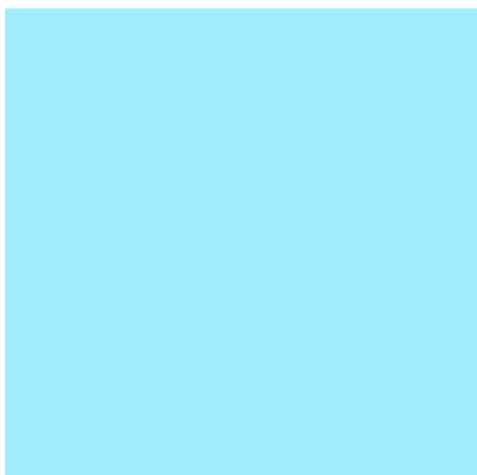


Comparación de Modelos Tradicionales y Algoritmos de Machine Learning en la Estimación de Propensity Scores



Universitat Oberta
de Catalunya



UNIVERSITAT DE
BARCELONA

Cristina Juárez Alía

Estudios Clínicos y Epidemiológicos

Máster en Bioinformática y Bioestadística

Nombre del director/a de TF:

Susana Pérez Álvarez

Nombre del/de la PRA:

Antonio Pita Lozano

4 de Junio de 2025



Esta obra esta sujeta a una licencia de Reconocimiento

<https://creativecommons.org/licenses/by-nc/3.0/es/>

Ficha Del Trabajo Final

Título del trabajo:	Comparación de Modelos Tradicionales y Algoritmos de Machine Learning en la Estimación de Propensity Scores
Nombre del autor/a:	Cristina Juárez Alía
Nombre del director/a de TF:	Susana Pérez Álvarez
Nombre del/de la PRA:	Antonio Pita Lozano
Fecha de entrega:	4 de Junio de 2025
Titulación o programa:	Máster en Bioinformática y Bioestadística
Área del trabajo final:	Estudios Clínicos y Epidemiológicos
Idioma del trabajo:	Castellano
Palabras clave:	machine learning, epidemiología, propensity score

Resumen del trabajo

La regresión logística (RL) es un método ampliamente utilizado para estimar el propensity score (PS) en estudios observacionales. Sin embargo, es frecuente omitir relaciones complejas o interacciones entre covariables en la fórmula de predicción. Este estudio evalúa si los algoritmos de machine learning (ML) pueden mejorar la estimación del PS y el balance de covariables detectando dichas relaciones complejas entre covariables de forma automática.

Se replicaron tres escenarios simulados a partir de estudios previos, incorporando una complejidad creciente en la relación entre tratamiento y covariables. Los PS fueron estimados mediante RL y tres modelos de ML: Random Forest (RF), Classification and Regression Trees (CART) y Gradient Boosting Machines (GBM). El balance de covariables se evaluó mediante la Diferencia de Medias Estandarizada Absoluta (ASMD) y su promedio (ASAM), utilizando dos métodos habituales de aplicación del PS: emparejamiento y IPW (ponderación por probabilidad inversa). Además, se aplicó una estrategia innovadora para la selección de hiperparámetros, basada en la minimización del ASAM en lugar de métricas clásicas de precisión, según estudios recientes.

Los resultados muestran que Random Forest alcanza un mejor balance de covariables que la regresión logística en el escenario más complejo. Estos hallazgos respaldan el uso de Random Forest como una alternativa robusta a la regresión logística para la estimación del PS en estudios observacionales.

Abstract

Logistic regression is a widely used method for estimating the propensity score (PS) in observational studies. However, it is frequently affected by model misspecification, as complex relationships or interactions among covariates are often omitted from the prediction model. This study evaluates whether machine learning (ML) algorithms can improve PS estimation and covariate balance in such misspecified settings.

Three simulated scenarios were replicated from previous studies, incorporating increasing complexity in the relationship between treatment and covariates. Propensity scores were estimated using logistic regression and the three ML models: Random Forest (RF), Classification and Regression Trees (CART), and Gradient Boosting Machines (GBM). Covariate balance was evaluated using the Absolute Standardized Mean Difference (ASMD) and its average (ASAM), under two common PS application methods: matching and inverse probability weighting (IPW). Additionally, innovative hyperparameter tuning strategy was applied, based on minimizing ASAM rather than maximizing prediction accuracy based on recent previous studies.

The results show that Random Forest achieves better covariate balance than logistic regression in the more complex scenario. These findings support the use of Random Forest as a robust alternative to logistic regression for PS estimation in observational research.

Índice general

1. Introducción	9
1.1. Contexto y justificación del trabajo	9
1.2. Objetivos del trabajo	11
1.3. Impacto en sostenibilidad, ético-social y de diversidad	11
1.4. Enfoque y método seguido	12
1.5. Planificación del trabajo	12
1.6. Breve resumen de productos obtenidos	12
2. Estado del Arte	14
2.1. Inferencia Causal en Epidemiología	14
2.1.1. Asociación y Causalidad	14
2.1.2. Estimación del efecto causal	16
2.1.3. Supuestos de la inferencia causal	17
2.1.4. Sesgo de confusión en estudios observacionales	18
2.1.5. Conceptualización del problema causal	20
2.2. <i>Propensity Score</i> como método de ajuste	21
2.2.1. Definición del <i>Propensity Score</i>	21
2.2.2. Métodos de aplicación del <i>Propensity Score</i>	22
3. Materiales y Métodos	23
3.1. <i>Workflow</i> general	23
3.2. <i>Software</i> y paquetes	24
3.3. Generación de datos simulados	24
3.3.1. Modelización de la asignación al tratamiento	24
3.3.2. Modelización del desenlace	24
3.3.3. Escenarios de simulación	25
3.4. Estimación de propensity scores	26
3.5. Selección de hiperparámetros en modelos de machine learning	26
3.6. Aplicación del <i>propensity score</i>	28
3.7. Evaluación del balance y sesgo en la estimación del efecto causal	28
4. Resultados	30
4.1. Balance inicial de los datos simulados	30
4.2. Evaluación de la estimación del propensity score	31

4.3. Balance global después del ajuste	32
4.4. Balance de covariables después del ajuste	32
5. Discusión	34
5.1. Limitaciones del estudio	35

Índice de figuras

1.	Publicaciones sobre técnicas de <i>machine learning</i> para la estimación de <i>propensity scores</i> en ciencias de la salud desde 1983 a 2023	10
2.	Cronograma de planificación del trabajo	12
3.	Asociación observada e inferencia causal	14
4.	Distinción entre asociación y causalidad	15
5.	Estimandos del efecto causal en el contexto del modelo contrafactual	17
6.	Ejemplo de relación causal y asociación por confusión.	19
7.	Control de la confusión en ensayos aleatorios y estudios observacionales.	20
8.	Ejemplo de grafo acíclico dirigido	21
9.	<i>Workflow</i> general del análisis	23
10.	Proceso de ajuste de hiperparámetros basado en el ASAM.	27
11.	Gráfico de dispersión de PS verdaderos frente a PS estimados por distintos métodos.	31
12.	Balance global (ASAM) tras la aplicación de los <i>propensity scores</i> estimados	32
13.	Loveplots del balance de covariables (ASMD) tras la aplicación de PS estimados mediante <i>matching</i> y IPW	33

Índice de cuadros

1.	Condiciones para el cumplimiento de los supuestos de inferencia causal en ensayos aleatorizados y observacionales	18
2.	Covariables y coeficientes para la modelización del tratamiento y el desenlace en escenarios simulados.	25
3.	Métodos de machine learning utilizados, paquetes en R y configuración de hiperparámetros	27
4.	Promedio y desviación estándar de ASAM y ASMD antes del ajuste.	30

Capítulo 1

Introducción

1.1. Contexto y justificación del trabajo

La distribución de enfermedades y sus determinantes no es aleatoria. Cada individuo posee unas características específicas (genéticas y ambientales) que lo predisponen o protegen de ciertas enfermedades. La epidemiología nace para descubrir estos patrones en poblaciones específicas, con el fin de mejorar la salud pública mediante la identificación de la etiología y los factores de riesgo, la determinación del pronóstico y alcance de las enfermedades, y la evaluación del impacto de medidas preventivas, terapias y servicios sanitarios [1].

Frecuentemente, para lograr dichos objetivos los estudios epidemiológicos están centrados en determinar si una *exposición* o *tratamiento* es causa directa de un *efecto*, en cuyo caso decimos que existe una relación *causal*. Frecuentemente, dicho análisis surge desde la observación previa de una *asociación*, que es la variación de la variable *efecto*, en función de la variable *exposición* o *tratamiento*.

Sin embargo, no todas las asociaciones se deben a relaciones causales. Las variables pueden estar asociadas debido a la existencia de otros factores que actúan como causa común de la *exposición* o *tratamiento*, como del *efecto*, distorsionando la relación causa-efecto. Dichos factores reciben el nombre de *factores de confusión*.

En ensayos aleatorizados (o experimentales), los grupos se diseñan de modo que la distribución de los factores de confusión esté equilibrada, lo que permite eliminar el sesgo antes de la exposición o tratamiento. Así, la diferencia observada entre los grupos se debe exclusivamente al tratamiento, permitiendo estimar directamente el efecto causal del mismo (**Figura 7**).

En estudios observacionales (o no experimentales), se utilizan *datos reales* (*Real World Data*, ampliamente conocido por sus siglas *RWD*), que se registran durante la práctica clínica habitual. Esto significa que las cohortes están definidas según las experiencias del *mundo real*, y por tanto cada una está sujeta a unas determinadas características, lo que da lugar al sesgo de confusión [2]p. 93-94. Si no se balancean las poblaciones adecuadamente, el efecto observado se debe en parte al factor de confusión, distorsionando el efecto del tratamiento.

Para controlar dicho sesgo, Rosebaum y Rubin introducen en 1983 el *Propensity Score* (en

español, "Puntuaje de Propensión"), un valor que se asigna a cada sujeto del estudio, y que se define como: "la probabilidad condicional de asignación al tratamiento dado un vector de covariables observadas" [3]. En su definición, el *vector de covariables* se refiere a las características del individuo.

Tradicionalmente, los *propensity scores* (PS) siempre se han estimado mediante modelos de regresión logística (RL), tal y como propusieron Rosenbaum y Rubin [3]. Dicho modelo ofrece una interpretación directa de los resultados (valores de probabilidad entre 0 y 1). Además, su implementación es relativamente sencilla desde *softwares* estadísticos populares (como R, STATA, o SAS) [4].

Sin embargo, a menudo las suposiciones de la RL han sido ignoradas (como la ausencia de multicolinealidad); y en la mayoría de los casos se establece en su forma convencional, es decir, sin considerar términos complejos en su fórmula (como términos de interacción, o polinomios de alto grado), lo cual puede limitar el ajuste óptimo del modelo [5].

En este contexto, el uso de métodos alternativos basados en *machine learning* (ML) para la estimación de PS ha ganado popularidad en los últimos años [4]. Las ventajas que ofrecen estas técnicas se pueden resumir en dos puntos: 1) No asumen suposiciones sobre las variables; y 2) Identifican y manejan relaciones complejas entre covariables, así como no linealidades, de forma implícita [4].

El uso de técnicas de ML para la estimación de PS destaca en ciencias de la salud con respecto a otras áreas (como las ciencias sociales, negocios, educación, o ingenierías) [4]. Entre los métodos más utilizados, se encuentran: Gradient Boosting Machines (GBM), Random Forests (RF), Classification/Regression trees (CART), y Neural Networks (**Figura 1**).

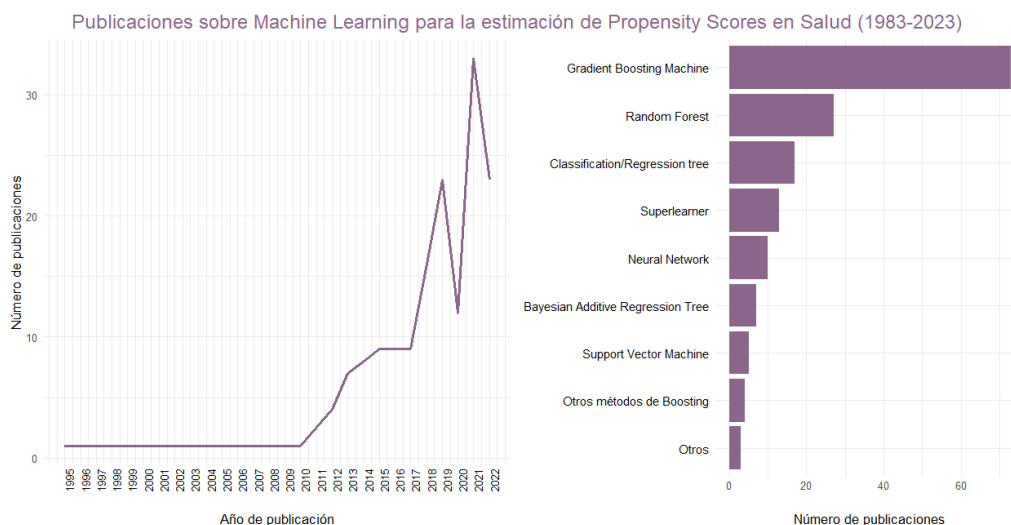


Figura 1: Publicaciones sobre técnicas de *machine learning* para la estimación de *propensity scores* en ciencias de la salud desde 1983 a 2023

Elaboración propia en R. Datos obtenidos del trabajo de Leite, W. *et al.*, 2024 [4].

Este trabajo contribuye a dicha discusión, mediante la comparación de las estimaciones para

PS realizadas con algunas técnicas de ML (RF, GBM y CART), y las obtenidas con el método tradicional de RL.

1.2. Objetivos del trabajo

Los objetivos principales del trabajo son los siguientes:

1. **Comparar distintos métodos de ML para la estimación de los PS con respecto a la RL.**
 - Implementar modelos de RL, CART, RF, y GBM para la estimación de los PS en R.
 - Evaluar el desempeño de los distintos métodos de estimación del PS en términos de balance de covariables, utilizando las métricas ASMD y ASAM.
2. **Indicar las ventajas y limitaciones de los métodos basados en ML con respecto a la RL en la estimación de PS.**
 - Interpretar el desempeño de los algoritmos de ML para estimar los PS, y la capacidad de las estimaciones para balancear covariables.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

Este trabajo considera la competencia de compromiso ético y global (CCEG) en sus tres dimensiones, desde el compromiso con los ODS (Objetivos de Desarrollo Sostenible, 2030):

1. Sostenibilidad

- ODS 9: Industria, Innovación e Infraestructura. En este trabajo se exponen técnicas estadísticas y computacionales innovadoras, destinadas a mejorar la eficiencia en análisis epidemiológicos.

2. Comportamiento ético y responsabilidad social

- ODS 16: Paz, Justicia e Instituciones Sólidas. Las conclusiones del estudio son aplicables a diversas instituciones, promoviendo un enfoque justo y equitativo en la toma de decisiones.
- ODS 8: Trabajo Decente y Crecimiento Económico. El trabajo propone mejoras en estudios epidemiológicos, cuyos resultados facilitan la implementación de medidas preventivas y decisiones políticas que optimizan la gestión de recursos y, en consecuencia, mejoran el estado económico de la población.

3. Diversidad y derechos humanos

- ODS 5: Igualdad de Género. La metodología evita sesgos de género y favorece resultados aplicables a contextos diversos, promoviendo la igualdad en el ámbito de la

salud pública.

- ODS 10: Reducción de las Desigualdades. La creación de un escenario generalista con datos simulados asegura que los resultados sean transferibles a cualquier población, contribuyendo a reducir desigualdades y promover la inclusión.

1.4. Enfoque y método seguido

Durante este trabajo se adopta la metodología *Scrum*. Dicha metodología consiste en la organización del trabajo mediante ciclos cortos (o *sprints*) con objetivos específicos. Concretamente, el proyecto se organiza en 5 ciclos, de 3 a 4 semanas de duración, en los cuales se persiguen los objetivos que se proponen en los enunciados de las Pruebas de Evaluación Continua (PEC) de la asignatura. Cada ciclo concluye con una entrega, que es evaluada y corregida por la tutora del trabajo. Las correcciones y/o cambios en el proyecto, se implementan en el producto final, que es fundamentalmente la memoria del trabajo.

En definitiva, esta metodología ágil para la gestión de proyectos, permite responder rápido a las necesidades del trabajo y persigue la mejora continua [6].

1.5. Planificación del trabajo

En la **Figura 2** se muestra la planificación de todas las tareas a lo largo del proyecto; así como los hitos, que se corresponden con cada una de las PECs.

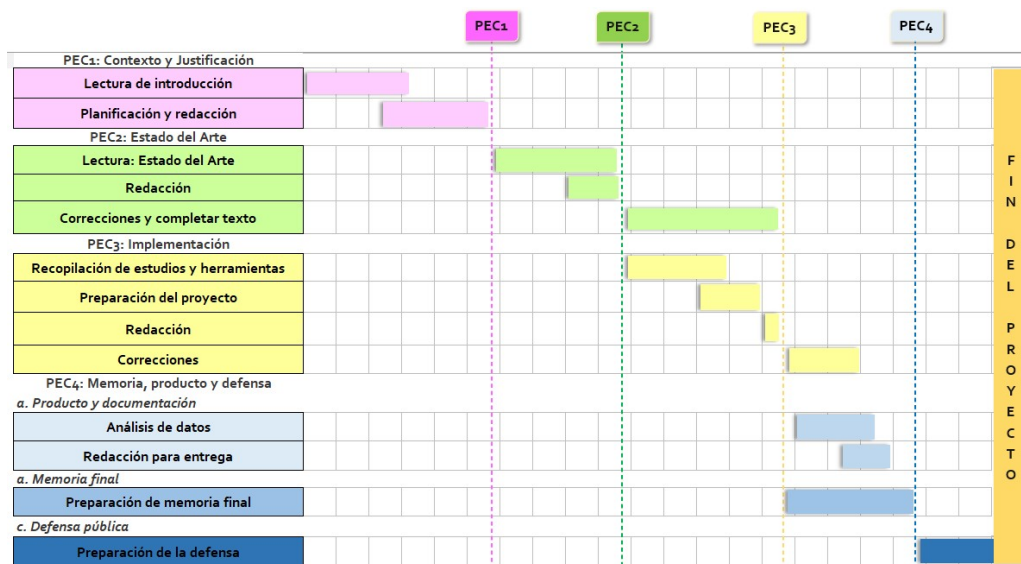


Figura 2: Cronograma de planificación del trabajo

1.6. Breve resumen de productos obtenidos

Como resultado de este trabajo, se obtienen los siguientes productos:

- **Memoria:** Contiene toda la información relativa al desarrollo del trabajo: describe los fundamentos teóricos sobre inferencia causal en epidemiología y la aplicación de los PS; la metodología seguida para realizar un estudio de comparación de la RL con varias técnicas de ML para la estimación de los PS, y la discusión y conclusiones sobre el tema del trabajo.
- **Informe de código en R:** Se adjunta junto con la memoria, e incluye los comandos utilizados para la generación del conjunto de datos simulados, para acompañar las explicaciones en la memoria sobre la implementación del estudio.
- **Repositorio en GitHub público:** Para permitir a cualquier usuario consultar la implementación completa del proyecto, reproducir los resultados, o establecer cambios para nuevos análisis adaptando la metodología propuesta.

Capítulo 2

Estado del Arte

2.1. Inferencia Causal en Epidemiología

La inferencia causal es el conjunto de procesos y métodos estadísticos que evalúan la existencia de una relación causa-efecto. Una *causa* se define como un evento o característica que precede a un *efecto*, y que un cambio en dicho evento o característica, modifica la probabilidad de ocurrencia o magnitud del efecto [2]. Por ejemplo, la infección por el virus de la hepatitis B se considera causa de la hepatitis crónica, ya que la probabilidad de desarrollar esta enfermedad es mayor tras haber contraído el virus.

Gran parte de los estudios epidemiológicos están centrados en esclarecer relaciones causales. Así, la causa suele corresponder a una exposición (*e.g.*: a un compuesto químico) o a un tratamiento (*e.g.*: como un fármaco, una intervención quirúrgica o la prestación de servicios sanitarios); mientras que el efecto se manifiesta típicamente como una enfermedad o un determinado estado de salud. Cuando la relación establecida por el tratamiento no es causal, o no conocemos si es causal, se utiliza el término *desenlace*.

2.1.1. Asociación y Causalidad

Comúnmente, el estudio de una relación causal se establece a partir de la observación de una *asociación*, que se da cuando la probabilidad de ocurrencia o valor de una variable varía con la de una o más variables. Sin embargo, es importante entender que la observación de una asociación no siempre implica causalidad (**Figura 3**).

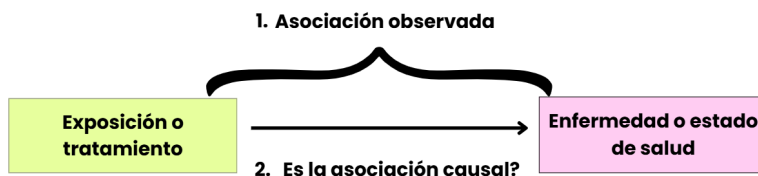


Figura 3: Asociación observada e inferencia causal

Adaptada de Celentano D. y Szlo M., 2019 [1]

Por ejemplo, supongamos que pacientes hipertensos tratados con un nuevo fármaco, presentan un promedio de niveles de colesterol más bajo que aquellos que reciben otro tratamiento. Podemos decir que observamos una asociación entre el nuevo fármaco y los niveles de colesterol. Sin embargo, dicha observación no implica que el nuevo fármaco sea la causa de la disminución de los niveles de colesterol en los pacientes. Para esclarecer dicha relación, se necesita plantear un problema de inferencia causal.

Es importante señalar que el análisis causal es diferente del análisis estadístico estándar. En general, el segundo estima los parámetros de las distribuciones de las poblaciones *tratada* y *no tratada* (e.g.: la media), y compara dichos parámetros (e.g.: mediante pruebas de hipótesis) para hallar relaciones asociativas, pero no causales. En cambio, el análisis causal estudia cómo cambia la distribución en estas poblaciones cuando se modifica el tratamiento, lo que permite inferir relaciones causales [7].

Supongamos que deseamos estudiar el efecto de un tratamiento A sobre un desenlace Y . En el análisis estadístico estándar, compararíamos los valores promedios de Y entre la población tratada ($A = 1$) y no tratada ($A = 0$). Dichos valores representarían esperanzas *condicionales*, porque se calculan bajo la condición de que el grupo esté tratado ($\mathbb{E}[Y \mid A = 1]$), o no lo esté ($\mathbb{E}[Y \mid A = 0]$). Sin embargo, en el análisis causal se comparan las esperanzas *marginales* (o *no condicionales*), que por lo contrario, no están definidas bajo la condición de pertenecer a ninguna de las dos poblaciones; sino que reflejan el valor promedio de Y en la población, cuando es tratada $\mathbb{E}[Y^{a=1}]$, y cuando no lo es $\mathbb{E}[Y^{a=0}]$. La **Figura 4** clarifica estos conceptos y representa la diferencia de la notación empleada para cada caso [8].

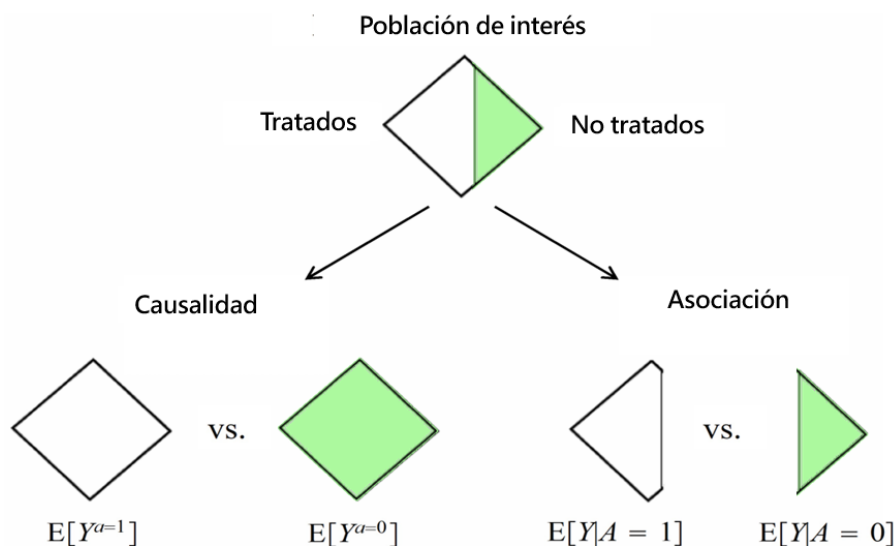


Figura 4: Distinción entre asociación y causalidad

El rombo completo representa la población, la cual se divide en dos grupos: tratados (blanco), y no tratados (verde). El análisis causal (izquierda) implica comparar toda la población si hubiera sido tratada, y si no lo hubiera sido. La relación asociativa, implica comparar el grupo de tratados y el de no tratados. Adaptada de Hernán M. A. & Robins J. M. (2020). [8]

Por tanto, la asociación se define por una diferencia en el desenlace observado en poblaciones diferentes; mientras que la causalidad se define por una variación del desenlace en la misma población. Así, las cuestiones que se formulan en cada caso son diferentes. Mientras estudiamos relaciones asociativas, respondemos a las cuestiones: “¿cuál es el efecto en los tratados?”, o “¿cuál es el efecto en los no tratados?”. En cambio, cuando el objetivo es realizar inferencia causal, las cuestiones se formulan como: “¿cuál hubiera sido el efecto si todos los individuos hubieran sido tratados?”; o bien: “¿cuál hubiera sido el efecto si ninguno de los individuos hubieran sido tratados?” [8].

2.1.2. Estimación del efecto causal

Un *estimador* es una función utilizada para estimar el valor de un parámetro desconocido a partir de datos observados. Dicho parámetro se conoce como *estimando*; y al valor resultante de aplicar el estimador, *estimación*. Así, cuando hablamos de estimación, asumimos la presencia de cierto grado de error [9].

Los estimandos del efecto causal se definen a partir del Modelo Causal de Rubin [10]. Dicho modelo supone un mundo *contrafactual*, en el que para cada individuo de la población i , existen dos desenlaces potenciales: el que hubiera ocurrido si el individuo hubiera sido tratado, $Y_i^{a=1}$; y el que se hubiera observado si no lo hubiese sido, $Y_i^{a=0}$ [10]. Por tanto, si $Y_i^{a=1} \neq Y_i^{a=0}$, asumimos que existe un efecto causal de A sobre Y en el individuo i .

A continuación, se definen los estimandos más habituales en el ámbito epidemiológico:

- **ATE** (*Average Treatment Effect*): Efecto promedio del tratamiento en la población total. Refleja la diferencia esperada en el desenlace si toda la población hubiera recibido el tratamiento, en comparación con el escenario en el que ningún individuo lo hubiera recibido.

$$ATE = \mathbb{E}[Y^{a=1} - Y^{a=0}]$$

- **ATT** (*Average Treatment Effect on the Treated*): Efecto promedio del tratamiento en los individuos que fueron tratados. Indica la diferencia en el desenlace observado si los individuos tratados no lo hubieran sido.

$$ATT = \mathbb{E}[Y^{a=1} - Y^{a=0} \mid A = 1]$$

- **ATU** (*Average Treatment Effect on the Untreated*): Efecto promedio que el tratamiento habría tenido en la población que no fue tratada. En este caso, se muestra la diferencia en el desenlace observado si los individuos que no han sido tratados lo hubieran sido.

$$ATU = \mathbb{E}[Y^{a=1} - Y^{a=0} \mid A = 0]$$

Sin embargo, en el mundo real sólo podemos observar uno de los dos *desenlaces potenciales*: si el individuo es tratado, se observa $Y^{a=1}$, pero no $Y^{a=0}$; y al contrario para los no tratados [10] [8]. Este fenómeno es conocido como el *problema fundamental de la inferencia causal*. La **Figura 5** resume estos conceptos.



Figura 5: Estimandos del efecto causal en el contexto del modelo contrafactual

En el mundo real, la población total se divide en dos grupos: tratados (blanco), y no tratados (verde). Los estimandos ATE, ATT y ATU asumen la existencia de ambos desenlaces potenciales ($Y^{a=1}$ y $Y^{a=0}$) en cada población. Las flechas negras señalan los desenlaces que se comparan con cada estimando. Sin embargo, sólo uno de ellos se observa en el mundo real; mientras que el otro es contrafactual. Elaboración propia basada en Hernán M. A. & Robins J. M. (2020). [8]

2.1.3. Supuestos de la inferencia causal

Dado que los estimandos causales están fundamentados en el modelo contrafactual, una estimación válida del efecto causal requiere aproximarse a este modelo teórico desde el mundo real, en el cual comparamos dos poblaciones diferentes. Esto se logra bajo el cumplimiento de tres supuestos clave [8]:

- **Ignorabilidad:** Asegura que los individuos tratados y no tratados sean comparables. Esto implica que el desenlace observado en uno de los grupos (*e.g.*: el tratado) sea el mismo que se hubiera observado en el otro grupo (*i.e.*: no tratado) si hubiera recibido el mismo tratamiento. A su vez, esto sólo es posible cuando no existen diferencias en las características de ambas poblaciones, que están definidas por el vector de covariables L .

Formalmente, la ignorabilidad establece que el desenlace potencial Y^a es independiente de la asignación al tratamiento A bajo la condición de L , lo que se expresa como:

$$Y^a \perp\!\!\!\perp A \mid L$$

- **Positividad:** Establece que todos los individuos tengan probabilidad positiva de recibir cualquiera de los tratamientos posibles. En otras palabras, garantiza que no exista ningún nivel en L para el cual la probabilidad de recibir cualquiera de los tratamientos sea nula:

$$1 > Pr[A = a \mid L] > 0$$

- **Consistencia:** Garantiza que el desenlace observado Y coincida con el desenlace bajo el tratamiento efectivamente recibido Y^a . Este supuesto se refiere a que el tratamiento esté bien definido, porque sólo así los desenlaces son *consistentes*:

$$Y = Y^a \quad \text{si } A = a.$$

El cumplimiento de estos supuestos es fundamental para realizar una inferencia causal válida. Los ensayos aleatorizados y los estudios observacionales utilizan distintas estrategias para lograr cumplirlos, resumidos en la **Tabla 1**.

Tabla 1: Condiciones para el cumplimiento de los supuestos de inferencia causal en ensayos aleatorizados y observacionales

Supuesto	Ensayos aleatorizados	Estudios observacionales
Ignorabilidad	Asignación aleatoria del tratamiento	Efectiva identificación de factores de confusión y aplicación de métodos estadísticos para el balance de covariables (e.g.: aplicación de <i>propensity scores</i>).
Positividad	Asignación aleatoria del tratamiento	Existencia de todos los niveles de tratamiento en todos los subgrupos de pacientes
Consistencia	Adherencia estricta al protocolo de intervención.	Definición clara y uniforme del tratamiento (e.g. mediante criterios de inclusión/exclusión efectivos).

En ensayos aleatorizados, los supuestos de ignorabilidad y positividad se cumplen por diseño, gracias a que la asignación aleatoria del tratamiento asegura una distribución de covariables equitativa entre los grupos tratado y no tratado [8]. Por ello, a menudo se hace referencia a los ensayos aleatorizados como el *gold standard* de los estudios epidemiológicos.

En cambio, en estudios observacionales, la asignación del tratamiento no es aleatoria, sino que está sujeta a las experiencias del mundo real: los individuos tratados presentan características diferentes a los no tratados. Esto se traduce en que la probabilidad de asignación de tratamiento es diferente en función de las características del individuo. Si además, dichas características influyen en el desenlace observado, se produce lo que se conoce como *sesgo de confusión*. Este es el principal motivo de la violación del supuesto de ignorabilidad en estudios observacionales. Además, también se relaciona con el incumplimiento de positividad, cuando la probabilidad de asignación del tratamiento es nula en pacientes con determinadas características [8].

2.1.4. Sesgo de confusión en estudios observacionales

Como se explicó anteriormente, las asociaciones no siempre son causales. Esto se debe principalmente a la existencia de *factores de confusión*: variables que actúan como causa común de

las variables asociadas (*e.g.*: tratamiento y el desenlace), haciendo que varíen en paralelo sin que exista una verdadera relación causa-efecto entre ellas.

Retomando el ejemplo anterior sobre el estudio del posible efecto causal de un nuevo fármaco sobre los niveles de colesterol: el sexo podría identificarse como factor de confusión si afecta tanto al tratamiento (*e.g.*: el fármaco se asigna con mayor probabilidad a mujeres); como al desenlace (*e.g.*: los niveles de colesterol son mayores en mujeres que en hombres en la población de estudio) (**Figura 6**).

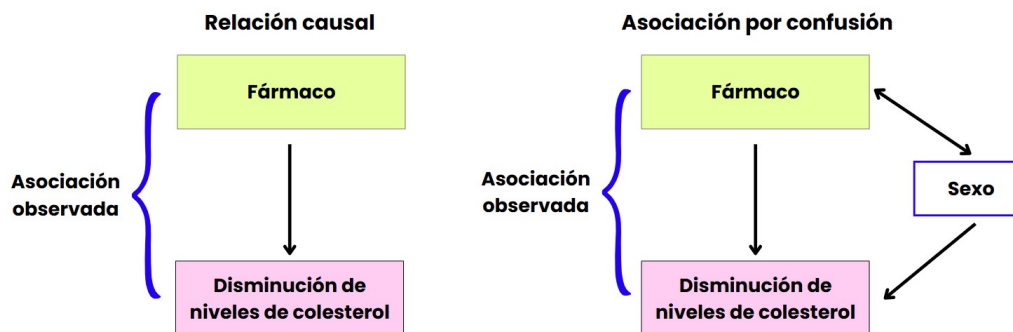


Figura 6: Ejemplo de relación causal y asociación por confusión.

Adaptada de Celentano D. y Szlo M., 2019 [1]

En estudios aleatorizados, los factores de confusión se controlan por diseño (tanto los identificados como los no identificados), porque la asignación del tratamiento es completamente aleatoria. En cambio, en los estudios observacionales, el control de la confusión representa uno de los principales desafíos metodológicos, dado que las cohortes tratada y no tratada suelen diferir en la distribución de estas variables. Esto se debe a que la probabilidad de asignación al tratamiento depende de las características del individuo (*i.e.*: no es aleatoria). Como resultado, el efecto causal se distorsiona; es decir, está sesgado. A dicho sesgo se lo conoce como *sesgo de confusión*. La corrección de la confusión en este caso consiste en la aplicación de métodos estadísticos (métodos de *ajuste de covariables*) (**Figura 7**).

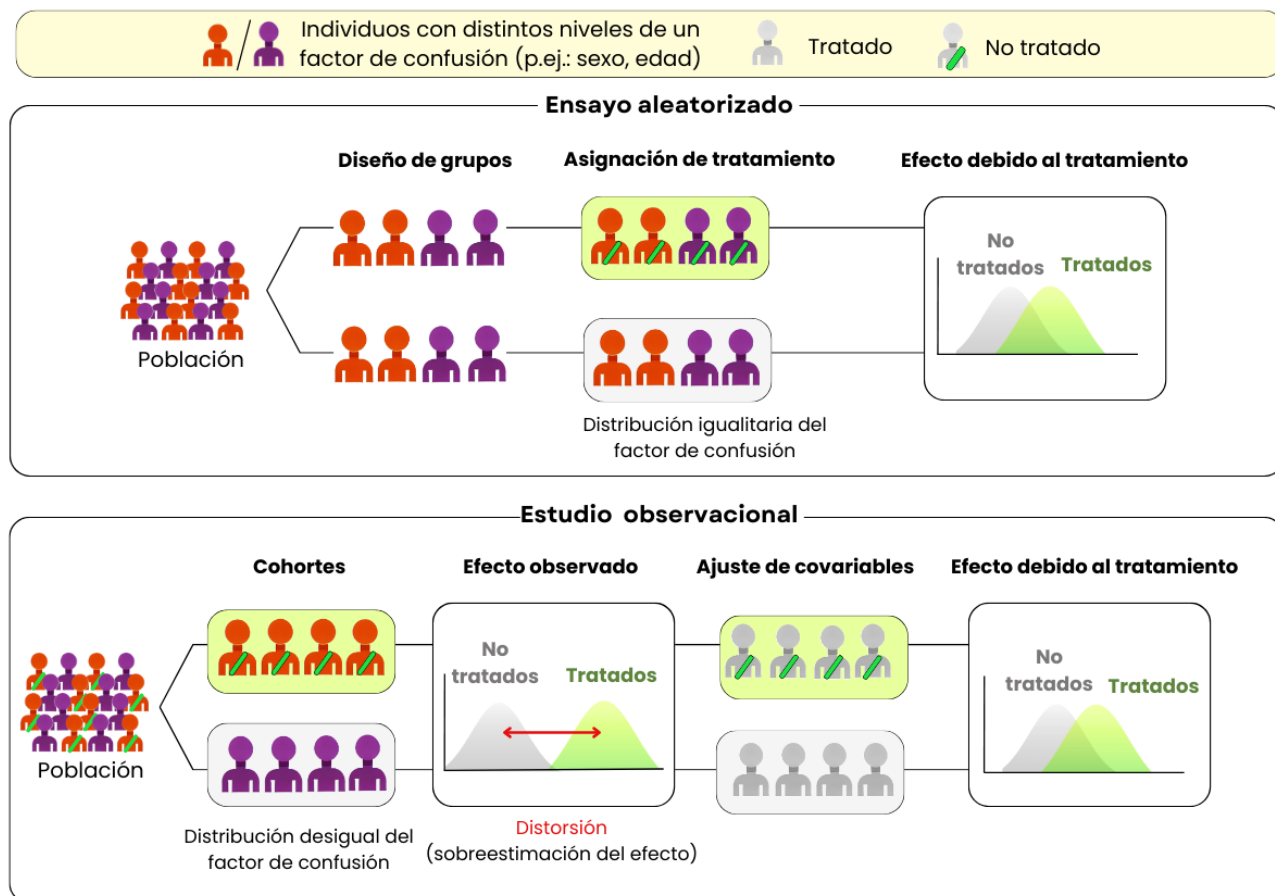


Figura 7: Control de la confusión en ensayos aleatorios y estudios observacionales.

Los colores (*naranja*, *morado*) representan individuos con distintos valores en un factor de confusión. En los estudios aleatorizados, se distribuyen equitativamente entre los grupos, lo que permite controlar la confusión. En cambio, en los estudios observacionales, los grupos difieren: predominan individuos *naranjas* entre los tratados y *morados* entre los no tratados, lo que introduce un sesgo en la estimación (en este caso, una sobreestimación). Los métodos de ajuste por covariables reducen este sesgo y permiten una estimación válida del efecto causal. Elaboración propia en *Canva*.

2.1.5. Conceptualización del problema causal

Antes de abordar cualquier problema de inferencia causal, es imprescindible conceptualizarlo: describir los posibles mecanismos por los que se establece la relación causal de interés, así como todos los factores que puedan influir en dicha relación. Si el problema no se conceptualiza adecuadamente, se pueden cometer errores al asumir las suposiciones anteriores. Por ejemplo, no identificar todos los posibles factores de confusión conllevaría a deficiencias en el diseño de cohortes o durante el análisis (no se tendría en cuenta el ajuste por dichas variables), lo que resultaría en la violación del supuesto de ignorabilidad, y por tanto, una estimación del efecto causal distorsionada.

Los grafos acíclicos dirigidos (Directed Acyclic Graphs, DAG) son una herramienta gráfica cada vez más utilizada en estudios epidemiológicos, ya que ofrece una representación intuitiva

y rigurosa de relaciones causales [12]. Cada elemento o nodo representa una variable, y cada flecha indica una posible relación causal. Los DAG permiten identificar sesgos que podrían distorsionar la relación causal. De esta forma, ayudan a plantear estrategias para su eliminación, ya sea en la fase de diseño del estudio o durante el análisis de datos. En la **Figura 8** se muestra un ejemplo de diagrama causal que resume los distintos tipos de variables que pueden afectar una relación causal.

Si bien existen múltiples reglas y conceptos específicos para la interpretación de grafos acíclicos dirigidos, en este texto nos limitaremos a explicar únicamente aquellos necesarios para la comprensión de los ejemplos y argumentos presentados, que están principalmente relacionados con el sesgo de confusión. Por ello, en el DAG de la **Figura 8**, se destaca el factor de confusión en un recuadro azul.

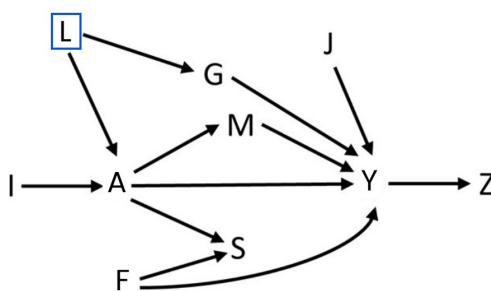


Figura 8: Ejemplo de grafo acíclico dirigido

Las rutas causales directas entre **A** (intervención) e **Y** (desenlace) se establecen mediante $A \rightarrow Y$ y $A \rightarrow M \rightarrow Y$. El resto de rutas entre **A** e **Y** no son causales porque existen elementos que bloquean el flujo causal. **L** es un factor de confusión; **G** puede bloquear el sesgo de confusión entre **A** e **Y**; **M** es un mediador; **S** es un colisionador; **I** es una variable instrumental (solo afecta a **A**); **J** es un modificador del efecto; **Z** es un descendiente de **Y**. Controlar o bloquear **M**, **L** o **S** puede introducir sesgos. Controlar o bloquear **L** elimina el sesgo de confusión. Imagen de Jean C. Digitale *et al.*, 2022 [12].

2.2. Propensity Score como método de ajuste

Como se ha explicado en apartados anteriores, en estudios observacionales es común que los individuos tratados y no tratados presenten características diferentes; es decir, distinta distribución de factores de confusión. Por ello, es fundamental aplicar métodos de ajuste por covariables que permitan equilibrar esas diferencias. Una de las herramientas más extendidas en los últimos años para este fin es el *propensity score* (PS).

2.2.1. Definición del Propensity Score

El PS se define como la probabilidad condicional de recibir alguno de los tratamientos (**A**) dado un vector de covariables **L** [3]:

$$e(L) = Pr[A \mid L]$$

El PS es un tipo concreto de *balancing score* (puntuación de balance), porque tiene la propiedad de que, condicionalmente al PS, la distribución de las covariables es la misma entre los grupos de tratamiento [3]. Esto se expresa como:

$$L \perp\!\!\!\perp A \mid e(L)$$

Dado que el PS es un parámetro desconocido en estudios observacionales, debe ser estimado a partir de los datos. Clásicamente, esta estimación se ha llevado a cabo mediante modelos de regresión logística, aprovechando su capacidad para modelar probabilidades condicionales.

Es importante tener en cuenta que la validez y eficacia del uso del PS dependen de una buena conceptualización del problema causal y del contexto del estudio (véase el Apartado 2.1.5). En particular, el principal supuesto del análisis con PS es que todos los factores de confusión estén correctamente identificados y medidos. Esto se debe a que la aplicación del PS está estrechamente ligada al supuesto de ignorabilidad de la inferencia causal [14].

2.2.2. Métodos de aplicación del Propensity Score

Existen varios métodos para aplicar el *propensity score* (PS) una vez estimado. Entre ellos, los más ampliamente utilizados en epidemiología son el emparejamiento (*matching*) y la ponderación por probabilidad inversa (*inverse probability weighting*, IPW) [15]. Algunos autores recomiendan evaluar distintos métodos y seleccionar aquel que proporcione un mejor balance de covariables y se adecúe a los objetivos específicos del estudio.

- **Emparejamiento:** Este método consiste en emparejar cada individuo tratado con uno (o varios) no tratados que presenten un valor de PS similar. El objetivo es formar pares o conjuntos comparables, de modo que las diferencias en el desenlace puedan atribuirse al tratamiento y no a los factores de confusión. El efecto del tratamiento se estima mediante la comparación de los resultados en los individuos emparejados [15].
- **Ponderación por probabilidad inversa (IPW):** Este enfoque asigna a cada individuo un peso inversamente proporcional a su probabilidad estimada de haber recibido el tratamiento que efectivamente recibió; es decir, $1/e(L)$ para los tratados y $1/(1 - e(L))$ para los no tratados. De este modo, se crea una población ponderada en la que la distribución de las covariables es independiente del tratamiento [15].

Capítulo 3

Materiales y Métodos

3.1. Workflow general

La **Figura 9** muestra un esquema general del trabajo realizado. En los siguientes apartados se detalla la metodología seguida en cada fase del análisis.

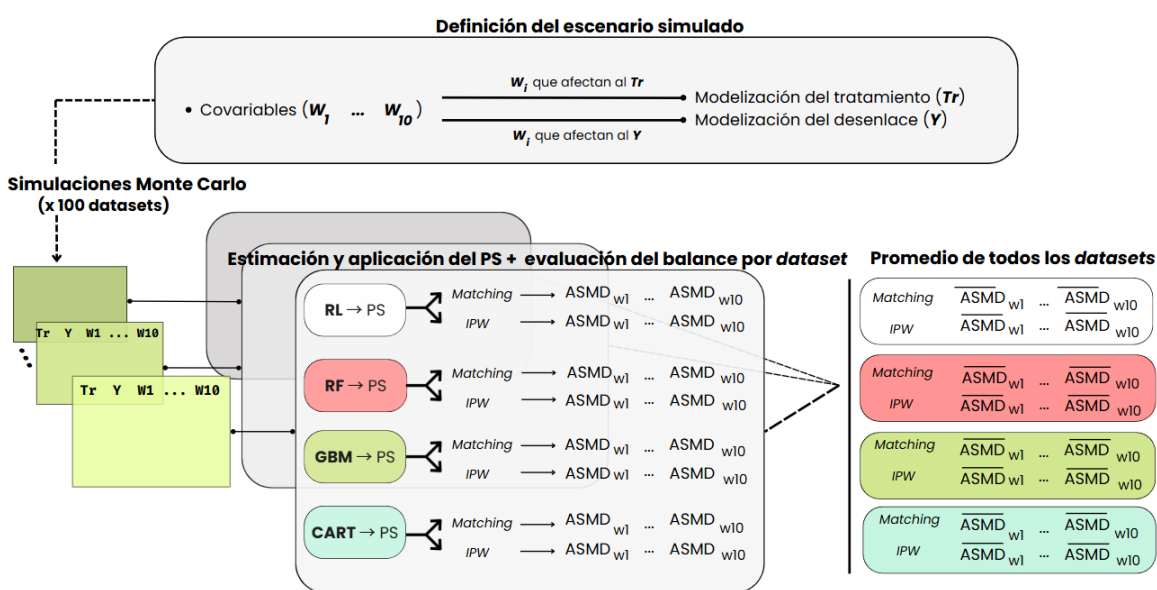


Figura 9: Workflow general del análisis

A partir de una serie de covariables W_i se define la asignación al tratamiento Tr y el desenlace Y . Se generan 100 *dataframes* (simulaciones Monte Carlo), que se analizan por separado. Para cada *dataframe* se estiman los PS por RL, RF, GBM y CART; y se aplican mediante *matching* y IPW. El balance logrado por cada estimación se evalúa según el ASMD de las covariables. El escenario se evalúa globalmente mediante el promedio de los ASMD obtenido en todos los *dataframes*. Elaboración propia en *Canva*.

Brevemente, se plantearon varios escenarios para los cuales se generaron 100 simulaciones Monte Carlo *datasets*. Para cada *dataset* de datos simulados, se realizó la estimación de los PS mediante varios métodos: RL, RF, GBM y CART. A su vez, los PS estimados se aplicaron a su vez me-

dian­te dos métodos diferentes: *matching* y IPW; y se evaluó el balance de covariables alcanzado tras el ajuste en cada caso mediante el ASMD. Los resultados correspondientes a cada escenario vienen dados como el promedio de los ASMD obtenidos para todos los *dataframes*.

Esta metodología basada en simulaciones Monte Carlo está inspirada por varios estudios con propósitos similares (evaluación de técnicas de ML para la estimación de PS) [17] [20] [21] [19] [18].

3.2. Software y paquetes

El análisis se desarrolló utilizando el *software* R (versión 4.4.2; R Core Team, 2024). Los principales paquetes empleados fueron: *MatchIt*, *cobalt*, *randomForest*, *gbm*, *rpart* y *ggplot2*.

El código completo se incluye en el Anexo. Además, el proyecto está disponible en un repositorio público de GitHub (<https://github.com/cjualia/TFM>), el cual ha sido configurado con el paquete *renv* que permite la instalación de todas las dependencias para reproducir el análisis.

3.3. Generación de datos simulados

Cada *dataset* está compuesto por una variable de tratamiento binaria (Tr), una variable de desenlace continua (Y), y un conjunto de 10 covariables (W_{1-10}). Las covariables W_{1-6} proceden de distribuciones estándar independientes, mientras que las covariables W_{7-10} derivan de las anteriores, representando transformaciones no lineales o interacciones (Tabla 2). Las variables Tr e Y se definen como función de W_i como se explica a continuación:

3.3.1. Modelización de la asignación al tratamiento

La modelización de la variable Tr se realiza en dos pasos:

1. Se establece un modelo de regresión logística para asignar a cada individuo la probabilidad de recibir el tratamiento dadas sus covariables W_i . Esto define el PS verdadero:

$$\text{logit } P(Tr = 1 \mid W_i) = \alpha_0 + \sum_i \alpha_i W_i$$

2. A cada individuo se le asigna un número aleatorio u procedente de una distribución continua uniforme entre 0 y 1. Si $PS > u$, entonces $Tr = 1$; si $PS > u$, $Tr = 0$.

3.3.2. Modelización del desenlace

El desenlace Y se define para cada individuo en función de W_i y Tr :

$$Y = \beta_0 + \sum_i \beta_i W_i + Tr_i \gamma$$

donde γ es el efecto del tratamiento. En este estudio, se ha fijado como $\gamma = -1$, siguiendo el diseño del estudio de Piracchio R. *et al.* (2018) [20]. Por ello, para todos los escenarios el efecto causal verdadero es el mismo para cualquier grupo de individuos, es decir: $ATE = ATT = ATC$.

3.3.3. Escenarios de simulación

Se plantearon tres escenarios: A , B y C , idénticos a los planteados por Piracchio R. *et al.* (2018) [20] Para cada uno de ellos, se generaron múltiples simulaciones de Monte Carlo, de 100 *datasets* con un tamaño de $n = 1000$.

Los coeficientes de α_i y β_i varían en cada escenario (Tabla 2), de modo que la modelización del tratamiento Tr y el desenlace Y varían en cada caso. El intercepto α_0 fue optimizado originalmente por Piracchio R. *et al.* (2018) [20] para que en todos los escenarios la prevalencia del tratamiento $Pr(Tr) \sim 0,5$.

Tabla 2: Covariables y coeficientes para la modelización del tratamiento y el desenlace en escenarios simulados.

		Escenario A		Escenario B		Escenario C	
		α	β	α	β	α	β
Intercepto		-3,5	-5,0	-3,5	-3,5	-1,0	-3,5
W_1	Normal (0,1)	1.0	0.5	1.0	0.4	0.8	0.4
W_2	Log-normal (0, 0.5)	1.0	0.5	1.0	0.03	0.06	0.03
W_3	Normal (0,10)	0.1	0.05	0.1	0.03	0.06	0.03
W_4	Binaria ($p = 0,5$)	2.0	1.0	2.0	0.75	1.5	0.75
W_5	Binaria ($p = 0,2$)	2.0	1.0	2.0	0.75	1.5	0.75
W_6	Catégorica ordinal	0.4	0.2	0.8	0.2	0.4	0.2
W_7	$\sin(W_1)$	0	0	0	0.4	0.8	0.4
W_8	W_2^2	0	0	0	0.02	0.04	0.02
W_9	$W_3 \cdot W_4$	0	0	0	0.04	0.08	0.04
W_{10}	$W_4 \cdot W_5$	0	0	1.0	0.5	1.0	0.5

α y β son los coeficientes utilizados en las fórmulas de modelización del tratamiento y desenlace, respectivamente. En verde, se marcan las filas correspondientes a las variables que actúan como confusoras en cada escenario. Escenarios originalmente diseñados por Piracchio R. *et al.* (2018) [20]

En resumen, los escenarios se diferencian en las siguientes propiedades:

- **Escenario A:** Representa el escenario base. El tratamiento Tr y el desenlace Y se definen mediante relaciones lineales de W_{1-6} .
- **Escenario B:** Se añade una interacción W_{10} como variable confusora, y otras interaccio-

nes y relaciones no lineales en la modelización de Y .

- **Escenario C:** Representa el escenario más complejo. Todas las covariables son confusoras W_{1-10} , lo que incluye términos no lineales e interacciones tanto en la modelización del tratamiento Tr como del desenlace Y .

3.4. Estimación de propensity scores

Se emplearon tres modelos de ML para la estimación de los PS: RF, GBM y CART. La implementación de cada modelo en R se realizó con los paquetes `randomForest`, `gbm` y `rpart`, respectivamente. Para la estimación por regresión logística se utilizó el paquete `glm`.

Uno de los aspectos metodológicos más importantes de este trabajo es que las fórmulas proporcionadas para la estimación de los PS únicamente incluyen las variables $W_1 - W_6$, que sólo aportan relaciones lineales. Así, el argumento `formula = Tr ~ w1 + w2 + w3 + w4 + w5 + w6` en cualquiera de las funciones que se utilizaron para la estimación (`randomForest`, `gbm`, `rpart` y `glm`).

En cambio, los términos más complejos: W_7 ($\sin(W_1)$) o W_8 (W_2^2); así como las interacciones: W_9 (W_3W_4) y W_{10} (W_4W_5), no se incorporaron explícitamente en los modelos de predicción, aunque participen en la modelización de la variable tratamiento Tr (que ocurre en los escenarios B y C) (Tabla 2).

Esta omisión es intencionada, ya que uno de los objetivos principales del estudio es evaluar en qué medida los modelos de ML son capaces de capturar estas relaciones no lineales de forma automática al predecir el tratamiento; una propiedad que no tienen los modelos de regresión logística.

3.5. Selección de hiperparámetros en modelos de machine learning

El desempeño de los métodos de ML puede variar en función de los hiperparámetros que se configuren. Por ello, uno de los aspectos más críticos a la hora de aplicar un algoritmo de ML es encontrar los hiperparámetros mediante los cuales se alcanzan mejores resultados en la predicción.

El proceso de búsqueda de hiperparámetros óptimos o *tuning* se adaptó de la metodología propuesta por Cannas & Arpino (2019) [19]. Dicha metodología se apoya en el hecho de que los hiperparámetros que conceden menor error de predicción en el modelo, no son necesariamente los que conceden mejor balance tras la aplicación de los PS estimados.

Brevemente, se procedió generar distintos modelos de ML, ajustados con distintos hiperparámetros. Los PS estimados en cada caso, se aplicaron mediante *matching* y *IPW*, hasta la evaluación del balance global mediante el ASAM. Finalmente, para el análisis, se seleccionó aquel modelo concedió menor ASAM en cada caso. Los pasos se resumen la Figura 10.

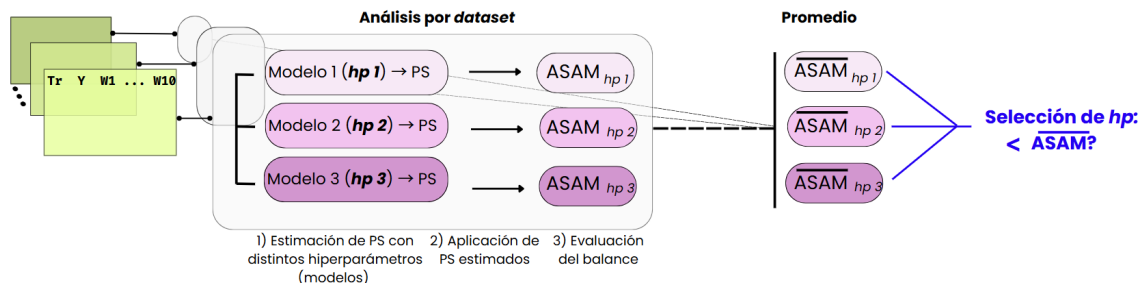


Figura 10: Proceso de ajuste de hiperparámetros basado en el ASAM.

De cada *dataframe* se realizan varias estimaciones de los PS, por diferentes modelos. Cada modelo se refiere a un determinado método de ML ajustado con unos hiperparámetros diferentes (hp). El balance de covariables obtenido después de aplicar los PS estimados en cada caso, se evalúa de forma global con el ASAM promedio de todos los *dataframes*. Elaboración propia en *Canva*.

Tabla 3: Métodos de machine learning utilizados, paquetes en R y configuración de hiperparámetros

Método	Función	Hiperparámetros
CART	rpart	cp = 0.01, 0.001
		minsplits = 20
		minbucket = round(minsplits/3)
		maxcompete = 4
		maxsurrogate = 5
		usesurrogate = 2
		surrogatestyle = 0
		maxdepth = 30
		xval = 10
RF	randomForest	mtry = 2, 3, 4, 5, 6
		ntree = 500
		nodesize = 1
GBM	gbm	shrinkage = 0.1, 0.01, 0.001
		interaction.depth = 4
		n.trees = 100
		n.minobsinnode = 10

Los hiperparámetros marcados en amarillo son aquellos que se testaron durante el proceso de *tuning*.

Si bien existen múltiples hiperparámetros para cada una de los métodos de ML empleados, en este estudio se ajustó únicamente uno por modelo: `cp` para `rpart`, `mtry` para `randomForest` y `shrinkage` para `gbm`. Los valores considerados para cada uno de estos hiperparámetros se detallan en la . El resto de hiperparámetros se mantuvieron en sus valores por defecto, con la única excepción de `interaction.depth` en `gbm`, que se fijó en 4, siguiendo las recomendaciones de Cannas y Arpino (2019) [19].

3.6. Aplicación del *propensity score*

Una vez estimados los PS, se aplicaron los métodos de *matching* y IPW como se explica a continuación:

- **Matching:** Se realizó mediante el método del vecino más cercano, con reemplazamiento y un *caliper* de 0.1. Para ello, se utilizó la función `MatchIt::matchit`, especificando los argumentos `method = "nearest"`, `replace = TRUE` y `caliper = 0.1`. El argumento `distance` permite introducir los PS estimados en cada caso.

A continuación, para realizar el balance, se utilizó la función `cobalt::bal.tab`, que toma como *input* el resultado de la función `matching` anterior.

- **IPW:** Se calcularon manualmente los pesos estabilizados mediante la siguiente fórmula:

$$\text{weight}_i = \begin{cases} \frac{P(Tr = 1)}{PS_i} & \text{si } Tr_i = 1 \\ \frac{P(Tr = 0)}{1 - PS_i} & \text{si } Tr_i = 0 \end{cases}$$

De forma similar, para aplicar el balance se empleó la función `cobalt::bal.tab`. En este caso, es necesario especificar el argumento `method = "weighting"` y proporcionar los pesos calculados anteriormente en el argumento `weights`.

3.7. Evaluación del balance y sesgo en la estimación del efecto causal

Para evaluar la calidad del ajuste logrado mediante el PS, se utilizan comúnmente dos métricas: la diferencia de medias estandarizada absoluta (*ASMD*) y su promedio global, la media de las diferencias de medias estandarizadas absolutas (*ASAM*).

El **ASMD** se calcula para cada covariable como:

$$\text{ASMD} = \left| \frac{\bar{W}_1 - \bar{W}_0}{\sqrt{\frac{s_1^2 + s_0^2}{2}}} \right|$$

donde \bar{W}_1 y \bar{W}_0 son las medias de la covariable en los grupos tratado y no tratado, respectivamente, y s_1^2 y s_0^2 sus varianzas.

El **ASAM** se define como el promedio de los ASMD calculados para todas las covariables incluidas en el modelo. Valores inferiores a 0.1 indican, en general, un buen balance.

Los valores de ASMD se pueden obtener directamente desde el balance realizado por la función `cobalt::bal.tab` en R.

Es importante indicar que el ASAM que figura en este análisis está calculado a partir de los ASMD de las variables confusoras; ya que se utiliza como métrica para evaluar el balance global de los confusores (por ejemplo, en el escenario *A*, el *ASAM* se calcula como la media de los *ASMD* de W_{1-6}).

Capítulo 4

Resultados

4.1. Balance inicial de los datos simulados

El balance inicial de los datos simulados se resume en la Tabla X. Como métrica global de evaluación del balance, se emplea el $\overline{\text{ASAM}}$, calculado como la media de los ASMD correspondientes a las covariables confusoras incluidas en cada escenario, ya que son estas las que deben quedar equilibradas tras el ajuste.

Tabla 4: Promedio y desviación estándar de ASAM y ASMD antes del ajuste.

	Escenario A	Escenario B	Escenario C
$\overline{\text{ASAM}}$	$0,324 \pm 0,020$	$0,293 \pm 0,016$	$0,289 \pm 0,020$
$\overline{\text{ASMD}} W_1$	$0,406 \pm 0,073$	$0,365 \pm 0,070$	$0,687 \pm 0,071$
$\overline{\text{ASMD}} W_2$	$0,208 \pm 0,056$	$0,181 \pm 0,061$	$0,045 \pm 0,064$
$\overline{\text{ASMD}} W_3$	$0,401 \pm 0,073$	$0,383 \pm 0,071$	$0,359 \pm 0,081$
$\overline{\text{ASMD}} W_4$	$0,468 \pm 0,066$	$0,435 \pm 0,072$	$0,267 \pm 0,071$
$\overline{\text{ASMD}} W_5$	$0,302 \pm 0,049$	$0,215 \pm 0,058$	$0,217 \pm 0,053$
$\overline{\text{ASMD}} W_6$	$0,159 \pm 0,061$	$0,278 \pm 0,062$	$0,171 \pm 0,064$
$\overline{\text{ASMD}} W_7$	$0,389 \pm 0,071$	$0,346 \pm 0,072$	$0,681 \pm 0,074$
$\overline{\text{ASMD}} W_8$	$0,186 \pm 0,046$	$0,163 \pm 0,058$	$0,036 \pm 0,069$
$\overline{\text{ASMD}} W_9$	$0,315 \pm 0,063$	$0,221 \pm 0,070$	$0,257 \pm 0,068$
$\overline{\text{ASMD}} W_{10}$	$0,281 \pm 0,041$	$0,197 \pm 0,034$	$0,167 \pm 0,040$

El promedio y la desviación estándar se calcula con los resultados obtenidos de los 100 *datasets* generados para cada escenario. Los valores en gris corresponden a covariables no confusoras, los cuales no se han incluido en el cálculo del ASAM.

En general, todas las variables confusoras se encuentran desequilibradas entre tratados y no tratados, a excepción de W_2 en el escenario C ($\overline{\text{ASMD}} < 0,1$).

4.2. Evaluación de la estimación del propensity score

En la **Figura 11** se presentan gráficos de dispersión que comparan los propensity scores verdaderos con los estimados por cada modelo. Esto permite evaluar la capacidad predictiva de cada método en un escenario concreto.

En general, los PS estimados mediante RL presentan mejor ajuste a los valores verdaderos que los obtenidos con los modelos ML. Entre estos últimos, RF ofrece las estimaciones más cercanas al PS verdadero, manteniendo la tendencia a través de todos los escenarios. En contraste, los modelos de GBM y CART tienden a generar PS concentrados en un rango muy estrecho, de modo que las estimaciones están considerablemente alejadas de los valores verdaderos.

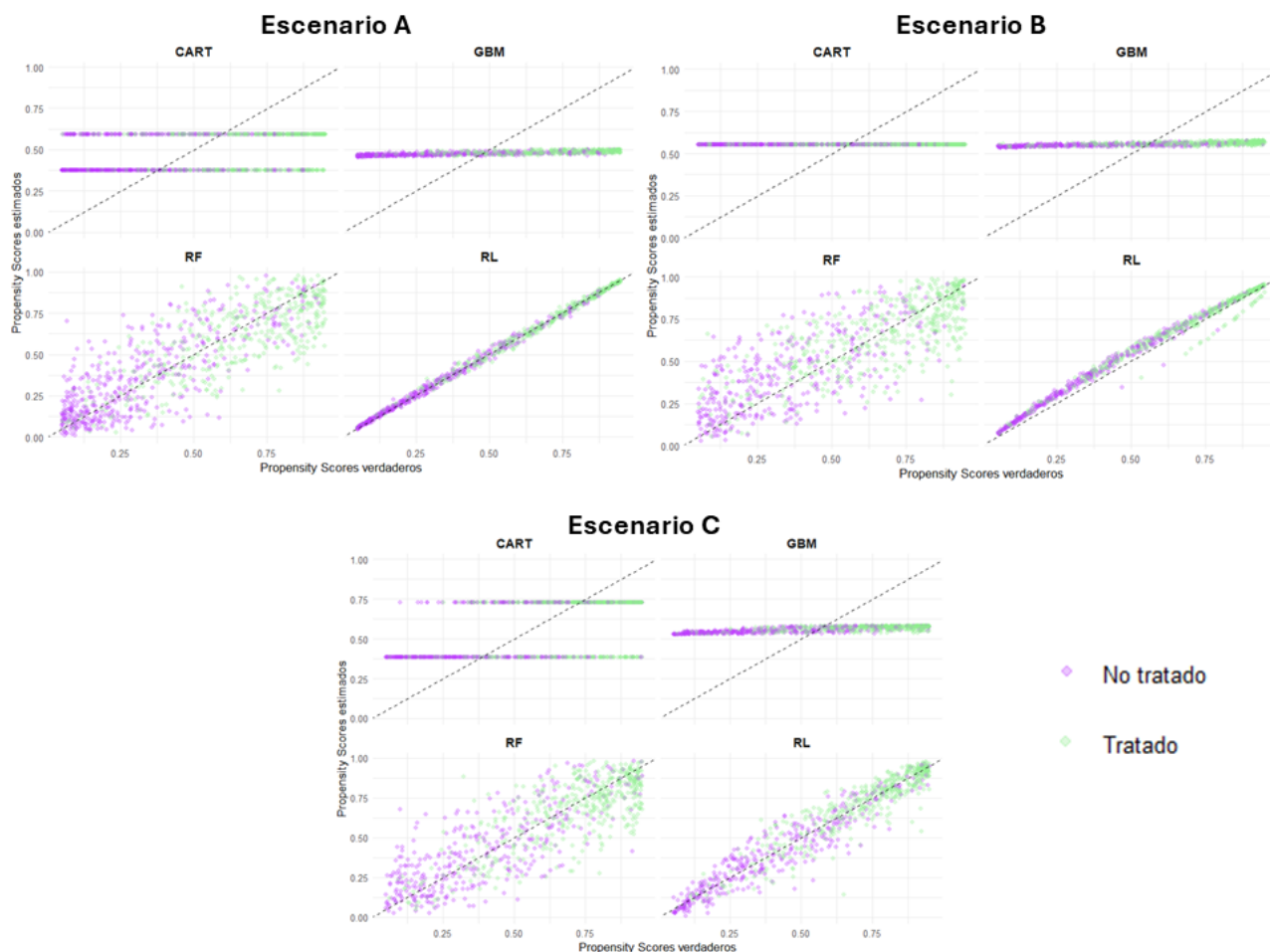


Figura 11: Gráfico de dispersión de PS verdaderos frente a PS estimados por distintos métodos.

Se representan los resultados de un único *dataset* de cada escenario ($n=1000$).

El ajuste de las estimaciones realizadas por RL es mejor en el escenario A porque representa el caso base, en el que la asignación del tratamiento (Tr) es función de las covariables W_{1-6} , que son especificadas en el modelo de predicción (como se explica en el apartado 3.4). En cambio, la dispersión aumenta en los escenarios B y C, en los cuales la asignación al tratamiento Tr depende además de las variables W_{7-10} , que aportan relaciones complejas e interacciones, las

cuales no se especificaron en la fórmula de predicción. Estos resultados reflejan las limitaciones de la regresión logística para capturar dichas relaciones cuando no se especifican en la fórmula de predicción del Tr .

Cabe señalar que los resultados mostrados corresponden a un único *dataset* de cada escenario; aunque dichas observaciones se aplican a todos los *datasets* generados.

4.3. Balance global después del ajuste

La evaluación del balance global se realizó mediante la comparación del *ASAM* promedio obtenido en los 100 *datasets* de cada escenario. Para el cálculo del *ASAM* sólo se tuvieron en cuenta los *ASMD* de las covariables confusoras correspondientes en cada escenario.

Exceptuando los resultados obtenidos con el modelo CART, el balance global alcanzado mediante el método IPW es notablemente superior al obtenido con *matching* en todos los escenarios, con valores de *ASAM* $< 0,1$, lo cual refleja una mejora considerable con respecto al desbalance inicial global (Tabla 4).

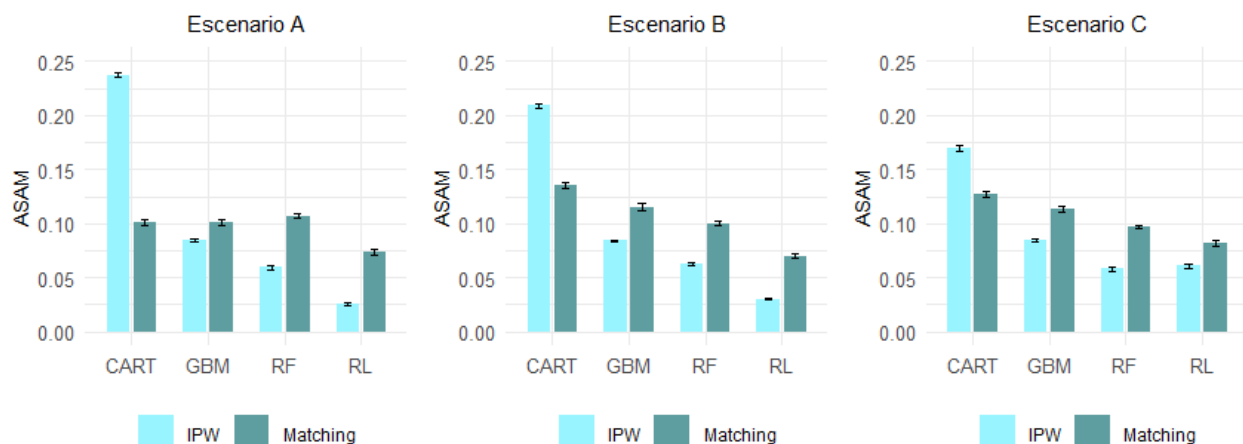


Figura 12: Balance global (*ASAM*) tras la aplicación de los *propensity scores* estimados. Promedio del *ASAM* calculado a partir de 100 *datasets*. Las barras de error representan el error estándar de la media.

4.4. Balance de covariables después del ajuste

En la **Figura 13** se muestran los resultados de los *ASMD* para cada variable, representando el promedio de los 100 *datasets* simulados para cada escenario.

En general, los PS estimados por RL conceden un buen balance en todas las variables confusoras de cada escenario; excepto para la covariable W_9 (W_3 W_4) en el escenario *C*. En cambio, los PS estimados por RF superan a la RL en este escenario; tanto en el caso de *matching* como en el IPW. Por otro lado, los resultados correspondientes al GBM también superan a los de la RL en el escenario *C* para W_9 .

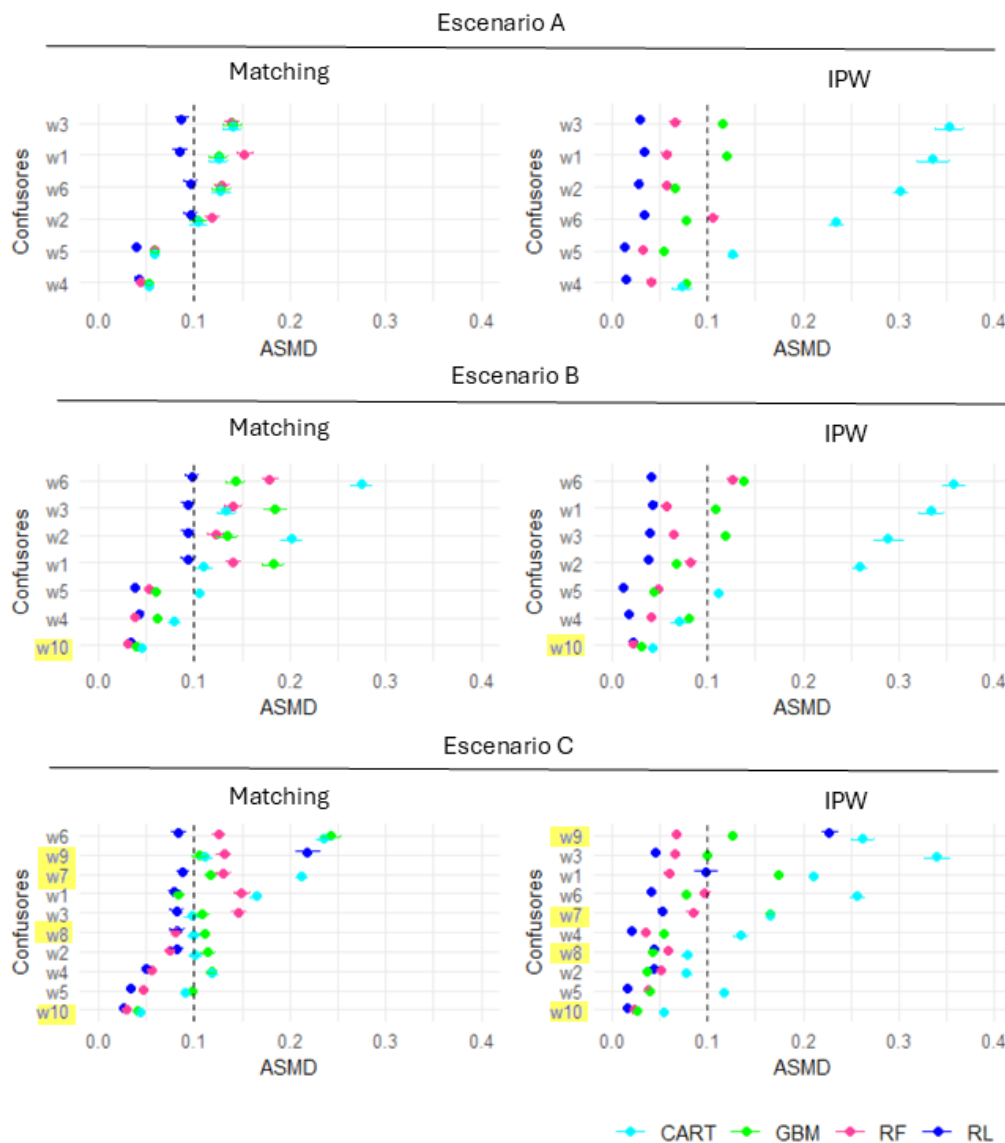


Figura 13: Loveplots del balance de covariables (ASMD) tras la aplicación de PS estimados mediante *matching* y IPW

Promedio del ASMD calculado a partir de 100 *datasets*. Las barras de error representan el error estándar de la media. Las covariables marcadas en amarillo son las que se ocultan del modelo de estimación (las que representan relaciones no lineales o interacciones)

Capítulo 5

Discusión

En este trabajo se han evaluado varios aspectos clave sobre la aplicación de PS estimados por diferentes métodos: RL, GBM, RF y CART. En primer lugar, se valoró el ajuste de los PS estimados a los PS verdaderos (Figura 11). En segundo lugar, se analizó el balance global en términos del ASAM tras la aplicación de los PS estimados mediante matching y IPW (Figura 12). Finalmente, se evaluó el ASMD de las variables confusoras (Figura 13).

Los tres escenarios simulados (A , B y C) utilizados en este trabajo son idénticos a los planteados por Piracchio R. *et al.* (2018) [20]. En su estudio, optimizan el balance de covariables mediante un modelo SuperLearner, el cual integra múltiples algoritmos de ML. Los mismos escenarios fueron también empleados por el estudio de Franklin J., *et al.* (2012) [21], en los que comparan la estimación de los PS realizada por modelos de RL correctamente especificados e incorrectamente especificados (con y sin inclusión de todas las covariables relevantes, respectivamente).

Sin embargo, no se han encontrado estudios que analicen dichos escenarios mediante técnicas de ML habituales, como GBM, RF y CART. Estos tres métodos son los más utilizados para la estimación de PS según la revisión sistemática realizada por Leite W. *et al.* (2024) [4], lo que justifica su elección en este trabajo. Además, el SuperLearner utilizado por Piracchio R. *et al.* (2018) [20] con estos datos, implica un elevado coste computacional; lo que limita su aplicación en la práctica.

Por otro lado, este trabajo adopta una metodología novedosa en comparación con otros estudios de simulación similares. En particular, se ha seguido el enfoque propuesto por [19] para la selección de los hiperparámetros en los modelos de machine learning. Esta aproximación parte de la idea de que la optimización de los hiperparámetros no debe orientarse a mejorar la capacidad predictiva del tratamiento, sino a maximizar el balance entre grupos, utilizando como criterio principal el ASAM, en lugar de las métricas en las que se basa habitualmente la evaluación de los modelos de ML, fundamentadas en el ajuste de predicción.

Entre los modelos de ML, el balance logrado mediante los PS estimados por RF fue superior en todos los escenarios (A , B y C). De todos los resultados obtenidos en este estudio, cabe destacar que, en el escenario C , se alcanzó un $ASMD < 0,1$ para todas las covariables confusoras al aplicar los PS estimados por RF con el método de IPW. En contraste, la RL no logró equilibrar adecuadamente la covariable W_9 , que representa una interacción entre W_4 y W_5 (Figura 13).

El resto de covariables no lineales (W_7 , W_8) y la interacción W_{10} , fueron ajustadas adecuadamente mediante RL en el escenario C , tanto al aplicar *matching* como IPW. Dichos resultados son coherentes con los obtenidos por Franklin J., *et al.* (2012) [21]. En su estudio, demuestran que los modelos mal especificados también pueden alcanzar niveles aceptables de balance, y atribuyen estos resultados a la contribución débil de estas covariables en la fórmula de asignación al tratamiento, ya que los coeficientes beta correspondientes a estas covariables son muy bajos (Tabla 2).

5.1. Limitaciones del estudio

Este estudio presenta una serie de limitaciones a tener en cuenta:

- Aunque se ha adoptado una metodología teóricamente más adecuada para el ajuste de hiperparámetros en los modelos de ML, basada en la minimización del ASAM; dicha estrategia es computacionalmente costosa si se desea aplicar al ajuste de varios hiperparámetros, ya que supone: 1) realizar la estimación con cada combinación de hiperparámetros, 2) aplicar el PS (mediante *matching*, IPW u otros métodos), y 3) hallar el ASAM tras el balance. Finalmente, seleccionar la combinación de hiperparámetros conveniente. La implementación realizada en este trabajo es una versión muy simplificada de esta estrategia; por ello sólo se han testado 2-3 valores para un hiperparámetro (`mtry`, `shrinkage`, `cp`) en cada método de ML (RF, GBM y CART, respectivamente). Recientemente han surgido nuevas técnicas de ajuste de hiperparámetros altamente eficaces, como ASHA [22], capaces de evaluar múltiples combinaciones de hiperparámetros simultáneamente; sin embargo, hasta la fecha, no existen implementaciones de este algoritmo en R.
- Los datos simulados en este estudio están alejados de un estudio observacional real en varios aspectos: en los escenarios simulados, se conocen todas las variables confusoras; las relaciones entre covariables, tratamiento y desenlace están completamente definidas y controladas; y no se manejan casos con datos faltantes. No obstante, trabajar en condiciones controladas nos permite evaluar mejor aspectos concretos de cada método.

Bibliografía

- [1] Celentano, D. D., & Szklo, M. (2019). *Gordis Epidemiology* (6^a ed.). Elsevier Inc. ISBN: 978-0-323-55229-5
- [2] Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern Epidemiology*. Lippincott Williams & Wilkins. ISBN: 978-0-7817-5564-1
- [3] Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41–55. <https://doi.org/10.1093/biomet/70.1.41>
- [4] Leite, W., Zhang, H., Collier, Z., Chawla, K., Kong, L., Lee, Y., Quan, J., & Soyoye, O. (2024). Machine learning for propensity score estimation: A systematic review and reporting guidelines (Pre-print Version 1.0 – 10/7/2024). *Preprint*. https://osf.io/a8prf/?view_only=c667f82b4c374bccb68fefab803c8b95
- [5] Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: Neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *Journal of Clinical Epidemiology*, 63(8), 826–833. <https://doi.org/10.1016/j.jclinepi.2009.11.020>
- [6] Asana. (s.f.). ¿Qué es Scrum? *Asana*. <https://asana.com/es/resources/what-is-scrum>
- [7] Pearl, P. (2010). An introduction to causal inference. *The International Journal of Biostatistics*, 6(2), Article 7. <https://doi.org/10.2202/1557-4679.1203>
- [8] Hernán M. A., Robins J. M. (2020). *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC. <https://miguelhernan.org/whatifbook>
- [9] Porta, M. (2008). *A dictionary of epidemiology* (5^a ed.). Oxford University Press/International Epidemiological Association. ISBN: 978-0-19-531450-2
- [10] Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5), 688–701. <https://doi.org/10.1037/h0037350>
- [11] Mori, M. (2024). Practical data considerations for the modern epidemiology student. *European Journal of Epidemiology*, 39, 349–352. <https://doi.org/10.1007/s10654-024-01138-0>

- [12] Digitale, J. C., Martin, J. N., & Glymour, M. M. (2022). Tutorial on directed acyclic graphs. *Journal of Clinical Epidemiology*, 142, 264–267. <https://doi.org/10.1016/j.jclinepi.2021.08.001>
- [13] Hernán, M. A. (2020). Misunderstandings between experimentalists and observationalists about causal inference. *European Journal of Epidemiology*, 35, 539–546. <https://doi.org/10.1007/s10654-020-00625-4>
- [14] Austin, P. C. (2011). An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. *Multivariate Behavioral Research*, 46(3), 399–424. <https://doi.org/10.1080/00273171.2011.568786>
- [15] Garrido, M. M., Kelley, A. S., Paris, J., Roza, K., Meier, D. E., Morrison, R. S., & Aldridge, M. D. (2014). Methods for Constructing and Assessing Propensity Scores. *Health Services Research*, 49(5), 1701–1720. <https://doi.org/10.1111/1475-6773.12182>
- [16] Lalani, N., Jimenez, R. B., & Yeap, B. (2020). Understanding Propensity Score Analyses. *International Journal of Radiation Oncology, Biology, Physics*, 107(3), 404–407. <https://doi.org/10.1016/j.ijrobp.2020.02.638>
- [17] Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., & Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and Drug Safety*, 17(6), 546–555. <https://doi.org/10.1002/pds.1555>
- [18] Lee, B. K., Lessler, J., & Stuart, E. A. (2010). Improving propensity score weighting using machine learning. *Statistics in Medicine*, 29(3), 337–346. <https://doi.org/10.1002/sim.3782>
- [19] Cannas, M., & Arpino, B. (2019). A comparison of machine learning algorithms and covariate balance measures for propensity score matching and weighting. *Biometrical journal. Biometrische Zeitschrift*, 61(4), 1049–1072. <https://doi.org/10.1002/bimj.201800132>
- [20] Pirracchio, R., & Carone, M. (2018). The Balance Super Learner: A robust adaptation of the Super Learner to improve estimation of the average treatment effect in the treated based on propensity score matching. *Statistical Methods in Medical Research*, 27(8), 2504–2518. <https://doi.org/10.1177/0962280216682055>
- [21] Franklin, J. M., Rassen, J. A., Ackermann, D., Bartels, D. B., & Schneeweiss, S. (2014). Metrics for covariate balance in cohort studies of causal effects. *Statistics in Medicine*, 33(10), 1685–1699. <https://doi.org/10.1002/sim.6058>
- [22] Asha, S. E., Mohd Noor, N., Phung, D., & Tighe, S. (2021). Propensity score methods in health technology assessment: a review of guidance documents. *International Journal of Technology Assessment in Health Care*, 37(e20), 1–7. <https://doi.org/10.1017/S0266462321000196>
- [23] Zhao, Q.-Y., Luo, J.-C., Su, Y., Zhang, Y.-J., Tu, G.-W., & Luo, Z. (2021). Propensity score matching with R: conventional methods and new features. *Annals of Translational Medicine*, 9(9), 812. <https://doi.org/10.21037/atm-20-3998>

- [24] Ferri-García, R., & Rueda, M. M. (2020). Propensity score adjustment using machine learning classification algorithms to control selection bias in online surveys. *PLOS ONE*, 15(4), e0231500. <https://doi.org/10.1371/journal.pone.0231500>
- [25] Chen, S., Zhan, Y., Wang, Z., Jin, L., & Zhang, L. (2024). Machine learning for propensity score matching and weighting in causal inference: a review and new insights. *NPJ Digital Medicine*, 7(1), 49. <https://doi.org/10.1038/s41746-024-00976-y>
- [26] Moccia, C., Moirano, G., Popovic, M., et al. (2023). Machine learning in causal inference for epidemiology. *European Journal of Epidemiology*, 39, 1097–1108. <https://doi.org/10.1007/s10654-024-01173-x>
- [27] Simoneau, G., Pellegrini, F., Debray, T. P. A., et al. (2022). Recommendations for the use of propensity score methods in multiple sclerosis research. *Multiple Sclerosis Journal*, 28(9), 1467–1480. <https://doi.org/10.1177/13524585221085733>
- [28] Brookhart, M. A., Schneeweiss, S., Rothman, K. J., Glynn, R. J., Avorn, J., & Stürmer, T. (2006). Variable selection for propensity score models. *American Journal of Epidemiology*, 163(12), 1149–1156. <https://doi.org/10.1093/aje/kwj149>
- [29] Garrido, M. M., Dowd, B., Hebert, P. L., & Maciejewski, M. L. (2016). Understanding Treatment Effect Terminology in Pain and Symptom Management Research. *Journal of Pain and Symptom Management*, 52(3), 446–452. <https://doi.org/10.1016/j.jpainsymman.2016.01.016>

Anexo

Cristina Juárez Alía

2025-06-04

Simulación de datos

La función `rep_simulation` permite generar las 100 simulaciones Monte Carlo de los escenarios que se especifican (para la elaboración de la memoria sólo se documentan los resultados de los escenarios A, B y C; pero originalmente se crearon también los escenarios D y E).

El diseño de la función se puede consultar en el apartado *Funciones*.

```
# `all_scenarios` es un dataframe que guarda todos los dataframes  
# utilizados  
# para el análisis. Cada columna representa un escenario distinto (A-E).  
  
all_scenarios <- as.data.frame(sapply(c("A", "B", "C", "D", "E"),  
  function(x){rep_simulation(100, 1000, x)}))  
  
# Datos guardados en la carpeta `data`.  
save(all_scenarios, file = "data/all_scenarios.rda")
```

Análisis

Balance de covariables inicial

La función `before` devuelve una lista en la que cada elemento guarda el vector de ASMD's para covariable (en total, 10) de cada dataframe (en total, 100). Convertimos la lista en una matriz 10 x 100 (dataframes x covariables) con `do.call`.

El diseño de la función `before` se puede consultar en el apartado *Funciones*.

```
ASMD_A <- do.call(cbind, before("A"))  
ASMD_B <- do.call(cbind, before("B"))  
ASMD_C <- do.call(cbind, before("C"))
```

Hallamos el promedio y la desviación de ASMD para cada covariable:

```
mean_ASMD_A <- apply(ASMD_A, 1, mean) ; sd_ASMD_A <- apply(ASMD_A, 1, sd)  
mean_ASMD_B <- apply(ASMD_B, 1, mean) ; sd_ASMD_B <- apply(ASMD_B, 1, sd)  
mean_ASMD_C <- apply(ASMD_C, 1, mean) ; sd_ASMD_C <- apply(ASMD_C, 1, sd)
```


De forma similar calculamos el ASAM para cada dataframe (media de ASMD's). Después, el promedio y la desviación entre todos los dataframes.

NOTA: Sólo incluimos en el ASAM las variables confusoras, porque es para las cuales nos interesa evaluar el balance. Las variables confusoras son distintas para cada escenario:

- A: w1-w6
- B: w1-w6 y w10
- C: w1-w10

```
ASAM_A <- apply(ASMD_A[c(1:6), ], 2, mean) # Confusores: w1-6
mean_ASAM_A <- mean(ASAM_A) ; sd_ASAM_A <- sd(ASAM_A)
```

```
ASAM_B <- apply(ASMD_B[c(1:6,10), ], 2, mean) # Confusores: w1-w6 y w10
mean_ASAM_B <- mean(ASAM_B) ; sd_ASAM_B <- sd(ASAM_B)
```

```
ASAM_C <- apply(ASMD_C, 2, mean) # Confusores: w1-w10
mean_ASAM_C <- mean(ASAM_C) ; sd_ASAM_C <- sd(ASAM_C)
```

Guardamos esta información en una lista:

```
inicial <- list("A" = list("asam_mean" = mean_ASAM_A,
                           "asam_sd" = sd_ASAM_A,
                           "asmd_mean" = mean_ASMD_A,
                           "asmd_sd" = sd_ASMD_A),
               "B" = list("asam_mean" = mean_ASAM_B,
                           "asam_sd" = sd_ASAM_B,
                           "asmd_mean" = mean_ASMD_B,
                           "asmd_sd" = sd_ASMD_B),
               "C" = list("asam_mean" = mean_ASAM_C,
                           "asam_sd" = sd_ASAM_C,
                           "asmd_mean" = mean_ASMD_C,
                           "asmd_sd" = sd_ASMD_C),
               "D" = list("asam_mean" = mean_ASAM_D,
                           "asam_sd" = sd_ASAM_D,
                           "asmd_mean" = mean_ASMD_D,
                           "asmd_sd" = sd_ASMD_D),
               "E" = list("asam_mean" = mean_ASAM_E,
                           "asam_sd" = sd_ASAM_E,
                           "asmd_mean" = mean_ASMD_E,
                           "asmd_sd" = sd_ASMD_E))
```

Que se almacena en la carpeta results del proyecto:

```
save(inicial, file = "results/inicial.rda")
```

Estimación de oropensity scores mediante regresión logística

NOTA: Ocultamos a los modelos de estimación las variables complejas. Esto es importante para cumplir con uno de los objetivos principales de este estudio, que es comprobar si técnicas de ML son capaces de mejorar la estimación de los PS gracias a su capacidad para detectar relaciones complejas/no lineales e interacciones, que a menudo se omiten en los modelos tradicionales (regresión logística).

Utilizamos la función `RL_matching` para realizar el ajuste por *matching* y `RL_weight` para IPW. Ambas son funciones customizadas cuyo diseño puede consultarse en el apartado *Funciones*.

+ Aplicando Matching

```
RL_m_A <- RL_matching("A")
RL_m_B <- RL_matching("B")
RL_m_C <- RL_matching("C")
```

+ Aplicando IPW

```
RL_w_A <- RL_weight("A")
RL_w_B <- RL_weight("B")
RL_w_C <- RL_weight("C")
```

Preparamos los resultados de cada escenario, calculando el promedio y desviación de los ASMD's obtenidos en todos los dataframes, así como del ASAM. El procedimiento es similar al anterior para el balance de covariables inicial:

+ Resultados del Matching

Creación de matrices

```
ASMD_A_RL_m <- do.call(cbind, RL_m_A)
ASMD_B_RL_m <- do.call(cbind, RL_m_B)
ASMD_C_RL_m <- do.call(cbind, RL_m_C)
```

Promedio y desviación de ASMD para cada covariable:

```
mean_ASMD_A_RL_m <- apply(ASMD_A_RL_m, 1, mean) ; sd_ASMD_A_RL_m <-
apply(ASMD_A_RL_m, 1, sd)
mean_ASMD_B_RL_m <- apply(ASMD_B_RL_m, 1, mean) ; sd_ASMD_B_RL_m <-
apply(ASMD_B_RL_m, 1, sd)
mean_ASMD_C_RL_m <- apply(ASMD_C_RL_m, 1, mean) ; sd_ASMD_C_RL_m <-
apply(ASMD_C_RL_m, 1, sd)
```

Promedio y desviación del ASAM:

```
ASAM_A_RL_m <- apply(ASMD_A_RL_m[c(1:6)], , 2, mean)
mean_ASAM_A_RL_m <- mean(ASAM_A_RL_m) ; sd_ASAM_A_RL_m <- sd(ASAM_A_RL_m)

ASAM_B_RL_m <- apply(ASMD_B_RL_m[c(1:6,10)], , 2, mean)
```

```
mean_ASAM_B_RL_m <- mean(ASAM_B_RL_m) ; sd_ASAM_B_RL_m <- sd(ASAM_B_RL_m)
```

```
ASAM_C_RL_m <- apply(ASMD_C_RL_m, 2, mean)
```

```
mean_ASAM_C_RL_m <- mean(ASAM_C_RL_m) ; sd_ASAM_C_RL_m <- sd(ASAM_C_RL_m)
```

+ Resultados del IPW

Creación de matrices

```
ASMD_A_RL_w <- do.call(cbind, RL_w_A)
```

```
ASMD_B_RL_w <- do.call(cbind, RL_w_B)
```

```
ASMD_C_RL_w <- do.call(cbind, RL_w_C)
```

Promedio y desviación de ASMD para cada covariable:

```
mean_ASMD_A_RL_w <- apply(ASMD_A_RL_w, 1, mean) ; sd_ASMD_A_RL_w <-  
apply(ASMD_A_RL_w, 1, sd)
```

```
mean_ASMD_B_RL_w <- apply(ASMD_B_RL_w, 1, mean) ; sd_ASMD_B_RL_w <-  
apply(ASMD_B_RL_w, 1, sd)
```

```
mean_ASMD_C_RL_w <- apply(ASMD_C_RL_w, 1, mean) ; sd_ASMD_C_RL_w <-  
apply(ASMD_C_RL_w, 1, sd)
```

Promedio y desviación del ASAM:

```
ASAM_A_RL_w <- apply(ASMD_A_RL_w[c(1:6), ], 2, mean)
```

```
mean_ASAM_A_RL_w <- mean(ASAM_A_RL_w) ; sd_ASAM_A_RL_w <- sd(ASAM_A_RL_w)
```

```
ASAM_B_RL_w <- apply(ASMD_B_RL_w[c(1:6,10), ], 2, mean)
```

```
mean_ASAM_B_RL_w <- mean(ASAM_B_RL_w) ; sd_ASAM_B_RL_w <- sd(ASAM_B_RL_w)
```

```
ASAM_C_RL_w <- apply(ASMD_C_RL_w, 2, mean)
```

```
mean_ASAM_C_RL_w <- mean(ASAM_C_RL_w) ; sd_ASAM_C_RL_w <- sd(ASAM_C_RL_w)
```

Guardamos esta información en listas:

Resultados del matching

```
RL_m <- list("A" = list("asam_mean" = mean_ASAM_A_RL_m,  
                        "asam_sd" = sd_ASAM_A_RL_m,  
                        "asmd_mean" = mean_ASMD_A_RL_m,  
                        "asmd_sd" = sd_ASMD_A_RL_m),  
            "B" = list("asam_mean" = mean_ASAM_B_RL_m,  
                        "asam_sd" = sd_ASAM_B_RL_m,  
                        "asmd_mean" = mean_ASMD_B_RL_m,  
                        "asmd_sd" = sd_ASMD_B_RL_m),  
            "C" = list("asam_mean" = mean_ASAM_C_RL_m,  
                        "asam_sd" = sd_ASAM_C_RL_m,  
                        "asmd_mean" = mean_ASMD_C_RL_m,  
                        "asmd_sd" = sd_ASMD_C_RL_m))
```

Resultados del IPW

```
RL_w <- list("A" = list("asam_mean" = mean_ASAM_A_RL_w,
                        "asam_sd" = sd_ASAM_A_RL_w,
                        "asmd_mean" = mean_ASMD_A_RL_w,
                        "asmd_sd" = sd_ASMD_A_RL_w),
            "B" = list("asam_mean" = mean_ASAM_B_RL_w,
                        "asam_sd" = sd_ASAM_B_RL_w,
                        "asmd_mean" = mean_ASMD_B_RL_w,
                        "asmd_sd" = sd_ASMD_B_RL_w),
            "C" = list("asam_mean" = mean_ASAM_C_RL_w,
                        "asam_sd" = sd_ASAM_C_RL_w,
                        "asmd_mean" = mean_ASMD_C_RL_w,
                        "asmd_sd" = sd_ASMD_C_RL_w))
```

Guardamos los resultados en un objeto dentro de la carpeta results del proyecto:

```
save(RL_m, RL_w, file = "results/RL_results.rda")
```

Tuning de modelos de machine learning

La función `hp_testing_m` realiza los siguientes pasos:

- - 1) Estimación de PS según el método de ML especificado (“randomForest”, “gbm”, “rpart”) ajustando por un hiperparámetro especificado (hp). En el caso de RF, es `mtry`; para GBM, `shrinkage`, y para CART, `cp`.
- - 2) Matching según los PS estimados.
- - 3) Devuelve los ASMD’s

EL diseño de la función se puede encontrar en el apartado *Funciones*.

Aplicamos dicha función con `lapply()` para testar varios hiperparámetros. El resultado es una lista en la que cada elemento guarda los resultados obtenidos con un hiperparámetro distinto. Los resultados, son dataframes 10 x 100 con los datos de ASMD para cada covariable (10 en total), en cada dataframe (100 en total).

Random Forest (RF)

```
### Para Los Random Forest, testaremos `mtry` = 2, 3, 4, 5, 6
```

```
### Resultados para Matching
```

```
hp_test_A_m <- lapply(c(2:6), function(x){ hp_testing_m(x, "A",
"randomForest")})
hp_test_B_m <- lapply(c(2:6), function(x){ hp_testing_m(x, "B",
"randomForest")})
hp_test_C_m <- lapply(c(2:6), function(x){ hp_testing_m(x, "C",
```

```
"randomForest"))})
```

Resultados para IPW

```
hp_test_A_w <- lapply(c(2:6), function(x){ hp_testing_w(x, "A",  
"randomForest"))})  
hp_test_B_w <- lapply(c(2:6), function(x){ hp_testing_w(x, "B",  
"randomForest"))})  
hp_test_C_w <- lapply(c(2:6), function(x){ hp_testing_w(x, "C",  
"randomForest"))})
```

Selección del hiperparámetro óptimo

#La función `opt_hp` está diseñada para indicar qué hiperparámetro de los testados

anteriormente da menor ASAM.

Resultados para Matching

```
opt_hp(hp_test_A_m, "A") # mtry = 2 para escenario A / matching  
opt_hp(hp_test_B_m, "B") # mtry = 2 para escenario B / matching  
opt_hp(hp_test_C_m, "C") # mtry = 2 para escenario C / matching
```

Resultados para IPW

```
opt_hp(hp_test_A_w, "A") # mtry = 3 para escenario A / IPW  
opt_hp(hp_test_B_w, "B") # mtry = 3 para escenario B / IPW  
opt_hp(hp_test_C_w, "C") # mtry = 2 para escenario C / IPW
```

Generalized Boosted Models (GBM)

Para Los Gradient Boosting Machines, testaremos `shrinkage` = 0.1, 0.01 y 0.001.

Resultados para Matching

```
gbm_hp_test_A_m <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_m(x, "A", "gbm")})  
gbm_hp_test_B_m <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_m(x, "B", "gbm")})  
gbm_hp_test_C_m <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_m(x, "C", "gbm")})
```

Resultados para IPW

```
gbm_hp_test_A_w <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_w(x, "A", "gbm")})  
gbm_hp_test_B_w <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_w(x, "B", "gbm")})
```

```
gbm_hp_test_C_w <- lapply(c(0.1, 0.01, 0.001), function(x){  
hp_testing_w(x, "C", "gbm")})
```

Selección del hiperparámetro óptimo

Resultados para Matching

```
opt_hp(gbm_hp_test_A_m, "A") # shrinkage = 0.1 para escenario A /  
matching  
opt_hp(gbm_hp_test_B_m, "B") # shrinkage = 0.1 para escenario B /  
matching  
opt_hp(gbm_hp_test_C_m, "C") # shrinkage = 0.001 para escenario C /  
matching
```

Resultados para IPW

```
opt_hp(gbm_hp_test_A_w, "A") # shrinkage = 0.1 para escenario A / IPW  
opt_hp(gbm_hp_test_B_w, "B") # shrinkage = 0.1 para escenario B / IPW  
opt_hp(gbm_hp_test_C_w, "C") # shrinkage = 0.1 para escenario C / IPW
```

Classification and Regression Trees (CART)

Para Los Classification and Regression Trees, testaremos `cp` = 0.01, 0.001

NOTA: Para cp = 0.1 no era posible obtener matches (en el proceso de matching)

utilizando un caliper = 0.1.

Resultados para Matching

```
cart_hp_test_A_m <- lapply(c(0.01, 0.001), function(x){ hp_testing_m(x,  
"A", "rpart")})  
cart_hp_test_B_m <- lapply(c(0.01, 0.001), function(x){ hp_testing_m(x,  
"B", "rpart")})  
cart_hp_test_C_m <- lapply(c(0.01, 0.001), function(x){ hp_testing_m(x,  
"C", "rpart")})
```

Resultados para IPW

```
cart_hp_test_A_w <- lapply(c(0.01, 0.001), function(x){ hp_testing_w(x,  
"A", "rpart")})  
cart_hp_test_B_w <- lapply(c(0.01, 0.001), function(x){ hp_testing_w(x,  
"B", "rpart")})  
cart_hp_test_C_w <- lapply(c(0.01, 0.001), function(x){ hp_testing_w(x,  
"C", "rpart")})
```

Selección del hiperparámetro óptimo

Resultados para Matching

```
opt_hp(cart_hp_test_A_m, "A") # cp = 0.01 para escenario A / matching
opt_hp(cart_hp_test_B_m, "B") # cp = 0.001 para escenario B / matching
opt_hp(cart_hp_test_C_m, "C") # cp = 0.001 para escenario C / matching
```

Resultados para IPW

```
opt_hp(cart_hp_test_A_w, "A") # cp = 0.001 para escenario A / IPW
opt_hp(cart_hp_test_B_w, "B") # cp = 0.001 para escenario B / IPW
opt_hp(cart_hp_test_C_w, "C") # cp = 0.001 para escenario C / IPW
```

Guardamos los resultados:

```
save(rf_hp_test_A_m, rf_hp_test_B_m, rf_hp_test_C_m, rf_hp_test_D_m,
     rf_hp_test_E_m,
     rf_hp_test_A_w, rf_hp_test_B_w, rf_hp_test_C_w, rf_hp_test_D_w,
     rf_hp_test_E_w,
     gbm_hp_test_A_m, gbm_hp_test_B_m, gbm_hp_test_C_m, gbm_hp_test_D_m,
     gbm_hp_test_E_m,
     gbm_hp_test_A_w, gbm_hp_test_B_w, gbm_hp_test_C_w, gbm_hp_test_D_w,
     gbm_hp_test_E_w,
     cart_hp_test_A_m, cart_hp_test_B_m, cart_hp_test_C_m,
     cart_hp_test_D_m, cart_hp_test_E_m,
     cart_hp_test_A_w, cart_hp_test_B_w, cart_hp_test_C_w,
     cart_hp_test_D_w, cart_hp_test_E_w,
     file = "results/hp_optimization.rda")
```

NOTA: Se pueden consultar los valores de ASAM promedio obtenidos para cada hp con el siguiente código; en este caso, para el escenario A utilizando IPW:

```
for (df in cart_hp_test_E_w){
  asam <- mean(apply(df, 2, mean))
  print(asam)
}
# Se imprimen Los ASAM en el orden en que se han especificado los hp.
# Es decir, si hemos testado para mtry= 2,3,4,5 y 6; el primer valor que se imprime
# corresponde a mtry=2.
```

Finalmente extraemos los datos con los que trabajaremos; es decir, los correspondientes a las estimaciones de los PS realizadas por el modelo ajustado con el hiperparámetro que ofrece menor ASAM; según los resultados anteriores.

```
rf_A_m <- rf_hp_test_A_m[[1]] ; rf_A_w <- rf_hp_test_A_w[[2]]
rf_B_m <- rf_hp_test_B_m[[1]] ; rf_B_w <- rf_hp_test_B_w[[2]]
rf_C_m <- rf_hp_test_C_m[[1]] ; rf_C_w <- rf_hp_test_C_w[[1]]

gbm_A_m <- gbm_hp_test_A_m[[1]] ; gbm_A_w <- gbm_hp_test_A_w[[1]]
```

```

gbm_B_m <- gbm_hp_test_B_m[[1]] ; gbm_B_w <- gbm_hp_test_B_w[[1]]
gbm_C_m <- gbm_hp_test_C_m[[3]] ; gbm_C_w <- gbm_hp_test_C_w[[1]]

cart_A_m <- gbm_hp_test_A_m[[1]] ; cart_A_w <- cart_hp_test_A_w[[2]]
cart_B_m <- gbm_hp_test_B_m[[2]] ; cart_B_w <- cart_hp_test_B_w[[2]]
cart_C_m <- gbm_hp_test_C_m[[2]] ; cart_C_w <- cart_hp_test_C_w[[2]]

# Reunimos los resultados por escenario y por método de aplicación:

## Escenario A / Matching                                ## Escenario A / IPW

A_m <- list(rf_A_m, gbm_A_m, cart_A_m) ; A_w <- list(rf_A_w, gbm_A_w,
cart_A_w)
names(A_m) <- c("rf", "gbm", "cart") ; names(A_w) <- c("rf", "gbm",
"cart")

## Escenario B / Matching                                ## Escenario B / IPW

B_m <- list(rf_B_m, gbm_B_m, cart_B_m) ; B_w <- list(rf_B_w, gbm_B_w,
cart_B_w)
names(B_m) <- c("rf", "gbm", "cart") ; names(B_w) <- c("rf", "gbm",
"cart")

## Escenario C / Matching                                ## Escenario C / IPW

C_m <- list(rf_C_m, gbm_C_m, cart_C_m) ; C_w <- list(rf_C_w, gbm_C_w,
cart_C_w)
names(C_m) <- c("rf", "gbm", "cart") ; names(C_w) <- c("rf", "gbm",
"cart")

```

balance_results simplemente resume los resultados para un determinado escenario. Devuelve una lista con el promedio y la desviación del ASAM, así como de los ASMD's.

Aplicamos la función `balance` a cada lista:

```

## Escenario A / Matching
results_A_m <- lapply(A_m, function(x){balance(x, "A")})

## Escenario A / IPW
results_A_w <- lapply(A_w, function(x){balance(x, "A")})

## Escenario B / Matching
results_B_m <- lapply(B_m, function(x){balance(x, "B")})

## Escenario B / IPW

```



```

results_C_w[["logr"]] <- list("asam_mean" = RL_w$C$asam_mean,
                             "asam_sd" = RL_w$C$asam_sd,
                             "asmd_mean" = RL_w$C$asmd_mean,
                             "asmd_sd" = RL_w$C$asmd_sd)

results_D_w[["logr"]] <- list("asam_mean" = RL_w$D$asam_mean,
                             "asam_sd" = RL_w$D$asam_sd,
                             "asmd_mean" = RL_w$D$asmd_mean,
                             "asmd_sd" = RL_w$D$asmd_sd)

results_E_w[["logr"]] <- list("asam_mean" = RL_w$E$asam_mean,
                             "asam_sd" = RL_w$E$asam_sd,
                             "asmd_mean" = RL_w$E$asmd_mean,
                             "asmd_sd" = RL_w$E$asmd_sd)

```

Guardamos los resultados:

```

save(results_A_m, results_A_w, results_B_m, results_B_w, results_C_m,
      results_C_w,
      results_D_m, results_D_w, results_E_m, results_E_w, file =
      "results/all_results.rda")

```

Distribución de propensity scores estimados

Seleccionamos un dataframe de cada escenario. Estos datos se utilizan para explorar la distribución de los PS estimados, así como su ajuste a los PS verdaderos, como se muestra en la memoria.

Para recuperar los PS estimados de estos dataframes empleamos funciones más sencillas `simple_RL`, `simple_RF`, `simple_GBM` y `simple_CART` cuyo diseño se encuentra en el apartado *Funciones*

Realizamos la estimación de los PS por cada técnica, para cada uno de estos

dataframes.

Para las técnicas de ML, utilizamos como hiperparámetros los seleccionados anteriormente:

Regresión Logística

```

eps_RL_A <- simple_RL(dfA) ; eps_RL_B <- simple_RL(dfB) ; eps_RL_C <-
simple_RL(dfC)

```

Random Forest

```

eps_RF_A <- simple_RF(dfA, 3) ; eps_RF_B <- simple_RF(dfB, 3) ; eps_RF_C
<- simple_RF(dfC, 3)

```

```
# GBM
```

```
eps_GBM_A <- simple_GBM(dfA, 0.001) ; eps_GBM_B <- simple_GBM(dfB, 0.001)  
; eps_GBM_C <- simple_GBM(dfC, 0.001)
```

```
# CART
```

```
eps_CART_A <- simple_CART(dfA, 0.1) ; eps_CART_B <- simple_CART(dfB, 0.1)  
; eps_CART_C <- simple_CART(dfC, 0.1)
```

```
simple_estimations <- list("A" = list("logreg" = data.frame("Tr" =  
dfA$Tr, "tps" = dfA$tps, "eps" = eps_RL_A),  
                           "rf" = data.frame("Tr" =  
dfA$Tr, "tps" = dfA$tps, "eps" = eps_RF_A),  
                           "gbm" = data.frame("Tr" =  
dfA$Tr, "tps" = dfA$tps, "eps" = eps_GBM_A),  
                           "cart" = data.frame("Tr" =  
dfA$Tr, "tps" = dfA$tps, "eps" = eps_CART_A)),  
                          "B" = list("logreg" = data.frame("Tr" =  
dfB$Tr, "tps" = dfB$tps, "eps" = eps_RL_B),  
                           "rf" = data.frame("Tr" =  
dfB$Tr, "tps" = dfB$tps, "eps" = eps_RF_B),  
                           "gbm" = data.frame("Tr" =  
dfB$Tr, "tps" = dfB$tps, "eps" = eps_GBM_B),  
                           "cart" = data.frame("Tr" =  
dfB$Tr, "tps" = dfB$tps, "eps" = eps_CART_B)),  
                          "C" = list("logreg" = data.frame("Tr" =  
dfC$Tr, "tps" = dfC$tps, "eps" = eps_RL_C),  
                           "rf" = data.frame("Tr" =  
dfC$Tr, "tps" = dfC$tps, "eps" = eps_RF_C),  
                           "gbm" = data.frame("Tr" =  
dfC$Tr, "tps" = dfC$tps, "eps" = eps_GBM_C),  
                           "cart" = data.frame("Tr" =  
dfC$Tr, "tps" = dfC$tps, "eps" = eps_CART_C)))
```

Guardamos estos resultados para realizar los gráficos:

```
save(simple_estimations, file = "results/simple_estimations.rda")
```

Funciones

A continuación se muestra el diseño de todas las funciones diseñadas para este estudio, y que han sido empleadas durante el análisis:

```
# Generación de datos simulados
```

```
simulation <- function(size, scenario){
```

```

# covariates
w1 <- rnorm(size,0,1)
w2 <- rlnorm(size,0,0.5)
w3 <- rnorm(size,0,10)
w4 <- rbinom(size,1,0.5)
w5 <- rbinom(size,1,0.2)
w6 <- sample(1:5, size, replace=T, prob = c(0.5,0.3,0.1,0.05,0.05))

# complex terms and interactions
w7 <- sin(w1)
w8 <- w2^2
w9 <- w3*w4
w10 <- w4*w5

# scenarios

if (scenario == "A"){
  tps <- (1 + exp(-(-3.5 + 1*w1 + 1*w2 + 0.1*w3 + 2*w4 + 2*w5 +
0.4*w6)))^ -1
  Y0 <- -5 + 0.5*w1 + 0.5*w2 + 0.05*w3 + 1*w4 + 1*w5 + 0.2*w6
}
if (scenario == "B"){
  tps <- (1 + exp(-(-3.5 + 1*w1 + 1*w2 + 0.1*w3 + 2*w4 + 2*w5 + 0.8*w6
+ 1*w10)))^ -1
  Y0 <- -3.5 + 0.4*w1 + 0.03*w2 + 0.03*w3 + 0.75*w4 + 0.75*w5 + 0.2*w6
+ 0.4*w7
+ 0.02*w8 + 0.04*w9 + 0.5*w10
}
if (scenario == "C"){
  tps <- (1 + exp(-(-1 + 0.8*w1 + 0.06*w2 + 0.06*w3 + 1.5*w4 + 1.5*w5 +
0.4*w6
+ 0.8*w7 + 0.04*w8 + 0.08*w9 + 1*w10)))^ -1
  Y0 <- -3.5 + 0.4*w1 + 0.03*w2 + 0.03*w3 + 0.75*w4 * 0.75*w5 + 0.2*w6
+ 0.4*w7
+ 0.02*w8 + 0.04*w9 + 0.5*w10
}
if (scenario == "D"){
  tps <- (1 + exp(-(-1.3 + 0.8*w1 + 0.06*w2 + 0.06*w3 + 1.5*w4 + 1.5*w5
+ 0.4*w6
+ 0.8*w7 + 0.04*w8 + 0.08*w9 + 1*w10)))^ -1
  Y0 <- -3.7 + 0.4*w1 + 0.03*w3 + 0.75*w4 * 0.75*w5 + 0.2*w6 + 0.4*w7
+ 0.04*w9 + 0.5*w10
}
if (scenario == "E"){
  tps <- (1 + exp(-(-1.3 + 0.8*w1 + 0.06*w2 + 0.06*w3 + 1.5*w4 + 1.5*w5
+ 0.4*w6
+ 0.8*w7 + 0.04*w8 + 0.08*w9 + 1*w10)))^ -1
  Y0 <- -3.7 + 0.4*w1 + 0.03*w3 + 0.75*w4 * 0.75*w5 + 0.2*w6 + 1*w7

```

```

    + 2*w9 + 1.5*w10
  }

  # Counterfactual Y1 corresponds to the outcome when treated,
  # therefore, the effect of the treatment is added (constant):
  effect <- -1
  Y1 <- Y0 + effect

  # Treatment assignment:
  unif1 <- runif(size,0,1) # Uniform continuous distribution (0,1)
  Tr <- ifelse(tps > unif1, 1, 0) # When PS > number, Tr=1 (treated);
  else, Tr=0 (non-treated)

  # continuous outcome Y
  Y <- Tr*Y1 + (1-Tr)*Y0

  sim <- as.data.frame(cbind(w1, w2, w3 ,w4, w5, w6, w7, w8, w9, w10, Tr,
Y, tps))

  sim <- sim[sim$tps >= 0.05 & sim$tps <= 0.95, ]

  return(sim)
}

rep_simulation <- function(nrep, size, scenario){
  all_sim <- list()
  for (r in seq(nrep)){
    df <- simulation(size, scenario)
    all_sim[[r]] <- df
  }
  return(all_sim)
}

# Balance de covariables inicial
before <- function(scenario){

  dfs <- all_scenarios[[scenario]]

  results <- list()

  for (i in 1:length(dfs)){ # Recorremos la lista de dataframes

    m.out0 <- matchit(Tr ~ w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 +
w10,
                      data = dfs[[i]],
                      method = NULL) # No aplicamos ningún método)

    summary <- summary(m.out0)

```

```

    asmd <- summary$sum.all[-1,3] # Extraemos Los ASMD's (ignoramos la
línea `distance`)

    results[[i]] <- asmd # Guardamos el resultado para cada dataframe
  }

  return(results)
}

# Regresión Logística: Estimación de Propensity Scores + Matching
RL_matching <- function(scenario){

  dfs <- all_scenarios[[scenario]]

  results <- list()

  for (i in 1:length(dfs)){ # Recorremos la lista de dataframes

    # Se realiza en 2 pasos:

    # 1) `matchit` estima los PS, y realiza el matching

    m.out <- matchit(Tr ~ w1 + w2 + w3 + w4 + w5 + w6, # Omitimos las
variables complejas
                     data = dfs[[i]],
                     method = "nearest", # Matching
                     replace = TRUE,
                     caliper = 0.1,
                     distance = "glm") # Método de estimación: regresión
Logística

    # 2) `bal.tab` calcula los ASMD's
    ## En este caso especificamos todas las variables porque nos interesa
    ## conocer el ASMD's de todas
    balance <- bal.tab(m.out,
                      add1 = ~ w7 + w8 + w9 + w10,
                      data = dfs[[i]])

    # Extraemos Los ASMD'S:
    asmds <- abs(balance$Balance$Diff.Adj)[2:11] # Omitimos registro de
`distance`
    results[[i]] <- asmds
  }
  return(results)
}

# Regresión Logística: Estimación de Propensity Scores + IPW

```

```

RL_weight <- function(scenario) {

  dfs <- all_scenarios[[scenario]]
  results <- list()

  for (i in seq_along(dfs)) {

    # Se realiza en 3 pasos:

    # 1) `matchit` estima los PS

    m.out <- matchit(Tr ~ w1 + w2 + w3 + w4 + w5 + w6, # Omitimos las
variables complejas
                    data = dfs[[i]],
                    distance = "glm") # Método de estimación: regresión
Logística

    eps <- m.out$distance

    # 2) Cálculo de weights

    p_Tr <- mean(dfs[[i]]$Tr == 1)
    p_noTr <- 1 - p_Tr
    weights <- ifelse(dfs[[i]]$Tr == 1,
                     p_Tr / eps,
                     p_noTr / (1 - eps))

    # 2. `bal.tab` calcula el balance
    balance <- bal.tab(
      Tr ~ w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10,
      data = dfs[[i]],
      weights = weights,
      method = "weighting")

    asmds <- abs(balance$Balance$Diff.Adj)
    results[[i]] <- asmds
  }

  return(results)
}

# Búsqueda de hiperparámetros óptimos (basado en el ASAM mínimo)

# `ML_eps` devuelve los PS estimados
ML_eps <- function(df, method, hp){ # Devuelve eps para 1 df con el hp
especificado

  if (method == "randomForest"){

```

```

rf <- randomForest(factor(Tr) ~ w1 + w2 + w3 + w4 + w5 + w6,
                    data = df,
                    mtry = hp)
eps <- predict(rf, type = "prob")[, 2]

} else if (method == "gbm"){
  gbm <- gbm(
    formula = Tr ~ w1 + w2 + w3 + w4 + w5 + w6,
    data = df,
    distribution = "bernoulli",
    n.trees = 100,
    interaction.depth = 3,
    shrinkage = hp,
    n.minobsinnode = 10,
    verbose= FALSE)
  eps <- predict(gbm, newdata = df, n.trees = 100, type = "response")

} else if (method == "rpart"){
  cart <- rpart(
    formula = Tr ~ w1 + w2 + w3 + w4 + w5 + w6,
    data = df,
    method = "class",
    cp = rpart.control(hp))
  eps <- predict(cart, newdata = df, type = "prob")[, 2]
}

return(eps)
}

# `matching` aplica el matching con los PS estimados y devuelve los
ASMD's
matching <- function(df, eps){
  m.out <- matchit(Tr ~ w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10,
                  data = df,
                  method = "nearest",
                  replace = TRUE,
                  caliper = 0.1,
                  distance = eps)

  balance <- bal.tab(m.out,
                    data = df)

  asmds <- abs(balance$Balance$Diff.Adj)[2:11]

  return(asmds)
}

# `ipw` aplica el IPW con los PS estimados y devuelve los ASMD's

```



```

ipw <- function(df, eps){

  eps <- pmin(pmax(eps, 0.01), 0.99)

  # Cálculo de los weights
  p_Tr <- mean(df$Tr == 1)
  p_noTr <- 1 - p_Tr
  weights <- ifelse(df$Tr == 1,
                    p_Tr / eps,
                    p_noTr / (1 - eps))

  # `bal.tab` calcula el balance
  balance <- bal.tab(
    Tr ~ w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10,
    data = df,
    weights = weights,
    method = "weighting")

  asmds <- abs(balance$Balance$Diff.Adj)

  return(asmds)
}

# `hp_testing_m` integra las funciones anteriores para devolver los
ASMD's
# resultantes de aplicar matching con los PS estimados por el método de
ML especificado,
# ajustándolo con un determinado hiperparámetro (`hp`).
hp_testing_m <- function(hp, scenario, method){

  dfs <- all_scenarios[[scenario]]
  results <- data.frame()

  for (i in seq_along(dfs)) {
    # Para cada dataframe del escenario:
    eps <- ML_eps(dfs[[i]], method, hp) # Se estiman los eps por el ML
especificado
    asmds <- matching(dfs[[i]], eps) # Se aplica matching, y se obtienen
los ASMD's
    results <- rbind(results, asmds)
  }

  return(results)
}

# `hp_testing_w` hace lo mismo pero aplicando los PS estimados con IPW.
hp_testing_w <- function(hp, scenario, method){

```

```

dfs <- all_scenarios[[scenario]]
results <- data.frame()

for (i in seq_along(dfs)) {
  eps <- ML_eps(dfs[[i]], method, hp)
  asmds <- ipw(dfs[[i]], eps)
  results <- rbind(results, asmds)
}

return(results)
}

# `opt_hp` ayuda a seleccionar el hiperparámetro óptimo.
opt_hp <- function(hp_tests, scenario){

  # Para calcular el ASAM utilizamos distintas covariables en cada
escenario;
  # ya que cada uno presenta diferentes variables confusoras.
  # Si el escenario es A, utilizamos w1-w6
  if (scenario == "A"){
    confounders <- c(1:6)
  } else if (scenario == "B"){
    confounders <- c(1:6, 10)
  } else if (scenario == "C"){
    confounders <- c(1:10)
  } else{
    confounders <- c(1, 3:7, 9, 10)
  }

  v <- c()
  for (test in hp_tests){
    asam_mean <- mean(apply(test[confounders], 1, mean))
    v <- c(v, asam_mean)
  }
  return(which.min(v))
}

# `balance_results` muy similar a la función anterior. Simplemente resume
los
# resultados para un determinado escenario. Devuelve una lista con el
promedio
# y la desviación del ASAM, así como de los ASMD's.

balance <- function(df, scenario){

  cov <- paste0("w", c(1:10))

  # Para calcular el ASAM utilizamos distintas covariables en cada

```

```

escenario;
# ya que cada uno presenta diferentes variables confusoras.

if (scenario == "A"){ # Si el escenario es A, utilizamos w1-w6
  confounders <- c(1:6)
} else if (scenario == "B"){ # Para B, utilizamos w1-w6 y w10
  confounders <- c(1:6, 10)
} else if (scenario == "C"){ # Para C, utilizamos w1-w10
  confounders <- c(1:10)
} else{
  confounders <- c(1, 3:7, 9, 10) # Para D y E, utilizamos w1, w3-w7,
w9 y w10
}

asmd_mean <- apply(df, 2, mean)
names(asmd_mean) <- cov

asmd_sd <- apply(df, 2, sd)
names(asmd_sd) <- cov

asam_mean <- mean(asmd_mean[confounders])
asam_sd <- sd(apply(df[confounders], 1, mean))

result <- list("asmd_mean" = asmd_mean,
              "asmd_sd" = asmd_sd,
              "asam_mean" = asam_mean,
              "asam_sd" = asam_sd)
return(result)
}

# Funciones de estimación de PS simples:
simple_RL <- function(df){
  m.out <- matchit(Tr ~ w1 + w2 + w3 + w4 + w5 + w6,
                  data = df,
                  distance = "glm")
  eps <- m.out$distance
  return(eps)
}
simple_RF <- function(df, hp){
  rf <- randomForest(factor(Tr) ~ w1 + w2 + w3 + w4 + w5 + w6,
                    data = df,
                    mtry = hp)
  eps <- predict(rf, type = "prob")[, 2]
  return(eps)
}
simple_GBM <- function(df, hp){
  gbm <- gbm(
    formula = Tr ~ w1 + w2 + w3 + w4 + w5 + w6,
    data = df,

```

```

distribution = "bernoulli",
n.trees = 100,
interaction.depth = 3,
shrinkage = hp,
n.minobsinnode = 10,
verbose= FALSE)
eps <- predict(gbm, newdata = df, n.trees = 100, type = "response")
return(eps)
}
simple_CART <- function(df, hp){
  cart <- rpart(
    formula = Tr ~ w1 + w2 + w3 + w4 + w5 + w6,
    data = df,
    method = "class",
    cp = rpart.control(hp))
  eps <- predict(cart, newdata = df, type = "prob")[, 2]
  return(eps)
}

```

*# `ate_estimation` devuelve la estimación del ATE para un dataframe.
 Estima los
 # PS por el método indicado (`method`), y aplica IPW (no matching).
 # Devuelve la estimación del ATE calculada con el paquete `survey`.*

```

ate_est <- function(df, method, hp){

  if (method == "randomForest"){
    eps = simple_RF(df, hp)
  } else if (method == "gbm"){
    eps = simple_GBM(df, hp)
  } else if (method == "cart"){
    eps = simple_CART(df, hp)
  } else if (method == "logreg"){
    eps= simple_RL(df)
  }

  # IPW

  eps <- pmin(pmax(eps, 0.01), 0.99)

  # Cálculo de los weights
  p_Tr <- mean(df$Tr == 1)
  p_noTr <- 1 - p_Tr
  weights <- ifelse(df$Tr == 1,
                    p_Tr / eps,
                    p_noTr / (1 - eps))

  d <- svydesign(ids = ~1, data = df, weights = ~weights)
  results <- svyglm(Y ~ Tr, design = d)

```

```
ate_est <- coef(results)["Tr"]  
  
ate <- mean(df[df$Tr == 1]$Y - df[df$Tr == 0]$Y)  
  
return(ate)  
}
```