kernel* máis utilizados aparecen recollidos en Shahriari e col. (2016).

Este enfoque podería ser utilizado para o axuste de modelos con diversos hiperparámetros dispoñibles en base a algún tipo de métrica relacionada co modelo, tal e como se recolle en Snoek e col. (2012). Neste caso, a función a optimizar sería algunha métrica de rendemento do modelo (no noso caso, o **AUC**) e a matriz de deseño estaría conformada polos posibles valores dos hiperparámetros de cada un dos modelos que busquemos axustar. Para facer isto, debemos asignar uns límites para cada hiperparámetro e un número máximo de iteracións do proceso. Utilizaremos a función `bayesOpt` do paquete `ParBayesianOptimization` (Wilson, 2020).

Deseguido móstranse os dominios, relativos aos posibles valores dos hiperparámetros, considerados para efectuar o proceso de optimización en función do tipo de modelo («CatBoost Documentation», 2026; Fernández-Casal e col., 2024; Shi e col., 2025):

- *Random Forest:* `randomForest()`
  - `ntrees`: Tomará o valor 500.
  - `mtry`: Seus posibles valores estarán comprendidos entre $[1, 5]$.
  - `maxnodes`: Asignarémoslle valores $[1, 10]$.
  - `nodesize`: Sexa n a cantidade de observacións da nosa mostra e n_e o número de éxitos, consideraremos o intervalo dado por:
$$\left[\text{mín}\left\{100, \frac{n_e}{20}\right\}, \text{mín}\left\{\frac{n}{50}, 2n_e\right\}\right].$$
  - `sampsize`: Os posibles valores estarán contidos no intervalo: $[0.3, 1]$.
- *XGBoost:* `xgb.train()`
  - `nrounds`: Seu valor será 1000.
  - `eta`: Tomará os valores inseridos no intervalo: $[0, 0.3]$.
  - `gamma`: Probaranse os valores contidos en: $[0, 10]$.
  - `min_child_weight`: Consideraremos o intervalo:
$$\left[0.6\,\text{mín}\left\{100, \frac{n_e}{20}\right\}, 0.6\,\text{mín}\left\{50000, \frac{n}{50}, 2n_e\right\}\right].$$
  - `max_delta_step`: Outorgarémoslle os valores contidos en $[0, 9]$.
  - `max_depth`: Asignarémoslle os valores inseridos no intervalo $[1, 10]$.
  - `subsample`: Consideramos os límites dados polo intervalo $[0.5, 0.8]$.
  - `colsample_bytree`: Consideraremos os valores contidos en $[0.5, 0.8]$.
  - `colsample_bylevel`: Tomamos valores dentro de $[0.5, 0.8]$.
  - `scale_pos_weight`: Tomaremos un valor contido no intervalo:
$$[1, w], \tag{4.2}$$
sendo $w = \frac{n - n_e}{n}$. É dicir, o número de fracasos da mostra de adestramento dividido entre o de éxitos.

- `early.stop.round`: Por defecto estará fixo en 50.

- *LightGBM:* `lgb.train()`
  - `nrounds`: Partiremos dun valor de 1000.
  - `learning_rate`: Outorgarémoslle os valores dentro de $[0.005, 0.3]$.
  - `min_gain_to_split`: Os valores propostos son os inseridos no intervalo $[0, 10]$.
  - `num_leaves`: Tomaremos os mesmos valores ca no parámetro `min_child_weight` do modelo *XGBoost*.
  - `min_data_in_leaf`: Consideraremos os mesmos valores que `num_leaves`.
  - `lambda_l1`: Imputaremos os valores contidos en $[0, 10]$.
  - `lambda_l2`: Asignarémoslle os mesmos valores que ao $L1$.
  - `max_depth`: Os seus valores estarán contidos en $[1, 10]$.
  - `bagging_freq`: Tomará valor 1 para activarse.
  - `feature_fraction`: Consideraremos modelos para valores contidos en $[0.5, 0.8]$.
  - `scale_pos_weight`: Tomará os valores dados por (4.2).
  - `early.stop.round`: Por ser análogo ao caso *XGBoost*, tomará o mesmo valor.

- *CatBoost:* `catboost.train()`
  - `iterations`: Tomará valor 1000.
  - `learning_rate`: Consideraremos os mesmos valores que *LightGBM*.
  - `depth`: Asignarémoslle os mesmos valores ca o modelo anterior.
  - `l2_leaf_reg`: Os valores cos que probaremos serán os contidos no intervalo: $[0, 10]$.
  - `rsm`: Probaremos cos parámetros descritos no modelo anterior para `feature_fraction`.
  - `class_weights`: Tomaremos os mesmos valores ca nas casuísticas anteriores..
  - `early_stopping_rounds`: Ten a mesma atribución que `early.stopping.rounds`, consecuentemente tomaremos o mesmo valor.

A idea consistirá en efectuar 20 iteracións do proceso de optimización bayesiana buscando maximizar o *AUC* de validación e mantendo un número fixo elevado de árbores e iteracións. Para finalizar e evitar o sobreaxuste, efectuaremos 5-*folds cross validation* para escoller o mellor deles para cada modelo. Notemos que todo este proceso debe ser personalizado, en particular, os *folds* deben verificar que non conteñen datos comúns de ningún cliente. É dicir, se un usuario presenta unha interacción nunha das particións, este non pode presentar rexistros noutro. Isto efectuámolo a fin de evitar problemas de dependencia. Utilizaremos tamén técnicas de parada temperá, detendo este proceso de validación cruzada unha vez non se produzan melloras representativas durante 20 iteracións.

4.4. Modelo de lecturas a curto prazo

Tal e como comentabamos anteriormente, este modelo resulta especialmente interesante. Isto débese a que unicamente considerará como éxitos aqueles *emails* que fosen lidos, como moi tarde, unha hora e media despois de ser enviados. A idea sería utilizar este modelo para predicir unha hora de envío en comunicacións urxentes ou nas que se pretenda conseguir un impacto inmediato sobre o cliente. Dado que buscamos predicir a hora con maior probabilidade de lectura a curto prazo, será necesario incluír unha variable relativa á hora de envío como explicativa.

4.4.1. *Random Forest*

Primeiramente efectuamos o noso proceso de selección de variables, descrito na Sección 4.2, e eliminamos aquelas que comentabamos anteriormente. Seguidamente, efectuamos un proceso de selección de hiperparámetros mediante optimización bayesiana e o 5-*folds cross validation* para seleccionar lo valor de `ntrees`. Obtendo os hiperparámetros que se recollen no Cadro 4.1.

maxnodes	nodesize	mtry	ntrees	sampsize
6	100	5	500	0.8000

Cadro 4.1: Hiperparámetros do modelo *Random Forest* óptimo para lecturas a curto prazo.

A curva *ROC* relativa ao modelo está recollida na Figura 4.1. Como vemos, a curva mostra un achatamento no centro. Isto é síntoma de que o modelo resultante presentará un desequilibrio entre sensibilidade e especificidade. Observemos que, no referente a **AUC**, se trata dun modelo bastante consistente sobre todas as mostras, o que é indicativo de que o modelo non está sobreaxustado.

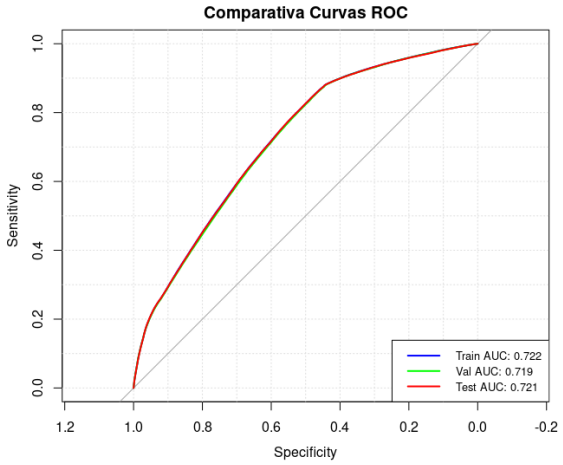


Figura 4.1: Curva *ROC* relativa ao modelo de lecturas a curto prazo *Random Forest*.

Seguidamente estudaremos a matriz de confusión do modelo para o seu limiar óptimo (*Listing* 4.1). Dita matriz permanecerá censurada por motivos de confidencialidade.

```
1 CONFIDENCIAL
```

Listing 4.1: Matriz de Confusión do modelo de lecturas a curto prazo *Random Forest*

Na Figura 4.2 represéntase a importancia de cada unha das 15 variables máis influentes do modelo [1].

Na Figura 4.3 mostramos o efecto das 5 variables explicativas máis importantes sobre a resposta. Por unha banda mostramos as frecuencias relativas (en azul) e, por outra, o *ratio* de aperturas a curto prazo relativas a cada categoría ou intervalo numérico. A liña vermella horizontal recolle a proporción de lecturas a curto prazo da mostra.

[1] A importancia representada sería a reescalada: $x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$. De aquí en diante, sempre nos referiramos á

Figura 4.2: Importancia das variables explicativas no modelo de lecturas a curto prazo *Random Forest*.

Figura 4.3: Variación da proporción de lecturas a curto prazo en función das cinco variables máis importantes do modelo *Random Forest*.

4.4.2. *XGBoost*

Para este modelo, consideráronse as mesmas variables que no caso anterior e, tras un proceso de selección de hiperparámetros e a comparación de modelos mediante o proceso descrito na Sección 4.3, concluímos que o modelo ideal sería aquel que presenta os parámetros recollidos no Cadro 4.2. Ademais seu *AUC* de test será de 0.741, que é lixeiramente mellor ao do *Random Forest*.

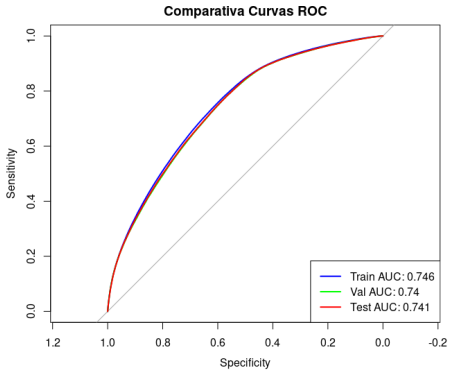
`maxnodes`	`eta`	`gamma`	`min_child_weigth`	`max_delta_step`
1000	0.07946	8.1386	15375	1.5175
`max_depth`	`subsample`	`colsample_bytree`	`colsample_bylevel`	`scale_pos_weight`
5	0.5859	0.5490	0.5703	6.191573

Cadro 4.2: Hiperparámetros do modelo *XGBoost* óptimo para lecturas a curto prazo.

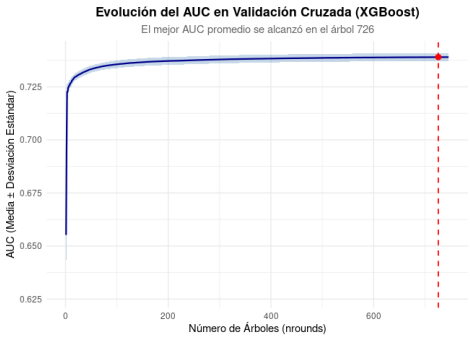
Na Figura 4.4b representamos a curva de variación do **AUC** medio durante o proceso de validación cruzada en función do número de árbores considerado. Sombreado recóllese a quasi-desviación típica dos mesmos. En vermello aparece indicado o **AUC** medio do óptimo seleccionado. Notemos que esta representación non se presenta para a casuística do *Random Forest*. Isto débese ao elevado custo computacional de axustala para unha árbore de tipo `regRF` (Malley e col., 2012). Por outra banda, as curvas *ROC* do modelo para as distintas mostras aparecen recollidas na Figura 4.4a. Tal e como vemos, este modelo presenta unhas curvas similar ao anterior, polo que tamén presentará un lixeiro desequilibrio entre sensibilidade e especificidade. Notemos que, tal e como sucedía co anterior, as tres mostras presentan valores de **AUC** similares. Isto é síntoma de que non se trata dun modelo sobre-

importancia será neste sentido

axustado ao adestramento.



(a) Curva *ROC* do modelo *XGBoost*.



(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *XGBoost*.

Figura 4.4: Información relativa á métrica de validación do modelo de lecturas a curto prazo *XGBoost*.

Obtemos, deseguido, o limiar óptimo e calculamos a matriz de confusión (*Listing* 4.2). A cal permanecerá censurada por motivos de confidencialidade.

Deseguido, estudaremos a importancia que presentan as distintas variables sobre o modelo. Notemos que, no caso do *XGBoost*, consideramos as variables categóricas como *dummy*. Polo tanto, cada categoría da variable actúa como un regresor propio. Isto tradúcese nun maior número variables explicativas, fronte ao modelo *Random Forest*. É por iso que, se analizamos a importancia das variables de tipo categórico, aparecerán as *dummy* relativas a cada clase. Na Figura 4.5 móstranse as 15 variables máis importantes xunto coa súa importancia.

```
1 CONFIDENCIAL
```

Listing 4.2: Matriz de Confusión do modelo de lecturas a curto prazo *XGBoost*.

Figura 4.5: Importancia das variables explicativas no modelo de lecturas a curto prazo *XGBoost*.

Como vemos, o *XGBoost* presenta unha maior capacidade para "distinguir" en función da hora. Isto resulta especialmente interesante no paradigma da presente memoria.

4.4.3. *LightGBM*

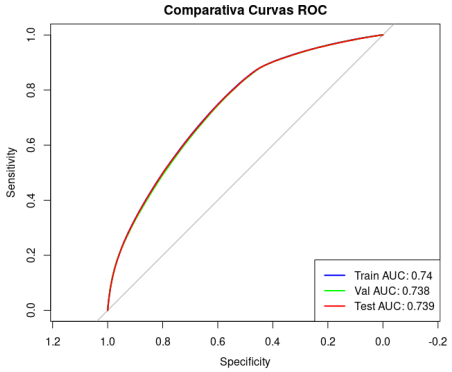
Para este modelo, empregaremos as mesmas variables explicativas ca nos anteriores. Tras efectuar o proceso de selección de hiperparámetros descrito anteriormente, obtivemos un modelo óptimo cuxos parámetros aparecen recollidos no Cadro 4.3.

Novamente, mostramos a curva de evolución do **AUC** medio de validación cruzada (Figura 4.6b). Nela observamos un aumento súbito do valor do **AUC** para os valores de iteracións máis baixos, seguido

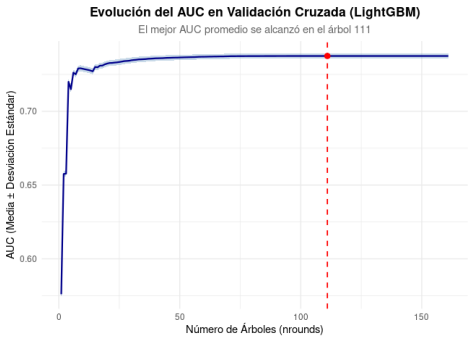
nrounds	learning_rate	min_gain_to_split	num_leaves	min_data_in_leaf	lambda_l1
111	0.0.14703	4.2421	3717	19968	4.8129
lambda_l2	max_depth	bagging_freq	feature_fraction	scale_pos_weight	early.stop.round
0.9993	9	1	0.6029	1.1534	50

Cadro 4.3: Hiperparámetros do modelo *LightGBM* óptimo para lecturas a curto prazo.

dun estancamento que finaliza provocado polo `early.stop.round`. A continuación, representamos na Figura 4.6a as curvas *ROC* do modelo para cada unha das segmentacións. A cal mostra un **AUC** de 0.739 para o test. Tal e como observamos, predí mellor que un clasificador aleatorio. A súa matriz de confusión será omitida por motivos de confidencialidade 4.3.



(a) Curva *ROC* do modelo *LightGBM*.



(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *LightGBM*.

Figura 4.6: Información relativa á métrica de validación do modelo de lecturas a curto prazo *LightGBM*.

```
1 CONFIDENCIAL
```

Listing 4.3: Matriz de Confusión do modelo de lecturas a curto prazo *LightGBM*.

Para as variables categóricas do *LightGBM* sucede o mesmo que para o modelo anterior, polo que veremos as importancias das 15 variables máis importantes incluíndo as *dummy*. Na Figura 4.7 represéntanse ditas importancias reescaladas.

Figura 4.7: Importancia das variables explicativas no modelo de lecturas a curto prazo *LightGBM*.

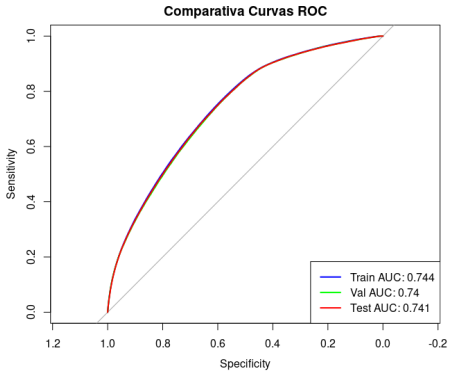
4.4.4. *CatBoost*

Finalmente, vexamos o que sucede co modelo *CatBoost*. Axustando as mesmas variables ca nos casos anteriores, obtemos un modelo óptimo cuxos parámetros aparecen recollidos no Cadro 4.4. Este presenta un *AUC* de test de 0.741, o máis alto (xunto co *XGBoost*) de entre todos os modelos.

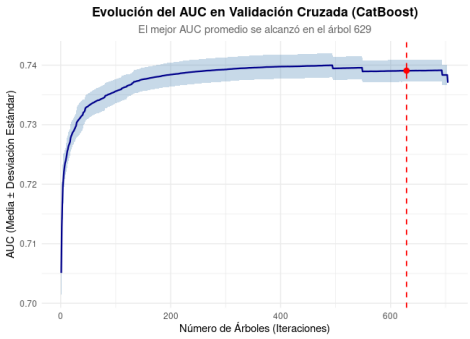
iterations	learning_rate	depth	l2_leaf_reg	rsm	class_weights
629	0.1442	4	6.4351	0.6462	6.191573

Cadro 4.4: Hiperparámetros do modelo *CatBoost* óptimo para lecturas a curto prazo.

Observamos, agora, a variación do **AUC** de validación durante o proceso de 5-*folds cross validation* (Figura 4.8b). O primeiro que chama a atención é como o punto de parada óptimo do proceso de validación cruzada sitúase nunha iteración onde a curva do **AUC** promedio comeza a descender. Isto débese a que escolleuse como punto óptimo o promedio dos relativos a cada un dos 5 *folds*. Isto, combinado ca maior variabilidade que presentou a media dos **AUC**, provoca esta aparente contradición. Non obstante, se atendemos á curva *ROC* deste modelo para as distintas mostras, (Figura 4.8a) observamos como o **AUC** de todas as mostras é practicamente idéntico (próximo a 0.74). Polo que consideraremos este valor óptimo de iteracións.



(a) Curva *ROC* do modelo *CatBoost*.



(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *CatBoost*.

Figura 4.8: Información relativa á métrica de validación do modelo de lecturas a curto prazo *CatBoost*.

Se aplicamos agora o seu limiar óptimo como regra de decisión, obtemos a matriz de confusión dada polo *Listing* 4.4. Na que non afondaremos por motivos de confidencialidade.

```
1 CONFIDENCIAL
```

Listing 4.4: Matriz de Confusión do modelo de lecturas a curto prazo *CatBoost*.

Na Figura 4.9 aparecen recollidas as importancias das 15 variables máis relevantes.

4.5. Modelos de lecturas

Outro enfoque consistiría en tratar de predicir se o cliente vai ler ou non un determinado *email*, independentemente de se o fai a curto ou longo prazo. É dicir, tratar de explicar a variable *LEIDO*.

Figura 4.9: Importancia das variables explicativas no modelo de lecturas a curto prazo *CatBoost*.

Para esta casuística incluiremos como variables explicativas as mesmas que no caso das lecturas a curto prazo, pero adaptadas á resposta. Tamén realizaremos unha análise similar á efectuada na Sección 4.4 para cada tipo de modelo.

4.5.1. *Random Forest*

Seguidamente procedemos coa selección de hiperparámetros para o modelo *Random Forest*, chegando a que o modelo óptimo en base ao exposto na Sección 4.3 sería o que presenta os hiperparámetros recollidos no Cadro 4.5. Ademais, consta dun **AUC** sobre a mostra de test de 0.7112.

maxnodes	nodesize	mtry	ntrees	sampsize
6	100	5	500	0.5

Cadro 4.5: Hiperparámetros do modelo *Random Forest* óptimo para lecturas.

Unha vez calculado o limiar óptimo da curva *ROC* asociada á mostra test, calculamos a matriz de confusión relativa ao mesmo (*Listing* 4.5). Non afondaremos nela por motivos de confidencialidade.

```
1 CONFIDENCIAL
```

Listing 4.5: Matriz de Confusión do Modelo *Random Forest*.

Na Figura 4.10 móstranse as curvas *ROC* do modelo para as diferentes segmentacións dos datos. Como vemos, o **AUC** é bastante equilibrado nas 3 mostras. Podendo así concluír que o modelo non presenta sobreaxuste.

Seguidamente, estudaremos a importancia das variables neste modelo. Na Figura 4.11 móstranse os 15 regresores máis influentes xunto coas súas importancias reescaladas.

Deseguido analizaremos o efecto que presentan estas variables sobre a resposta, *LEIDO*. Na Figura 4.12 representamos a variación da proporción da variable resposta en base ás cinco variables máis importantes do modelo.

4.5.2. *XGBoost*

Unha vez consideradas as variables anteriores efectuamos a selección de hiperparámetros onde, en base aos criterios recollidos na Sección 4.3, seleccionamos un modelo óptimo. O cal, presenta un **AUC** de test de 0.7403 e cuxos hiperparámetros aparecen recollidos no Cadro 4.6.

Na Figura 4.13b representamos a curva de do **AUC** relativo ao proceso de 5-*folds cv*. Nela vemos como, a medida que aumentan as interaccións, os valores tenden a estabilizarse. Acadando un máximo na iteración número 1000. As curvas *ROC* relativas ao modelo para cada un dos segmentos que dividen a nosa mostra móstranse na Figura 4.13a. Observamos que o comportamento do modelo é mellor que

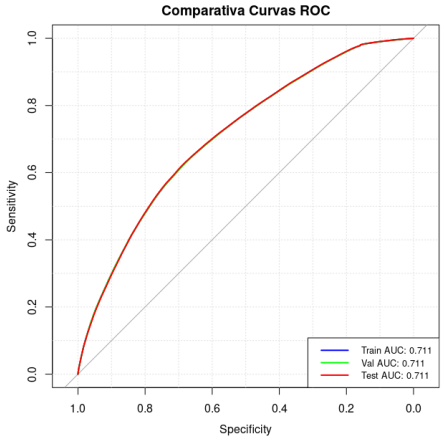


Figura 4.10: Curva *ROC* do modelo de lecturas *Random Forest*.

Figura 4.11: Importancia das variables explicativas no modelo de lecturas *Random Forest*.

un clasificador aleatorio, emporiso, as diferenzas entre os **AUC** das mostras de test e validación son lixeiramente menores aos de adestramento. Ademais, vemos como a curva non está tan achatada, o que é indicativo dun maior equilibrio entre especificidade e sensibilidade.

Seguidamente, tras fixar un limiar óptimo con valor, analizaremos as métricas da mostra test do modelo de clasificación mediante a súa matriz de confusión 4.6.

```
1 CONFIDENCIAL
```

Listing 4.6: Matriz de Confusión do Modelo *XGBoost*.

Agora, estudaremos a importancia das variables sobre o modelo. Na Figura 4.14 represéntanse as importancias das 15 con maior influencia sobre el.

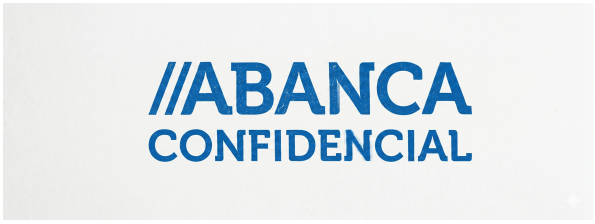
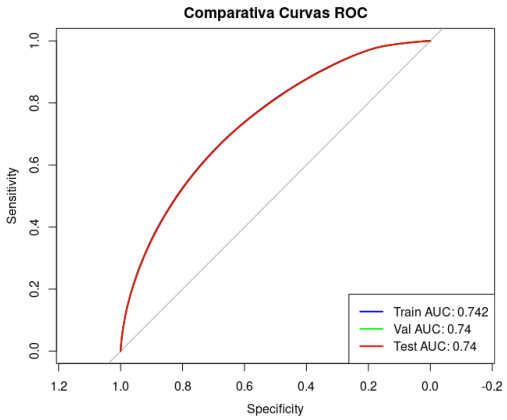


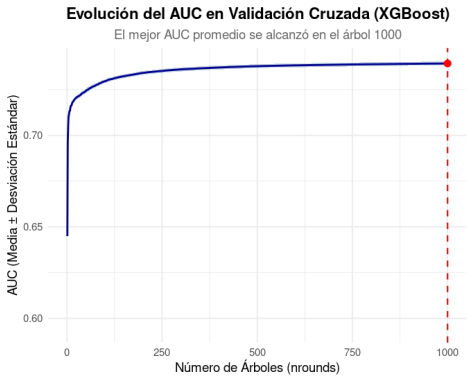
Figura 4.12: Variación da proporción de lecturas en función das cinco variables máis importantes do modelo de lecturas *Random Forest*.

maxnodes	eta	gamma	min_child_weigth	max_delta_step
1000	0.0880	9.0818	2845.202	0.7026
max_depth	subsample	colsample_bytree	colsample_bylevel	scale_pos_weight
5	0.5208	0.5141	0.7906	0.958185

Cadro 4.6: Hiperparámetros do modelo *XGBoost* óptimo para lecturas.



(a) Curva *ROC* do modelo *XGBoost*.



(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *XGBoost*.

Figura 4.13: Información relativa á métrica de validación do modelo *XGBoost* de lecturas.

Figura 4.14: Importancia das variables explicativas no modelo de lecturas *XGBoost*.

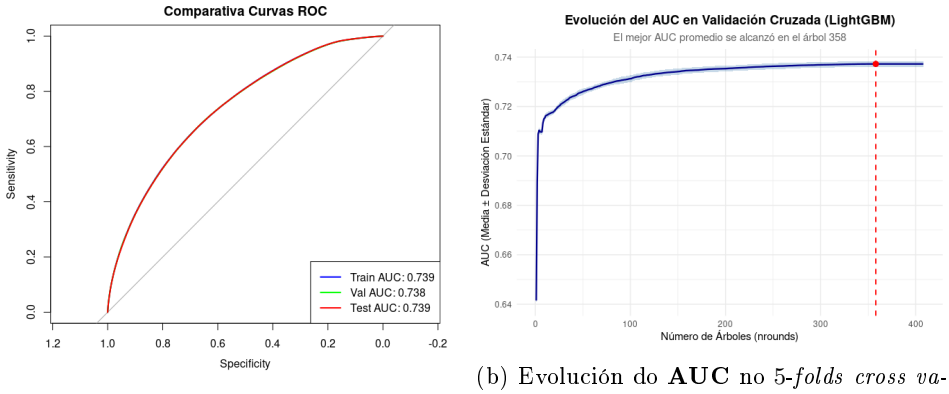
4.5.3. *LightGBM*

Vexamos, entón, que sucede se axustamos un modelo *LightGBM* para as variables anteriores. Se efectuamos a selección de hiperparámetros obtemos como modelo óptimo aquel que presenta os valores recollidos no Cadro 4.7.

Na Figura 4.15b vemos como evoluciona o valor do **AUC** de validación medio do proceso 5-*folds cross-validation* para cada iteración. Observamos como alcanza un óptimo para o valor 111. Este modelo, cuxas curvas *ROC* recollidas na Figura 4.15a para cada un dos estratos da mostra de datos, presenta un **AUC** de test moderado con valor 0.739. Así mesmo; os valores para a mostra de adestramento, test e validación presentan valores similares. O que é síntoma de que o modelo non padece problemas sobreaxuste. Se estudamos agora as curvas *ROC*, vemos como estas non son tan achatadas, polo que é de esperar que sexa un modelo relativamente equilibrado entre sensibilidade e especificidade.

nrounds	learning_rate	min_gain_to_split	num_leaves	min_data_in_leaf	lambda_l1
358	0.2004	1.5361	3626	6358	10
lambda_l2	max_depth	bagging_freq	feature_fraction	scale_pos_weight	early.stop.round
8.5584	3	1	0.7496	1	50

Cadro 4.7: Hiperparámetros do modelo *LightGBM* óptimo para lecturas.



(a) Curva *ROC* do modelo *LightGBM*.

(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *LightGBM*.

Figura 4.15: Información relativa á métrica de validación.

Seguidamente, validaremos o modelo baseándonos nas súas métricas mediante a súa matriz de confusión 4.8.

```
1 CONFIDENCIAL
```

Listing 4.7: Matriz de Confusión do modelo de lecturas *LightGBM*.

Na Figura 4.16 móstranse as 15 variables máis influentes do modelo xunto coa súa importancia reescalada.

Figura 4.16: Importancia das variables explicativas no modelo de lecturas *LightGBM*.

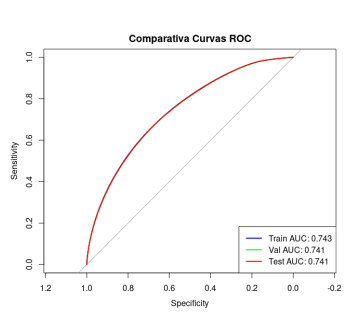
4.5.4. *CatBoost*

Finalmente, axustaremos un modelo de tipo *CatBoost* coas variables consideradas ao inicio da sección. Obtemos un modelo óptimo, con **AUC** con valor 0.7315, para os hiperparámetros recollidos no Cadro 4.8.

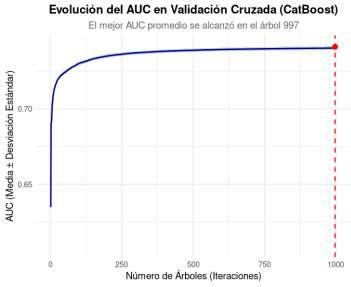
iterations	learning_rate	depth	l2_leaf_reg	rsm	class_weights
997	0.2477	3	5.6196	0.5795	1

Cadro 4.8: Hiperparámetros do modelo *CatBoost* óptimo para lecturas.

Na Figura 4.17b vemos a evolución do **AUC** durante o proceso de 5-*folds cv* a medida que aumenta o número de iteracións. Vemos como tende a estabilizarse para valores altos. Tamén calculamos a curva *ROC* e, en base a ela, o seu limiar óptimo. As curvas *ROC* das tres mostras aparecen recollidas na Figura 4.17a e, tal e como sucedía nos exemplos anteriores, o **AUC** do modelo é estable para as 3 mostras. Polo tanto, podemos concluír que o modelo non está sobreaxustado. Ademais, dada a forma da curva, podemos sospeitar que este modelo presentará un maior equilibrio entre especificidade e sensibilidade que os de lecturas a curto prazo.



(a) Curva *ROC* do modelo *CatBoost*.



(b) Evolución do **AUC** no 5-*folds cross validation* do modelo *CatBoost*.

Figura 4.17: Información relativa á métrica de validación do modelo de lecturas *CatBoost*.

En base ao limiar anterior, podemos calcular as estatísticas da mostra test do modelo mediante a súa matriz de confusión (*Listing* 4.8).

```
1 CONFIDENCIAL
```

Listing 4.8: Matriz de Confusión do modelo de lecturas *CatBoost*.

Para concluír, estudaremos a importancia das variables do modelo. Na Figura 4.18 aparecen recollidas as 15 máis influentes xunto coas súas importancias reescaladas.

Figura 4.18: Importancia das variables explicativas no modelo de lecturas *CatBoost*.

4.6. Valoración dos modelos

Tal e como comentabamos anteriormente, os nosos modelos non resultaban os máis axeitados á hora de clasificar o suceso da interacción. Malia isto, ese non era o obxectivo perseguido nesta memoria; senón que buscabamos estimar a probabilidade do éxito condicionada ás variables explicativas. Polo tanto, mesmo se as métricas de clasificación tradicionais (como o κ, PPV, ACC, etc) mostren un desempeño moderado, o **AUC** confirma que o modelo posúe unha alta capacidade de discriminación. Por iso argumentabamos que os elevados **AUC** permitían confirmar a adecuación do modelo. Se ben non podemos efectuar unha valoración dos modelos en termos de estudar a súa capacidade para ofrecer unha hora de envío axeitada, si que poderíamos estudar a súa adecuación a partir dos seus **AUC**.

No Cadro 4.9 móstranse os **AUC** relativos ás mostras de validación para cada un dos modelos. Tal e como vemos todos eles parecen devolver uns **AUC** relativamente similares, sendo lixeiramente inferiores os pertencentes aos Bosques Aleatorios. Isto unido ao custo computacional que presentan os mesmos, fannos decantar polos restantes modelos.

	Random Forest	*XGBoost*	*LightGBM*	*CatBoost*
LEIDOS_CP	0.721	0.741	0.739	0.741
LEIDOS	0.711	0.740	0.739	0.741

Cadro 4.9: **AUC**'s de mostras de validación.

Centrémonos agora nos modelos restantes con variable resposta *LEIDO_CP*: *XGBoost*, *LightGBM* e *CatBoost*. Se ben os tres mostran niveis de **AUC** similares e equilibrados para as nosas tres mostras, hai dous factores que nos fan decantarnos polo último deles.

- **Eficiencia computacional:** Dos catro modelos que escollemos axustar ao longo desta memoria, aqueles que presentan tempos de computación máis razoables son o *CatBoost* e o *LightGBM*. Polo que, ante valores de **AUC** similares, escolleremos aqueles que aceleren máis o proceso.
- **Importancia de variables:** Unha vez xa temos seleccionados os dous modelos comentados no ítem anterior, fixámonos na importancia das variables para cada modelo. O *CatBoost* outorga un maior peso á variable *HORA_CIRC_C*, polo que é de esperar que esta inflúa de xeito máis significativo sobre a nosa resposta. Isto fai que decantemos a balanza por este modelo.

Seguidamente, efectuamos o mesmo procedemento para o modelo de lecturas, tanto a curto, como a longo prazo. Seguindo a mesma lóxica que no caso anterior podemos resumir a nosa elección nos seguintes dous puntos:

- **Eficiencia computacional:** Do mesmo xeito que no caso anterior, ante **AUC** similares, escollemos *LightGBM* e *CatBoost*.
- **Importancia de variables:** Non obstante, nesta casuística, ambos modelos sitúan a *HORA_CIRC_C* como cuarta variable en importancia. Consecuentemente, seleccionaremos o *LightGBM* por mostrarse máis eficiente computacionalmente.

4.7. Calibrado

Unha vez seleccionados os nosos modelos definitivos, calibraremos as probabilidades do modelo para seguir avaliando as súas capacidades. No caso das lecturas a curto prazo, efectuaremos o estudo e

implementación sobre o modelo *CatBoost* e, para a outra variable resposta, efectuaremos dito estudo sobre o modelo *LightGBM* sendo totalmente análogos para o resto. Co fin de efectuar este proceso de calibrado empregaremos **regresión isotónica** (Barlow & Brunk, 1972; Brunk, Barlow, Bartholomew & Bremner, 1973) aplicada sobre o *score* dos nosos modelos (Fonseca & Lopes, 2017; Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002). Esta clase de regresión insírese dentro das metodoloxías non paramétricas, impoñendo unicamente a condición de que a función estimada sexa monótona. No noso caso; a probabilidade condicionada é, por definición, monótona crecente. Polo que esta metodoloxía resulta especialmente axeitada para esta casuística.

Supoñamos que queremos aproximar unha función estritamente non decrecente e, para iso, contamos con pares de observacións $\{(y_i, x_i)\}_{i=1,\dots,n}$ de xeito que:

$$y_i = f(x_i) + \varepsilon_i, \quad \forall i \in \{1, \dots, n\}.$$

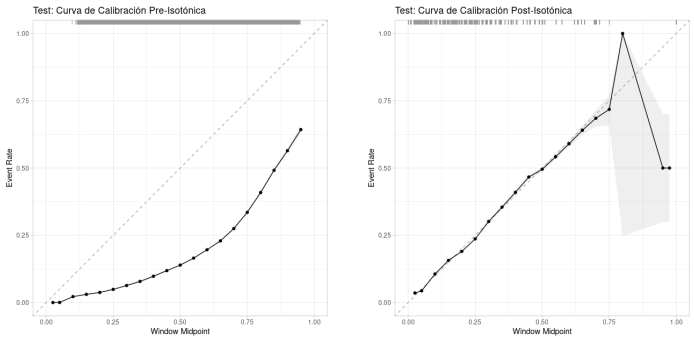
Entón buscamos minimizar unha determinada métrica do erro (xeralmente o **RSS**) suxeita á condición de monotonía. O algoritmo máis utilizado para solucionar este problema é o que se coñece como *Pair-Adjacent Violators* (**PAV**) (Barlow & Brunk, 1972; Fonseca & Lopes, 2017; Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002). Dito algoritmo busca atopar unha función en chanzos e constante por tramos para aproximar f minimizando o erro anteriormente mencionado, emporiso, non afondaremos máis nos pormenores deste algoritmo [2]. Para a nosa casuística, a función que buscamos aproximar sería $\mathbb{P}(\mathbf{Y} = 1|\mathbf{X} = \mathbf{x})$ e os nosos pares de observacións serían os *scores* achegados polos modelos (variable explicativa) xunto co valor de clase observado (variable resposta). Este enfoque é, precisamente, proposto para as calibracións de probabilidades a partir dos *scores* dos modelos de *Machine Learning* (Barlow & Brunk, 1972; Fonseca & Lopes, 2017; Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002). Para implementar esta metodoloxía acudiremos ao paquete `probably` de *R* (Kuhn, Vaughan & Ruiz, 2026).

Modelo de lecturas a curto prazo

Primeiramente, calculamos a curva de calibración da mostra test; a cal queda recollida na Figura 4.19a. Como vemos, a curva é monótona crecente. Isto indica que as probabilidades obtidas a partir do modelo son adecuadas no sentido de que, a maior probabilidade asignada ao cliente, maior é $\mathbb{P}(\mathbf{Y} = 1|\mathbf{X} = \mathbf{x})$. Adicionalmente calculamos o ***Brier Score*** do modelo sobre a mostra test, sendo este de 0.208. O cal é moi inferior ao valor de 0.25 que presentaría un modelo ao azar (entendendo por azar aquel que asigne unha probabilidade do 50 % a cada suceso). Non obstante, isto non implica que, necesariamente, sexa unha boa calibración (Hoessly, 2026). Ademais, observamos como a curva sitúase por debaixo da diagonal. Isto significa que o modelo tende a sobreestimar esta probabilidade.

Para solucionalo, propoñemos efectuar unha calibración isotónica (Fonseca & Lopes, 2017; Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002) sobre a mostra de validación e, posteriormente, aplicar esta calibración sobre as predicións propostas polo modelo. Na Figura 4.19b observamos a curva de calibración das probabilidades tras aplicar regresión isotónica. Vemos como dita curva achégase á diagonal, reflectindo unha mellora substancial na fiabilidade das probabilidades preditas. Deste xeito, o *Brier Score* redúcese a 0.108 e, tendo en conta que a proporción da clase positiva sobre a mostra é do 14 %, un modelo aleatorio obtería unha puntuación de 0.12. Polo que o noso modelo parecería máis axeitado. Por outra banda, presenta unha **log-loss** de 0.356 o cal tradúcese nunha mellora dun 12.1 % con respecto a un modelo que prediga sempre a proporción de éxitos sobre a mostra. Tendo todo isto en conta, podemos concluír que as probabilidades estimadas para a posterior proposta de hora óptima son fiables. Cabe destacar que si presenta maior variabilidade e problemática para valores máis elevados debido á ausencia de datos neses niveis. Non obstante, a nivel de negocio, esta problemática non supón un obstáculo á hora de cumprir co noso cometido.

[2] Para máis información, consultar Barlow e Brunk (1972) e Brunk e col. (1973).



(a) Curva de calibración das probabilidades preditas por *CatBoost* sobre a mostra test previo á calibración isotónica.

(b) Curva de calibración das probabilidades preditas por *CatBoost* sobre a mostra test tras aplicar regresión isotónica.

Figura 4.19: Curvas de calibración para o modelo de lecturas a curto prazo.

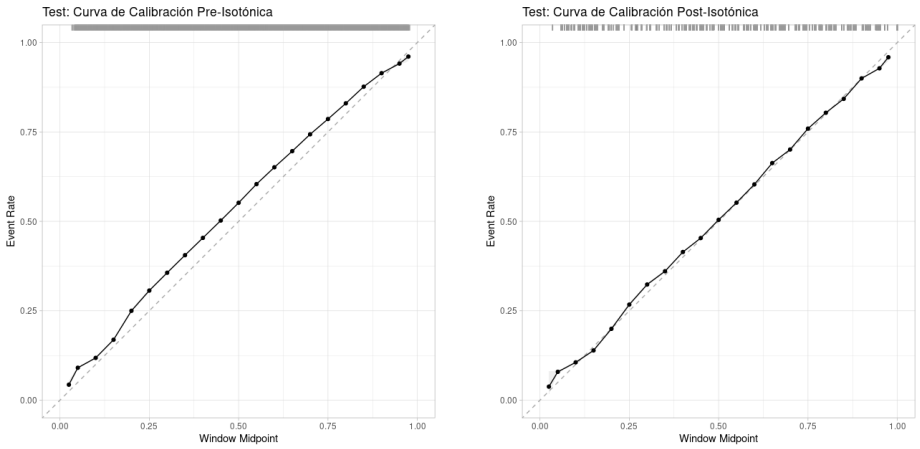
Modelo de lecturas

Deseguido, repetimos o proceso anterior para o modelo seleccionado para estudar as lecturas. Na Figura 4.20a recóllese a curva de calibración dos *scores* previos á regresión isotónica (Barlow & Brunk, 1972; Brunk e col., 1973; Fonseca & Lopes, 2017; Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002). Tal e como vemos, sitúanse lixeiramente por riba da liña diagonal. Isto quere dicir que o noso modelo tende a subestimar moi sutilmente a probabilidade de lectura. O seu *Brier Score* tomará un valor de 0.204 o cal, tendo en conta que a proporción de éxitos da mostra é de 0.55, supón unha redución dun 18 % cun respecto a un modelo que prediga sempre a probabilidade esperada de éxitos (o 55 %). Unha vez aplicada a calibración isotónica (Figura 4.20b), observamos como as estimacións xa se adecúan máis. Malia que o *Brier Score* mellora moi levemente (0.202), a *Log-Loss* do modelo toma un valor de 0.588. Isto significa unha mellora do 14 % con respecto a un modelo que asigne sempre a proporción de éxitos da mostra.

Concluíndo así que se trata dun modelo axeitado para o noso cometido. Notemos que, ao contar cun maior número de éxitos, non se da a casuística do apartado anterior. Isto débese a que contamos cunha densidade de datos axeitada para todos os niveis de probabilidade.

	Brier Score	*Log-loss*
Post-calibrado lecturas a curto prazo	0.108	0.356
Post-calibrado lecturas	0.202	0.588

Cadro 4.10: Métricas de calibrado.



(a) Curva de calibración das probabilidades preditas por *LightGBM* sobre a mostra test previo á calibración isotónica.

(b) Curva de calibración das probabilidades preditas por *LightGBM* sobre a mostra test tras aplicar regresión isotónica.

Figura 4.20: Curvas de calibración para o modelo de lecturas.

Capítulo 5

Posta en produción

Neste capítulo exporemos a parte máis técnica do proceso, que consistiría na aplicación dos modelos previamente calculados sobre o noso problema concreto. Tal e como comentamos na introdución da memoria (Capítulo 1), o obxectivo do traballo consistía en asignar para cada cliente unha hora de envío óptima co propósito de aumentar o impacto das comunicacións sobre o usuario. O procedemento para levalo a cabo consistiría en estimar a probabilidade dun cliente de interactuar cun *email* enviado nun determinado instante e, posteriormente, comprobar cal deles maximiza dita probabilidade. Presentaremos o caso das lecturas a curto prazo, emporiso, o caso de lecturas en xeral será totalmente análogo.

Primeiro, debemos preparar un *dataset* sobre o que efectuar as nosas predicións. Para este proceso, consideraremos os rexistros de clientes entre as datas: "21-11-2025" e "20-02-2026". Ademais, imporemos a condición de que sexan clientes que recibisen máis de 10 comunicacións e, de entre eles, tomaremos os 100 que maior proporción de interacción mostren. O motivo desta restrición é seleccionar usuarios cun número de interaccións significativas. Seguidamente, será necesario calcular as distintas variables relativas a eles, polo que deberemos replicar o proceso efectuado no adestramento do modelo. Notemos que, neste caso, poderemos considerar todos os datos da mostra. Isto débese a que buscamos predicir un acontecemento futuro e, consecuentemente, ningún dos rexistros da nosa mostra outorgará directamente información da resposta.

Entón o conxunto de datos presentará unha estrutura como a seguinte:

- **Variables relativas ás interaccións:** Nesta sección recóllense as variables: *PROP_LEIDO_CP_HORA*, *PROP_LEIDO_HORA*, *RETARDO_MEDIANO_HORA*, *RETARDO_APROX_LEIDO_CP*, *RETARDO_APROX_LEIDO*, *PERFECCION_LEIDO_CP*, *PERFECCION_LEIDO*, *NUM_DIAS_ACCESO* e *TELEFONO1*. Para o cálculo destas utilizaremos toda a mostra anterior, particularizando os resultados para cada cliente.

- **Variables recollidas da táboa de datos do cliente:** Nesta categoría englóbanse as variables: *NIVEL2*, *SEGMENTO_ID*, *EDAD*, *SEXO*, *AMBITO_CLIENTE*, *VINCULACION_TRANS_CT*, *VINCULACION_NEG_CT*, *MESES_ANTIGUEDAD_CLIENTE*, *IMP_NOMINA_SUA*, *IMP_PENSION_SUA* e *IN_DIGITAL*. Buscaremos, entón, o rexistro máis recente para cada cliente nestas datas. A idea é ter a información máis actualizada posible sobre o usuario.

- **Variables simuladas:** Estas serán aquelas variables ás que lles outorgaremos valores "artificiais", en función do fenómeno que buscamos predicir. Neste segmento incluiríamos ás variables: *CATEGORÍA_UNIF* e *HORA_CIRC_C*.

En resumo, construiremos unha táboa onde se recollan as variables anteriores asignando valores para as que notamos como **variables simuladas**. Por exemplo, se estamos interesados en coñecer a hora ideal de envío para un cliente determinado, inserimos a *CATEGORIA_UNIF* e consideramos as

24 horas de envío posibles para *HORA_CIRC_C*. Unha vez feito isto, calculamos as variables dependentes da hora de envío e aquelas baseadas no histórico do cliente. Finalmente, efectuamos predicións e seleccionamos aquela hora que maximice a probabilidade de interacción.

Unha vez temos un *dataset* sobre o que traballar, será necesario cargar o modelo para realizar as nosas predicións. Comezaremos considerando o caso de lecturas a curto prazo onde, lembremos, seleccionamos o modelo *CatBoost* como preferible. Non obstante, o proceso sería totalmente análogo para o resto de modelos. Dito procedemento será realizado mediante a librería `catboost`. No *Listing* 5.1 recóllese un esquema de como funcionaría o procedemento en *R* para a variable *LEIDO_CP* unha vez xa temos o noso modelo (`modelo_leidos_cp`) e o obxecto de calibrado (`calibrado`).

```
1 CONFIDENCIAL
```

Listing 5.1: *Script* para a obtención da hora ideal.

Deste xeito, teremos un *dataset* onde se recollen *CLIENTE_ID* e as probabilidades de apertura en función da hora de envío (recollidas nas columnas). Dita táboa tería a forma dada polo *Listing* 5.2.

```
1 CONFIDENCIAL
```

Listing 5.2: *Query* para a obtención da hora ideal.

Se ben resulta interesante coñecer a hora de envío ideal global, deberíamos ter en conta que dende a entidade non poden efectuar envíos a calquera hora do día. Para resolver esta casuística propoñemos a construción dunha función que faga o seguinte: dado un *CLIENTE_ID* xunto con todas as súas características (recollidas no *data frame* `produccion`), un intervalo horario e unha calibración de probabilidades efectuada sobre a mostra de validación (`calibracion`); obter as probabilidades estimadas do modelo para cada hora contida no intervalo horario. A implementación de dita función, que denominamos `prob.leido_cp()`, está recollida no *Listing* 5.3.

```
1 CONFIDENCIAL
```

Listing 5.3: *Script* para a obtención da hora ideal.

Agora describiremos o procedemento para as lecturas, considerando o modelo *LightGBM*; o cal mudará lixeiramente con respecto ao anterior. Neste caso, utilizaremos o modelo *LightGBM* que seleccionamos no capítulo anterior. A implementación está recollida no *Listing* 5.4. A cal devolve unha táboa como a recollida no *Listing* 5.5.

```
1 CONFIDENCIAL
```

Listing 5.4: *Script* para a obtención da hora ideal.

```
1 CONFIDENCIAL
```

Listing 5.5: *Script* para a obtención da hora ideal.

Como vemos, agora as horas de envío propostas polo modelo difiren bastante daquelas que teñen lugar a curto prazo. De feito, algunha delas podería non ser unha hora realista para o envío de *emails*. Por iso, creamos unha función que permita, ademais, establecer un rango de horas nas que se pretende efectuar o envío. Dita función aparece recollida no *Listing* 5.6.

```
1 CONFIDENCIAL
```

Listing 5.6: *Script* para a obtención da hora ideal.

Capítulo 6

Conclusións e liñas de estudo

Neste capítulo trataremos algúns aspectos relativos á realización do traballo, así como as conclusións e futuras liñas de estudo para o mesmo que non puideron ser abarcadas nesta memoria.

6.1. Conclusións

Neste traballo buscamos achegar unha solución a un problema crítico para a entidade, como é a política de comunicacións cos seus clientes vía *email*. Primeiramente, efectuamos unha análise exhaustiva dos rexistros de *emails* recollidos pola entidade ao longo dun ano. Proseguimos identificando as variables resposta (*LEIDO* e *LEIDO_CP*) e definindo, mediante técnicas de *feature engineering*, as explicativas; das que tamén estudamos o seu efecto sobre as nosas respostas. Dúas das principais dificultades do noso traballo residiron, por un lado, na enorme cantidade de rexistros cos que era necesario traballar e, por outro, o nesgo producido polo patrón de envíos actual da empresa. Para solucionar estes dous puntos, conxuntamente, deseñouse unha técnica de *subsampling*. A continuación; escollemos metodoloxías que puidesen resultar axeitadas para o noso cometido (no noso caso, *Random Forest*, *XGBoost*, *LightGBM* e *CatBoost*), emporiso, valoráronse outras alternativas ás mesmas. Seguidamente, efectuouse unha selección de variables e hiperparámetros para as metodoloxías escollidas e, tras o axuste dos modelos, seleccionamos aqueles que mostrasen mellores aptitudes. Finalmente, mediante técnicas de calibrado, calculáronse *scorings* fiables e implementáronse os procedementos indicados para a estimación da hora ideal de envío. Polo tanto, podemos destacar os seguintes puntos clave da memoria:

- ***Feature engineering*:** Construímos novas variables para capturar o efecto da hora do envío e o histórico do cliente, agrupándoas nunha táboa de adestramento para axustar os modelos.

- **Estimación da probabilidade:** Mediante metodoloxías de clasificación binaria e calibrado fomos capaces de obter estimacións robustas da probabilidade de interacción dun cliente en función da hora de envío.

- **Implementación práctica:** Adicionalmente, propuxéronse procedementos para o cálculo destas estimacións mediante *software R*.

Deste xeito, conseguimos modelos con unha gran robustez analítica, sinxelos de implementar e altamente eficientes.

6.2. Liñas de estudo

Adicionalmente, presentamos certas liñas de actuación a futuro para estender as capacidades no noso modelo:

- **Test A/B** : O seguinte paso natural do traballo desenvolto na presente memoria é comprobar se o noso modelo é efectivo. Para iso sería necesaria a elaboración dun Test A/B. Deste xeito, comprobamos se o noso modelo presenta unha mellora estatisticamente significativa con respecto a unha política de envíos aleatoria. Para facelo, seleccionaremos unha mostra de clientes no contexto dunha determinada campaña. O 50 % deles recibirá as comunicacións en horarios aleatorios, mentres que ao outro 50 % contactarémoslles nas horas proporcionadas polo modelo. Seguidamente, definiremos as métricas que serán estudadas durante o período que dure o estudo. Inicialmente propoñemos: a **taxa de lecturas media dos clientes**, a **taxa de aperturas total**[1] e a de ***clicks***. A idea sería comprobar se algunha destas métricas mellora directamente, como podería ser o caso das lecturas, ou indirectamente, como sería o caso dos *clicks*. Antes do lanzamento de dito proceso, será preciso realizar unha análise de potencia estatística para escoller o tamaño de mostra axeitado para garantir unha correcta potencia e significación estatística. Unha vez rematado o período de envíos, efectuaremos un contraste de hipóteses para proporcións a fin de comprobar se o noso procedemento presenta algunha mellora significativa.

  Malia que a realización deste proceso resulta fundamental para a validación da metodoloxía, a súa implementación excede o alcance e as limitacións temporais desta memoria. Isto débese á alta complexidade organizativa que require a posta en marcha deste test, ao ter que coordinar diversas áreas da entidade simultaneamente (*C.R.M. Omnicanal*, *Marketing*, etc.). Non obstante, esta será a próxima liña de traballo dentro do departamento.

- **Política de envíos aleatorios:** Sería interesante propoñer unha política de envío que contacte aos usuarios a diferentes horas do día, de xeito que contemos cunha mostra onde todas as horas estean equitativamente representadas.

- **Inclusión de modelos de propensión:** Unha importante limitación do proxecto consistía na ausencia de variables que recollan o contido da mensaxe e a propensión do cliente ante o mesmo. Non obstante, na base de datos da entidade non existían métodos efectivos para acceder ao contido da mensaxe nin á propensión do cliente cara ese tipo de produto. Polo que este enfoque escápase das pretensións da presente memoria.

- ***Data Drift*:** O ***Data Drift*** é un fenómeno que ten lugar cando a distribución dos datos relativos ás variables explicativas do noso modelo varía con respecto á que presentaban cando o modelos foi adestrado (Mannapur, 2025). Dado o paradigma baixo o que estamos traballando, sería interesante contar cunha medida de control deste efecto; pois os hábitos de clientes poden sufrir enormes variacións co paso do tempo. En particular; propoñemos adaptar o enfoque de Zamzmi e col. (2025) onde, mediante técnicas de Control Estatístico da Calidade, monitorizan este *drift* para unha mostra de imaxes radiolóxicas.

[1] Para ambas taxas consideraremos, por unha banda, as lecturas a curto e,por outra, aquelas que sexan independentes da tardanza de apertura.

Bibliografía

0xdata, Inc. (2013). H2O Documentation. https://h2o-release.s3.amazonaws.com/h2o/master/1757/docs-website/index.html. Documentación técnica de H2O 2.9.0.1757.

Araújo, C., Soares, C., Pereira, I., Coelho, D., Rebelo, M. Â. & Madureira, A. (2022). A Novel Approach for Send Time Prediction on Email Marketing. *Applied Sciences*, *12*(16). doi:10.3390/app12168310

Barlow, R. E. & Brunk, H. D. [H. D.]. (1972). The Isotonic Regression Problem and its Dual. *Journal of the American Statistical Association*, *67*(337), 140-147. doi:10.1080/01621459.1972.10481216. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1972.10481216

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. doi:10.1023/A:1010933404324

Brunk, H. D. [Hugh D.], Barlow, R. E., Bartholomew, D. J. & Bremner, J. M. (1973). Statistical inference under order restrictions : the theory and application of isotonic regression. *International Statistical Review*, *41*, 395. Consultado desde https://api.semanticscholar.org/CorpusID:120349543

CatBoost Documentation. (2026). https://catboost.ai/docs/en/. Accedido: 11-03-2026.

Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *CoRR*, *abs/1603.02754*. arXiv: 1603.02754. Consultado desde http://arxiv.org/abs/1603.02754

Chou, Y.-L. (1977). *Análisis estadístico* (2ª ed). México [etc: Interamericana.

Deligiannis, A., Argyriou, C. & Kourtesis, D. (2020a). Building a Cloud-based Regression Model to Predict Click-through Rate in Business Messaging Campaigns. *International Journal of Modeling and Optimization*, *10*, 26-31. doi:10.7763/IJMO.2020.V10.742

Deligiannis, A., Argyriou, C. & Kourtesis, D. (2020b). Predicting the Optimal Date and Time to Send Personalized Marketing Messages to Repeat Buyers. *International Journal of Advanced Computer Science and Applications*, *11*, 90-99. doi:10.14569/IJACSA.2020.0110413

Dietterich, T. G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, *40*, 139-157. Consultado desde https://api.semanticscholar.org/CorpusID:12394453

Dorogush, A. V., Gulin, A., Gusev, G., Kazeev, N., Prokhorenkova, L. O. & Vorobev, A. (2017). Fighting biases with dynamic boosting. *CoRR*, *abs/1706.09516*. arXiv: 1706.09516. Consultado desde http://arxiv.org/abs/1706.09516

Dunn, P. K. & Smyth, G. K. (2018). *Generalized Linear Models With Examples in R* (1st ed. 2018.). New York, NY: Springer New York.

Ellering, N. (2025). What 10 Studies Say About The Best Time To Send Email. Consultado el 25 de febrero de 2026, desde https://coschedule.com/content-marketing/best-time-to-send-email

Empresa, S. I. (2026). *¿Qué es un KPI y para qué sirve?* Informacón sobre la definición y uso de los indicadores clave de rendimiento (KPI). Consultado desde https://www.impulsa-empresa.es/diccionario/kpi/

Fernández Casal, R., Cao, R. & Costa, J. (2023). Técnicas de simulación y remuestreo. Consultado desde https://github.com/rubenfcasal/simbook?tab=readme-ov-file

Fernández-Casal, R., Costa, J. & Oviedo de la Fuente, M. (2024). *Métodos predictivos de aprendizaje estadístico*. Consultado el 12 de febrero de 2026, desde http://hdl.handle.net/2183/37227

Fonseca, P. G. & Lopes, H. D. (2017). Calibration of Machine Learning Classifiers for Probability of Default Modelling. arXiv: 1710.08901 [econ.EM]. Consultado desde https://arxiv.org/abs/1710.08901

Friedman, J. (2002). Stochastic Gradient Boosting. *Computational Statistics & Data Analysis*, *38*, 367-378. doi:10.1016/S0167-9473(01)00065-2

Friedman, J., Hastie, T. & Tibshirani, R. (2000). Additive Logistic Regression: A Statistical View of Boosting. *The Annals of Statistics*, *28*(2), 337-407.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189-1232. Consultado desde https://api.semanticscholar.org/CorpusID:39450643

Haan, K. & Watts, R. (2026). 49 Top Email Marketing Statistics. Consultado el 24 de febrero de 2026, desde https://www.forbes.com/advisor/business/software/email-marketing-statistics-feb-26/

Härdle, W., Werwatz, A., Müller, M. & Sperlich, S. (2004). *Nonparametric and Semiparametric Models* (1.ª ed.). Series ISSN: 0172-7397. doi:10.1007/978-3-642-17146-8

Hastie, T., Friedman, J. & Tibshirani, R. (2001). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction* (1st ed. 2001.). New York, NY: Springer New York.

Hoessly, L. (2026). On misconceptions about the Brier score in binary prediction models. *Global Epidemiology*, *11*, 100242. doi:10.1016/j.gloepi.2025.100242

Hothorn, T., Hornik, K. & Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, *15*, 651-674. Consultado desde https://api.semanticscholar.org/CorpusID:6074128

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning : with Applications in R* (1st ed. 2013.). New York, NY: Springer New York.

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *29*(2), 119-127. Consultado el 27 de marzo de 2026, desde http://www.jstor.org/stable/2986296

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. En *Neural Information Processing Systems*. Consultado desde https://api.semanticscholar.org/CorpusID:3815895

Kuhn, M., Vaughan, D. & Ruiz, E. (2026). *probably: Tools for Post-Processing Predicted Values*. R package version 1.2.0. Consultado desde https://probably.tidymodels.org

Liaw, A., Wiener, M., Breiman, L. & Cutler, A. (2022). *Package 'randomForest': Breiman and Cutler's Random Forests for Classification and Regression*. R package version 4.7-1.2. Consultado desde https://CRAN.R-project.org/package=randomForest

Loh, W.-Y. (2009). Improving the precision of classification trees. *The Annals of Applied Statistics*, *3*(4). doi:10.1214/09-aoas260

Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G. & Ziegler, A. (2012). Probability machines: consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, *51*(1), 74-81. doi:10.3414/ME00-01-0052

Mannapur, S. B. (2025). Understanding Data Drift and Concept Drift in Machine Learning Systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *11*(1), 318-330. doi:10.32628/CSEIT25111239

Niculescu-Mizil, A. & Caruana, R. (2005). Predicting good probabilities with supervised learning. En *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625-632). doi:10.1145/1102351.1102430

Nocedal, J. & Wright, S. J. (2006). *Numerical Optimization* (2nd). doi:10.1007/978-0-387-40065-5

Pal, A., Bansal, S., Singh, S., Hiran, S. & Yadav, P. (2022). Dynamic Best Send Time Prediction for Marketing Email Campaigns. En *2022 International Joint Conference on Information and Communication Engineering (JCICE)* (pp. 92-99). doi:10.1109/JCICE56791.2022.00029

Paralič, J., Kaszoni, T. & Mačina, J. (2020). Predicting Suitable Time for Sending Marketing Emails. En J. Świątek, L. Borzemski & Z. Wilimowska (Eds.), *Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems*

Architecture and Technology – ISAT 2019 (pp. 189-196). Cham: Springer International Publishing.

Rueda, C., Fernández, M. A., Barragán, S., Mardia, K. V. & Peddada, S. D. (2016). Circular Piecewise Regression with an Application to Cell-cycle Biology. *Biometrics*, *72*(4), 1266-1274.

Saleh Abbas, D. & Al-Jailawi, M. (2024). Data-Driven Strategies for Improving Email Campaign Engagement: A Send Time Optimization Approach. Danielsson, Fredrik e Giselsson, Pontus.

Schapire, R. E. (2013). Explaining AdaBoost. En B. Schölkopf, Z. Luo & V. Vovk (Eds.), *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik* (pp. 37-52). doi:10.1007/978-3-642-41136-6_5

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, *104*(1), 148-175. doi:10.1109/JPROC.2015.2494218

Shao, J. & Tu, D. (1995). *The jackknife and bootstrap* (1.ª ed.). New York, NY: Springer.

Shi, Y., Ke, G., Soukhavong, D., Lamb, J., Meng, Q., Finley, T., ... Mayer, M. (2025). *lightgbm: Light Gradient Boosting Machine*. R package version 4.6.0. Consultado desde https://cran.r-project.org/web/packages/lightgbm/refman/lightgbm.html

Sinha, M., Vinay, V. & Singh, H. (2018). Modeling Time to Open of Emails with a Latent State for User Engagement Level. En *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 531-539). doi:10.1145/3159652.3159683

Snoek, J., Larochelle, H. & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. arXiv: 1206.2944 [stat.ML]. Consultado desde https://arxiv.org/abs/1206.2944

Wilson, S. (2020). *ParBayesianOptimization: Parallel Bayesian Optimization of Hyperparameters*. R package version 1.2.6. Consultado desde https://CRAN.R-project.org/package=ParBayesianOptimization

Yuan, J., Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., ... Lara-Knuutila, K. (2026). *Package 'xgboost': Extreme Gradient Boosting*. R package version 3.2.1.1. Consultado desde https://github.com/dmlc/xgboost

Zadrozny, B. & Elkan, C. P. (2002). Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. Consultado desde https://api.semanticscholar.org/CorpusID:3349576

Zamzmi, G., Venkatesh, K., Nelson, B., Prathapan, S., Yi, P., Sahiner, B. & Delfino, J. G. (2025). Out-of-Distribution Detection and Radiological Data Monitoring Using Statistical Process Control. *Journal of Imaging Informatics in Medicine*, *38*, 997-1015. doi:10.1007/s10278-024-01212-9