Uso responsable de la IA para las políticas públicas:
Manual de ciencia de datos

Felipe González
Teresa Ortiz
Roberto Sánchez

OECD BID

Uso responsable de la IA para las políticas públicas:

Manual de ciencia de datos

Felipe González, Teresa Ortiz y Roberto Sánchez Ávalos

https://www.iadb.org/

Banco Interamericano de Desarrollo (BID) - Sector Social

El Sector Social (SCL) está conformado por un equipo multidisciplinario que actúa con la convicción de que la inversión en las personas permite mejorar sus vidas y superar los desafíos del desarrollo en América Latina y el Caribe. Junto con los países de la región, el Sector Social formula soluciones de política pública para reducir la pobreza y mejorar la prestación de servicios de educación, trabajo, protección social y salud. El objetivo es construir una región más productiva donde predominen la igualdad de oportunidades para hombres y mujeres, y una mayor inclusión de los grupos más vulnerables. www. iadb.org/en/about-us/departments/scl

Banco Interamericano de Desarrollo (BID) – BID Lab

BID Lab es el laboratorio de innovación del Grupo BID. Allí se movilizan financiamiento, conocimiento y conexiones para catalizar la innovación orientada a la inclusión en América Latina y el Caribe. Para BID Lab, la innovación es una herramienta poderosa que puede transformar la región al crear oportunidades sin precedentes para las poblaciones en situación vulnerable por las condiciones económicas, sociales y ambientales en que se encuentran. https://bidlab.org/

Organización para la Cooperación y el Desarrollo Económicos

La OECD (por sus siglas en inglés) es una organización internacional que trabaja para construir mejores políticas para una vida mejor. Nuestro objetivo es elaborar políticas que fomenten la prosperidad, la igualdad, las oportunidades y el bienestar para todos.

Junto con los gobiernos, los responsables políticos y los ciudadanos, trabajamos en el establecimiento de normas internacionales basadas en pruebas y en la búsqueda de soluciones a una serie de retos sociales, económicos y medioambientales. Un ejemplo de establecimiento de normas son los Principios de la OECD sobre Inteligencia Artificial (IA), que son los primeros principios de este tipo adoptados por los gobiernos. Estos principios promueven una IA innovadora y fiable que respete los derechos humanos y los valores democráticos. https://ia-latam.com/portfolio/principios-de-la-OECD-sobre-ia/

OECD.AI Policy Observatory

El Observatorio OECD.AI es un hub de políticas públicas sobre inteligencia artificial (IA). Ayuda a los países a fomentar, alimentar y supervisar el desarrollo y el uso de una IA confiable.

Utilizado por los responsables políticos y otras partes interesadas en más de 170 países, OECD.AI se ha convertido en un centro reconocido para la evidencia, el debate y la orientación orientada a la política, con el apoyo de fuertes asociaciones con actores de todos los grupos de interés y con otras organizaciones internacionales. Ofrece un análisis basado en evidencias sobre la IA.

OECD.AI es una fuente única de datos y visualizaciones en tiempo real sobre la evolución de la IA. También contiene una base de datos de políticas de IA de más de 60 países que permite a los gobiernos comparar las respuestas políticas y desarrollar buenas prácticas. OECD.AI mide nuestro progreso colectivo hacia una IA digna de confianza; su red de expertas y expertos y el blog AI Wonk facilitan los debates colaborativos sobre políticas de IA.

Se puede acceder a otros trabajos de la OECD relacionados con la aplicación específica de la IA en el sector público a través del Observatorio de la Innovación del Sector Público (OPSI) de la OECD (https://oecd-opsi.org/), que ofrece una visión general de las medidas de IA aplicadas en el sector público, incluidas las relativas a la elaboración de políticas de gobernanza de la IA y el diseño y la prestación de servicios públicos.

fAIr LAC

Iniciativa fAIr LAC

El Banco Interamericano de Desarrollo (BID), en colaboración con socios y aliados estratégicos, lidera la iniciativa fAIr LAC, mediante la cual se busca promover la adopción responsable de la Inteligencia Artificial (IA) y los sistemas de soporte de decisión para mejorar la prestación de servicios sociales y crear oportunidades de desarrollo en aras de atenuar la desigualdad social. Este manual es parte de un grupo de documentos y herramientas para equipos técnicos y responsables de la formulación de políticas públicas para guiarlos en la mitigación de los retos de los sistemas de soporte de decisión y promover una adopción responsable de la IA (Pombo, Cabrol, González, & Sánchez, 2020).

Agradecimientos

Por su tiempo y valiosos aportes, expresamos un agradecimiento especial a Cristina Pombo, coordinadora de la iniciativa fAIr LAC del BID y al Prof. Ricardo Baeza-Yates, Director de Ciencia de Datos en Northeastern University, Campus de Silicon Valley, e integrante del Grupo de Expertos y Expertas de fAIr LAC. Los autores también agradecen por las contribuciones recibidas de Karine Perset, administratora del OECD.AI, y de Luis Aranda, analista de políticas del OECD.AI.

Agradecemos igualmente el apoyo prestado y los comentarios recibidos de Luis Tejerina, Elena Arias Ortiz, Natalia González Alarcón, Tetsuro Narita, Constanza Gómez-Mont, Daniel Korn, Ulises Cortés, José Antonio Guridi Bustos, César Rosales y Sofía Trejo.

TABLA DE CONTENIDO

Resumen ejecutivo 7
- ¿Por qué este manual? 8
- ¿Para quién es este manual? 8
- Glosario 9

Introducción 10
- Machine Learning (ML) y sistemas de toma/soporte de decisiones 10
- Componentes de un sistema de IA para políticas públicas 12
- Retos del ciclo de vida del ML 13

1. Conceptualización y diseño 16
- 1.1 Definición correcta del problema y de la respuesta de política pública 17
- 1.2 Principios de IA responsable 17

2. Recolección y procesamiento de datos 20
- 2.1 Calidad y relevancia de los datos disponibles 21
- 2.2 Cualificación y exhaustividad de los datos para la población objetivo 23

3. Desarrollo del modelo y validación 28
- 3.1 Ausencia o uso inadecuado de muestras de validación 29
- 3.2 Fugas de información 30
- 3.3 Modelos de clasificación: probabilidades y clases 32
- 3.4 Sub y sobreajuste 35
- 3.5 Errores no cuantificados y evaluación humana 36
- 3.6 Equidad y desempeño diferencial de predictores 37

4. Uso y monitoreo 40
- 4.1 Degradación de desempeño 41
- 4.2 Experimentos y evaluación del modelo 42

5. Rendición de cuentas 43
- 5.1 Interpretabilidad y explicación de predicciones 44
- 5.2 Trazabilidad 46

Herramientas 48
- **Herramienta 1:** Lista de verificación de IA robusta y responsable 49
- **Herramienta 2:** Perfil de datos 55
- **Herramienta 3:** Perfil del modelo (Model Card) 57

Cuadernillos de trabajo 59
- Recolección y procesamiento de datos 60
- Desarrollo de modelo y validación 71
- Rendición de cuentas 94

Referencias 99

RESUMEN EJECUTIVO

Desde las finanzas y los seguros hasta la agricultura y el transporte, la inteligencia artificial (IA) se está difundiendo a gran velocidad en todos los sectores, creando oportunidades, pero también planteando nuevos problemas de política pública. En el sector público, la IA promete generar ganancias de productividad y mejorar la calidad de los servicios públicos. Al analizar la actividad de las redes sociales en tiempo real, los responsables de política pública pueden, por ejemplo, aprovechar los sistemas de IA para obtener una evaluación más precisa y basada en pruebas de los problemas y necesidades sociales más acuciantes. Los resultados y predicciones de los sistemas de IA pueden servir como base para la formulación, aplicación y evaluación de políticas.

En este contexto, los gobiernos de todo el mundo se están dotando de los conocimientos técnicos necesarios para aprovechar el poder de la IA en apoyo del desarrollo de las políticas públicas. Sin embargo, dado que estas políticas basadas en la IA pueden tener un impacto significativo en la vida y el bienestar de las personas, se necesita un enfoque sistémico que garantice la existencia de las salvaguardias adecuadas para aprovechar las oportunidades que ofrece el uso de estos sistemas por parte de los equipos de políticas públicas y hacer frente a los retos que plantea.

Utilizando el ciclo de vida de los sistemas de IA como marco de análisis, este conjunto de herramientas ofrece orientación técnica a los equipos de políticas públicas que deseen utilizar las tecnologías de IA para mejorar sus procesos de toma de decisiones y sus resultados. Para cada fase del ciclo de vida del sistema de IA –"conceptualización y diseño", "recolección y procesamiento de datos", "desarrollo y validación de modelos", y "uso y monitoreo"–, el conjunto de herramientas identifica los retos más comunes del uso de la IA en contextos de políticas públicas y esboza mecanismos prácticos para detectar y mitigar estos retos.

Los responsables políticos y sus equipos técnicos deben responsabilizarse del buen funcionamiento de un sistema de IA en cada fase de su ciclo de vida. En este sentido, uno de los capítulos de la caja de herramientas está dedicado a analizar los problemas relacionados con la responsabilidad en el uso de la IA para las políticas públicas y a esbozar mecanismos prácticos para abordarlos.

Fiel a su objetivo de promover el uso responsable de la IA para la elaboración de políticas públicas, cada sección del conjunto de herramientas incluye listas de comprobación para ayudar a orientar la aplicación práctica. También se ofrece una herramienta de "perfil de datos" y una "ficha del modelo" para ayudar a evaluar los problemas de datos y documentar las características de un sistema de IA, las suposiciones realizadas y las medidas de mitigación de riesgos aplicadas a lo largo del ciclo de vida. Además, el conjunto de herramientas ofrece una sección con un cuaderno de trabajo que contiene ejemplos prácticos de algunos de los retos y estrategias de mitigación tratados en el informe, así como el código pertinente para aplicarlos utilizando R u otros lenguajes de programación.

A través de la iniciativa fAIr LAC y el Observatorio de Políticas de IA de la OECD, el BID y la OECD se han asociado para ayudar a que el debate sobre políticas de IA pase de los principios de alto nivel a la práctica y la implementación. Este conjunto de herramientas es un paso concreto en esta dirección.

¿Por qué este manual?

A pesar de que existe un número importante de principios que buscan una IA ética, solo proporcionan una orientación de alto nivel sobre lo que debe o no hacerse en su desarrollo y existe muy poca claridad sobre cuáles son las mejores prácticas para ponerlas en funcionamiento (Vayena, 2019). El objetivo de este manual es proveer esas recomendaciones y buenas prácticas técnicas con el fin de evitar resultados contrarios (muchas veces inesperados) a los objetivos de los tomadores de decisiones. Esos fines son variados: pueden referirse a consecuencias no deseables desde el punto de vista de los tomadores de decisiones, desaprovechamiento de recursos debido a focalizaciones inadecuadas o cualquier otro objetivo que el tomador de decisiones esté buscando lograr.[1]

¿Para quién es este manual?

Este manual está pensado para equipos técnicos que trabajan en la aplicación de algoritmos de aprendizaje automático para políticas públicas. Sin embargo, todos los retos que cubre son comunes a cualquier aplicación de esta tecnología. Se asume que el lector cuenta con conocimientos básicos de estadística y programación, aunque cuando se nombran conceptos se incluyen descripciones breves y se comparte bibliografía adicional. El manual incluye cuadernillos de trabajo con varios ejemplos de los retos y soluciones explicadas. Se usan distintos tipos de modelos (lineales, basados en árboles y otros) y distintas implementaciones (R, Keras, Xgboost) para mostrar que estos problemas se presentan independientemente de la elección de herramientas particulares. Aunque los códigos y ejemplos se desarrollaron en R, todos los temas y metodologías aplicadas y descritas en este manual pueden implementarse en cualquier otro lenguaje de programación.[2]

1 Este manual no pretende reglamentar o explicar cuáles deben ser los fines y objetivos de organismos y actores que toman las decisiones.

2 Todo el material de este documento es reproducible según instrucciones en el repositorio https://github.com/EL-BID/Manual-IA-Responsable, que contiene un archivo Dockerfile que describe las dependencias de infraestructura para su replicación. Se utiliza el lenguaje de programación R y los siguientes paquetes: tidyverse, recipes, themis, rsample, parsnip, yardstick, workflows, tune, knitr, patchwork.

GLOSARIO

- **Aprendizaje automático:** conjunto de técnicas que permiten a un sistema aprender comportamientos de forma automatizada a través de patrones e inferencias en lugar de instrucciones explícitas o simbólicas introducidas por un ser humano (OECD, 2019c).
- **Atributo protegido:** una característica o variable protegida es aquella en que queremos que se cumpla cierto criterio de equidad en las predicciones. En un conjunto de datos podemos tener más de una variable protegida, como edad, género, raza, etc.
- **Criterio de justicia algorítmica:** representación matemática de una definición de justicia específica que se incorpora en el proceso de ajuste y selección del modelo. Es importante tomar en cuenta que estas definiciones pueden ser excluyentes, es decir, satisfacer una podría implicar no satisfacer las demás (Verma & Rubin, 2018).
- **Estructura predictiva:** se usa para hablar en general del tipo de modelos empleados para hacer predicciones (lineales, bosques aleatorios, redes neuronales), las características que utiliza y cómo las usa el modelo (interacciones, transformaciones no lineales).
- **Garantías probabilísticas:** en muestras diseñadas con aleatorización es posible, bajo ciertos supuestos, caracterizar el comportamiento de estimadores y procedimientos (con alta probabilidad). Por ejemplo, un intervalo de confianza de 95 % para las métricas de desempeño que contiene al valor real que será observado.
- **Inequidad algorítmica:** falla técnica en los modelos que produce disparidad de resultados para grupos protegidos que deben evaluarse con la definición de justicia algorítmica determinada en un punto anterior (podría ser más de una).
- **Inteligencia artificial:** sistema computacional que es capaz de influir en el entorno y producir un resultado (predicciones, recomendaciones o decisiones) para un conjunto de objetivos determinado. Utiliza datos e insumos de fuentes humanas o sensores para (i) percibir entornos reales y/o virtuales; (ii) abstraer estas percepciones en modelos mediante el análisis en forma automatizada (por ejemplo, con aprendizaje automático), o manual; y (iii) utilizar la inferencia del modelo para formular resultados. Los sistemas de IA están diseñados para funcionar con distintos niveles de autonomía (adaptado de OECD 2019c).
- **Población objetivo:** conjunto de elementos que se pretende intervenir (personas, hogares, zonas geográficas, etc.). Los modelos se construyen con el fin de aplicarse a la población objetivo.
- **Sistemas de soporte de decisión:** relacionados con el concepto de inteligencia asistida o aumentada, se utilizan para describir los sistemas en donde la información generada por los modelos de aprendizaje automático se usa como insumo para la toma de decisiones por un ser humano.
- **Sistemas de toma de decisión:** estos sistemas se relacionan con el concepto de inteligencia automatizada y autónoma. Las decisiones finales y su consecuente acción se toman sin intervención humana directa. Es decir, el sistema pasa a realizar tareas previamente desarrolladas por un ser humano.
- **Subpoblaciones de interés o subpoblaciones protegidas:** son subpoblaciones de la población objetivo para las cuales se quiere tener evaluaciones concretas del desempeño de estimaciones o de los modelos.

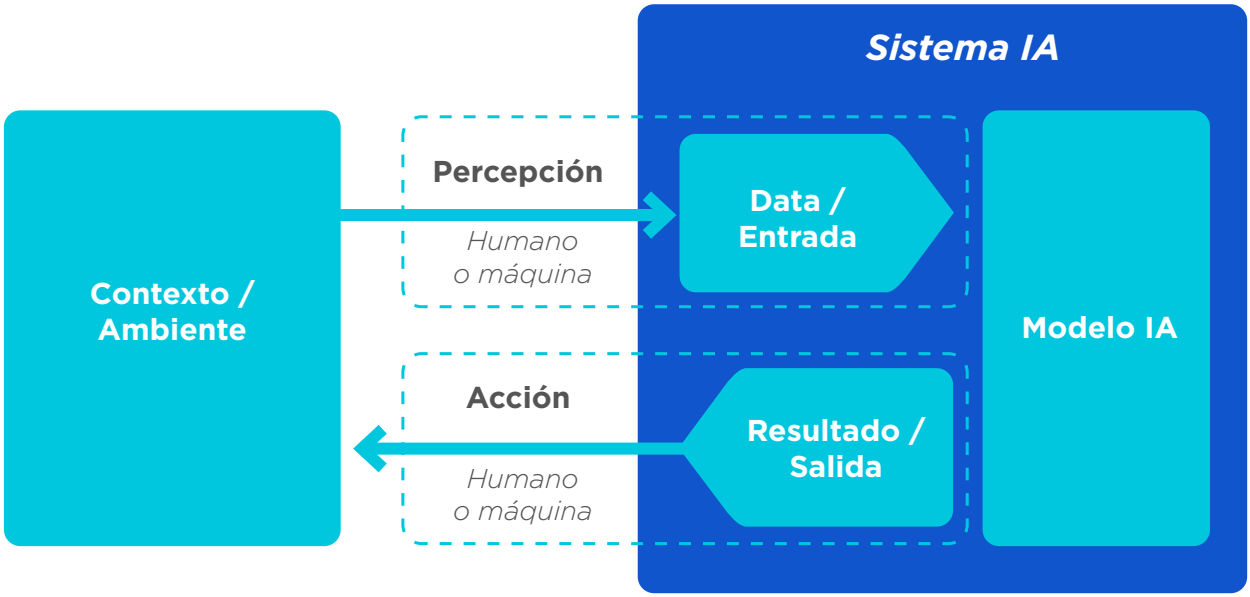
INTRODUCCIÓN

Los métodos de aprendizaje automático (que para resumir en este documento llamamos ML, por sus siglas en inglés, Machine Learning), como un subconjunto de lo que se conoce como inteligencia artificial, son cada vez más requeridos y utilizados por tomadores de decisiones para informar acciones o intervenciones en varios contextos, desde negocios hasta política pública. En la práctica, estos métodos se han utilizado con diversos grados de éxito y con esto ha aparecido la preocupación creciente de cómo entender el desempeño e influencia positiva o negativa de estos métodos en la sociedad (Barocas and Selbst, 2016; Suresh and Guttag, 2019).

Machine Learning (ML) y sistemas de toma/soporte de decisiones

La Organización para la Cooperación y el Desarrollo Económico (OECD por sus siglas inglés) describe la IA como un sistema computacional que es capaz de influir en el entorno produciendo un resultado (predicciones, recomendaciones o decisiones) para un conjunto de objetivos determinado. Utiliza datos e insumos de fuentes humanas o sensores para (i) percibir entornos reales y/o virtuales; (ii) abstraer estas percepciones en modelos mediante el análisis de forma automatizada (por ejemplo, con aprendizaje automático), o manual; y (iii) utilizar la inferencia del modelo para formular resultados. Los sistemas de IA están diseñados para funcionar con distintos niveles de autonomía (Adaptado de OECD 2019c).

Figura 1. Vista conceptual de un sistema de IA



Aunque los métodos de aprendizaje automático no son el único tipo de algoritmos que pueden utilizar los sistemas de IA, sí son los que han tenido más crecimiento de los últimos años. Se trata de un conjunto de técnicas para permitir que un sistema aprenda comportamientos de manera automatizada a través de patrones e inferencias en lugar de instrucciones explícitas o simbólicas introducidas por un ser humano (OECD 2019c).

Este manual analiza algunos de los retos más comunes en el uso de las tecnologías de aprendizaje automático para la toma de decisiones o el apoyo a las mismas. Entre ellos se encuentran la detección y mitigación de errores y de sesgos y la evaluación de resultados no deseados por una empresa, institución del sector público o sociedad.

Se consideran dos arquetipos de inclusión de aprendizaje automático en el proceso de toma de decisiones:[3]

1. **Sistemas de soporte de decisión:** relacionados con el concepto de inteligencia asistida o aumentada, se utilizan para describir los sistemas en donde la información generada por los modelos de aprendizaje automático se usa como insumo para la toma de decisiones por un ser humano.

2. **Sistemas de toma de decisión:** estos sistemas se relacionan con el concepto de inteligencia automatizada y autónoma. Las decisiones finales y su consecuente acción se toman sin intervención humana directa. Es decir, el sistema pasa a realizar tareas previamente desarrolladas por un ser humano. En muchos contextos se emplea ADM para denominar estos sistemas por su sigla en inglés: *Automated Decision Making*.

Para el desarrollo de un sistema de toma/soporte de decisión exitoso basado en aprendizaje automático debe considerarse que existe una gran variedad de técnicas, conocimiento experto del tema y de modelación en general. En este manual no se pretende discutir métodos particulares de aprendizaje automático ni de procesos específicos de ajuste de hiperparámetros –ver por ejemplo (Hastie, Tibshirani, and Friedman, 2017; Kuhn and Johnson, 2013; Gelman and Hill, 2006)–, sino concentrarse en su evaluación y en los retos más importantes que los sistemas comparten sin importar el tipo de algoritmo o tecnología utilizada.

Por otra parte, la evaluación de un sistema de aprendizaje no tiene sentido fuera de su contexto. Preguntas como ¿cuál es la tasa de error apropiada? o ¿cuáles sesgos son poco aceptables?, entre otras, solo pueden considerarse y responderse dentro del contexto específico de su aplicación, de los propósitos y motivaciones de los tomadores de decisiones, así como por el riesgo que se presenta en los usuarios finales. Es decir, muchos de los criterios técnicos tienen que entenderse a la luz del problema específico. Los sistemas de toma/soporte de decisión nunca son perfectos, pero si se conocen sus sesgos y sus limitaciones incluso un sistema con una precisión baja podría ser útil y utilizarse responsablemente. En el caso contrario, tener un sistema con métricas de evaluación altas no elimina el riesgo de un uso irresponsable si no se entienden sus limitaciones.

Objetivos

- Este manual se centra en el subconjunto de desafíos que están relacionados con los procesos técnicos a lo largo del ciclo de vida de los sistemas de IA utilizados para la toma y soporte de decisiones de política pública.

- Este manual describe cómo diferentes sesgos y deficiencias pueden ser causados por los datos de entrenamiento, por decisiones en el desarrollo del modelo o tomadas durante el proceso de validación y monitoreo.

3 Estos dos tipos de sistemas son genéricos, es decir, no utilizan necesariamente el aprendizaje automático. Además, estos sistemas pueden ser interactivos y aprender dinámicamente mediante técnicas de aprendizaje por refuerzo, pero en este manual solo consideramos los sistemas no interactivos.

Componentes de un sistema de IA para políticas públicas

Ciclo de vida de la política pública con IA

La IA no sustituye a la política pública, pues por sí misma no soluciona el problema social. Su función es asistir proveyendo información para la toma o soporte de decisiones.
El ciclo de política pública asistido por IA lo componen las siguientes etapas:

1. **Identificación del problema:** todo proyecto de IA debe iniciar identificando correctamente el problema social al que la política pública busca impactar, detallando sus posibles causas y consecuencias.

2. **Formulación de intervención:** se explicita la intervención o política que se está considerando aplicar a ciertas personas, unidades o procesos. Se supone generalmente que se tiene evidencia del beneficio de esa política cuando se aplica a la población objetivo.

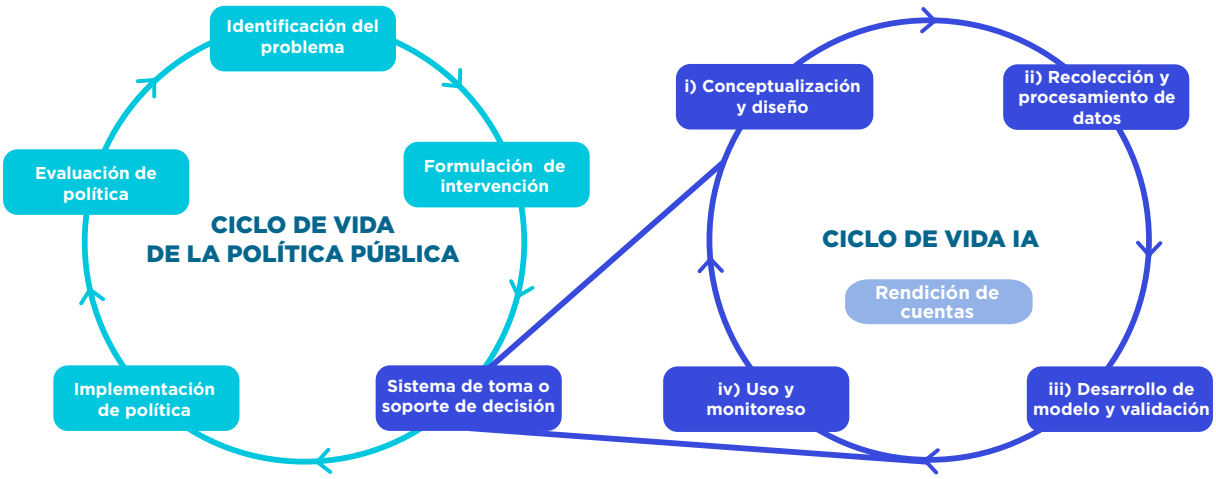
3. **Sistema de toma/soporte de decisión:** una vez definida la intervención, se inicia el ciclo de la IA con el diseño y desarrollo del sistema de toma/soporte de decisión, cuyo resultado se utilizará para focalizar u orientar la intervención elegida en el punto anterior.[4]

4. **Implementación de política:** se pone en funcionamiento la política pública, ya sea como proyecto piloto y/o con una escala mayor.

5. **Evaluación de política:** se evalúan la eficacia, la fiabilidad, el costo, las consecuencias previstas y no previstas y otras características pertinentes de la medida de política en cuestión. Si sus resultados son positivos, se escala o continúa la intervención.

En paralelo con el ciclo de elaboración de políticas públicas, el desarrollo de un sistema de IA tiene su propio ciclo de vida que incluye las siguientes etapas (OECD, 2019c): (i) Conceptualización y diseño; (ii) Recolección y procesamiento de datos; (iii) Desarrollo y validación de modelos, y (iv) Uso y monitoreo. Estas fases suelen tener lugar en forma iterativa y no son necesariamente secuenciales (Figura 2).

Figura 2. Ciclo de vida de las políticas públicas asistido por un sistema de toma/soporte a la toma de decisiones.



Fuente: Preparado por los autores.

4 La IA puede utilizarse de distintas maneras. Algunas de ellas pueden ser: i) Sistemas de alerta temprana o detección de anomalías: predicción de deserción escolar o alertas de fenómenos hidrometeorológicos; ii) Sistemas de recomendación o personalización: recomendación para vacantes laborales o personalización de materiales educativos, y iii) Sistemas de reconocimiento, diagnóstico de enfermedades, detección de objetos o reconocimiento biométrico.

En la interrelación de estos dos ciclos se generan importantes retos que deben ser evaluados y considerados durante el desarrollo y uso de sistemas de IA robustos y responsables.

Retos del ciclo de vida del ML

Para la construcción de sistemas de toma/soporte de decisión robustos y responsables es necesario efectuar varias tareas: considerar las posibles fuentes de sesgo y deficiencias que pueden causar los datos de entrenamiento y problemas y decisiones en el desarrollo del modelo; definir en forma clara los objetivos de los sistemas y los criterios de justicia que se buscará cumplir; entender las limitantes y errores en el contexto del proyecto específico y establecer medidas de monitoreo de los sistemas para evitar que se produzcan resultados indeseables e inequidad en la toma de decisiones.

Para lograrlo, este manual presenta los retos y errores usuales en la construcción y aplicación de métodos de aprendizaje automático durante el ciclo de vida de la IA. Cinco secciones describen los problemas más comunes que pueden encontrarse, diagnósticos para detectarlos y sugerencias para mitigarlos:

1. **Conceptualización y diseño:** se refiere a la información y criterios necesarios que debe obtener el tomador de decisiones de política pública para iniciar un proyecto de IA.
2. **Recolección y procesamiento de datos:** se enfoca en el proceso de generación de datos, la selección y el control de las distintas fuentes, y la identificación y mitigación de las deficiencias y sesgos.
3. **Desarrollo del modelo y validación:** alude a métodos y principios importantes para construir modelos robustos y validados correctamente.
4. **Uso y monitoreo:** es la evaluación del modelo en producción y seguimiento de los principios clave para evitar una degradación inesperada.

Además, un quinto aspecto, la Rendición de cuentas, es una dimensión transversal en el ciclo de vida del sistema de IA que se refiere a las medidas de transparencia y explicabilidad para promover la comprensión de los mecanismos a través de los cuales un sistema de IA produce un resultado, la reproducibilidad del resultado y la capacidad del usuario para identificar y cuestionar errores o resultados inesperados.

Los actores de la IA en cada etapa del ciclo de vida deben ser responsables del buen funcionamiento de un sistema de IA en función de sus roles en el desarrollo y del contexto del uso del sistema.

Se proponen tres herramientas para acompañar el desarrollo del sistema de IA:

- **Herramienta 1:** Lista de verificación de IA robusta y responsable. Esta herramienta consolida las principales preocupaciones por la dimensión de riesgo del ciclo de vida de IA. Esta lista debe revisarla en forma continua el equipo técnico acompañado por el tomador de decisiones.
- **Herramienta 2:** Perfil de datos. Este perfil es un análisis exploratorio inicial durante la fase de Recolección y procesamiento de datos del ciclo de vida de IA. Brinda información para evaluar la calidad, integridad, temporalidad, consistencia y posibles sesgos, daños potenciales e implicaciones de su uso.
- **Herramienta 3**: Perfil del modelo. Es la descripción final de un sistema de IA; reporta los principales supuestos, las características más importantes del sistema y las medidas de mitigación implementadas.

Recuadro 1. Fuentes de sesgo un sistema de IA

Uno de los conceptos más importantes para los retos del ciclo de vida de la IA es el de los sesgos, pues muchas de las medidas de mitigación y retos que tienen que contemplarse durante el desarrollo de los modelos depende de su correcta comprensión y tratamiento. Para abordar tempranamente este problema es conveniente tener revisiones específicas en las distintas etapas del ciclo de vida. En cada revisión debe invitarse a los expertos y usuarios finales del sistema que corresponda para verificar y defender las hipótesis realizadas durante cada etapa. Esto permite enriquecer los puntos de vista, encontrar suposiciones erradas y agregar aspectos no considerados.

El **error del sistema** es la diferencia entre el valor predicho, resultado del modelo, y el valor real de la variable que se está estimando. Si el error es sistemático en una dirección o en un subconjunto específico de los datos, se llama **sesgo**[5]. Por ejemplo, si una balanza siempre pesa un kilo más, está sesgada; o si un valor es siempre menor, como el salario de las mujeres para un trabajo equivalente al que realizan los hombres, la variable salario está sesgada. Por otro lado, cuando el **error** es aleatorio, se llama **ruido**.

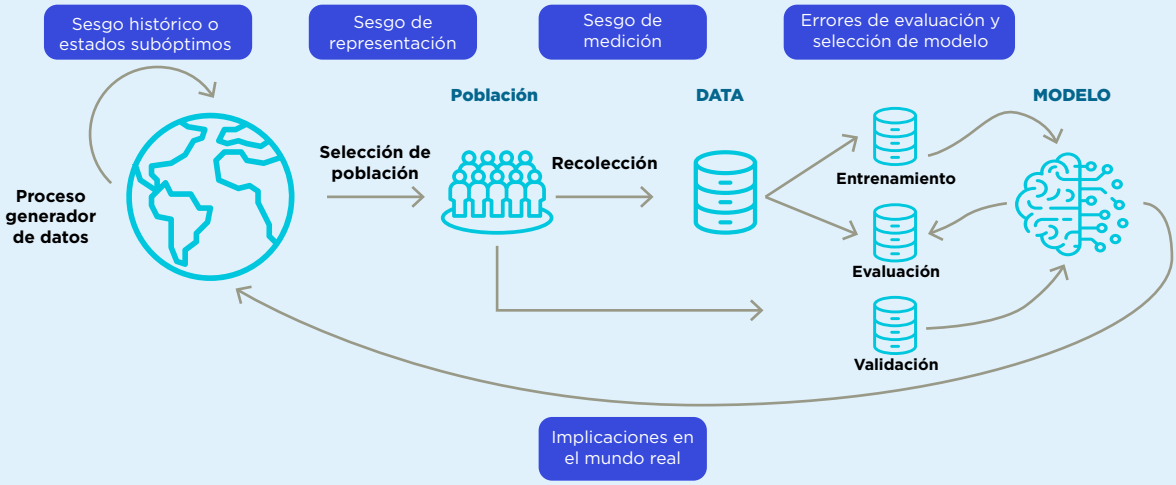
El sesgo de un sistema de IA puede tener implicaciones éticas cuando sus resultados se utilizan para formular políticas públicas que pueden considerarse injustas o perjudiciales para determinados subgrupos de la población. Esta evaluación del sesgo está sujeta a una definición específica de equidad algorítmica, que deben determinar los responsables de las políticas públicas.

Una definición de equidad algorítmica es una representación matemática de un objetivo de política pública que se incorpora al proceso de selección y ajuste del modelo. Por ejemplo, en algunos casos el objetivo de un sistema puede estar ligado a criterios como la paridad demográfica, la igualdad de posibilidades y tener representación por cuotas, entre otros muchos criterios. En algunas ocasiones, el cumplimiento de una definición de equidad algorítmica hace imposible el cumplimiento de otra, es decir, pueden ser parcial o totalmente excluyentes. La definición de equidad algorítmica es una tarea de los responsables de las políticas públicas y no de los equipos técnicos. El equipo técnico solo tiene la tarea de realizar validaciones para garantizar su cumplimiento. La sección 3 de este manual analiza en profundidad las diferentes definiciones de equidad algorítmica y sus implicaciones.

Hay diferentes **fuentes de sesgo**. Algunos sesgos son intrínsecos a los datos, como los sesgos históricos o los estados indeseables, que son patrones preexistentes en la sociedad o en los datos recolectados que no es deseable reproducir en el modelo. El **sesgo de representación** se produce cuando hay información incompleta debido a la falta de atributos, al diseño de la muestra o a la ausencia total o parcial de datos de subpoblaciones. Los **sesgos** de medición surgen por la omisión (inclusión) de variables que deberían (no) estar incluidas en el modelo (Suresh y Guttag, 2019). Otros sesgos aparecen debido a errores metodológicos: por ejemplo, durante el entrenamiento debido a errores en los procesos de validación, definición de métricas y evaluación de resultados (**sesgo de evaluación**), o **debido a supuestos** erróneos sobre la población objetivo que pueden afectar la definición del modelo; también pueden surgir debido al mal uso y seguimiento de los modelos, ya sea por interpretaciones inadecuadas de sus resultados o por cambios temporales en los patrones del mundo real o en los

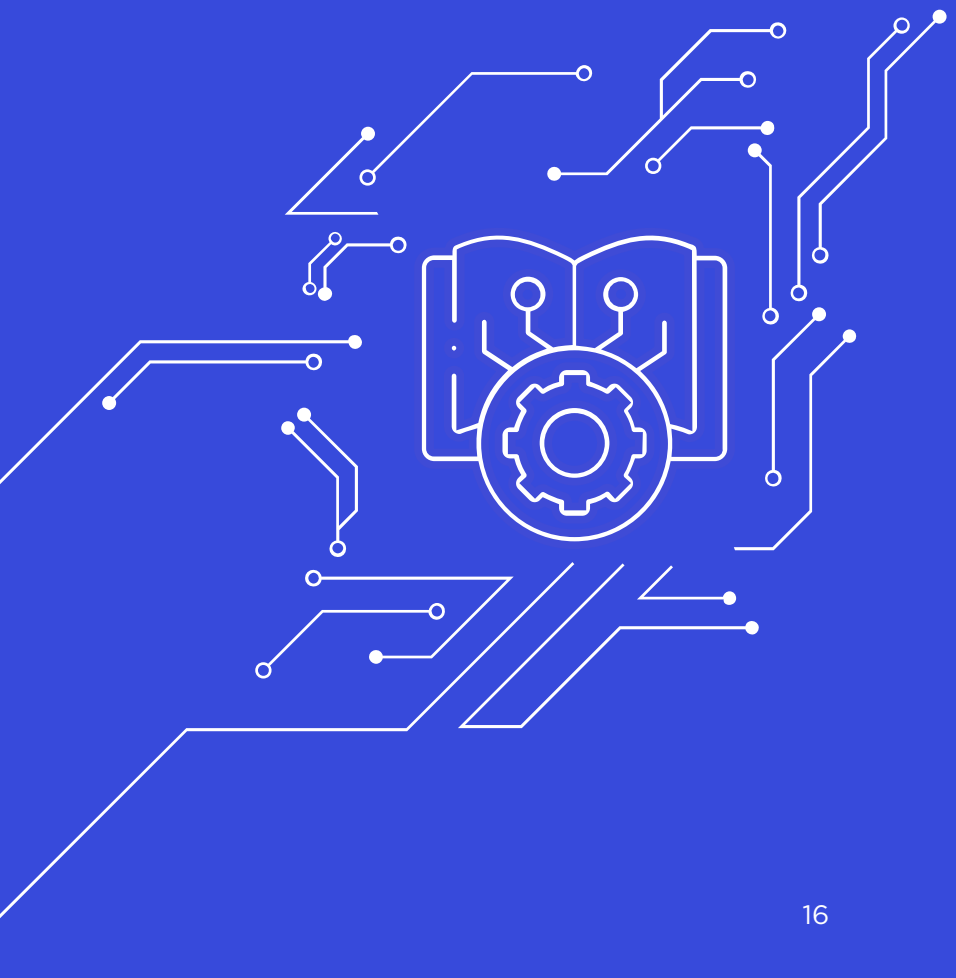
5 En modelos de predicción existe una compensación entre la varianza y el sesgo que capta el modelo y su objetivo de generalización de aprendizaje. Por un lado, un modelo con sesgo alto puede crear sistemas que subajustan y aprenden muy poco de los datos observados, pero modelos con alta varianza pueden tener el efecto contrario y sobreajustar, aprendiendo perfectamente los datos de entrenamiento. La sección de 'Desarrollo de modelos y validación' de este manual describe estos fenómenos con mayor detalle y ofrece medidas para mitigar sus riesgos.

métodos de captación de datos. A lo largo de las diferentes secciones de este manual se presentarán las principales razones de estos sesgos y se propondrán diferentes medidas para mitigarlos.



Fuente: basado en Suresh Guttag (2019).

1. CONCEPTUALIZACIÓN Y DISEÑO



1. Conceptualización y diseño

La implementación de una solución de IA no puede ir separada del ciclo de vida de la política pública con IA[6]. La IA es una herramienta que debe estar condicionada a un buen diseño de la intervención o acción que se tomará con los resultados del sistema. La IA en ningún momento sustituye la política pública. Esto implica que cualquier proyecto de IA robusto y responsable debe partir del problema y no desde la tecnología.

1.1 Definición correcta del problema y de la respuesta de política pública

Este manual asume que existen, al menos, dos actores involucrados en el desarrollo de los sistemas: el tomador de decisiones de políticas públicas y el equipo técnico que los implementará. La definición de la intervención siempre debe ser responsabilidad del tomador de decisiones, que es quien tiene un conocimiento del problema social.

Sin embargo, el equipo técnico debe poder entender el problema para que pueda vincular los resultados del modelo a la intervención deseada. Así mismo, es responsable de guiar y orientar en el diseño del sistema, explicando lo que es viable y definiendo claramente las limitantes y riesgos del sistema, por lo que se requiere una comunicación constante entre ambos actores.
Un caso concreto es la definición de la población donde se aplicará el sistema, la definición de grupos protegidos, así como las medidas de justicia algorítmica por aplicar[7]. Estas definiciones tienen un impacto directo en la forma como puede evaluarse la calidad y cobertura de los datos o el posible sesgo en los resultados del modelo.

1.2 Principios de IA responsable

Aunque la IA tiene un importante potencial para agilizar procesos y ampliar la capacidad del Estado, también hay que señalar que no es una bala de plata. Una vez definidos el problema y el tipo de intervención, es necesario contextualizar y replantear el uso de la IA y el aprendizaje automático en consonancia con los Principios de la IA de la Organización para la Cooperación y el Desarrollo Económico (OECD por sus siglas en inglés; ver recuadro 2).

Es importante tener en cuenta la gobernanza más amplia que enmarca la aplicación de un sistema de IA, incluidas las normas y leyes de la jurisdicción donde va a implantarse el sistema. También es importante establecer los requisitos adecuados durante la conceptualización y diseño del sistema, porque pueden definir o limitar las opciones de desarrollo para el equipo técnico. Por ejemplo, los requisitos de explicabilidad en las predicciones podrían limitar el uso de algunos algoritmos para los que sea muy difícil interpretar los resultados.

6 Ver sección “Componentes de un sistema de IA para políticas públicas”.

7 La sección 3 de este manual discute a profundidad distintas definiciones de justicia algorítmica y sus implicaciones.

Recuadro 2. Principios de IA responsable OECD

Los Principios de la IA de la Organización para la Cooperación y el Desarrollo Económicos (OECD por sus siglas en inglés) promueven el uso responsable de la Inteligencia Artificial (IA), respetando los derechos humanos y los valores democráticos. Los principios establecen normas para la IA que son suficientemente prácticas y flexibles para resistir el paso del tiempo. Incluyen cinco principios basados en valores para la gestión de una IA responsable:

- **Crecimiento inclusivo, desarrollo sostenible y bienestar:** las partes interesadas deben comprometerse a crear una IA confiable que contribuya a inducir resultados benéficos para las personas, así como para el planeta.
- **Valores centrados en el ser humano y la equidad:** los valores de los derechos humanos, la democracia y el Estado de derecho deben incorporarse a lo largo del ciclo de vida del sistema de IA, permitiendo al mismo tiempo la intervención humana mediante mecanismos de salvaguardia.
- **Transparencia y explicabilidad:** los actores que desarrollan u operan los sistemas de IA deben proporcionar información para fomentar una comprensión general de los sistemas entre las partes interesadas que permita a las personas afectadas por los sistemas de IA comprender el resultado y cuestionar la decisión cuando sea necesario.
- **Robustez, seguridad y protección:** los sistemas de IA deben funcionar adecuadamente durante todo su ciclo de vida. Los actores deben garantizar la trazabilidad y aplicar enfoques sistemáticos de gestión de riesgos para mitigarlos.
- **Responsabilidad:** los actores que desarrollen, desplieguen u operen sistemas de IA deben respetar los principios y ser responsables del buen funcionamiento de esos sistemas.

Los Principios de la IA de la OECD contienen cinco recomendaciones para las políticas nacionales y la cooperación internacional: (1) Invertir en investigación y desarrollo de IA; (2) Fomentar un ecosistema digital para la IA; (3) Configurar un entorno político propicio para la IA; (4) Desarrollar la capacidad humana y prepararse para la transformación del mercado laboral, y (5) Promover la cooperación internacional para una IA digna de confianza (OECD, 2019b). Los principios fueron adoptados en mayo de 2019 por los países miembros de la OECD y son la primera norma internacional sobre IA suscrita por los gobiernos. Más allá de los miembros de la OECD, otros países como Argentina, Brasil, Costa Rica, Malta, Perú, Rumania, Ucrania, Singapur y Egipto han adherido a los Principios de la IA y son bienvenidas más adhesiones. En junio de 2019, el G20 adoptó los Principios de IA centrados en el ser humano que se basan en los Principios de IA de la OECD.

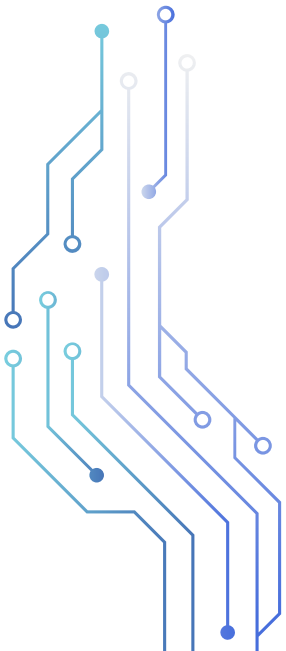
Recuadro 3. Lista de verificación - Conceptualización y diseño

☑ Definición correcta del problema y de la respuesta de la política pública:

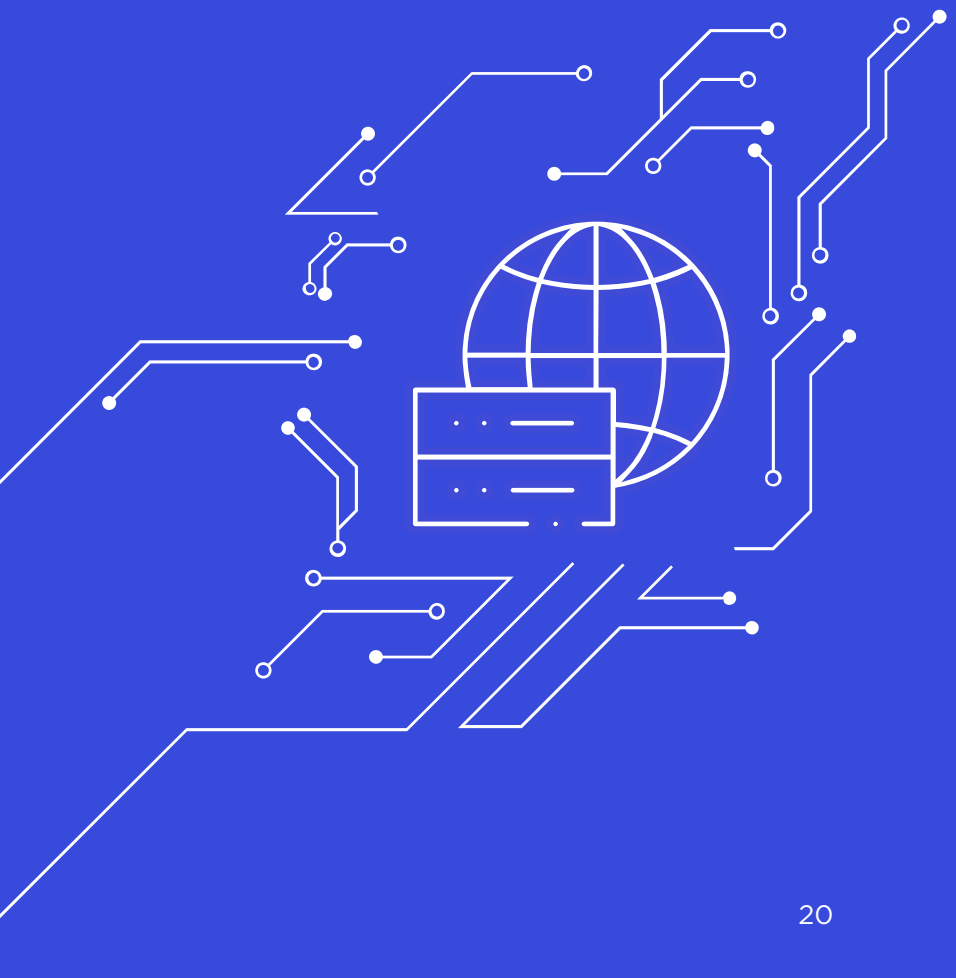
- (Cualitativo) ¿Está claramente definido el problema de política pública?
- (Cualitativo) Describir cómo se aborda actualmente este problema –considerando las respuestas de las instituciones relacionadas– y cómo el uso de la IA mejoraría la respuesta gubernamental a ese problema.
- (Cualitativo) ¿Se identificaron los grupos o los atributos protegidos dentro del proyecto (por ejemplo, edad, género, nivel educativo, raza, nivel de marginación, etc.)?
- (Cualitativo) ¿Se definieron las acciones o intervenciones que se llevarán a cabo en función del resultado del sistema de IA?

☑ Principios de la IA

- (Cuantitativo) ¿Se ha justificado la necesidad de un sistema de IA, teniendo en cuenta otras posibles soluciones que no requieran el uso de datos personales y decisiones automatizadas?
- (Cuantitativo) ¿Existen pruebas de que tanto la acción de las políticas públicas como la recomendación del sistema de IA supondrán un beneficio para las personas y el planeta al impulsar el crecimiento inclusivo, el desarrollo sostenible y el bienestar?
- (Cualitativo) ¿Se han identificado proyectos similares y se han analizado para obtener aprendizajes y establecer errores comunes?
- (Cuantitativo) ¿Se ha considerado la posibilidad de minimizar la exposición de información personal identificable (por ejemplo, anonimizando o no recogiendo información no relevante para el análisis)?



2. RECOLECCIÓN Y PROCESAMIENTO DE DATOS



2. Recolección y procesamiento de datos

No siempre, sin embargo, los datos recolectados tienen una frecuencia, desagregación o cobertura que los haga relevantes, o carecen de la calidad necesaria para utilizarse para la toma de decisiones. Por ejemplo, las encuestas diseñadas mediante muestreo probabilístico especifican por su diseño el tipo de análisis que se puede hacer con ellas, pero este tipo de herramientas suelen levantarse con poca frecuencia y pueden resultar insuficientes para captar el movimiento de los patrones que van a estudiarse. Por otro lado, la información proveniente de registros administrativos o datos provenientes de internet (interacción en redes sociales, visitas y otras medidas en páginas web, etc.) y telefonía (llamadas, ubicación por GPS, etc.) suelen tener una frecuencia mucho mayor, pero en pocos casos cubre a la población en su conjunto, por lo que no es siempre posible utilizarla con el fin de tomar decisiones para toda la población.

Ya sea que se esté implementando un modelo supervisado o no supervisado, los datos de entrenamiento son un punto muy importante de cualquier sistema de ML. La calidad de los datos puede analizarse mediante criterios como volumen, completitud, validez, relevancia, precisión, puntualidad, accesibilidad, comparabilidad e interoperabilidad de distintas fuentes. Definir con precisión estos criterios en general es difícil, pues el contexto de cada problema tiene particularidades sutiles. La relevancia y precisión se refieren a calidad de medición y utilidad para informar la decisión, mientras que la puntualidad alude a que los datos ocurren con la temporalidad necesaria para informar el problema que va a decidirse. Accesibilidad, comparabilidad e interoperabilidad se refieren a que los datos pueden extraerse oportunamente y a que distintas fuentes de datos tienen la congruencia necesaria para aplicarse conjuntamente en el análisis.[8]

En esta sección se abordan dos problemas comunes para los sistemas de aprendizaje automático durante la fase de recolección y procesamiento de datos[9]:

1. Calidad y relevancia de los datos disponibles, y

2. Cualificación y exhaustividad de los datos para la población objetivo.

Las secciones 2.1 y 2.2 abordan algunas de las cuestiones destacadas en los Principios de buenas prácticas para la ética de los datos en el sector público de la OECD en relación con la calidad y la cualificación de los datos. Los Principios de Buenas Prácticas tienen por objeto apoyar a los funcionarios públicos en la aplicación de la ética de los datos en los proyectos, productos y servicios del gobierno digital, de modo que: i) la confianza se sitúe en el centro de su diseño y entrega y ii) la integridad pública se mantenga a través de medidas específicas adoptadas por los gobiernos, las organizaciones públicas y, a un nivel más granular, los funcionarios públicos (OECD, 2021).

2.1 Calidad y relevancia de los datos disponibles

Los algoritmos de aprendizaje automático captan patrones y relaciones observadas a partir de los datos con los que se han entrenado. Su objetivo es identificar estos mismos patrones para nuevos casos no observados durante el entrenamiento del modelo. Por esta razón,

8 En esta etapa se recomienda el llenado del Perfil de datos de la sección de herramientas de este manual.

9 Aunque no se tratan en detalle en esta sección, otras cuestiones relacionadas con los datos –como el dominio y la estructura de los datos– se incluyen en el perfil de datos.

los datos de entrenamiento determinan la forma como se comportará el algoritmo. Sin embargo, los datos disponibles no siempre son ideales para todos los casos de uso. Dos de los principales problemas son:

1. Estados indeseables o subóptimos en datos recolectados.
2. Mala correspondencia entre variables disponibles y variables ideales.

2.1.1 Estados indeseables o subóptimos en datos recolectados

El primer reto es no tomar en cuenta que los datos con los que entrenamos un modelo de ML pueden haber captado estados indeseables del mundo real. Esos "estados indeseables" pueden ser sesgos e inequidades perjudiciales para subgrupos, pero también puede ser cualquier otro patrón que se considere subóptimo o no deseable desde un punto de vista de política social.

Ejemplo

Un caso de este reto se dio en 2015 cuando Amazon experimentó con un sistema de recomendación de recursos humanos a partir de técnicas de aprendizaje supervisado. El modelo entrenaba con una base de datos de los procesos de selección de candidatos de la compañía almacenados durante los diez años anteriores. En esa base de datos se identificaba si un candidato había sido aceptado o rechazado para el trabajo por el departamento. El sistema se basaba en la hipótesis de que el algoritmo podría captar buenos candidatos y reducir el trabajo del departamento de recursos humanos al hacer una primera selección de los candidatos. Lo que el equipo no había tomado en cuenta es que la industria de la tecnología se ha caracterizado por ser predominantemente masculina, por lo que el sistema recomendaba una mayor proporción de hombres, pues más hombres habían sido aceptados en esos puestos históricamente, lo que creaba un sesgo que parecía mostrar que los hombres eran más exitosos, cuando en realidad estaba captando una inequidad.

Recuadro 4. Lista de verificación - Estados indeseables o subóptimos en datos recolectados

- (Cualitativo) Discutir las posibles desigualdades sociales históricas en el caso de uso con especialistas en la materia.
- (Cuantitativo) Realizar un análisis exploratorio de los datos disponibles con los que se entrenará el modelo para identificar posibles sesgos históricos o estados indeseables.

2.1.2 Mala correspondencia entre variables disponibles y variables ideales

Cuando se toman decisiones de política pública, se hacen a partir de la definición de una o varias variables objetivo "ideales" que tiene en mente el tomador de decisiones. Sin embargo, las variables ideales pueden o no estar disponibles en los datos a los que se tiene acceso. En muchas ocasiones es necesario el uso de variables sustitutas o sucedáneas (proxy) que nos ayude a aproximarnos a la variable ideal. Cuando introducimos este tipo de variables dentro de modelos de ML podemos estar aprendiendo sesgos implícitos que pueden no ser

deseables. Por ejemplo, una beca escolar que busque beneficiar a los estudiantes más inteligentes (variable ideal) se encontrará con el problema de definir ese concepto y encontrar una variable que pueda describirlo. Un examen de IQ asigna un valor mediante una prueba estandarizada que se describe como una variable proxy de la inteligencia. Sin embargo, el examen mide únicamente algunas dimensiones de la inteligencia, por lo que subestimará la inteligencia de algunas personas (Wilson, 2014).

Las variables objetivo deben plantearse claramente, aunque sean ideales. Las variables disponibles deben analizarse para entender qué tan adecuadas son para utilizarse como proxy de la variable ideal. Se deben identificar sesgos sistemáticos dentro del contexto de su uso.

Ejemplo

El sistema de salud de Estados Unidos implementó un algoritmo para predecir las necesidades de cuidado médico que necesitaban distintos pacientes. En este caso, el tomador de decisiones de política pública quería una herramienta que le indicara de forma preventiva qué pacientes tenían un alto riesgo de requerir mayores cuidados médicos utilizando la información histórica de los hospitales. Dado que la variable ideal de riesgo de complicación no estaba disponible, utilizaron como variable proxy el gasto en que incurrieron los pacientes durante su enfermedad, en la hipótesis de que personas más enfermas terminarían gastando más en tratamientos médicos para superar la enfermedad. (Obermeyer, Powers, Vogeli, & Mullainathan, 2019) demostraron que este sistema tenía un sesgo racial porque subestimaba el número de pacientes negros con necesidades de atención médica. El sesgo racial se ocasionaba porque esta subpoblación gastaba, en promedio, menos dinero que los pacientes blancos. Al utilizar el gasto como variable proxy de riesgo de complicación los pacientes blancos más saludables parecían requerir más cuidados de salud que pacientes negros más enfermos. En este caso usar el gasto en salud como medida sustituta de necesidad de cuidado médico fue poco apropiado, pues se encontraba sesgada por una variable omitida de desigualdad económica.

Recuadro 5. Mala correspondencia entre variables disponibles y variables ideales

- (Cualitativo) Las variables objetivo ideales deben estar claramente establecidas. Las variables recogidas/disponibles deben analizarse para comprender hasta qué punto son adecuadas para sustituir a la variable objetivo. Deben identificarse los sesgos sistemáticos o la validez de la métrica sustituto.
- (Cualitativo) ¿Se ha justificado claramente el uso de la variable de respuesta seleccionada para los fines de la intervención?

2.2 Cualificación y exhaustividad de los datos para la población objetivo

Los modelos de ML pretenden generar información para tomar acciones o políticas para una población objetivo. La mayor parte del tiempo las fuentes de datos no incluyen a toda la población (como sería el caso de un censo), por lo que es usual que solo se tenga disponible un subconjunto o muestra de la misma (una encuesta, una base de datos administrativa, etc.), a partir de la cual se debe buscar desarrollar extrapolaciones, predicciones o estimaciones que ayuden en la toma de decisiones.

2.2.1 Muestras probabilísticas y naturales

En estadística, una muestra es un subconjunto de casos o individuos de una población. En ella existen dos extremos posibles:

1. **Muestreo probabilístico:** se llama así a una muestra en donde los casos se seleccionan a partir de un diseño probabilístico. Por ejemplo: muestra aleatoria simple, estratificada, por conglomerados, etc. En este caso, todas las predicciones y estimaciones que pretende aplicarse a la población objetivo pueden evaluarse en cuanto a su precisión con garantías probabilísticas. Es decir, podemos dar rangos de error para estimaciones de cantidades asociadas a toda la población objetivo.

   Por ejemplo: una encuesta nacional de hogares con diseño probabilístico generalmente consiste en una definición de estratificación, unidades de selección aleatoria a distintos niveles (unidades primarias, secundarias, etc.). Cada hogar se selecciona con una probabilidad conocida. Aunque la muestra se diseñe de manera no representativa (por ejemplo, más hogares en zonas rurales o de ingresos bajos), es posible hacer inferencias para toda la población con ciertas garantías acerca del tamaño de error de estimación.

2. **Muestras naturales (no probabilísticas):** por otro lado, una muestra natural o no probabilística se da cuando los casos no se seleccionan en forma aleatoria, sino por un proceso natural mal o parcialmente conocido. En este caso, no es posible saber qué va a pasar cuando se aplique una política que resulte de un modelo en la población general y no puedan construirse rangos de error de predicciones y estimaciones mediante métodos estadísticos que tengan garantías probabilísticas. Es decir, las cantidades y predicciones estimadas tienen error desconocido, los modelos y características útiles en la muestra pueden no aplicar en la población objetivo, y la situación puede agravarse para grupos protegidos subrepresentados. (Williams, 1981), por ejemplo, muestra que valores predictivos de anemia pueden ser distintos para diferentes grupos raciales y que predicciones desarrolladas para un grupo pueden tener desempeño pobre en otro.

   Un caso usual de este tipo de muestras se da cuando por el canal de captación de la información se excluyen subgrupos particulares de población (sesgo de selección). Por ejemplo, con aplicaciones donde la población que no tenga acceso a Internet o a un teléfono inteligente será excluida. Este es el caso de la información proveniente de redes sociales, registros de llamadas telefónicas, etc.

   Las muestras naturales de datos pueden dar lugar a:

   - Errores o sesgos de estimación y/o predicción.
   - Estructuras predictivas diferentes de las que observaríamos en la población objetivo (modelos no válidos).
   - Extrapolaciones que no están respaldadas por los datos.
   - Subrepresentación o sobrerrepresentación de subconjuntos de la población.

El muestreo probabilístico sería la situación ideal para la mayoría de los proyectos de aprendizaje automático. En este caso, puede entenderse exactamente qué subpoblaciones se muestrearon, a qué tasas y cómo se relacionan esas tasas con las tasas poblacionales. El diseño de la muestra determina el alcance inferencial. Sin embargo, tener una muestra probabilística no es siempre posible.

Esto no quiere decir que las muestras naturales no sean útiles; en muchas ocasiones son la única fuente de datos disponible para la toma de decisiones. Sin embargo, es importante entender de dónde provienen los datos para poder tomar en cuenta sus limitantes e identificar los riesgos implícitos al tomar decisiones para toda la población.

Un caso típico son las muestras de datos que provienen de redes sociales en las que la composición demográfica de los usuarios difiere sustancialmente de la población general. Un estudio para el Reino Unido descubrió que, en promedio, los usuarios de Twitter y Facebook son considerablemente más jóvenes que la población general y tienen más probabilidades de tener niveles de educación más altos que los no usuarios (Prosser & Mellon, 2016). Cualquier estudio con estos datos debe explicar cómo estas particularidades pueden afectar los resultados.

Algo importante que hay que tomar en cuenta es que tener muestras equilibradas en términos de características de la población no es tampoco una condición ni necesaria ni suficiente para calificar como apropiada la base de datos para la construcción de modelos de ML. Por ejemplo, en el caso de información recolectada por redes sociales, el hecho de que se tenga una muestra que contenga 50 % hombres y 50 % mujeres no dice nada sobre el tipo de conclusiones que pueden extraerse con esos datos porque la selección de esas observaciones, al no darse mediante un proceso probabilístico, podría presentar un sesgo en alguna otra dimensión y no necesariamente generalizarse a la población total.

Recuadro 6. Lista de verificación - Muestras probabilísticas y naturales

- (Cualitativo) ¿Se han analizado las posibles diferencias entre la base de datos y la población para la que se está desarrollando el sistema de IA? (Utilice la bibliografía relacionada con el tema y la información de los expertos. Estudie en particular los sesgos de selección no medidos).
- (Cuantitativo) Aunque los modelos pueden construirse con diversas fuentes de datos, diseñadas o naturales, lo ideal es que la validación se realice con una muestra que permita la inferencia estadística a la población. La muestra de validación debe cubrir adecuadamente la población objetivo y las subpoblaciones de interés.

2.2.2 Atributos faltantes o incompletos

Muchos proyectos de aprendizaje automático están destinados a fallar por la poca calidad de los datos con los que se cuenta. Cuando se recolectan datos del mundo real a través de muestras no probabilísticas es muy común que algunas observaciones tengan datos faltantes, es decir, observaciones para las que no se tienen todos los atributos.

Los atributos faltantes o incompletos son un fenómeno que puede tener un efecto significativo en las conclusiones extraíbles de los datos. Por un lado, cuando información crucial acerca de las unidades es totalmente desconocida, esto puede resultar en modelos de desempeño pobre, con poca utilidad para la toma de decisiones y, por otro lado, la ausencia de información puede también estar asociada a características relevantes de las unidades para las que se quiere predecir.

Cuando existen observaciones faltantes es posible implementar distintos métodos de imputación, pero es importante explorar las razones o el "mecanismo de censura" por el que una observación puede tener valores faltantes. En la literatura existen tres principales supuestos (Rubin, 2002):

- **Valores faltantes completamente aleatorios (Missing Completely at Random - MCAR):** se da cuando la probabilidad de faltar es la misma para todas las observaciones. Es decir, la censura o falta se produce totalmente al azar.
- **Valores faltantes aleatorios (Missing at Random - MAR):** se da cuando los valores faltantes no dependen de los valores que toma esa variable, pero sí existe una relación entre los valores faltantes y otros datos observados del individuo.
- **Valores faltantes no aleatorios (Missing Not at Random - MNAR):** se da cuando los valores faltantes dependen de los valores que toma esa variable o de datos no observados. Por ejemplo, es un fenómeno conocido que cuando se levantan encuestas de ingreso autorreportado las personas con mayor ingreso tienden a no revelarlo.

Recuadro 7. Atributos faltantes o incompletos

- (Cualitativo) ¿Se ha realizado un análisis de valores faltantes y de variables omitidas?
- (Cualitativo) ¿Se ha identificado si existen variables omitidas importantes para las cuales no se cuenta con mediciones asociadas (en caso de existir)?
- (Cualitativo) ¿Se ha identificado las razones por las que existen observaciones faltantes (en caso de existir)?
- (Cuantitativo) Los procesos de imputación tienen que evaluarse en cuanto a su sensibilidad a supuestos y datos. De preferencia, deben utilizarse métodos de imputación múltiple que permitan evaluar incertidumbre en la imputación (Little & Rubin, 2002), (Buuren & Groothuis-Oudshoorn, 2011).

2.3 Comparación causal

Cuando los humanos racionalizan el mundo intentan comprenderlo en términos de causa y efecto: si entendemos por qué ocurrió algo, podemos alterar nuestro comportamiento para cambiar resultados futuros.

Un modelo de ML nos puede dar resultados que parecerían describir relaciones causales sin que necesariamente lo sean. Si la política se aplica en función de hallazgos en términos de las variables incluidas en el modelo, la derivación de políticas a partir de esos modelos puede llevar a decisiones erróneas.

Técnicas econométricas, como los experimentos aleatorios controlados o RCTs (randomized controlled trials), experimentos naturales, diferencia en diferencias y variables instrumentales se utilizan con estos objetivos para controlar fenómenos como sesgo por selección o endogénesis por variables omitidas, entre otros. En los últimos años trabajos como Athey (2018) han comenzado a introducir en algoritmos de ML estas técnicas y procesos experimentales tipo A/B testing han empezado a utilizarse de forma masiva en contextos digitales por la facilidad de crear experimentos masivos en Internet. Sin embargo, en la mayoría de los casos los algoritmos de ML no buscan describir relaciones causales y es necesario ser muy cuidadosos con este tipo de uso (Stuart, 2008).

Recuadro 8. Lista de verificación - Comparación causal

- (Cualitativo) Comprender y describir las razones por las que la variable de respuesta está correlacionada con variables conocidas y desconocidas. Describir los posibles sesgos basados en el conocimiento y el análisis de los expertos.
- (Cualitativo) En caso de que no se haya trabajado para asegurar la causalidad en los resultados, ¿se comunicaron explícitamente las limitaciones de los resultados al responsable de las políticas públicas?
- (Cuantitativo) En caso de que se intente la inferencia causal con modelos deben describirse las hipótesis, consideraciones o métodos utilizados para apoyar una interpretación causal. Deben realizarse y documentarse las comprobaciones de robustez.

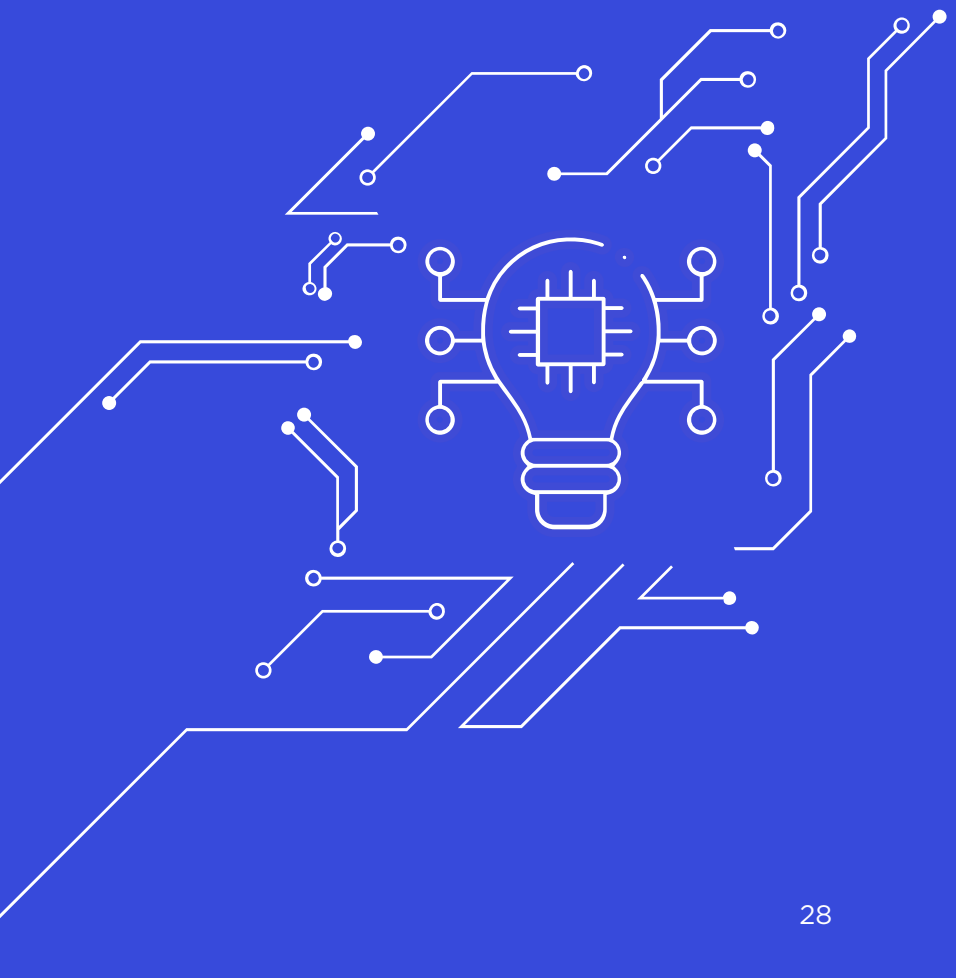
Actividad:

Se recomienda el llenado del Perfil de datos durante la fase de conocimiento y preparación de datos del ciclo de vida de IA (ver Herramienta 2).
Al terminar esta fase se recomienda el llenado de la sección de fuente y manejo de datos del Perfil del modelo y llevar a cabo una discusión con el tomador de decisiones de política pública.

VER HERRAMIENTA 2

VER HERRAMIENTA 3

3. DESARROLLO DEL MODELO Y VALIDACIÓN



3. Desarrollo del modelo y validación

El proceso de desarrollo de un modelo conlleva muchas decisiones que tienen implicaciones en sus resultados. Algunas decisiones pueden llevar a cometer errores metodológicos que generen sesgos o que eviten que el sistema generalice en forma adecuada. Entre estos encontramos fugas de información, sobreajuste y subajuste.

Además, hay otro grupo de decisiones que no son necesariamente problemas metodológicos que pueden cambiar sustancialmente la forma como se comporta el sistema: ¿cómo elegir entre dos modelos?, ¿qué tipo de errores reportar?, ¿qué definición de justicia algorítmica se elegirá? Al comienzo del manual se comentó que ninguna de estas preguntas tiene sentido fuera del contexto de la aplicación específica. Lo que sí es posible es crear un marco de entendimiento de estos errores para que puedan ser discutidos entre los equipos técnicos y los tomadores de decisiones de política pública.

En esta sección del manual se exponen retos que aparecen durante los procesos de entrenamiento y validación de los sistemas de soporte y toma de decisión. En este caso la mayoría de los errores se deben a fallos metodológicos en la evaluación y a no plantear en forma correcta el objetivo de ajuste del sistema o las métricas que se busca optimizar.

3.1 Ausencia o uso inadecuado de muestras de validación

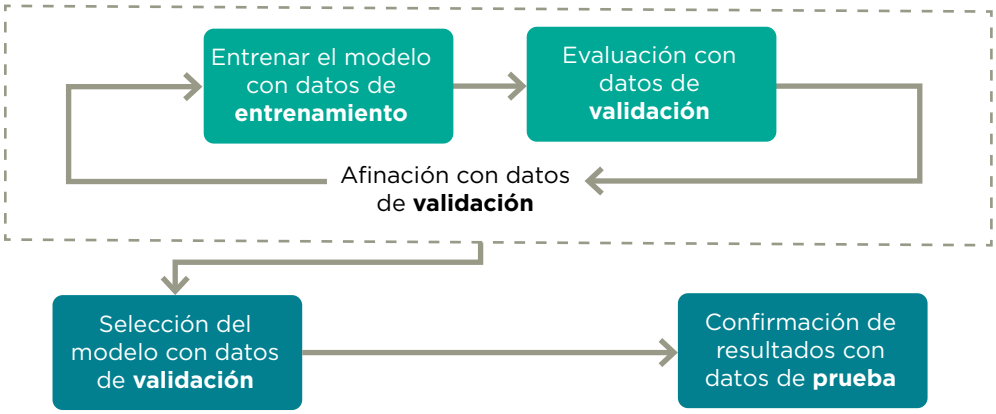
Los modelos de aprendizaje automático se entrenan principalmente para crear predicciones en casos no observados. De nada sirve evaluar un sistema en su desempeño de predicción de las observaciones con las que se entrenó, pues el sistema podría únicamente memorizar cada respuesta.[10] Su utilidad se encuentra en la medida en la que el sistema logra generalizar un aprendizaje para predecir con datos fuera del conjunto de entrenamiento (out-of-sample). La validación generalmente involucra al menos dos muestras (1 y 2), y de preferencia tres:

1. **Datos de entrenamiento:** subconjunto de los datos utilizados para entrenar el modelo.
2. **Datos de validación:** subconjunto de los datos con los que se evalúa el entrenamiento en forma iterativa.
3. **Datos de prueba:** subconjunto de los datos que deben mantenerse ocultos hasta después de seleccionar el modelo y que son usados para confirmar los resultados.

Para evitar que una partición aleatoria en datos de entrenamiento y validación favorezca o perjudique la evaluación, en general se hace una validación cruzada. Esta consiste en dividir los datos en *k* pedazos, calculando el promedio de k evaluaciones, donde los datos de validación son cada uno de los pedazos y los *k*-1 restantes son los datos de entrenamiento. Esto en inglés se llama *k*-fold evaluation y normalmente se escoge *k*=5 o *k*=10.

10 Este fenómeno está relacionado con el sobreajuste que se verá más adelante.

Figura 3. Etapas de evaluación



Fuente: Construcción propia

El primer reto es no tener un proceso de validación apropiado o que incluso sea inexistente. En este caso, los resultados del modelo se presentarían únicamente con el desempeño del conjunto de datos de entrenamiento. Las métricas de desempeño de este conjunto no deberían utilizarse como indicador del potencial comportamiento del modelo para casos nuevos, pues se podría estar sobreestimando su desempeño.

Una validación exitosa también se relaciona con los criterios de calidad, tales como la completitud y representatividad de la información que se vio en el capítulo anterior, porque si la población objetivo es distinta a la representada por datos utilizados durante el entrenamiento, aunque el proceso de evaluación se haya realizado en forma correcta, es posible tener un comportamiento completamente distinto.

Recuadro 9. Lista de verificación - Ausencia o uso inadecuado de muestras de validación

- (Cuantitativo) ¿Se construyeron las muestras de validación y prueba adecuadamente, considerando un tamaño apropiado que cubra a subgrupos de interés y protegidos y evite fugas de información durante su implementación?
  - La construcción de la muestra de validación debe producirse según un diseño muestral que permita inferencia a la población objetivo (Lohr, 2009).
  - La muestra de validación debe cubrir a subgrupos de interés y protegidos, de manera que sea posible hacer inferencia a sus subpoblaciones. Eso incluye tamaños de muestras adecuados según metodología de muestreo (Lohr, 2009).
  - Si no está disponible tal muestra, es indispensable un análisis de riesgos y limitaciones de la muestra natural, conducida por expertos y personas que conozcan el proceso que generó esos datos muestrales.

3.2 Fugas de información

La fuga de datos se produce cuando el modelo observa información adicional al conjunto de entrenamiento (Kaufman, Rosset, & Perlich, 2011). Esta información adicional modifica el proceso de aprendizaje y pone en duda la validación del modelo como forma de estimar el rendimiento de producción del sistema.

Esto ocurre de dos maneras:

- **Contaminación entrenamiento-validación:** la muestra de entrenamiento recibe fugas de los datos de validación, lo que implica el uso de datos de validación en entrenamiento e invalida la estimación del error de predicción.

- **Fugas de datos no disponibles en la predicción:** muestras de validación y entrenamiento tienen agrupaciones temporales o de otro tipo que no se conservan en el proceso de entrenamiento y validación. En este caso, entrenamiento y validación reciben fugas de información que no estará disponible en el momento de hacer predicciones.

3.2.1 Contaminación entrenamiento-validación

La contaminación entrenamiento-validación se produce cuando todas o una parte de las muestras de validación o de prueba se utilizan para la construcción de los modelos durante el entrenamiento. Este error suele dar lugar a niveles de rendimiento poco realistas en el conjunto de validación, pues el modelo está haciendo predicciones basadas en observaciones que ha visto anteriormente.

Este error suele producirse al aplicar metodologías de preprocesamiento que agregan y comparten información de la composición de la base de datos a las observaciones individuales: por ejemplo, escalando una variable, creando covariables con promedios o recuentos, metodologías de sobre o submuestreo, etc. Estos procesos deben realizarse después de dividir el conjunto de datos de entrenamiento y validación.

Recuadro 10. Lista de verificación - Contaminación entrenamiento-validación

- (Cuantitativo) Cualquier procesamiento y preparación de datos de entrenamiento debe evitar usar los datos de validación o prueba de alguna manera. Debe mantenerse una barrera sólida entre entrenamiento vs. validación y prueba. Esto incluye recodificación de datos, normalizaciones, selección de variables, identificación de datos atípicos y cualquier otro tipo de preparación de cualquier variable que vaya a incluirse en los modelos. Esto incluye también ponderaciones o balance de muestras basadas en sobre/sub muestreo.

3.2.2 Fugas de datos no disponibles en la predicción

Este error se produce cuando se entrena un modelo con información que no estará disponible de la misma manera o con la misma calidad cuando el modelo se ponga en producción. Generalmente, tiene que ver con la temporalidad de los datos o las agrupaciones. En casos más sutiles, este error puede ser difícil de detectar, porque la variable está presente, pero la información se actualiza en forma retroactiva.

Un ejemplo de esto puede verse en las estadísticas de delincuencia y mortalidad. Las denuncias de un robo pueden tardar en aparecer en las bases de datos de las autoridades debido a procesos burocráticos o administrativos, y la incidencia observada en un periodo podría aumentar sistemáticamente con el paso del tiempo. En este ejemplo, la variable objetivo está disponible en la producción, pero puede no estar completa debido a ciertos desfases inherentes a la notificación. Si no se tiene en cuenta este fenómeno durante el entrenamiento y se utilizan datos que ya están completos, la evaluación del modelo puede parecer precisa, pero en producción la precisión de los datos se degradará considerablemente.

Recuadro 11. Fugas de datos no disponibles en la predicción

El esquema de validación **debe replicar tan cerca como sea posible** el esquema con el cual se aplicarán las predicciones. Esto incluye que hay que replicar:

- Ventanas temporales de observación y registro de variables y ventanas de predicción.
- Si existen grupos en los datos, considerar si habrá información disponible de cada grupo cuando se haga la predicción, o si será necesario predecir para nuevos grupos.

3.3 Modelos de clasificación: probabilidades y clases

En aprendizaje automático, los algoritmos de clasificación supervisada son sistemas cuyo objetivo es asignar una categoría o etiqueta de clase a las nuevas observaciones. Se denomina clasificación binaria cuando la variable objetivo tiene dos clases (por ejemplo, clasificar un correo electrónico como spam o no spam), y clasificación multiclase cuando hay más de dos clases (por ejemplo, algoritmo de identificación de especies vegetales).

3.3.1 Datos desbalanceados

En un problema de clasificación, un conjunto de datos desbalanceados se produce cuando la distribución de las observaciones entre las clases conocidas no está distribuida por igual. Estos tipos de conjuntos de datos tendrán una o más clases con muchos ejemplos, denominadas clases mayoritarias, y una o más clases con menos observaciones, denominadas clases minoritarias (por ejemplo, grupos con menos de 1 % de las observaciones totales). Estos últimos grupos presentan dificultades considerables para los modelos de predicción, pues puede haber poca información sobre ellos.

En datos muy desbalanceados, los predictores de clase pueden tener un rendimiento pobre (por ejemplo, nunca predicen la clase minoritaria), aunque las medidas de rendimiento sean buenas. En particular, si siempre predecimos la clase mayoritaria, la precisión será igual al porcentaje de elementos de esta clase.

Ejemplos

- **Considere que se tiene un millón de datos: 999.000 negativos y 1000 positivos. Puede ser buena idea submuestrear los negativos por una fracción dada (por ejemplo, 10 %) ponderando cada caso muestreado por 10 en el ajuste y el posproceso.**
- **Considere que se tiene un millón de datos: 999.950 negativos y 50 positivos. Puede ser imposible discriminar apropiadamente los 50 datos positivos. Construir conjuntos de validación empeora la situación: no es posible validar el desempeño predictivo ni construir un modelo que tenga buen desempeño.**

Recuadro 12. Lista de verificación – Datos desbalanceados

- (Cuantitativo) Hacer **predicciones de probabilidad** en lugar de clase. Estas probabilidades pueden ser incorporadas al proceso de decisión posterior como tales. Evitar puntos de corte estándar de probabilidad como 0.5, o predecir según máxima probabilidad.
- (Cuantitativo) Cuando el número absoluto de casos minoritarios es muy reducido, puede ser muy difícil encontrar información apropiada para discriminar esa clase. Se requiere **recolectar más datos de la clase minoritaria**.
- (Cuantitativo) Submuestrear la clase dominante (ponderando hacia arriba los casos para no perder calibración) puede ser una estrategia exitosa para reducir el tamaño de los datos y el tiempo de entrenamiento sin afectar el desempeño predictivo.
- (Cuantitativo) Replicar la clase minoritaria, para balancear mejor las clases (sobremuestreo).
- (Cuantitativo) Algunas técnicas de aprendizaje automático permiten ponderar cada clase por un peso distinto para que el peso total de cada clase quede balanceado. Si esto es posible, es preferible a sub o sobremuestreo.

3.3.2 Puntos de corte arbitrarios

En los problemas de clasificación para la toma de decisiones se recomienda utilizar algoritmos de clasificación probabilística en lugar de clasificar la observación solo con su clase más probable. El resultado de un algoritmo de clasificación probabilística es una distribución de probabilidad sobre el conjunto de clases. Estos métodos pueden proporcionar información al responsable de toma de decisiones sobre la incertidumbre relativa a la clasificación.

Para tomar la decisión sobre si la observación debe clasificarse como positiva o negativa, el equipo técnico debe elegir el umbral a partir del cual la observación se clasifica como perteneciente a cada clase. A menudo se acepta erróneamente un punto de corte de 0,5 para las clasificaciones binarias, pues es el valor por defecto de muchos modelos de aprendizaje automático. Esta decisión puede tener importantes implicaciones si se toma fuera del contexto del problema en cuestión, por lo que es importante que se discuta y se seleccione teniendo en cuenta los tipos de errores y sus implicaciones.

Recuadro 13. Lista de verificación - Puntos de corte arbitrarios

- (Cuantitativo) El uso de **algoritmos de clasificación probabilística** es más adecuado para que la toma de decisiones incorpore la incertidumbre con respecto a la clasificación.
- (Cuantitativo) Evitar los puntos de corte de probabilidad estándar, como el 0,5. Elegir una interpretación óptima de las probabilidades predichas, analizando las métricas de error.

3.3.3 Idoneidad de las métricas de evaluación

En problemas de clasificación, los puntos de corte se toman con criterios relacionados con el contexto de la decisión. La mayoría se construye mediante el análisis de la matriz de confusión de clasificación.

Tabla 1. Matriz de confusión

		Real	
		Positivo	**Negativo**
Predicho	**Positivo**	**Verdadero Positivo (VP)**	**Falso Positivo (FP)**
	Negativo	**Falso Negativo (FN)**	**Verdadero Negativo (VN)**

Los errores de un modelo de clasificación pueden dividirse en falsos positivos y falsos negativos. Un falso positivo es una observación para la que el modelo predice incorrectamente la clase positiva. Y un falso negativo es una observación para la que el modelo predice incorrectamente la clase negativa. Estas métricas pueden combinarse de diferentes maneras en función del caso de uso y del objetivo de la política social. Las métricas más utilizadas son:

1. **Exactitud (Accuracy):** una de las métricas más utilizadas para evaluar los modelos de clasificación es la fracción de las predicciones que el modelo tuvo correctas:

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN}$$

2. **Precisión:** fracción de aquellos clasificados como positivos por el modelo que en realidad eran positivos:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

3. **Sensibilidad (Recall):** fracción de positivos que el modelo clasificó correctamente:

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

4. **Especificidad:** fracción de negativos que el modelo clasificó correctamente:

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

Es necesario tener en cuenta el contexto del problema a la hora de definir los criterios para evaluar los modelos de clasificación. Por ejemplo, si el modelo clasifica la prevalencia de una enfermedad mortal, el costo de no diagnosticar (falso negativo) es mucho mayor que el costo de enviar a una persona sana a hacerse más pruebas (falso positivo). En otras palabras, dependiendo de la aplicación, el costo de los falsos negativos puede ser muy diferente del costo de los falsos positivos. Por ello, se recomienda el uso del análisis costo-beneficio, porque compara el resultado del modelo en el contexto de la toma de decisiones.

Estos criterios también pueden ser engañosos en función de la composición de la base de datos de entrenamiento y evaluación. Por ejemplo, cuando se utilizan datos desbalanceados,

una precisión de 95 % puede significar en realidad un rendimiento inferior del modelo. Las soluciones parciales a este problema incluyen el uso de medidas que combinan la precisión y la sensibilidad, como la métrica F1 o la curva *Precision-Recall*, que pueden ayudar a analizar el equilibrio entre los verdaderos positivos y los falsos positivos en el contexto de la aplicación.

5. Recuadro 14. Lista de verificación - Idoneidad de las métricas de evaluación

- (Cualitativo) ¿Se cuestionaron las implicaciones de los diferentes tipos de errores para el caso de uso específico, así como la forma correcta de evaluarlos?
- (Cualitativo) ¿Se explicó en forma clara las limitantes del modelo, identificando tanto los falsos positivos como falsos negativos y las implicaciones que una decisión del sistema tendría en la vida de la población beneficiaria?
- (Cuantitativo) ¿Se implementó un análisis costo-beneficio del sistema contra el statu quo u otras estrategias de toma/soporte de decisión (cuando es posible)?

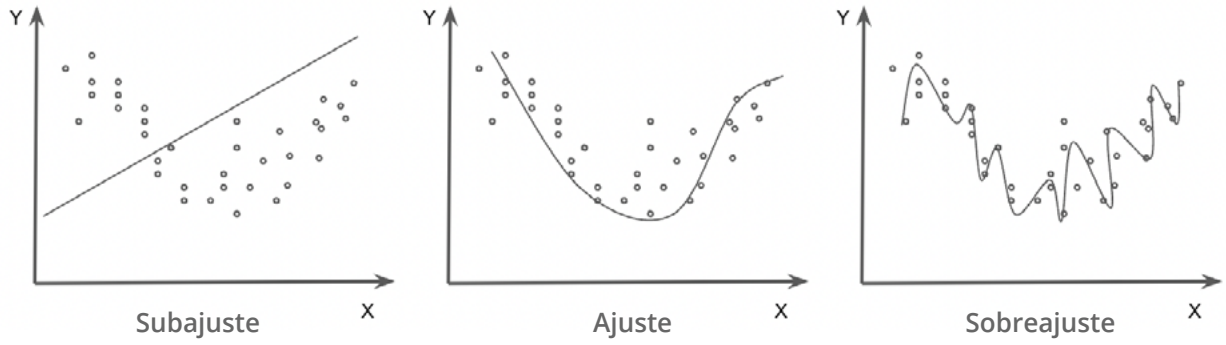
3.4 Sub y sobreajuste

Sub y sobreajuste ocurren cuando la información predictiva en los datos se usa de manera poco apropiada para el objetivo final del aprendizaje automático, que es la generalización del aprendizaje y su uso en conjuntos de datos no observados en el entrenamiento.

- El **sobreajuste** se produce cuando el modelo memoriza las particularidades de los datos de entrenamiento, pero es incapaz de generalizar a ejemplos no vistos. Un modelo demasiado complejo para los datos disponibles tiende a captar características no informativas como parte de la estructura predictiva. Esto se refleja a menudo en un modelo que funciona muy bien con los datos de entrenamiento, pero que tiene un rendimiento pobre en el conjunto de datos de validación.

- El **subajuste** se produce cuando el modelo es incapaz de funcionar bien con los datos de entrenamiento o de generalizar con los nuevos datos. Esto ocurre cuando las características individuales de las observaciones se agrupan en exceso y se les da poca importancia. Un modelo poco ajustado tiende a ignorar los patrones de la estructura predictiva. Esto se refleja en errores sistemáticos e identificables, por ejemplo, sub/sobre predicción sistemática para ciertos grupos o valores de las variables de entrada.

Figura 4. Sub y sobreajuste



Fuente: Preparado por los autores

Recuadro 15. Lista de verificación - Sub y sobreajuste

- (Cuantitativo) Sobreajuste: debe evitarse modelos cuya brecha validación - entrenamiento sea grande (indicios de sobreajuste). De ser necesario, debe afinarse métodos para moderar el sobreajuste como regularización, restricción del espacio funcional de modelos posibles, usar más datos de entrenamiento o perturbar los datos de entrenamiento, entre otros (Hastie, Tibshirani, & Friedman, 2017).
- (Cuantitativo) Subajuste: deben revisarse subconjuntos importantes de casos (por ejemplo, grupos protegidos) para verificar que no existan errores sistemáticos indeseables.

3.5 Errores no cuantificados y evaluación humana

En muchos casos, existirán aspectos del modelo que no son medidos por las métricas de desempeño que se han escogido.

Por ejemplo, en un sistema de búsqueda en documento que, aunque tenga buen desempeño de validación en las métricas, seleccione documentos que tiendan a ser demasiado cortos, produzcan resultados poco útiles o imparciales para búsquedas particulares, o prefiera documentos de tipo promocional o propagandístico. Las razones pueden ir desde errores de preprocesamiento (algunos atributos mal calculados) hasta la selección de atributos para hacer las predicciones que consideran solo una parte del problema.

3.5.1 Fallas no medidas por el modelo

Algunos algoritmos producen resultados de baja calidad que escapan a la evaluación de las métricas de validación. Estos modelos pueden tener un bajo rendimiento cuando se ponen en producción. Las razones para ello son, entre otras, las siguientes:

- Errores de preprocesamiento en el momento de calcular predicciones.
- Tratamiento de los datos que excluyen métricas importantes para hacer predicciones de calidad o no injustas.
- Ausencia de métricas que midan cierto tipo de errores particulares graves.

Este puede ser un problema difícil, pues por su naturaleza son errores no visibles o medibles directamente. Es necesario descubrir estos sesgos o errores fuera del contexto técnico de evaluación y, de ser posible, incluir métricas adicionales de evaluación que consideren estos problemas.

Recuadro 16. Lista de verificación - Fallas no medidas por el modelo

- (Cualitativo) ¿Se realizó una evaluación humana con expertos del caso de uso para buscar sesgos o errores conocidos? (Por ejemplo, pueden usarse paneles de revisores que examinen predicciones particulares y consideren si son razonables o no. Estos paneles deben balancearse en cuanto al tipo de usuarios que se prevén, incluyendo tomadores de decisiones, si es necesario).

3.6 Equidad y desempeño diferencial de predictores

Métodos basados en aprendizaje automático pueden producir resultados injustos o discriminatorios para subgrupos (Buolamwini & Gebru, 2018) (Barocas & Selbst, 2014) (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016). Estos resultados pueden ocasionarlos todos los retos antes mencionados, tanto de la fuente y manejo de los datos como de errores en el diseño del modelo.

Ejemplos de desempeño diferencial e inequidad incluyen distintas tasas de aceptación para recibir beneficios en distintos grupos o errores de detección en rostros humanos que son diferentes dependiendo de la raza.

Es importante recordar que la evaluación de los resultados de un sistema de toma/soporte de decisiones se realiza teniendo en cuenta los objetivos del tomador de decisiones, que pueden ser distintos e incluso contradictorios a los objetivos desde el punto de vista del problema de aprendizaje automático. Por ejemplo, un tomador de decisiones podría sacrificar el desempeño global de un modelo para mejorar el desempeño del modelo en un subgrupo, aunque este subgrupo sea pequeño en comparación con la población en su conjunto (por ejemplo, una acción afirmativa para corregir alguna discriminación social existente).

Aunque el análisis de las implicaciones éticas en los modelos de aprendizaje automático y la relación que estas tienen con una definición de justicia es aún un campo de estudio abierto, existe una importante literatura que busca implantar definiciones matemáticas de equidad en los modelos para describir su imparcialidad o discriminación entre subgrupos y tomar decisiones que mitiguen resultados no deseados.

3.6.1 Definición de justicia y equidad algorítmicas

Lo que se entiende por "justicia" puede cambiar según una cultura o tradición y también puede ser específico de un proyecto o problema de política pública. Por ejemplo, en algunos casos las políticas buscan la inclusión social a través de acciones afirmativas –como las cuotas de diversidad y las políticas de reparación– mientras que en otros casos estas políticas se basan simplemente en argumentos regionales o territoriales. Estos criterios deben integrarse en el proceso de diseño, en el análisis de los datos de formación, durante el proceso de evaluación de errores y en la salida del sistema. Este proceso puede dividirse en dos importantes etapas:

- **Definición de justicia algorítmica:** representación matemática de una definición de justicia específica que se incorpora en el proceso de ajuste y selección del modelo. Es importante tomar en cuenta que estas definiciones pueden ser excluyentes, es decir, satisfacer una podría implicar no satisfacer las demás (Verma & Rubin, 2018).

- **Inequidad algorítmica:** fallas técnicas en los modelos que producen disparidad de resultados para grupos protegidos que deben evaluarse según la definición de justicia algorítmica determinada en el punto anterior (podría ser más de una).

El objetivo del modelador es establecer lineamientos para evitar que deficiencias en los modelos produzcan disparidades indeseables según los distintos subgrupos asociados a una variable protegida (por ejemplo, género, raza o nivel de marginación). Para ello, es necesario seleccionar de antemano una definición de justicia algorítmica. Algunas de las más utilizadas son las siguientes (aunque pueden definirse otras, dependiendo del problema particular y los objetivos de los tomadores de decisiones):

1) Omisión de variables protegidas

Una estrategia muy cuestionada para prevenir disparidades entre los grupos de una variable protegida es ignorar la variable . En este proceso se pretende eliminar la posibilidad de disparidad no incluyendo la variable en el proceso de construcción de predictores. Este enfoque NO resuelve el problema porque:

- Típicamente existen otros atributos asociados a que pueden producir resultados similares, aunque no se considere (por ejemplo, zona geográfica o código postal y nivel socioeconómico).

- Puede haber razones importantes para incluir en los modelos predictivos. Por ejemplo, en el caso de presión arterial, existen variaciones en los grupos raciales () en cuanto a predisposición a presión alta (Lackland, 2014); por lo tanto, un modelo que evalúe riesgo sería más preciso y adecuado si incluye la variable .

(2) Paridad demográfica

La paridad demográfica establece, por su parte, que la proporción de cada segmento de una clase protegida (por ejemplo, el género o ciertos rangos de edad) debe obtener un resultado positivo en una misma proporción (como, por ejemplo, la asignación de becas escolares). Esto es poco deseable por sí mismo: por ejemplo, si quisiéramos construir un clasificador para cierta enfermedad, necesitaríamos considerar que es posible que mujeres y hombres fueran afectados de manera distinta. Sin embargo, la paridad demográfica puede ser un objetivo de los tomadores de decisiones y eso debe tenerse en cuenta en el momento de tomar la decisión asociada a la predicción.

(3) Equidad de posibilidades

El concepto de equidad de posibilidades (Hardt, 2016) es uno menos dependiente de los objetivos de los tomadores de decisiones; se refiere al desempeño predictivo a lo largo de distintos grupos definidos por una variable protegida (Verma & Rubin, 2018). Si es la variable que queremos predecir, e es nuestra predicción, decimos que nuestra predicción satisface equidad de posibilidades cuando y son independientes, dado el valor verdadero.

Esto quiere decir que no debe influir en la predicción cuando conocemos el valor verdadero o, dicho de otra manera, la pertenencia o no pertenencia al grupo protegido A no debe influir en el resultado de la clasificación.

Se considera entonces que predictores que se alejan mucho de este criterio son susceptibles de incluir disparidades asociadas a la variable protegida A. Una implicación de este criterio es que, bajo el supuesto de equidad de posibilidades, las tasas de error predictivo sobre cada subgrupo de A son similares y para clasificación binaria las tasas de falsos positivos y de falsos negativos son similares.

Por ejemplo, supongamos que quiere crearse un sistema para la selección de beneficiarios de una beca escolar para una universidad reconocida. La institución define como variable protegida la pertenencia a una comunidad indígena (que supondremos en este caso que toma dos valores: se autodenomina indígena o no se autodenomina indígena). El predictor satisface equidad de posibilidades cuando tanto la tasa de falsos positivos como la de falsos negativos son iguales para personas indígenas como para personas que no lo son.

(4) Justicia contrafactual

Esta medida considera que un predictor es "justo" si su resultado sigue siendo el mismo cuando se toma el valor del atributo protegido y se cambia a otro valor posible del atributo protegido (como, por ejemplo, introducir un cambio de raza, género u otra condición).

En la práctica no existe una respuesta única ni una medida de justicia algorítmica que funcione para todos los problemas. En la mayoría de los casos buscar el cumplimiento de una implica no cumplir totalmente con las demás, por lo que su elección debe hacerse en el contexto del problema y deben justificarse sus razones. Equidad de oportunidad muchas veces es un criterio aceptable, que introduce criterios de justicia algorítmica permitiendo también optimizar otros resultados deseables.

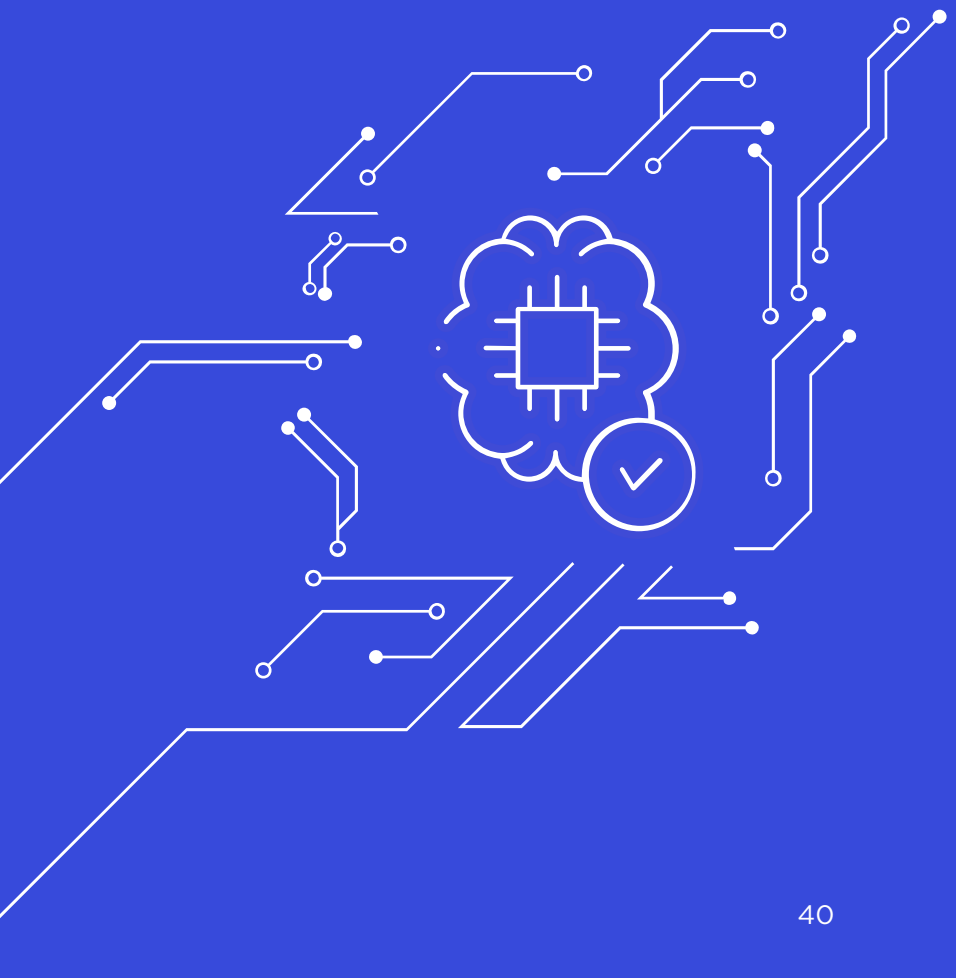
Recuadro 17. Lista de verificación - Definición de justicia y equidad algorítmicas

- (Cualitativo) Identificar grupos o atributos protegidos. Por ejemplo: edad, género, raza, nivel de marginación, etc.
- (Cualitativo) ¿Se definió con expertos y tomadores de decisiones la medida de justicia algorítmica que va a usarse en el proyecto?
- (Cuantitativo) Cuando existen atributos protegidos debe evaluarse qué tanto se alejan las predicciones de la definición de justicia algorítmica elegida.
- (Cuantitativo) Posprocesar adecuadamente las predicciones, si es necesario, para lograr el criterio de justicia algorítmica elegido (por ejemplo, equidad de posibilidades, oportunidad).
- (Cuantitativo) En el caso de clasificación, puntos de corte para distintos subgrupos pueden ajustarse para lograr equidad de oportunidad.
- (Cuantitativo) Recolectar información más relevante de subgrupos protegidos (tanto casos como características) para mejorar el desempeño predictivo en grupos minoritarios.

Actividad: Al terminar esta fase se recomienda el llenado de las secciones de Desarrollo del modelo del Perfil del modelo y llevar a cabo una discusión con el tomador de decisiones de políticas públicas (ver Herramienta 3).

VER HERRAMIENTA 3

4. USO Y MONITOREO



4. Uso y monitoreo

Una vez que los métodos de aprendizaje automático comienzan a utilizarse para tomar decisiones, es necesario:

- Monitorear, en general, desempeño y características usadas en el tiempo.
- Monitorear, en particular, resultados indeseables que pueden provenir de la interacción de usuarios con sistemas de toma/soporte de decisiones.
- Evaluar la recolección y procesamiento de datos para mejorar desempeño o evaluar resultados.

4.1 Degradación de desempeño

El desempeño de un modelo puede degradarse con el tiempo por múltiples razones:

- Los modelos de ML que asumen una relación estática entre las variables de entrada y de salida pueden degradar la calidad de sus predicciones por cambios en las relaciones subyacentes del contexto de estudio.
- También puede deberse a un cambio en la calidad de los datos por la forma de recolección o incluso redefiniciones metodológicas utilizadas para recolectar información. Por ejemplo, en registros administrativos un ministerio o secretaría podría cambiar los procesos de recolección de datos, digitalizar sistemas, sistematizar limpieza o procesamiento que haga que el aprendizaje de un sistema ya no sea relevante.
- También esto ocurre en sistemas interactivos donde el sistema y sus usuarios forman un ciclo de realimentación cerrado, con lo cual el sistema se va degradando porque los usuarios solo pueden interactuar con elementos que son decididos por el sistema.

Para mitigar estos posibles errores es necesario monitorear el comportamiento de las variables de entrada y actualizar supuestos con tomadores de decisión y conocimiento experto.

También debe vigilarse el comportamiento de las métricas de error en el tiempo: cantidades con tasa total de positivos y negativos (incluyendo desagregaciones por otras variables protegidas o de interés), distribución de predicciones y atributos.

Recuadro 18. Lista de verificación - Degradación de desempeño

☑ Degradación de desempeño:

- (Cualitativo) ¿Existe un plan para monitorear el desempeño del modelo y la recolección de información a lo largo del tiempo?
- (Cuantitativo) Monitorear varias métricas asociadas a las predicciones, en subgrupos definidos con antelación (incluyendo variables protegidas).
- (Cuantitativo) Monitorear la deriva en distribuciones de características con respecto al conjunto de entrenamiento.
- (Cuantitativo) Monitorear cambios en la metodología de levantamiento y procesamiento de datos que pueden reducir la calidad de las predicciones.
- (Cuantitativo) Idealmente, planear para recolectar datos de la variable no observada para reajustar modelos y mantener el desempeño.
- (Cualitativo) Cuando sea aplicable y factible, una fracción de las predicciones deberán examinarlas seres humanos y calificarlas según alguna rúbrica o mediciones de las variables que se busca predecir.

4.2 Experimentos y evaluación del modelo

La forma y los datos que se recopilan para el mantenimiento de los algoritmos de predicción debe planearse con el objeto de mejorar en lo posible y entender mejor las consecuencias del uso de los modelos.

Las mejoras que se esperan en el proceso pueden ser difíciles de evaluar sin contrafactuales sólidos.

Pruebas con diseño experimental pueden planearse, por ejemplo, de tipo A/B u otras (Vaver and Koehler, 2011), cuando sea posible, para entender qué cambios particulares, deseables o indeseables, introduce el uso de los modelos.

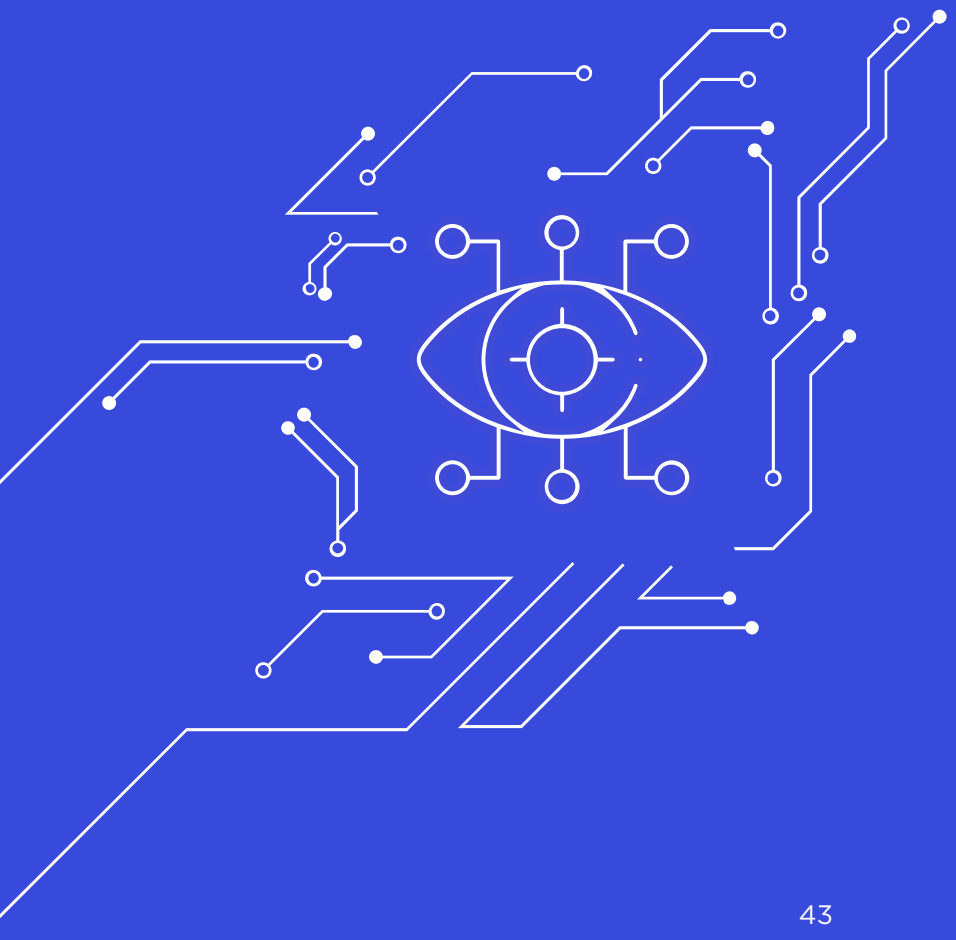
Recuadro19. Lista de verificación - Experimentos y evaluación del modelo

- (Cuantitativo) Cuando sea posible, planear asignar bajo diseños experimentales tratamientos aleatorios (o según el statu quo) a algunas unidades. Hacer comparaciones de desempeño y comportamiento entre esta muestra y los resultados bajo el régimen algorítmico.
- (Cuantitativo) Identificar las variables no observadas y buscar la forma de medirlas. Si es posible, volver a ajustar el modelo y evaluar su rendimiento, utilizando esta nueva información.

Actividad: Al terminar esta fase se recomienda el llenado de la sección de Uso y monitoreo del Perfil del modelo (ver Herramienta 3) y llevar a cabo una discusión con el tomador de decisiones de políticas públicas.

VER HERRAMIENTA 3

5. RENDICIÓN DE CUENTAS



5. Rendición de cuentas

Regulaciones como el Reglamento General de Protección de Datos (RGPD) de la Unión Europea definen la responsabilidad como el requisito para que las organizaciones pongan en marcha medidas técnicas y organizativas adecuadas y sean capaces de demostrar lo que hicieron y su eficacia cuando se les solicite.

Aunque el desarrollo de estándares y normas técnicas para los sistemas de IA es todavía una tarea en proceso para la comunidad de la IA, este manual describe los principales aspectos técnicos y medidas para evitar y mitigar los sesgos durante el ciclo de vida de la IA. Sin embargo, quedan varios retos relacionados con los requisitos sociales y legales que conlleva el uso de estos sistemas en aplicaciones del mundo real.

Esta sección revisa los conceptos de interpretabilidad, explicabilidad y trazabilidad de los sistemas de IA.

5.1 Interpretabilidad y explicación de predicciones

5.1.1 Interpretabilidad

Es difícil dar una definición técnica de interpretabilidad o explicabilidad, términos que en general se refieren a hacer inteligible para los seres humanos el funcionamiento de un algoritmo y sus resultados (Molnar, 2019). Cuanto más interpretable sea un modelo, más fácil será para un individuo entender el proceso que lo ha llevado a una determinada decisión (Miller, 2019). Un modelo con alta interpretabilidad es deseable en una aplicación de política social de alto riesgo donde el criterio de responsabilidad se vuelve fundamental.

Hay varias razones por las que es importante tener cierto grado de interpretabilidad en los modelos que se usan para tomar decisiones (Molnar, 2019):

1. Aprendizaje acerca del dominio del problema.
2. Logro de aceptación social del uso de los sistemas.
3. Detección de sesgos potenciales de los algoritmos.
4. Depuración y mejora de los modelos.

Los algoritmos complejos, como las redes neuronales profundas, pueden tener millones de relaciones entre sus parámetros, por lo que obtener la interpretabilidad del modelo en estos algoritmos sigue siendo un campo abierto en el aprendizaje automático. Cuando es necesaria una alta interpretabilidad se recomienda usar métodos intrínsecamente interpretables como la regresión lineal, la regresión logística y los árboles de decisión.

5.1.2 Explicabilidad de predicciones individuales

Existe en muchos casos la necesidad legal y/o ética de dar explicaciones individuales acerca de cómo fueron tomadas ciertas decisiones (por ejemplo, por qué a una persona no se le otorgó un crédito o por qué alguien no califica para un programa social). [11]

11 En la Unión Europea, por ejemplo, el artículo 22 de GDPR describe el derecho de una persona a rebatir la decisión de un sistema, especialmente cuando es automática.

En áreas de investigación, como visión artificial y procesamiento del lenguaje natural, las implementaciones más exitosas suelen estar desarrolladas con modelos de alta complejidad, como redes neuronales profundas, que son en principio poco transparentes en cuanto a cómo se hacen las predicciones subyacentes (Carrillo, Cantú, & Noriega, 2020).

Aunque se trata de un área de investigación en curso, ya existen varios métodos para aumentar la explicabilidad de las predicciones (Molnar, 2019). Pueden utilizarse métodos como el de explicaciones contrafactuales (Wachter, Mittelstadt, & Russell, 2017), valores de Shapley (Lundberg & Lee, 2017) o gradientes integrados para redes profundas (Sundararajan, Taly, & Yan, 2017).

Recuadro 20. Explicabilidad de predicciones individuales

- (Cualitativo) ¿Se analizaron los requerimientos legales y éticos de explicabilidad e interpretabilidad necesarios para el caso de uso?
- .(Cualitativo) ¿Se ha definido un plan de respuesta en caso de que algún usuario se vea perjudicado por los resultados?
- (Cualitativo) ¿Existe algún proceso para dar explicaciones a un individuo en particular sobre por qué se tomó una decisión?
- (Cualitativo) ¿Se discutieron los pros y contras de los algoritmos según su nivel de interpretabilidad y explicabilidad para elegir el más apropiado?
- (Cuantitativo) Para modelos más simples (por ejemplo, lineales o árboles de decisión), pueden construirse explicaciones ad hoc.
- (Cuantitativo) Utilizar métodos como explicaciones contrafactuales, valores de Shapley o gradientes integrados para redes profundas.

5.1.3 Modelos parsimoniosos

Está muy extendida la idea de que un modelo ML es siempre mejor cuando se utilizan más covariables; esto es parcialmente correcto, porque el modelo puede encontrar patrones entre la interrelación de las variables. Sin embargo, cuando se tiene en cuenta la interpretabilidad, los métodos más parsimoniosos que utilizan menos características, pero relevantes, son preferibles a los modelos que utilizan muchas características, pero quizás menos relevantes. Pueden producirse sesgos potenciales cuando se utilizan características o variables de los datos que, aunque sean válidas para un momento y un conjunto de datos determinados, son fácilmente susceptibles de cambiar cuando evoluciona el proceso de generación de datos. Los algoritmos o métodos de predicción que utilizan muchos atributos irrelevantes corren un mayor riesgo de fallar, tanto explícita como silenciosamente, cuando cambian las fuentes de datos o los procesos de generación de datos.

Ejemplos de ello pueden ser el uso de variables que están siendo activamente influenciadas por alguna política que no continuará en el futuro o el aprendizaje de características a partir de un conjunto de entrenamiento no exhaustivo (por ejemplo, en el reconocimiento de imágenes, el reconocimiento de especies animales por el contexto en el que se recogió la información, como un zoológico, una cámara trampa, un paisaje, etc.).

Este tipo de sesgo perjudica la explicabilidad de un sistema y puede ser difícil de detectar, pero los métodos parsimoniosos y el conocimiento experto pueden mitigar el riesgo.

Recuadro 21. Lista de verificación – Modelos parsimoniosos

- (Cualitativo) Incluir todas las características disponibles para construir modelos aumenta el riesgo de que se generen sesgos. Las variables por incluirse en el proceso de aprendizaje deben tener algún sustento teórico o explicación de por qué pueden ayudar en la tarea de predicción.
- (Cuantitativo) Métodos más parsimoniosos, que usan menos características, son preferibles a modelos que utilizan muchas características.
- (Cuantitativo) Métodos como gráficas de dependencia parcial (Friedman, 2001) o importancia basada en permutaciones (Breiman, 2001) (Molnar, 2019) pueden señalar variables problemáticas que reciben mucho peso en la predicción, en contra de observaciones pasadas o conocimiento experto.

5.2 Trazabilidad

Un proceso de datos para decisiones que es poco **trazable** es uno que contiene pasos con documentación deficiente acerca de su ejecución: incluyen procesos manuales o decisiones de operadores pobremente especificadas, extraen datos de fuentes no documentadas o no accesibles, omiten códigos o materiales necesarios, o no explican los ambientes de cómputo para garantizar resultados reproducibles.

La trazabilidad permite a los usuarios comprender los procesos seguidos por un sistema de IA para llegar a un resultado, incluidas las deficiencias y limitaciones del sistema. Cuando hay poca trazabilidad en un modelo, los riesgos señalados a lo largo de este documento pueden ser difíciles de identificar e incluso agravarse. Por el contrario, todos los pasos, desde la recolección de datos hasta la toma de decisiones, están claramente documentados y especificados sin ambigüedades en un proyecto trazable.

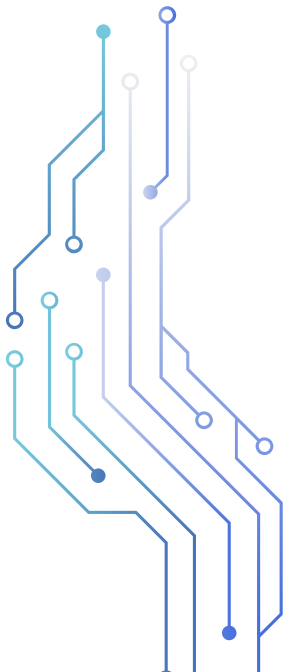
Recuadro 22. Lista de verificación - Trazabilidad

- (Cuantitativo) ¿Está bien documentado el proceso de ingesta, transformación, modelado y toma de decisión (incluyendo fuente de datos, infraestructura y dependencias, código, métricas e interpretación de resultados)?

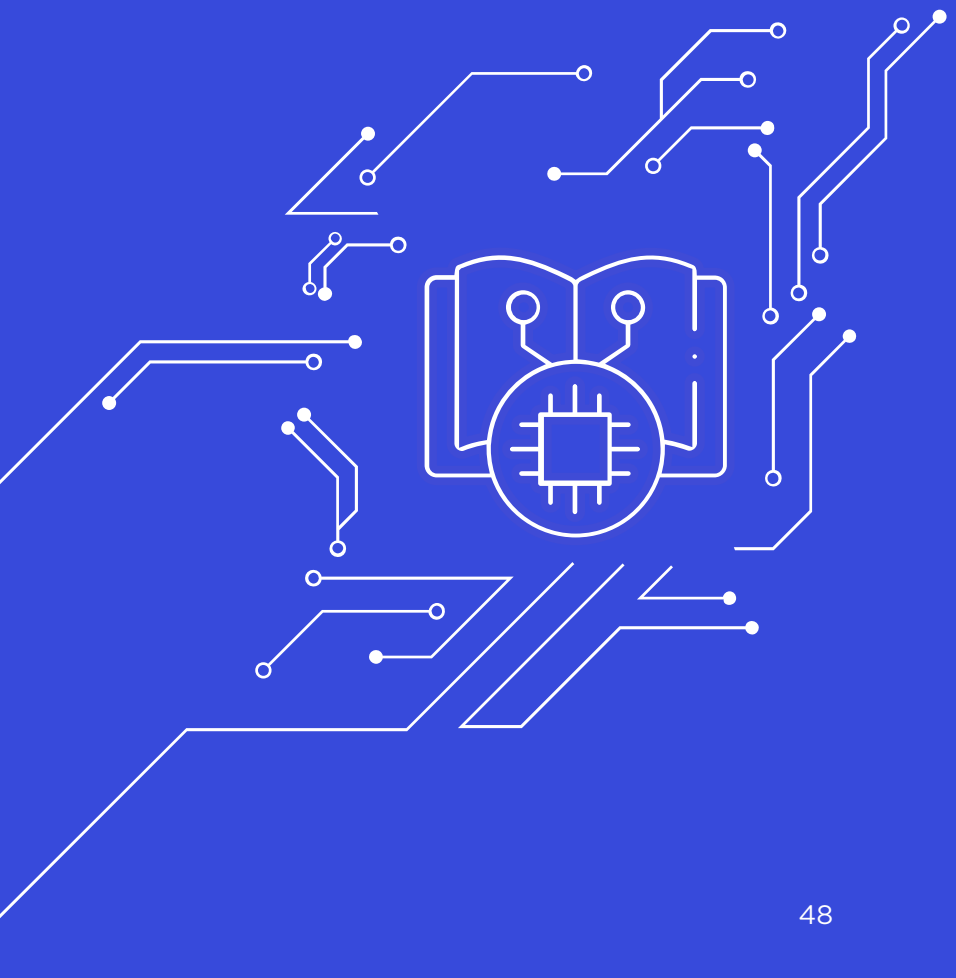
  1. **Fuentes de datos,** incluidos los metadatos de los conjuntos de datos, los procesos de recolección de datos y etapas de su procesamiento (ver la Herramienta 2).
  2. **Código completo y adecuadamente documentado**, que define las bibliotecas necesarias y sus versiones adecuadas para permitir que cualquier tercero comprenda la finalidad de cada parte del código.
  3. **Información sobre cómo debe ejecutarse el código**, incluyendo una documentación detallada de los parámetros y requisitos informáticos. Esta información debe garantizar la reproducibilidad de los resultados originales por parte de un tercero.
  4. **Información sobre cómo se utilizaron los resultados del sistema y se incluyeron** en el proceso de toma de decisiones de política pública.
  5. **Información sobre la estrategia de monitoreo**, que incluya detalles sobre las métricas de rendimiento y los umbrales, así como el comportamiento esperado del modelo y las acciones de mitigación.

Lo ideal es que un tercero pueda replicar los pasos mencionados con una intervención mínima o nula de los creadores y operadores del sistema original.

- (Cualitativo) ¿Se han comunicado las deficiencias, limitaciones y sesgos del modelo a las partes interesadas para que se tengan en cuenta en la toma de decisiones y el apoyo a las mismas?
- (Cualitativo) ¿Ha completado el equipo técnico el Perfil de datos (ver la Herramienta 2) y el Perfil del modelo (ver la Herramienta 3), y se ha definido un proceso para la actualización continua de estas herramientas?



HERRAMIENTAS



Herramienta 1: Lista de verificación de IA robusta y responsable

Esta herramienta consolida las principales preocupaciones referidas a la dimensión de riesgo del ciclo de vida de IA. La lista de verificación debe revisarla continuamente el equipo técnico, acompañado por el tomador de decisiones (Fritzler, 2015; drivendata, 2019).

Conceptualización y diseño

☑ Definición correcta del problema y de la respuesta de la política pública:

- (Cualitativo) ¿Está claramente definido el problema de política pública?
- (Cualitativo) Describir cómo se aborda actualmente este problema –considerando las respuestas de las instituciones relacionadas– y cómo el uso de la IA mejoraría la respuesta gubernamental a este problema.
- (Cualitativo) ¿Se identificaron los grupos o los atributos protegidos dentro del proyecto (por ejemplo, edad, género, nivel educativo, raza, nivel de marginación, etc.)?
- (Cualitativo) ¿Se definieron las acciones o intervenciones que se llevarán a cabo en función del resultado del sistema de IA?

☑ Principios de la IA

- (Cuantitativo) ¿Se ha justificado la necesidad de un sistema de IA, teniendo en cuenta otras posibles soluciones que no requieran el uso de datos personales y decisiones automatizadas?
- (Cuantitativo) ¿Existen pruebas de que tanto la acción de las políticas públicas como la recomendación del sistema de IA supondrán un beneficio para las personas y el planeta al impulsar el crecimiento inclusivo, el desarrollo sostenible y el bienestar?
- (Cualitativo) ¿Se han identificado y analizado proyectos similares para obtener aprendizajes y errores comunes?
- (Cuantitativo) ¿Se ha considerado la posibilidad de minimizar la exposición de información personal identificable (por ejemplo, anonimizando o no recogiendo información no relevante para el análisis)?

Ciclo de vida

☑ Recolección y procesamiento de datos

Calidad y relevancia de los datos disponibles
Estados indeseables o subóptimos en datos recolectados

- (Cualitativo) Discutir las posibles desigualdades sociales históricas en el caso de uso con especialistas en la materia.
- (Cuantitativo) Realizar un análisis exploratorio de los datos disponibles con los que se entrenará el modelo para identificar posibles sesgos históricos o estados indeseables.

Mala correspondencia entre variables disponibles y variables ideales

- (Cualitativo) Las variables objetivo ideales deben estar claramente establecidas. Las variables recogidas/disponibles deben analizarse para comprender hasta qué punto son adecuadas para sustituir a la variable objetivo. Deben identificarse los sesgos sistemáticos o la validez de la métrica sustituto.
- (Cualitativo) ¿Se ha justificado claramente el uso de la variable de respuesta seleccionada para los fines de la intervención?

☑ **Cualificación y exhaustividad de los datos para la población objetivo**

Muestras probabilísticas y naturales

- (Cualitativo) ¿Se han analizado las posibles diferencias entre la base de datos y la población para la que se está desarrollando el sistema de IA? (Utilizar la bibliografía relacionada con el tema y la información de los expertos. Estudiar en particular los sesgos de selección no medidos).
- (Cuantitativo) Aunque los modelos pueden construirse con diversas fuentes de datos, diseñadas o naturales, lo ideal es que la validación se realice con una muestra que permita la inferencia estadística a la población. La muestra de validación debe cubrir adecuadamente la población objetivo y las subpoblaciones de interés.

Atributos faltantes o incompletos

- (Cualitativa) ¿Se ha realizado un análisis de valores faltantes y de variables omitidas?
- (Cualitativa) ¿Se ha identificado si existen variables omitidas importantes para las cuales no se tiene mediciones asociadas (en caso de existir)?
- (Cualitativa) ¿Se ha identificado las razones por las que existen observaciones faltantes (en caso de existir)?
- (Cuantitativa) Los procesos de imputación tienen que evaluarse en cuanto a su sensibilidad a supuestos y datos. De preferencia, deben utilizarse métodos de imputación múltiple que permitan evaluar incertidumbre en la imputación (Little & Rubin, 2002) (Buuren & Groothuis-Oudshoorn, 2011).

☑ **Comparación causal**

- (Cualitativo) Comprender y describir las razones por las que la variable de respuesta está correlacionada con variables conocidas y desconocidas. Describir los posibles sesgos basados en el conocimiento y el análisis de los expertos.
- (Cualitativo) En caso de que no se haya trabajado para asegurar la causalidad en los resultados, ¿se comunicaron explícitamente las limitaciones de los resultados al responsable de las políticas públicas?
- (Cuantitativo) En el caso de que se intente la inferencia causal con modelos, deben describirse las hipótesis, consideraciones o métodos utilizados para apoyar una interpretación causal. Deben realizarse y documentarse las comprobaciones de robustez.

Desarrollo del modelo y validación

☑ Ausencia o uso inadecuado de muestras de validación

- (Cuantitativa) ¿Se construyeron adecuadamente las muestras de validación y prueba, considerando un tamaño apropiado, cubriendo a subgrupos de interés y protegidos y evitando fugas de información durante su implementación?

☑ Fugas de información

Contaminación entrenamiento-validación

- (Cuantitativa) Cualquier procesamiento y preparación de datos de entrenamiento debe evitar de cualquier manera usar los datos de validación o prueba. Debe mantenerse una barrera sólida entre entrenamiento vs. validación y prueba. Esto comprende recodificación de datos, normalizaciones, selección de variables, identificación de datos atípicos y cualquier otro tipo de preparación de cualquier variable que va a incluirse en los modelos, lo que también abarca ponderaciones o balance de muestras basadas en sobre/sub muestreo.

Fugas de datos no disponibles en la predicción

- El esquema de validación debe replicar tan cerca como sea posible el esquema con el cual se aplicarán las predicciones. Esto incluye que hay que replicar:
- Ventanas temporales de observación y registro de variables y ventanas de predicción.
- Si existen grupos en los datos, considerar si se tendrá información disponible de cada grupo cuando se haga la predicción, o si es necesario predecir para nuevos grupos.

☑ Probabilidades y clases

Datos desbalanceados

- (Cuantitativa) Hacer predicciones de probabilidad en lugar de clase. Estas probabilidades pueden incorporarse al proceso de decisión posterior como tales. Evitar puntos de corte estándar de probabilidad como 0.5, o predecir según máxima probabilidad.
- (Cuantitativa) Cuando el número absoluto de casos minoritarios es muy reducido, puede ser muy difícil encontrar información apropiada para discriminar esa clase. Se requiere recolectar más datos de la clase minoritaria.
- (Cuantitativa) Submuestrear la clase dominante (ponderando hacia arriba los casos para no perder calibración) puede ser una estrategia exitosa para reducir el tamaño de los datos y el tiempo de entrenamiento sin afectar el desempeño predictivo.
- (Cuantitativa) Replicar la clase minoritaria, para balancear mejor las clases (sobremuestreo).
- (Cuantitativa) Algunas técnicas de aprendizaje automático permiten ponderar cada clase por un peso distinto para que el peso total de cada clase quede balanceado. Si esto es posible, es preferible a sub o sobremuestreo.

Puntos de corte arbitrarios

- (Cuantitativo) El uso de algoritmos de clasificación probabilística es más adecuado para que la toma de decisiones incorpore la incertidumbre con respecto a la clasificación.
- (Cuantitativo) Evitar los puntos de corte de probabilidad estándar, como el 0,5. Elegir una interpretación óptima de las probabilidades predichas analizando las métricas de error.

Idoneidad de las métricas de evaluación

- (Cualitativa) ¿Se cuestionaron las implicaciones de los diferentes tipos de errores para el caso de uso específico, así como la forma correcta de evaluarlos?
- (Cualitativa) ¿Se explicó en forma clara las limitantes del modelo, identificando tanto los falsos positivos como los falsos negativos y las implicaciones que una decisión del sistema tendría en la vida de la población beneficiaria?
- (Cuantitativa) ¿Se implementó un análisis costo-beneficio del sistema contra el statu quo u otras estrategias de toma/soporte de decisión (cuando es posible)?

Sub y sobreajuste

- (Cuantitativa) Sobreajuste: debe evitarse modelos cuya brecha validación - entrenamiento sea grande (indicios de sobreajuste). De ser necesario, deben afinarse métodos para moderar el sobreajuste como regularización, restricción del espacio funcional de modelos posibles, usar más datos de entrenamiento o perturbar los datos de entrenamiento, entre otros (Hastie, Tibshirani, & Friedman, 2017).
- (Cuantitativa) Subajuste: deben revisarse subconjuntos importantes de casos (por ejemplo, grupos protegidos) para verificar que no existen errores sistemáticos indeseables.

☑ **Errores no cuantificados y evaluación humana**

Fallas no medidas por el modelo

- (Cualitativa) ¿Se realizó una evaluación con expertos del caso de uso para buscar sesgos o errores conocidos? (Por ejemplo, pueden usarse paneles de revisores que examinen predicciones particulares y consideren si son razonables o no. Estos paneles deben ser balanceados en cuanto al tipo de usuarios que se prevén, incluyendo tomadores de decisiones, si es necesario).

☑ **Equidad y desempeño diferencial de predictores**

Definición de justicia y equidad algorítmicas

- (Cualitativa) ¿Se definió con expertos y tomadores de decisiones la medida de justicia algorítmica que va a usarse en el proyecto?
- (Cuantitativa) Cuando existen atributos protegidos, debe evaluarse qué tanto se alejan las predicciones de la definición de justicia algorítmica elegida.
- (Cuantitativa) En el caso de clasificación, pueden ajustarse puntos de corte para distintos subgrupos con el fin de lograr equidad de oportunidad.

Uso y monitoreo

☑ Degradación de desempeño

- (Cualitativo) ¿Existe un plan para monitorear el desempeño del modelo y la recolección de información a lo largo del tiempo?
- (Cuantitativo) Monitorear varias métricas asociadas a las predicciones en subgrupos definidos con antelación (incluyendo variables protegidas).
- (Cuantitativo) Monitorear deriva en distribuciones de características con respecto al conjunto de entrenamiento.
- (Cuantitativo) Monitorear cambios en la metodología de levantamiento y procesamiento de datos que pueden reducir calidad de las predicciones.
- (Cuantitativo) Cuando sea posible, planear asignar bajo diseños experimentales tratamientos aleatorios (o según el statu quo) a algunas unidades. Hacer comparaciones de desempeño y comportamiento entre esta muestra y los resultados de acuerdo con el régimen algorítmico.
- (Cuantitativo) Identificar las variables no observadas y buscar la forma de medirlas. Si es posible, volver a ajustar el modelo y evaluar su rendimiento utilizando esta nueva información.

Rendición de cuentas

☑ Interpretabilidad y explicación de predicciones

Explicabilidad de predicciones individuales

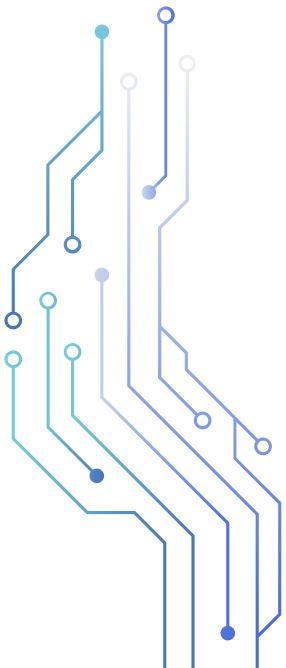
- (Cualitativo) ¿Se analizaron los requerimientos legales y éticos de explicabilidad e interpretabilidad necesarios para el caso de uso?
- (Cualitativo) ¿Existe algún proceso para dar explicaciones a un individuo en particular sobre por qué se tomó una determinada decisión?
- (Cualitativo) ¿Se discutieron los pros y contras de los algoritmos según su nivel de interpretabilidad y explicabilidad para elegir el más apropiado?

Modelos parsimoniosos

- (Cualitativa) Incluir todas las características disponibles para construir modelos aumenta el riesgo de que se generen sesgos. Las variables por incluirse en el proceso de aprendizaje deben tener algún sustento teórico o explicación de por qué pueden ayudar en la tarea de predicción.
- (Cuantitativa) Métodos más parsimoniosos, que usan menos características, son preferibles a modelos que utilizan muchas características.
- (Cuantitativa) Métodos como gráficas de dependencia parcial (Friedman, 2001) o importancia basada en permutaciones (Breiman, 2001) (Molnar, 2019) pueden señalar variables problemáticas que reciben mucho peso en la predicción, en contra de observaciones pasadas o conocimiento experto.

Trazabilidad

- (Cuantitativa) ¿Está bien documentado el proceso de ingesta, transformación, modelado y toma de decisión (incluyendo fuente de datos, infraestructura y dependencias, código, métricas e interpretación de resultados)?
- (Cualitativo) ¿Se han comunicado las deficiencias, limitaciones y sesgos del modelo a las partes interesadas para que se tengan en cuenta en la toma de decisiones y el apoyo a las mismas?
- (Cualitativo) ¿Ha completado el equipo técnico el Perfil de datos (ver la Herramienta 2) y el Perfil del modelo (ver la Herramienta 3), y se ha definido un proceso para la actualización continua de estas herramientas?



Herramienta 2: Perfil de datos

El perfil de datos es un análisis exploratorio que brinda información para evaluar la calidad, integridad, temporalidad, consistencia y posibles sesgos de un conjunto de datos que se uti lizará para entrenar un modelo de aprendizaje automático (Gebru et al., 2018).

Fuente de recolección y origen de los datos

- Nombre del conjunto de datos utilizado.
- ¿Qué institución creó el conjunto de datos?
- ¿Con qué propósito creó la institución el conjunto de datos empleado?
- ¿Qué mecanismos o procedimientos se usaron para recoger los datos (por ejemplo, encuesta de hogares, sensor, software, API)? ¿Cumplen la normativa vigente en materia de protección de datos?
- ¿Cuál es la escala del conjunto de datos?
- Obtener documentación para cada variable del conjunto de datos. Proporcionar una breve descripción, incluyendo su nombre y tipo, lo que representa, cómo se mide, etc.

Gobernanza de los datos

- ¿Cuál es el dominio de los datos (por ejemplo, propietario, público, personal)?
- Si son personales, ¿los datos están identificados, tienen seudónimos, cuentan con seudónimos no vinculados, son anónimos o agregados?
- Si son privados, ¿se tiene en cuenta los derechos de propiedad intelectual y la protección de datos personales?

Estructura de los datos

- ¿Los datos son estáticos o dinámicos? Si son dinámicos, ¿con qué frecuencia se actualizarán?
- Captar la frecuencia (diaria, semanal, mensual) o el número medio de observaciones por individuo. ¿Qué versión del conjunto de datos se está utilizando?
- ¿Es el conjunto de datos más adecuado disponible, teniendo en cuenta el problema en cuestión?

Calidad de los datos

- ¿Cómo se han obtenido los datos (observados, derivados, sintéticos o proporcionados por personas u organizaciones)?
- ¿Son los datos representativos de la población de interés?
- Describir el tipo de muestreo utilizado para obtener los datos.
- Analizar la cobertura espacial y temporal de los datos.
- Analizar la cobertura de los grupos protegidos (sexo, raza, edad, etc.).
- Describir las dimensiones importantes en las que la muestra de datos puede diferir de la población, en particular los sesgos de selección no medidos. Utilizar la literatura relacionada con el tema y la información de los expertos.

- Identificar los posibles “estados indeseables” en los datos, que podrían dar lugar a sesgos e inequidades perjudiciales para un determinado subgrupo, o cualquier otro patrón que se considere subóptimo o indeseable desde el punto de vista de la política social.
- ¿Hay valores faltantes? Si es así, explicar las razones por las que no se dispone de esa información (esto incluye la información eliminada intencionalmente). Identificar las razones de los datos que faltan y pensar si los datos faltantes se asocian a la variable que va a predecirse. Documentar cualquier proceso de imputación utilizado para sustituir los datos que faltan.

Herramienta 3: Perfil del modelo (Model Card)

La rúbrica presentada aquí es una tarjeta de seguimiento que resume las características principales de un sistema de toma/soporte de decisiones basado en ML y destaca los principales supuestos, las características más importantes del sistema y las medidas de mitigación implementadas (Mitchell et al., 2019).

Conceptualización y diseño de política pública

1. Información básica
   - Personas que desarrollaron el modelo, fecha, versión, tipo.
2. Casos de uso
   - Antecedentes.
   - Población objetivo y horizonte de predicciones.
   - Actores y componentes que interactuarán con los resultados.
   - Casos de uso considerados durante el desarrollo.
   - Usos no considerados y advertencias relacionadas.
   - Definición de grupos protegidos.

Fuente y manejo de datos

3. Datos de entrenamiento
   - Conjunto de datos usados y su etiquetado.
   - Pasos de preprocesamiento o preparación de datos.
   - Sesgos y deficiencias potenciales según el caso de uso.

Desarrollo del modelo

4. Modelación
   - Algoritmos que se usaron para entrenar, parámetros supuestos o restricciones.
5. Métricas de desempeño
   - Métricas técnicas usadas para seleccionar y evaluar modelos.
   - Análisis costo-beneficio del modelo para su caso de uso.
   - Definición de grupos protegidos y medidas de equidad seleccionadas.
6. Datos de validación
   - Conjuntos de datos usados y su etiquetado.
   - Pasos de preprocesamiento.
   - Evaluación de adaptación de datos de validación según el caso de uso.
   - Sesgos y deficiencias potenciales según el caso de uso.
7. Resumen de análisis cuantitativo
   - Error de validación reportado.
   - Resumen de análisis costo-beneficio.
   - Reporte de medidas de equidad para grupos protegidos.

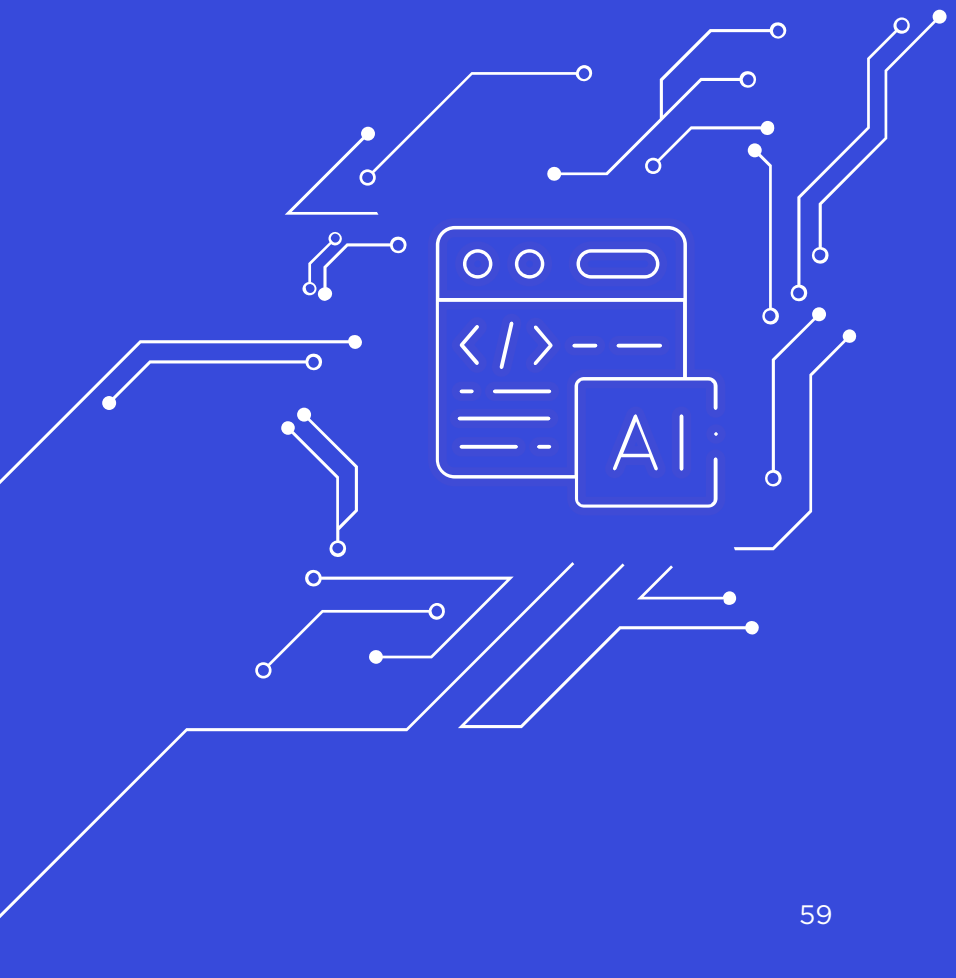
Uso y monitoreo

8. Recomendaciones de monitoreo
   - Estrategia de monitoreo y mejora en producción.
   - Estrategias de monitoreo humano de predicciones (si aplica).

Rendición de cuentas

9. (Opcional) Explicabilidad de predicciones
   - Estrategia para explicar predicciones particulares (si es necesario).
   - Estrategia para entender la importancia de distintos atributos.
10. Otras consideraciones éticas, recomendaciones y advertencias.

CUADERNILLOS DE TRABAJO



Cuadernillos de trabajo

Esta sección ofrece varios ejemplos de los retos y soluciones explicadas en el documento principal. Se utilizan distintos tipos de modelos (lineales, basados en árboles y otros) y distintas implementaciones (R, keras, xgboost) para mostrar que estos problemas se presentan independientemente de la elección de herramientas particulares.

Los cuadernillos utilizan notación con punto decimal para mantener consistencia con los paquetes que así lo usan. Emplea el lenguaje de programación R y los siguientes paquetes: tidyverse, recipes, themis, rsample, parsnip, yardstick, workflows, tune, knitr, patchwork. Todo el material es reproducible según instrucciones en este repositorio, el cual contiene un archivo Dockerfile que describe las dependencias de infraestructura para su replicación.

Recolección y procesamiento de datos

Mala correspondencia entre variables disponibles y variables ideales

Usar modelos que predicen la métrica incorrecta puede llevar a tomar decisiones erradas. A veces el problema es claro, cuando la métrica sustituto tiene deficiencias obvias; en otras, puede ser más sutil.

En el ejemplo que aparece a continuación se busca predecir la demanda de cierto producto (pensemos en vacunas o alguna medicina) para poder tomar decisiones de abastecimiento. Se cuenta con datos históricos de inventario (80 semanas), ventas y una variable asociada a ventas (en el caso de las vacunas podría ser temperatura) y otra de agotamiento del inventario. Separamos los datos en entrenamiento y prueba, ajustando el modelo con el subconjunto de datos de entrenamiento. En este caso se utiliza un modelo lineal con variable dependiente ventas y covariables de semana y la covariable .

```
entrena <- ventas %>% filter(semana < 60)

prueba <- ventas %>% filter(semana >= 60, semana <= 80)

entrena %>% select(-demanda) %>%  head() %>% kable()
```

Semana	Inventario	Ventas	Predictor	Agotamiento
1	153	110	-27.7014124	0
2	170	148	0.7664636	0
3	158	130	-15.2606032	0
4	162	142	4.2461227	0
5	159	159	28.5107593	1
6	162	162	14.8895964	1

```
mod_lineal <- lm(ventas ~ semana + predictor, data = ventas)

mod_lineal
```

```
##

## Call:

## lm(formula = ventas ~ semana + predictor, data = ventas)

##

## Coefficients:

## (Intercept)       semana    predictor

##    140.9935       0.8166       0.5535
```

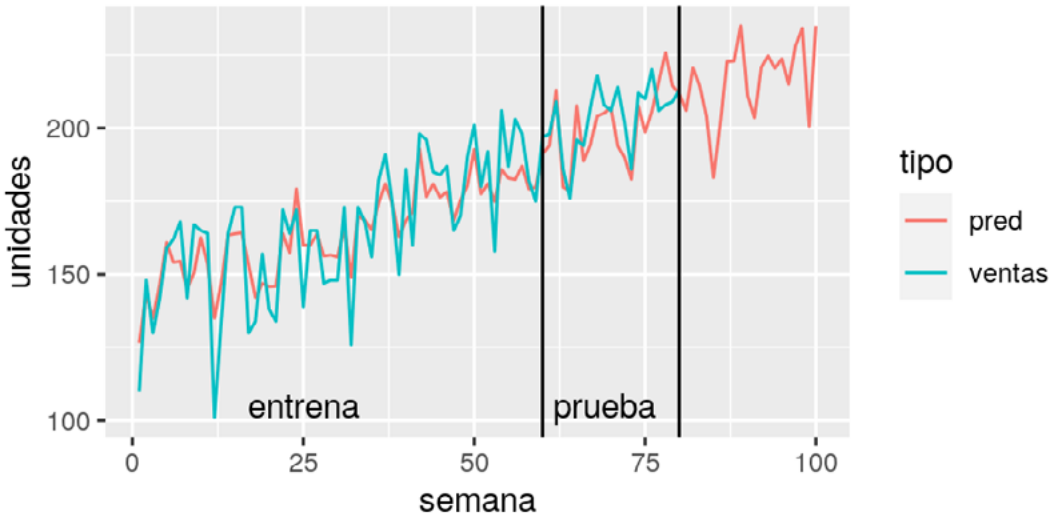
Evaluamos el error de predicción.

```
preds <- predict(mod_lineal, newdata = prueba)

round(mean(abs(preds - prueba$ventas))/mean(prueba$ventas), 3)
```

```
## [1] 0.04
```

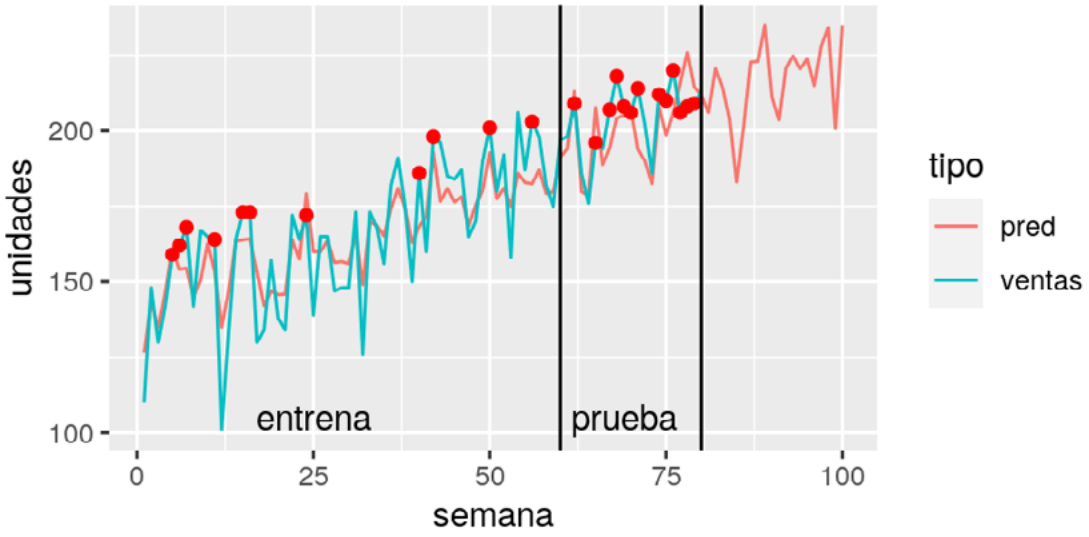
El error porcentual es bajo. Los datos ajustados y predicciones se ven como sigue:

```
preds <- predict(mod_lineal, newdata = ventas)

ventas_larga <- ventas %>% mutate(pred = preds) %>%

  pivot_longer(cols = all_of(c("ventas","pred")), names_to = "tipo", values_to
= "unidades")

ggplot(ventas_larga %>% mutate(unidades = ifelse(tipo=="ventas" & semana > 80,
NA, unidades)),

       aes(x = semana, y = unidades, group = tipo, colour = tipo)) +

  geom_line() +

  geom_vline(xintercept = 80) +

  geom_vline(xintercept = 60) +

 annotate("text", x = 25, y=105, label = "entrena") +

  annotate("text", x = 69, y=105, label = "prueba")
```



Sin embargo, tomar decisiones de demanda o inventario es equivocado. La razón es que existe una diferencia entre la variable ideal (demanda real de medicinas) y la variable observada (venta de medicinas). La diferencia radica en que existen agotamientos de inventario, es decir, periodos en los que aunque existía demanda, no había suficiente inventario para todos los compradores. Esto se ve marcado con rojo en la siguiente gráfica.

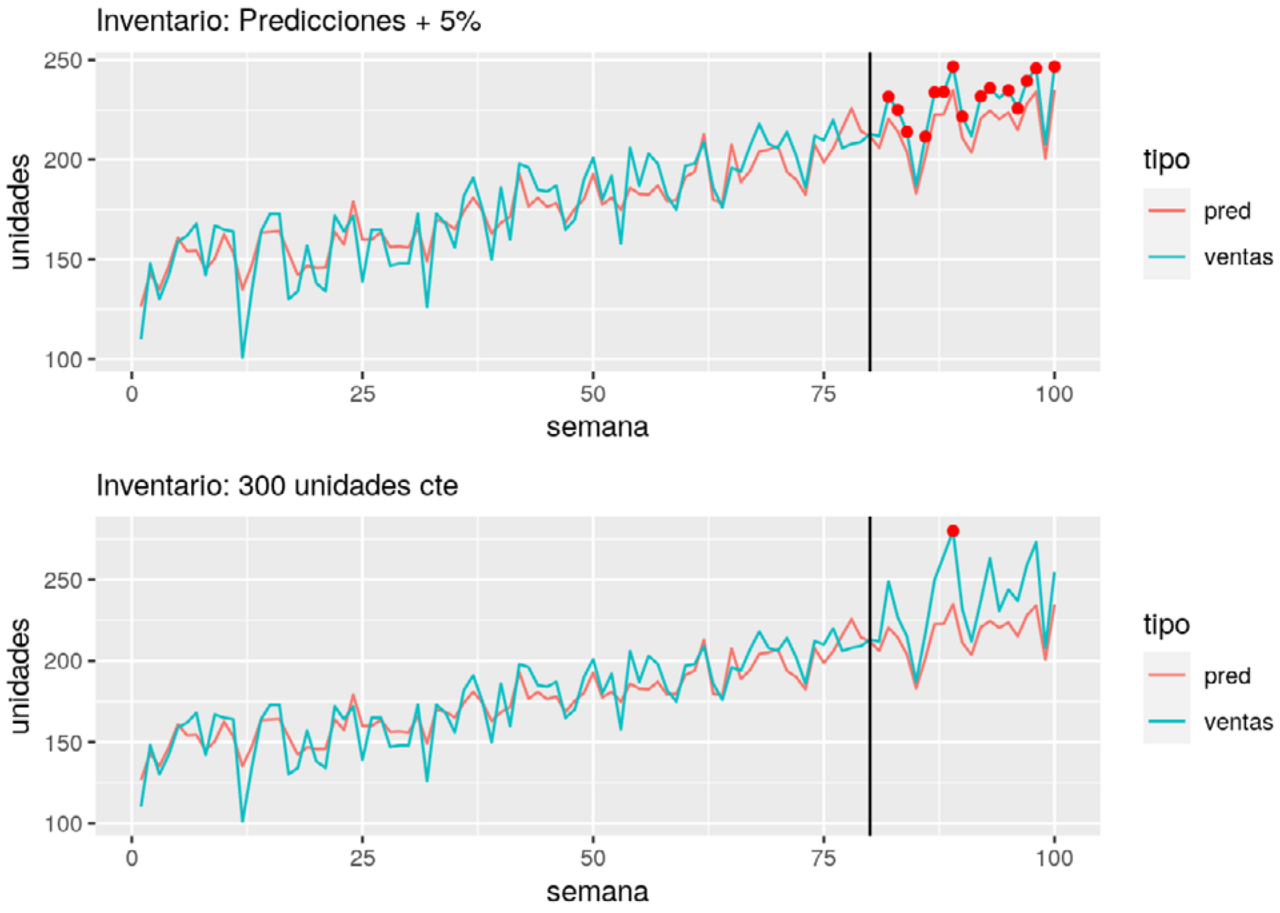
```
preds <- predict(mod_lineal, newdata = ventas)

ventas_larga <- ventas %>% mutate(pred = preds) %>%

  pivot_longer(cols = all_of(c("ventas","pred")), names_to = "tipo", values_to
= "unidades")

ggplot(ventas_larga %>% mutate(unidades = ifelse(tipo=="ventas" & semana > 80,
NA, unidades)), aes(x = semana)) +

  geom_line(aes(group = tipo, colour = tipo, y = unidades)) +

  geom_point(data = filter(ventas, agotamiento==1, semana < 80), aes(y = ventas),
colour = "red") +

  geom_vline(xintercept = 80) +

  geom_vline(xintercept = 60) +

  annotate("text", x = 25, y=105, label = "entrena") +

  annotate("text", x = 69, y=105, label = "prueba")
```



Si usáramos la política sugerida por las predicciones (por ejemplo, 5 % más), veríamos las ventas de la primera gráfica a continuación. Sin embargo, si utilizáramos una política de inventario con 280 unidades, observaríamos:

```
preds <- predict(mod_lineal, newdata = ventas)

ventas_obs <- ventas %>% mutate(pred = preds) %>%

  mutate(inventario = 1.05 *  pred) %>%

  mutate(ventas = ifelse(semana > 80, pmin(inventario, demanda), ventas))

ventas_larga <- ventas_obs %>%

  pivot_longer(cols = all_of(c("ventas","pred")), names_to = "tipo", values_to
= "unidades")

g1 <- ggplot(ventas_larga, aes(x = semana)) +

  geom_line(aes(group = tipo, colour = tipo, y = unidades)) +

  geom_point(data = filter(ventas_obs, ventas == inventario, semana > 80), aes(y
= ventas), colour = "red") +

geom_vline(xintercept = 80) + labs(subtitle = "Inventario: Predicciones + 5%")

preds <- predict(mod_lineal, newdata = ventas)

ventas_obs <- ventas %>% mutate(pred = preds) %>%

  mutate(inventario = 280) %>%

  mutate(ventas = ifelse(semana > 80, pmin(inventario, demanda), ventas))
```

```
ventas_larga <- ventas_obs %>%

  pivot_longer(cols = all_of(c("ventas","pred")), names_to = "tipo", values_to
= "unidades")

g2 <- ggplot(ventas_larga, aes(x = semana)) +

  geom_line(aes(group = tipo, colour = tipo, y = unidades)) +

  geom_point(data = filter(ventas_obs, ventas == inventario, semana > 80), aes(y
= ventas), colour = "red") +

  geom_vline(xintercept = 80)+ labs(subtitle = "Inventario: 300 unidades cte")

g1 / g2
```



Entonces,

- La política basada en las predicciones exacerba el problema de los agotamientos.
- Un uso no pensado de datos, sin considerar el proceso generador de los mismos, puede producir errores grandes en las decisiones.
- En este caso, la confusión proviene de no separar los conceptos de demanda y ventas. Otros indicadores de demanda o modelos más adecuados ayudarían a resolver el problema.
- Soluciones simplistas, como solo tomar los datos donde no ocurren agotamientos, pueden empeorar aún más la situación: incrementan el sesgo (seleccionamos semanas donde las ventas tienden a ser bajas) y reducen la precisión.

Muestras probabilísticas y naturales

- Cuando las muestras de entrenamiento son diferentes a las poblaciones donde van a aplicarse los modelos, existen dificultades en validar correctamente las predicciones.
- Para este ejemplo se utilizarán datos de la encuesta nacional de ingresos y gastos en hogares de México (INEGI, 2014), para simular el escenario que queremos ejemplificar:

```
set.seed(128)

encuesta_ingreso <- read_csv("datos/enigh-ejemplo.csv")

datos_ingreso <- encuesta_ingreso %>%

  mutate(num_focos = FOCOS) %>%

  mutate(ingreso_miles = (INGCOR / 1000)) %>%

  mutate(tel_celular = ifelse(SERV_2 == 1, "Sí", "No")) %>%

  mutate(piso_firme = ifelse(PISOS != 1 | is.na(PISOS), "Sí", "No")) %>%

  mutate(lavadora = ifelse(LAVAD != 1 | is.na(LAVAD), "Sí", "No")) %>%

  mutate(automovil = VEHI1_N > 0) %>%

  mutate(marginacion = fct_reorder(marginación, ingreso_miles, median)) %>%

  rename(ocupadas = PEROCU) %>%

  rename(educacion_jef = NIVELAPROB) %>%

  select(ingreso_miles, num_focos, tel_celular,

  marginacion, ocupadas, piso_firme, lavadora, automovil, educacion_jef)

  ingreso_split <- initial_split(datos_ingreso, prop = 0.7)

   entrena <- training(ingreso_split)

prueba <- testing(ingreso_split)
```

Supóngase que interesa estimar el ingreso de los hogares. Para ello se usa una encuesta por teléfono celular; más aún, supóngase que solo se accede a zonas que no tienen marginación muy alta.

```
muestra_sesgada <- filter(entrena,

                          tel_celular == "Sí",

                          marginacion=="Muy bajo")

sesgados_split <- initial_split(muestra_sesgada)

entrena_sesgo <- training(sesgados_split)

validacion_sesgo <- testing(sesgados_split)
```

Se construye un modelo lineal para el logaritmo de ingresos con los datos disponibles.

```
library(splines)

formula <- as.formula("log(ingreso_miles) ~ ns(num_focos, 3) +

              ns(ocupadas, 3) +  lavadora + automovil + piso_firme +

              ns(educacion_jef, 3)")

mod_sesgo <- lm(formula, data = entrena_sesgo)

# tomamos una muestra representativa para comparar, del mismo tamaño que la sesgada

mod_representativa <- lm(formula, data = sample_n(entrena, nrow(entrena_sesgo)))
```

Y se evalúa el error en una muestra de prueba construida con datos con las mismas características sesgadas que los datos de entrenamiento (hogares con teléfono celular y grado de marginación muy bajo).

```
preds_val <- predict(mod_sesgo, newdata = validacion_sesgo)

mean(abs(preds_val - log(1 + validacion_sesgo$ingreso_miles))) %>% round(2)
```

```
## [1] 0.37
```

El error en una muestra más similar a la población donde se pretende aplicar el algoritmo es mayor:

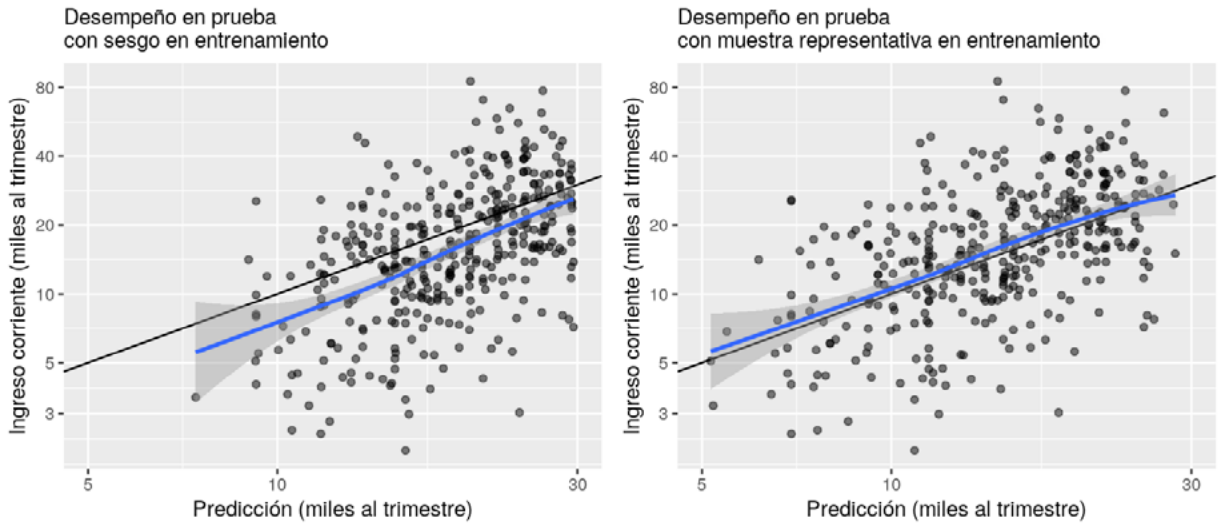
```
preds_prueba_sesgo <- predict(mod_sesgo, newdata = prueba)

preds_prueba <- predict(mod_representativa, newdata = prueba)
```

```
prueba$pred_sesgada <- preds_prueba_sesgo

prueba$pred_rep <- preds_prueba

mean(abs(preds_prueba_sesgo - log(1 + prueba$ingreso_miles))) %>% round(2)
```

```
## [1] 0.42
```

Sin embargo, el principal problema se refleja en la siguiente gráfica, donde se usan escalas logarítmicas para hacer comparaciones multiplicativas, que interesan por la naturaleza del ingreso. Cada punto representa un hogar. La muestra es más similar a la población donde se aplicará la metodología. En el eje horizontal se grafica la predicción de los hogares, utilizando el modelo, mientras que el eje vertical corresponde al ingreso de cada hogar. Como referencia se agrega la recta y = x, y un suavizador. El foco es en el desempeño para los hogares de ingresos relativamente bajos (menos de 10.000 MXN al mes):

```
breaks_y <- c(3, 5, 10, 20, 40, 80)

g_sesgo <- ggplot(prueba %>% filter(pred_sesgada < log(30)),

  aes(x = exp(pred_sesgada), y = ingreso_miles)) +

  geom_point(alpha = 0.5) +

  geom_abline() + geom_smooth(method = "loess", span = 1) +

  scale_x_log10(limits=c(5, 30)) + scale_y_log10(breaks = breaks_y) +

  xlab("Predicción (miles al trimestre)") +

  ylab("Ingreso corriente (miles al trimestre)") +

  labs(subtitle = "Desempeño en prueba \ncon sesgo en entrenamiento")

g_representativa <- ggplot(prueba %>% filter(pred_sesgada < log(30)),

  aes(x = exp(pred_rep), y = ingreso_miles)) +

  geom_point(alpha = 0.5) +

  geom_abline() + geom_smooth(method = "loess", span = 1) +

  scale_x_log10(limits = c(5, 30)) + scale_y_log10(breaks = breaks_y) +

  xlab("Predicción (miles al trimestre)") +

  ylab("Ingreso corriente (miles al trimestre)") +

   labs(subtitle = "Desempeño en prueba \ncon muestra representativa en entre-
namiento")

g_sesgo + g_representativa
```



Aunque comúnmente se espera sobrepredecir valores observados relativamente bajos, y lo contrario para valores relativamente altos, para quienes tienen ingresos de menos de 10.000 MXN mensuales el modelo sesgado sobrepredice el ingreso verdadero alrededor de 40 %:

```
prueba_bajo <- prueba %>% filter(ingreso_miles < 3*10)

sesgo <- mean(exp(prueba_bajo$pred_sesgada))/mean(prueba_bajo$ingreso_miles) - 1

round(sesgo,3)
```

```
## [1] 0.412
```

Al compararse con el mismo modelo entrenado con una muestra representativa, donde el efecto es considerablemente menor:

```
prueba_bajo <- prueba %>% filter(ingreso_miles < 3*10)

sesgo <- mean(exp(prueba_bajo$pred_rep))/mean(prueba_bajo$ingreso_miles) - 1

round(sesgo,3)
```

```
## [1] 0.152
```

Se tiene, entonces, dos problemas:

1. El sesgo produce un error considerablemente más grande en la implementación que en la validación.
2. Peor aún, el sesgo es mayor para hogares de menores ingresos (las predicciones son altas), lo cual puede producir una focalización mediocre si se busca identificar hogares de menores ingresos.

Muestras naturales: comparaciones causales

Este ejemplo está tomado de (Hastie, Tibshirani, & Friedman, 2017) y (Rossouw, 1983). Se consideran los siguientes datos, donde el objetivo es predecir enfermedad del corazón (chd)[12]:

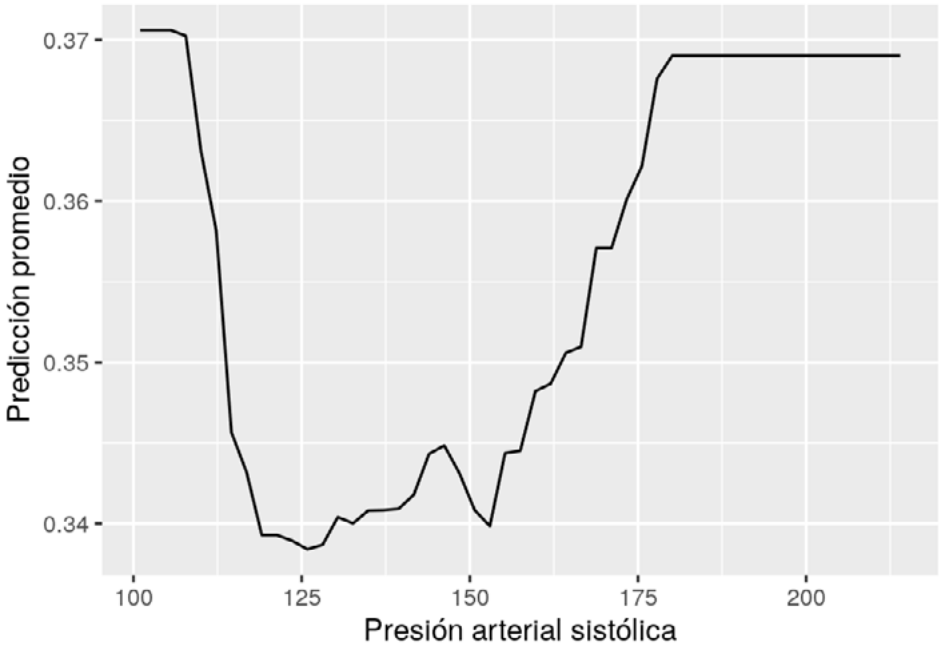
```
sa_heart <- read_csv("datos/sa-heart.csv")

sa_heart <- sa_heart %>%

  rename(presion_arterial = sbp, tabaco = tobacco, colesterol_ldl = ldl,

         adiposidad = adiposity, historia_fam = famhist, tipo_a = typea, obesi-
dad = obesity,

         edad = age, enf_coronaria = chd)

sa_heart

## # A tibble: 462 x 10

##    presion_arterial tabaco colesterol_ldl adiposidad historia_fam tipo_a

##               <dbl>  <dbl>          <dbl>      <dbl> <chr>         <dbl>

##  1              160  12              5.73       23.1 Present          49

##  2              144   0.01           4.41       28.6 Absent           55

##  3              118   0.08           3.48       32.3 Present          52

##  4              170   7.5            6.41       38.0 Present          51

##  5              134  13.6            3.5        27.8 Present          60

##  6              132   6.2            6.47       36.2 Present          62

##  7              142   4.05           3.38       16.2 Absent           59

##  8              114   4.08           4.59       14.6 Present          62

##  9              114   0              3.83       19.4 Present          49

## 10              132   0              5.8        31.0 Present          69

## # … with 452 more rows, and 4 more variables: obesidad <dbl>, alcohol <dbl>,

## #   edad <dbl>, enf_coronaria <dbl>

library(recipes)

set.seed(125)

sa_split <- rsample::initial_split(sa_heart, prop = 0.75)

sa_split
```

12 Datos accesibles en http://archive.ics.uci.edu/ml/datasets/heart+Disease

```
## <Training/Validation/Total>

## <347/115/462>

receta_sa <- training(sa_split) %>%

  recipe(enf_coronaria ~ .) %>%

  step_dummy(historia_fam) %>%

  step_mutate(enf_coronaria = factor(enf_coronaria)) %>%

prep()

sa_entrena <- receta_sa %>% juice

sa_boosted <- boost_tree(trees = 3000, mode = "classification",

                         learn_rate = 0.001, tree_depth = 2,

                         sample_size = 0.5) %>%

  set_engine("xgboost") %>%

  fit(enf_coronaria ~ ., data = sa_entrena)
```

Se puede evaluar este modelo y también afinar parámetros. Aquí interesa interpretar el efecto de las variables en este modelo. Para eso se considera la gráfica de dependencia parcial de la prevalencia de enfermedad de corazón y la variable obesidad:

```
library(pdp)

pdp_ob <- pdp::partial(sa_boosted$fit,  pred.var = "presion_arterial",

  plot = TRUE, plot.engine = "ggplot2", prob = TRUE,

  train = sa_entrena %>% dplyr::select(-enf_coronaria))

pdp_ob + xlab("Presión arterial sistólica") + ylab("Predicción promedio")
```



La interpretación correcta de esta gráfica de dependencia parcial (Hastie, Tibshirani, & Friedman, 2017) depende del hecho de que este es un estudio retrospectivo, donde algunos pacientes con riesgo de enfermedad de corazón tuvieron intervenciones para reducir su riesgo, entre los que está tomar medicinas para reducir la presión. Una interpretación causal de reducciones de la presión arterial como promotora de enfermedades del corazón es incorrecta y potencialmente peligrosa.

Desarrollo de modelo y validación

Contaminación entrenamiento-validación

Se presentan a continuación varios ejemplos de cómo las fugas de entrenamiento a validación producen estimaciones sesgadas del desempeño de predictores.

Selección de variables antes de dividir los datos

Cualquier paso de preprocesamiento debe hacerse sin usar datos de validación. Esto incluye cuando se usan métodos como validación cruzada.

Este ejemplo es originalmente de (Hastie, Tibshirani, & Friedman, 2017). Se utilizarán datos sintéticos, generados con el siguiente proceso:

1. Simulando variables respuesta y con distribución binomial,
2. Simulando 1000 covariables independientes, cada una con distribución normal estándar.

```
simular <- function(n = 100, p = 500, prob = 0.5){
  datos <- map(1:p, ~ rnorm(n)) %>%
    bind_cols()
  datos$y <- rbinom(n, 1, prob)
  datos
}
set.seed(8234)
datos_entrena <- simular(n = 200, p = 1000)
datos_prueba <- simular(n = 2000, p = 1000)
dim(datos_entrena)
```

```
## [1]  200 1001
```

```
datos_entrena %>% group_by(y) %>% tally() %>% kable()
```

y	n
0	113
1	87

La selección de variables está dada por la siguiente función. Esta función selecciona las variables más correlacionadas con la variable objetivo:

```
seleccionar <- function(datos, num_var = 10){
  correlaciones <- datos %>%
   pivot_longer(cols = matches("V"), names_to = "variable", values_to = "x") %>%
```

```
    group_by(variable) %>%
    summarise(corr = abs(cor(y, x))) %>%
    arrange(desc(corr))
  # seleccionar
  seleccionadas <- correlaciones %>%
    top_n(num_var, wt = corr) %>%
    pull(variable)
  datos %>% select(one_of(c("y", seleccionadas)))
}
```

Método erróneo

Aquí se ven las 10 variables que fueron seleccionadas. Por sí solo este método no es incorrecto, pero cuando se ejecuta sobre los datos que se usarán en validación (validación cruzada), entonces la estimación de desempeño es optimista:

```
datos_filtrados <- seleccionar(datos_entrena)

datos_filtrados %>% head %>%
  mutate_if(is.numeric, round, 3) %>% kable()
```

y	V337	V464	V984	V461	V525	V732	V39	V774	V491	V682
0	1.592	-0.587	1.763	-0.847	0.452	-0.604	-0.400	-1.146	-0.938	0.136
0	1.782	0.604	0.739	-0.533	1.752	0.945	1.142	-0.638	-0.342	-1.308
0	1.528	0.635	-0.326	0.734	-0.207	-0.974	1.574	2.401	0.428	0.176
0	0.799	-1.436	0.724	0.366	1.680	0.476	0.376	-1.673	-0.683	0.161
0	0.759	-0.208	-0.373	0.208	-1.009	-0.028	-1.209	0.759	2.038	1.402
1	-0.377	-1.044	1.358	-0.223	0.469	1.221	0.582	0.378	-0.116	0.173

Para cualquier corte de validación que se haga (ya sea que se separa un conjunto de datos o se hace validación cruzada), el porcentaje de aciertos parece ser mayor a 0.5:

```
corte_validacion <- datos_filtrados %>% sample_frac(0.7)

valida <- anti_join(datos_filtrados, corte_validacion)
```

```
modelo_1 <- glm(y ~ ., corte_validacion, family = "binomial")

mean(as.numeric(predict(modelo_1, valida) > 0) == valida$y) %>% round(2)
```

```
## [1] 0.73
```

Sin embargo, el desempeño real del modelo será:

```
mean(as.numeric(predict(modelo_1, datos_prueba) > 0) == datos_prueba$y) %>%
round(2)
```

```
## [1] 0.49
```

Método correcto

La selección de variables debe hacerse en cada vuelta de validación cruzada:

```
corte_validacion <- datos_entrena %>% sample_frac(0.7)

datos_filtrados_corte <- seleccionar(corte_validacion)

valida <- anti_join(datos_entrena, corte_validacion)

modelo_1 <- glm(y ~ ., datos_filtrados_corte, family = "binomial")

mean(as.numeric(predict(modelo_1, valida) > 0) == valida$y) %>% round(2)
```

```
## [1] 0.52
```

Sobremuestrear antes de particionar

Una de las formas para resolver problemas de desbalance de clases es la técnica de sobremuestreo. Sin embargo, se tiene que ser muy cuidadoso para evitar errores de fuga de información al aplicar estas técnicas.

En este ejemplo se verá que sobremuestrear una clase reducida antes de separar datos de validación o hacer validación cruzada puede producir estimaciones demasiado optimistas del error de predicción.

Supongamos que tenemos desbalance severo entre nuestras dos clases:

```
set.seed(99134)

datos_desbalance <- simular(n = 500, p = 20, prob = 0.1) %>%

  mutate(y = factor(y, levels = c(1, 0)))

datos_desbalance %>% group_by(y) %>% tally() %>% kable()
```

y	n
1	41
0	459

Manera incorrecta

Supóngase que primero se aplica (SMOTE) (Chawla, 2002) para intentar balancear los datos:

```
receta_balance <- recipe(y ~ ., datos_desbalance) %>%
step_smote(y) %>%
  prep()
datos_smote <- juice(receta_balance)
```

Obteniendo así,

```
datos_smote %>% group_by(y) %>% tally() %>% kable()
```

y	n
1	459
0	459

Ahora se separa entrenamiento y validación:

```
sep_datos_smote <- initial_split(datos_smote)
entrena_smote <- training(sep_datos_smote)
prueba_smote <- testing(sep_datos_smote)
```

Y se genera un método de clasificación usando un bosque aleatorio de árboles de decisión:

```
metricas <- metric_set(accuracy, recall, precision)
bosque <- rand_forest(trees = 500, mtry = 20, mode = "classification") %>%
  set_engine("ranger") %>%
  fit(y ~ ., data = entrena_smote)
bosque %>%
  predict(prueba_smote) %>%
  bind_cols(prueba_smote) %>%
  metricas(truth = y, estimate = .pred_class) %>%
  mutate_if(is.numeric, round, 3) %>% kable
```

.metric	.estimator	.estimate
accuracy	binary	0.926
recall	binary	0.973
precision	binary	0.887

En primera instancia parece que el desempeño es muy bueno. Se sabe que esto es ficticio, pues no hay relación de y con el resto de las covariables.

Manera correcta

Antes de hacer el rebalanceo de clases se separan entrenamiento y validación. Si se quiere, esta parte puede hacerse usando muestreo estratificado, por ejemplo, pero aquí se construye con muestreo aleatorio simple:

```
sep_datos <- initial_split(datos_desbalance, prop = 0.5)

entrena <- training(sep_datos)

prueba <- testing(sep_datos)

receta_balance <- recipe(y ~ ., data = entrena) %>%

  step_smote(y) %>%

  prep()

entrena_balanceado <- juice(receta_balance)

bosque_1 <- rand_forest(trees = 500, mtry = 20, mode = "classification") %>%

  set_engine("ranger") %>%

  fit(y ~ ., data = entrena_balanceado)

bosque_1 %>%

predict(prueba) %>%

bind_cols(prueba) %>%

  metricas(truth = y, estimate=.pred_class) %>%

  mutate_if(is.numeric, round, 3) %>%

  kable()
```

.metric	.estimator	.estimate
accuracy	binary	0.828
recall	binary	0.000
precision	binary	0.000

Aunque el accuracy parece alto, la precisión y la sensibilidad son cero. Un clasificador trivial que siempre predice la clase dominante puede tener mejor exactitud que el que hemos construido.

Variables no disponibles en el momento de predicción

En este caso mostramos un ejemplo donde se utiliza erróneamente una variable que no estará disponible en el momento de hacer las predicciones (datos de (Greene, 2003)).

```
credito <- read_csv("datos/AER_credit_card_data.csv") %>%
  rename(gasto = expenditure, dependientes = dependents, ingreso = income,
         edad = age, propietario = owner) %>%
  mutate(propietario = fct_recode(propietario, c(si = "yes")))
credito %>% head %>%
  mutate_if(is.numeric, round, 1) %>% kable()
```

card	reports	edad	ingreso	share	gasto	propietario	selfemp	dependientes	months	majorcards	active
yes	0	37.7	4.5	0.0	125.0	si	no	3	54	1	12
yes	0	33.2	2.4	0.0	9.9	no	no	3	34	1	13
yes	0	33.7	4.5	0.0	15.0	si	no	4	58	1	5
yes	0	30.5	2.5	0.1	137.9	no	no	0	25	1	7
yes	0	32.2	9.8	0.1	546.5	si	no	2	64	1	5
yes	0	23.2	2.5	0.0	92.0	no	no	0	54	1	1

Se quiere construir un modelo para predecir qué solicitudes fueron aceptadas y automatizar el proceso de selección. Se usa una regresión logística con Keras y penalización L2:

```
set.seed(823)
credito_split <- initial_split(credito)
entrena <- training(credito_split)
prueba <- testing(credito_split)
# preparacion de datos
credito_receta <- recipe(card ~ ., credito) %>%
```

```
  step_normalize(all_numeric()) %>%
  step_dummy(all_nominal(), -card)
# modelo
modelo_regularizado <-
  logistic_reg(penalty = 1) %>%
  set_engine("keras", epochs = 500, verbose = FALSE)  %>%
  set_mode("classification")
# ajustar parametros de preprocesamiento
receta_prep <- credito_receta %>% prep(entrena)
# preprocesar datos
entrena_prep <- bake(receta_prep, entrena)
prueba_prep <- bake(receta_prep, prueba)
# ajustar modelo
ajuste <- modelo_regularizado %>%
  fit(card~ gasto + dependientes + ingreso + edad + propietario_si, data = entre-
na_prep)
# evaluar
metricas <- metric_set(accuracy, recall, precision)
ajuste %>% predict(prueba_prep) %>%
  bind_cols(prueba) %>%
  metricas(truth = factor(card), estimate = .pred_class) %>%
  mutate_if(is.numeric, round, 3) %>%
                                    kable()
```

.metric	.estimator	.estimate
accuracy	binary	0.833
recall	binary	0.393
precision	binary	0.892

Y parece tener un desempeño razonable. Si quitamos la variable gasto ("expenditure") se degrada totalmente el desempeño del modelo:

```
ajuste_2 <- modelo_regularizado %>%
  fit(card~ dependientes + ingreso + edad + propietario_si, data = entrena_prep)
ajuste_2 %>% predict(prueba_prep) %>%
  bind_cols(prueba) %>%
  metricas(truth = factor(card), estimate = .pred_class) %>%
  mutate_if(is.numeric, round, 3) %>%
kable()
```

.metric	.estimator	.estimate
accuracy	binary	0.745
recall	binary	0.000
precision	binary	NA

La sensibilidad es muy mala y la precisión no puede calcularse, pues el modelo no hace predicciones positivas para el conjunto de prueba.

La razón de esta degradación en el desempeño es que gasto se refiere a uso de tarjetas de crédito. Esto incluye la tarjeta para la que se quiere hacer predicción de aceptación:

```
entrena %>%
  mutate(algun_gasto = gasto > 0) %>%
  group_by(algun_gasto, card) %>%
  tally() %>%
  kable()
```

algun_gasto	card	n
FALSE	no	212
FALSE	yes	19
TRUE	yes	759

Lo que indica que algún gasto probablemente incluye el gasto en la tarjeta actual. La variable gasto es medida posteriormente a la entrega de la tarjeta:

- El desempeño de este modelo para nuevas aplicaciones será muy malo, pues la variable gasto, en el momento de la aplicación, evidentemente no cuenta cuánto va a gastar cada cliente en el futuro.

Puntos de corte arbitrarios

Las mejores decisiones de punto de corte pueden hacerse con análisis de costo beneficio, con curvas tipo lift, como las del ejemplo anterior, basadas en ganancias y pérdidas de cada decisión. Aunque esta información muchas veces no está disponible, es la situación ideal para evaluar cómo ayuda el modelo y cuánto valen las acciones que pretendemos tomar. Es posible hacer este análisis con valores inciertos de costo beneficio.

Supóngase que estamos pensando en un tratamiento para retener estudiantes en algún programa de entrenamiento o mejora.

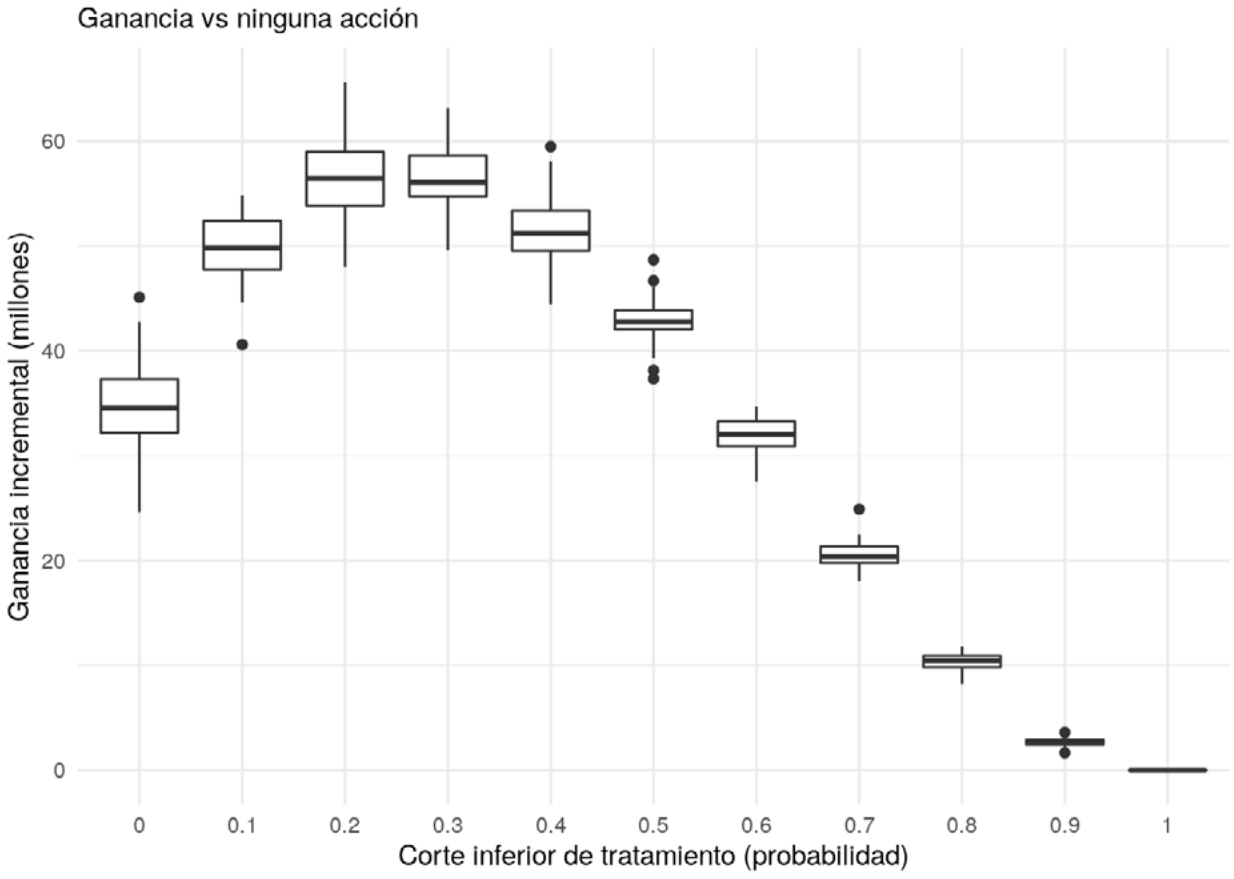
- El tratamiento de retención cuesta 5000 MXN por alumno.
- Se estima mediante experimentos o algún análisis externo que el tratamiento reduce la probabilidad de abandono en 60 %.
- Se tiene algún tipo de valuación del valor social de que un alumno persista en el programa.

Se puede evaluar el modelo en el contexto del problema en la siguiente forma:

- Suponiendo que se tratará un porcentaje de los estudiantes con mayor probabilidad de rotar.
- Se calcula el costo esperado si se trata a un porcentaje de los estudiantes: se simula, reduciendo su probabilidad de abandono por el tratamiento y se suman los costos de tratarlos.
- Se compara contra el escenario de no aplicar ningún tratamiento.

No es necesario usar medidas muy técnicas para dar un resumen de cómo puede ayudar el tratamiento y modelo para mantener el valor de la cartera:

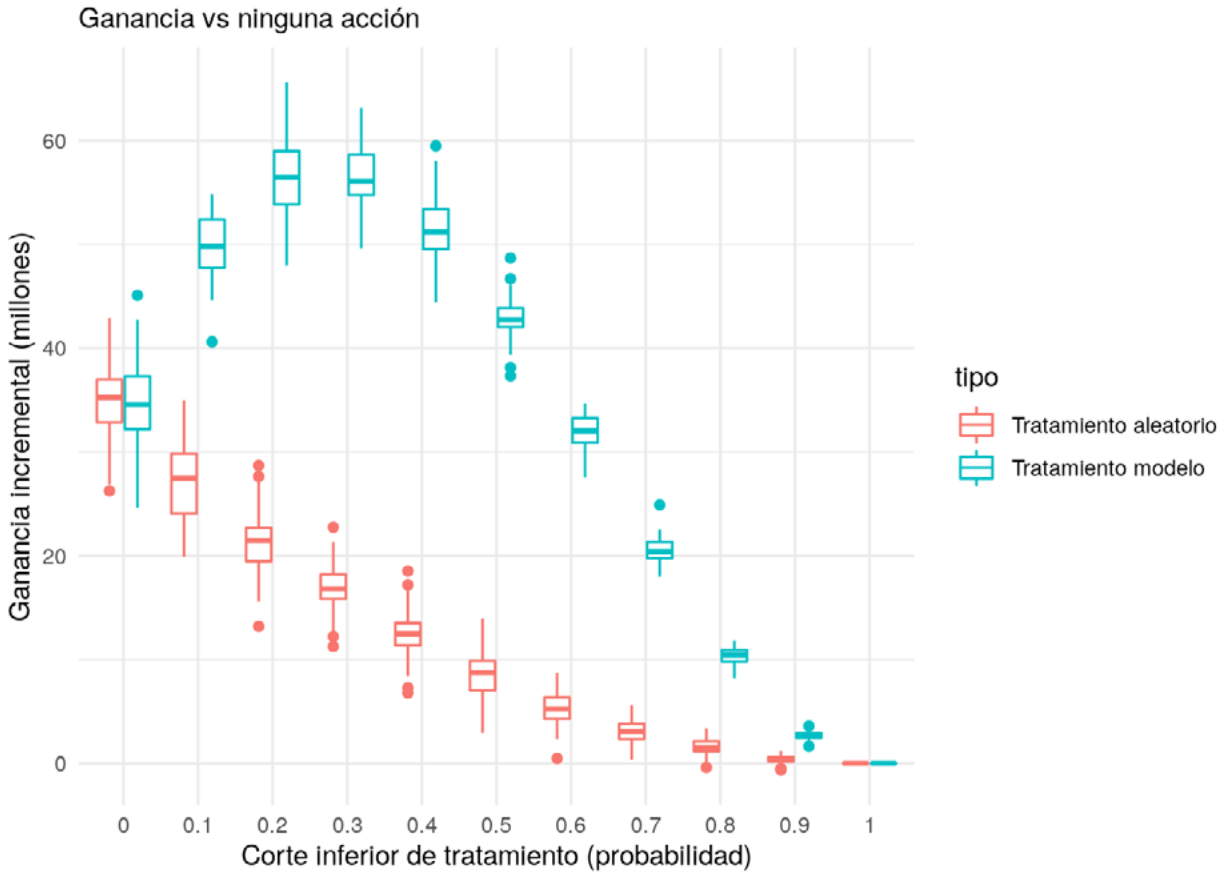
```
ggplot(filter(perdidas_sim, tipo=="Tratamiento modelo"),
       aes(x = factor(corte), y = - perdida / 1e6)) +
  geom_boxplot() + ylab("Ganancia incremental (millones)") +
  xlab("Corte inferior de tratamiento (probabilidad)") +
  labs(subtitle = "Ganancia vs ninguna acción") + theme_minimal()
```



Se puede escoger un punto de corte entre 0.2 y 0.3, por ejemplo, o hacer más simulaciones para refinar la elección.

Si quiere separarse el efecto del tratamiento y el efecto del tratamiento aplicado según el modelo, puede compararse con la acción que consiste en tratar a los estudiantes al azar:

```
ggplot(perdidas_sim, aes(x = factor(corte), y = - perdida / 1e6,
                         group = interaction(tipo, corte), colour = tipo)) +
  geom_boxplot() + ylab("Ganancia incremental (millones)") +
  xlab("Corte inferior de tratamiento (probabilidad)") +
  labs(subtitle = "Ganancia vs ninguna acción") + theme_minimal()
```



- La conclusión es que el modelo **ayuda considerablemente a la focalización del programa** (el área entre las dos curvas mostradas arriba).

Desbalance de clases

Cuando se tiene un desbalance severo de clases se puede enfrentar dos problemas: uno, existen en términos absolutos muy pocos elementos de una clase para poder discriminarla de manera efectiva (aun cuando se tengan los atributos o *features* correctos) y, dos, métodos usuales de evaluación de predicción son deficientes para evaluar el desempeño de predicciones.

Considérense los siguientes datos:

- "Los datos contienen 5822 registros de clientes reales. Cada registro consta de 86 variables, que contienen datos sociodemográficos (variables 1-43) y de propiedad del producto (variables 44-86). Los datos sociodemográficos se derivan de los códigos postales. Todos los clientes que viven en zonas con el mismo código postal tienen los mismos atributos sociodemográficos. La variable 86 (compra) indica si el cliente adquirió una póliza de seguro de caravanas" (James, 2017).[13]

13 Datos y más información accesible en http://www.liacs.nl/~putten/library/cc2000/data.html

Se quiere predecir la variable *purchase*:

```
caravan <- read_csv("datos/caravan.csv") %>%
  mutate(MOSTYPE = factor(MOSTYPE),
         MOSHOOFD = factor(MOSHOOFD)) %>%
  mutate(Compra = fct_recode(Purchase, si = "Yes", no = "No")) %>%
  mutate(Compra = fct_rev(Compra)) %>%
  select(-Purchase)
nrow(caravan)
```

```
## [1] 5822
```

```
caravan %>% count(Compra) %>%
  mutate(pct = 100 * n / sum(n)) %>%
  mutate(pct = round(pct, 2))
```

```
## # A tibble: 2 x 3
##   Compra     n   pct
##   <fct>  <int> <dbl>
## 1 si       348  5.98
## 2 no      5474 94.0
```

Esta es la distribución natural de respuesta que se ve en los datos y hay relativamente pocos datos en la categoría "Si".

Se usará muestreo estratificado para obtener proporciones similares en conjuntos de entrenamiento y prueba:

```
set.seed(823)
caravan_split = initial_split(caravan, strata = Compra, prop = 0.9)
caravan_split
```

```
## <Training/Validation/Total>
## <5240/582/5822>
```

```
entrena <- training(caravan_split)
prueba <- testing(caravan_split)
```

Y se usará regresión logística (lo mismo aplica para otros métodos que produzcan probabilidades de clase, como *boosting*, árboles aleatorios o redes neuronales):

```
library(tune)

# preparacion de datos

caravan_receta <- recipe(Compra ~ ., entrena) %>%

  step_dummy(all_nominal(), -Compra)

caravan_receta_prep <- caravan_receta %>% prep

# modelo

modelo_log <-

  logistic_reg() %>%

  set_engine("glm")  %>%

  set_mode(«classification») %>%

  fit(Compra ~ ., data = caravan_receta_prep %>% juice)
```

Análisis incorrecto

La matriz de confusión de los datos de entrenamiento es:

```
predictions_ent_glm <- modelo_log %>%

  predict(new_data = juice(caravan_receta_prep)) %>%

  bind_cols(juice(caravan_receta_prep) %>% select(Compra))

predictions_ent_glm %>%

  conf_mat(Compra, .pred_class)
```

```
##           Truth

## Prediction   si   no

##         si    6    9

##         no  299 4926
```

Y los de prueba:

```
prueba_procesado <- bake(caravan_receta_prep, prueba)

predictions_glm <- modelo_log %>%

  predict(new_data = prueba_procesado) %>%

  bind_cols(prueba_procesado %>% select(Compra))

predictions_glm %>%
```

```
  conf_mat(Compra, .pred_class)
```

```
##           Truth
## Prediction  si  no
##         si   0   4
##         no  43 535
```

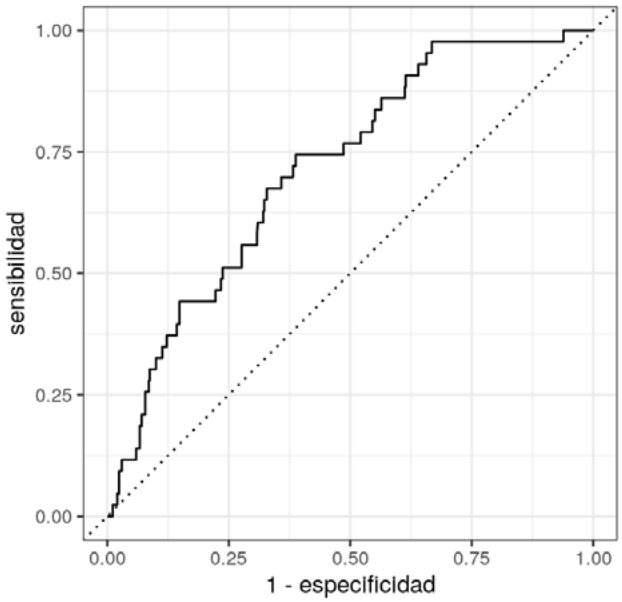
Y se obtiene un desempeño pobre según esta matriz de confusión (prueba y entrenamiento). La sensibilidad es muy baja, aunque la especificidad (tasa de correctos negativos) sea alta. Una conclusión típica es que *el modelo no tiene valor predictivo*, o que es *necesario sobremuestrear la clase de ocurrencia baja*.

Análisis correcto

En lugar de empezar con sobre/sub muestreo, que modifica las proporciones naturales de las categorías en los datos, es posible trabajar con probabilidades en lugar de predicciones de clase con punto de corte de 0.5.

Por ejemplo, puede visualizarse con una curva ROC (o curva lift, precision-recall, o alguna otra similar que tome en cuenta probabilidades):

```
predictions_prob <- modelo_log %>%
  predict(new_data = prueba_procesado, type = "prob") %>%
  bind_cols(prueba_procesado %>% select(Compra)) %>%
  select(.pred_si, Compra)
datos_roc <- roc_curve(predictions_prob, Compra, .pred_si)
autoplot(datos_roc) +
  xlab("1 - especificidad") + ylab("sensibilidad")
```



Donde se ve que es posible alcanzar buenos niveles de sensibilidad si se acepta alguna degradación en la especificidad, que originalmente es muy alta. Por ejemplo, cortando en 0.05 se puede obtener especificidad y sensibilidad que posiblemente sean adecuadas para el problema:

```
datos_roc %>% filter(abs(.threshold - 0.04) < 1e-4) %>% round(4)
```

```
## # A tibble: 2 x 3
##   .threshold specificity sensitivity
##        <dbl>       <dbl>       <dbl>
## 1     0.0399       0.553       0.744
## 2     0.0399       0.555       0.744
```

¿Qué pasa si se hace sub/sobremuestreo?

Se sobremuestrea:

```
caravan_receta_smote <- recipe(Compra ~ ., entrena) %>%
  step_dummy(MOSTYPE, MOSHOOFD) %>%
  step_smote(Compra)
smote_prep <- prep(caravan_receta_smote)
# modelo
entrena_1 <- juice(smote_prep)
entrena_1 %>% count(Compra)
```

```
## # A tibble: 2 x 2
##   Compra     n
##   <fct>  <int>
## 1 si      4935
## 2 no      4935
```

```
modelo_log_smote <-
  logistic_reg() %>%
  set_engine("glm")  %>%
  set_mode(«classification») %>%
  fit(Compra ~ ., data = entrena_1)
```

En entrenamiento la matriz de confusión es aparentemente mejor:

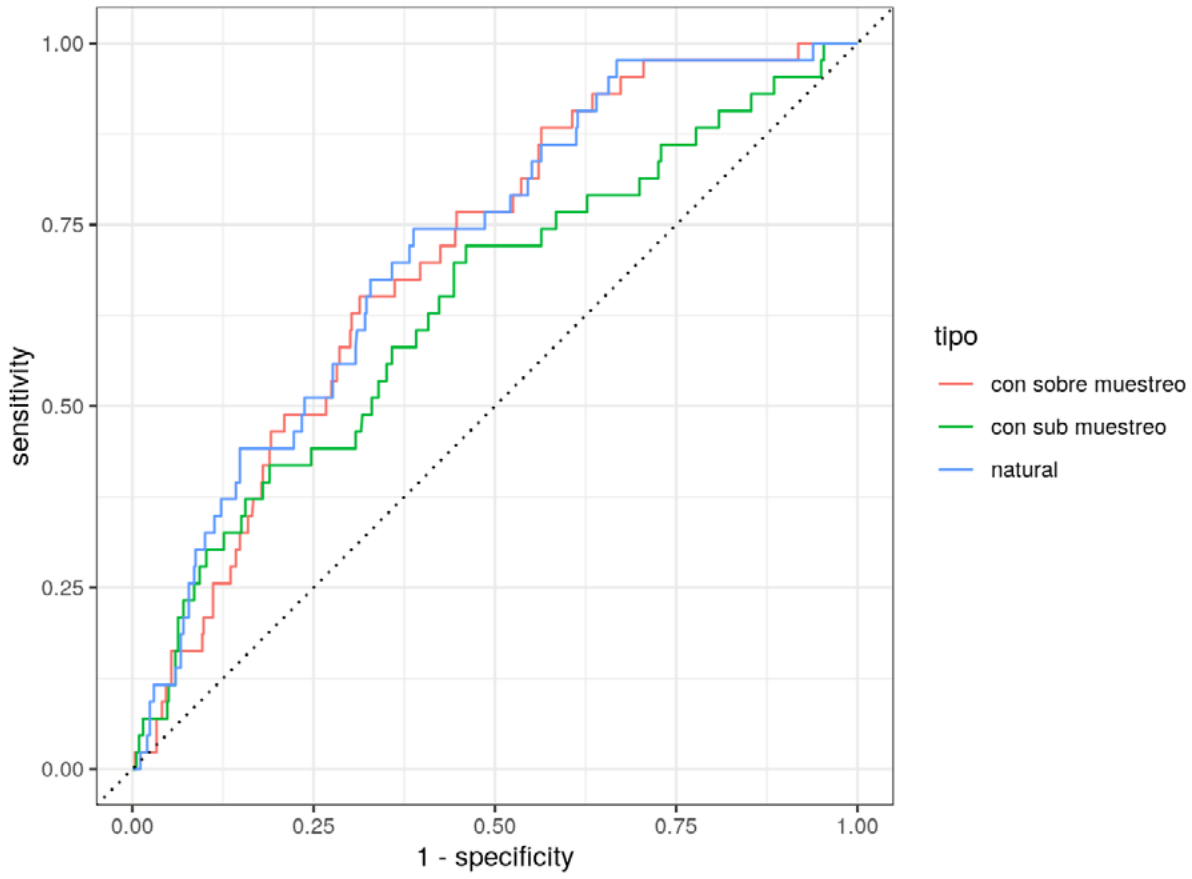
```
predictions_ent_glm <- modelo_log_smote %>%
  predict(new_data = entrena_1) %>%
bind_cols(entrena_1 %>% select(Compra))
predictions_ent_glm %>%
  conf_mat(Compra, .pred_class)
```

```
##            Truth
## Prediction   si   no
##          si 3854 1271
##          no 1081 3664
```

Pero en prueba, los resultados son muy similares. Se agrega también el modelo construido submuestreando la clase dominante:

```
entrena_sub <- caravan_receta %>% step_downsample(Compra) %>% prep() %>% juice
modelo_log_sub <-
  logistic_reg() %>%
  set_engine("glm")  %>%
  set_mode("classification") %>%
  fit(Compra ~ ., data = entrena_sub)
predictions_prob <- modelo_log_smote %>%
  predict(new_data = prueba_procesado, type = "prob") %>%
  bind_cols(prueba_procesado %>% select(Compra)) %>%
  select(.pred_si, Compra)
predictions_prob_sub <- modelo_log_sub %>%
  predict(new_data = prueba_procesado, type = "prob") %>%
  bind_cols(prueba_procesado %>% select(Compra)) %>%
 select(.pred_si, Compra)
datos_roc_smote <- roc_curve(predictions_prob, Compra, .pred_si)
datos_roc_sub <- roc_curve(predictions_prob_sub, Compra, .pred_si)
datos_roc_comp <- bind_rows(datos_roc %>% mutate(tipo = "natural"),
              datos_roc_smote %>% mutate(tipo = "con sobre muestreo"),
```

```
                datos_roc_sub %>% mutate(tipo = "con sub muestreo")
)

ggplot(datos_roc_comp,
       aes(x = 1 - specificity, y = sensitivity, colour = tipo)) +
  geom_path() +
  geom_abline(lty = 3) +
  coord_equal() +
  theme_bw()
```



- **El problema original no era que el ajuste no funcionaba, sino que se evaluó el punto de corte incorrecto**. Un punto de corte de 0.5 con SMOTE equivale a uno mucho más reducido sin SMOTE.

- Peor aún, l**as probabilidades del modelo construido con sobremuestreo no reflejan las tasas de ocurrencia de la respuesta que interesa**, lo cual puede producir resúmenes engañosos de las tasas de respuesta que se espera observar en producción.

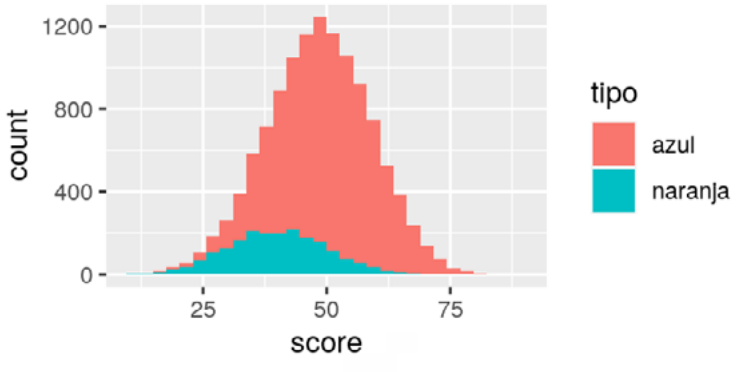
Equidad con atributos protegidos

El siguiente ejemplo es derivado de (Hardt, 2016). Supóngase que se tiene un atributo protegido A que tiene dos valores: azul y naranja. Naranja es el grupo minoritario desaventajado. Se usan datos simulados como sigue: el atributo score está asociado al atributo protegido.

```
 _logit <- function(x){

  1 / (1 + exp(-x))

}

simular_datos <- function(n = c(10000, 2000)){

  score_azul <- pmax(rnorm(n[1], 50, 10), 0)

  score_naranja <- pmax(rnorm(n[2], 40, 10), 0)

  azul <- tibble(tipo = "azul", score = score_azul)

  naranja <- tibble(tipo = "naranja", score = score_naranja)

  datos <- bind_rows(azul, naranja) %>%

    mutate(coef_0 = ifelse(tipo == "azul", 0.0, 0),

           prob_real_pos = inv_logit(-1 + coef_0 + 0.1 * (score-40))) %>%

    mutate(atr_1 = rpois(nrow(.), 3))

  datos %>% select(-coef_0) %>%

    mutate(paga = map_dbl(prob_real_pos, ~ rbinom(1, 1, .x))) %>%

    select(-prob_real_pos)

}

set.seed(1221)

tbl_datos <- simular_datos()
```

Usando un histograma para el score se obtiene un grupo minoritario con valores de la variable score más baja:

```
ggplot(tbl_datos, aes(x = score, fill = tipo)) + geom_histogram()
```



Se ajusta un modelo simple de regresión logística:

```
reg_log <- glm(paga ~  score + atr_1 + tipo, tbl_datos, family = "binomial")

tbl_datos <- tbl_datos %>% mutate(prob_pos = predict(reg_log, type = "response"))
```

Las tasas reales de cumplimiento son iguales para los dos grupos. En primer lugar, se considera una estrategia donde se aplica el mismo punto de corte para todos los grupos:

```
resultado_cortes <- function(tbl_datos, cortes){

  resultado <- tbl_datos %>%

     mutate(recibe = ifelse(tipo == "azul", prob_pos > cortes[1], prob_pos >
cortes[2]),

            decision = ifelse(recibe, "Aceptado", "Rechazado"))

  resultado %>% group_by(tipo, decision, paga) %>% count() %>%

    ungroup()

}

resultados_conteo <- resultado_cortes(tbl_datos, c(0.6, 0.6))

resultados_conteo
```

```
## # A tibble: 8 x 4

##   tipo    decision   paga     n

##   <chr>   <chr>     <dbl> <int>

## 1 azul    Aceptado      0   905

## 2 azul    Aceptado      1  2400

## 3 azul    Rechazado     0  4149
```

```
## 4 azul    Rechazado     1  2546
## 5 naranja Aceptado      0    47
## 6 naranja Aceptado      1   101
## 7 naranja Rechazado     0  1353
## 8 naranja Rechazado     1   499
```

```
resultados_conteo %>%
  group_by(tipo, decision) %>%
  summarise(n = sum(n)) %>%
  mutate(total = sum(n)) %>%
  mutate(prop = n / total) %>%
  filter(decision == "Aceptado")
```

```
## # A tibble: 2 x 5
## # Groups:   tipo [2]
##   tipo    decision     n total  prop
##   <chr>   <chr>    <int> <int> <dbl>
## 1 azul    Aceptado  3305 10000 0.330
 ## 2 naranja Aceptado   148  2000 0.074
```

Nótese que el grupo naranja ha recibido considerablemente menos aceptaciones que el grupo azul, tanto en totalidad como en proporción. Más aún, con la precisión o tasa de verdaderos positivos es posible evaluar qué proporción de los que cumplirían si fueran aceptados fueron aceptados según el punto de corte:

```
resultados_conteo %>%
  filter(paga == 1) %>%
  group_by(tipo) %>%
  mutate(tvp = n / sum(n)) %>%
  filter(decision == "Aceptado")
```

```
## # A tibble: 2 x 5
## # Groups:   tipo [2]
##   tipo    decision  paga     n   tvp
##   <chr>   <chr>    <dbl> <int> <dbl>
```

```
## 1 azul    Aceptado     1  2400 0.485
## 2 naranja Aceptado     1   101 0.168
```

Y se ve que el grupo naranja también está en desventaja, pues entre los que cumplen hay menos decisiones de aceptación.

El siguiente paso es considerar la paridad demográfica. En este caso, se decide dar el mismo número de préstamos a cada grupo, dependiendo de su tamaño:

```
calcular_puntos_paridad <- function(tbl_datos, prop){
  tbl_datos %>% group_by(tipo) %>%
    summarise(corte = quantile(prob_pos, 1 - prop))
}
cortes_paridad_tbl <- calcular_puntos_paridad(tbl_datos, 0.45)
cortes_paridad_tbl
```

```
## # A tibble: 2 x 2
##   tipo    corte
##   <chr>   <dbl>
## 1 azul    0.521
## 2 naranja 0.297
```

El corte para azul es más exigente que para naranja. En sí, eso no es un problema, pero se observa:

```
cortes_paridad <- cortes_paridad_tbl %>% pull(corte)
resultados_conteo <- resultado_cortes(tbl_datos, cortes_paridad)
resultados_conteo %>%
  filter(paga == 1) %>%
  group_by(tipo) %>%
  mutate(tvp = n / sum(n)) %>%
  filter(decision == "Aceptado")
```

```
## # A tibble: 2 x 5
## # Groups:   tipo [2]
##   tipo    decision  paga     n   tvp
##   <chr>   <chr>    <dbl> <int> <dbl>
```

```
## 1 azul    Aceptado     1  3094 0.626
## 2 naranja Aceptado     1   410 0.683
```

Y así que además de ser más exigente con el grupo azul, a los que cumplen del grupo azul también se les otorgan menos decisiones de aceptación. Además, se aceptan considerablemente menos personas de la población.

La solución de igualdad de oportunidad es cortar de forma que la tasa de aceptación dentro del grupo de los que pagan sea similar para ambas poblaciones, lo que ocurre aproximadamente en 0.35:

```
calcular_cortes_oportunidad <- function(tbl_datos, prop){
  tbl_datos %>%
    filter(paga==1) %>%
    group_by(tipo) %>%
    mutate(rank_p = rank(prob_pos) / length(prob_pos) ) %>%
    filter(rank_p < prop) %>%
    top_n(1, rank_p) %>%
    select(tipo, corte = prob_pos)
}
cortes_op <- calcular_cortes_oportunidad(tbl_datos, 0.35)
resultados_conteo <- resultado_cortes(tbl_datos, cortes_op %>% pull(corte))
resultados_conteo %>%
  filter(paga == 1) %>%
  group_by(tipo) %>%
  mutate(tvp = n / sum(n)) %>%
  filter(decision == "Aceptado")
```

```
## # A tibble: 2 x 5
## # Groups:   tipo [2]
##   tipo    decision  paga     n   tvp
##   <chr>   <chr>    <dbl> <int> <dbl>
## 1 azul    Aceptado     1  3215 0.650
## 2 naranja Aceptado     1   391 0.652
```

Nota: es importante notar que si la variable de resultado positivo es injustamente asignada, entonces este método no resuelve el problema. En este caso es relevante entender cuáles son los criterios con los que se considera un resultado exitoso dependiendo del grupo del atributo protegido (por ejemplo, si a un segmento particular se le permite mayores atrasos en los pagos y a otro menos, o un grupo se considera un delincuente reincidente por una ofensa mucho menor que otros grupos).

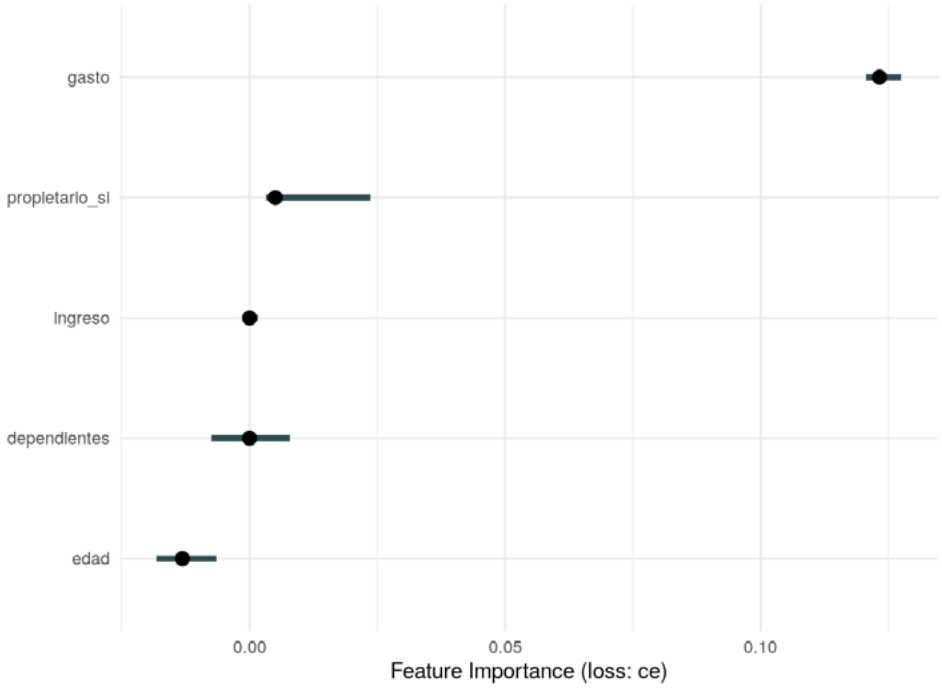
Rendición de cuentas

Interpretabilidad

Se pueden usar medidas como importancia de permutaciones para examinar modelos. En este ejemplo, se regresa al ejercicio de predicción de aceptación de solicitudes de crédito y se considera la importancia basada en permutaciones (Molnar, 2019):

```
set.seed(823)

credito_split <- initial_split(credito)

entrena <- training(credito_split)

prueba <- testing(credito_split)

# preparacion de datos

credito_receta <- recipe(card ~ ., credito) %>%

  step_normalize(all_numeric()) %>%

  step_dummy(all_nominal(), -card)

# modelo

modelo_regularizado <-

  logistic_reg(penalty = 1) %>%

  set_engine("keras", epochs = 500, verbose = FALSE)  %>%

  set_mode("classification")

# ajustar parametros de preprocesamiento

receta_prep <- credito_receta %>% prep(entrena)

# preprocesar datos

entrena_prep <- bake(receta_prep, entrena)

prueba_prep <- bake(receta_prep, prueba)

# ajustar modelo

ajuste <- modelo_regularizado %>%
```

```
  fit(card~ gasto + dependientes + ingreso + edad + propietario_si, data = en-
trena_prep)

library(iml)

modelo <- ajuste$fit

entrena_x <- entrena_prep %>% dplyr::select(gasto, dependientes, ingreso, edad,
propietario_si)

predictor <- Predictor$new(modelo, data = entrena_x, y = ifelse(entrena_prep$-
card == "yes",2,1) ,

                           type = "prob")

imp <- FeatureImp$new(predictor, loss = "ce", compare = "difference")

plot(imp) + theme_minimal()
```



- Se ve que, para esta red sin capas ocultas, la importancia se concentra en un solo predictor, *gasto*, que como se ve representa una fuga de información. Este diagnóstico es útil en general y, aunque no tan dramático como este ejemplo, puede señalar cuáles variables es importante considerar con cuidado.

- Es primordial considerar también el efecto de variables asociadas a grupos protegidos y, de ser necesario, examinar con cuidado cómo afectan las predicciones.

- Modelos parsimoniosos, que usan menos atributos, facilitan el análisis, el mantenimiento del flujo de datos y reducen la exposición a problemas de fugas o efectos indeseables.

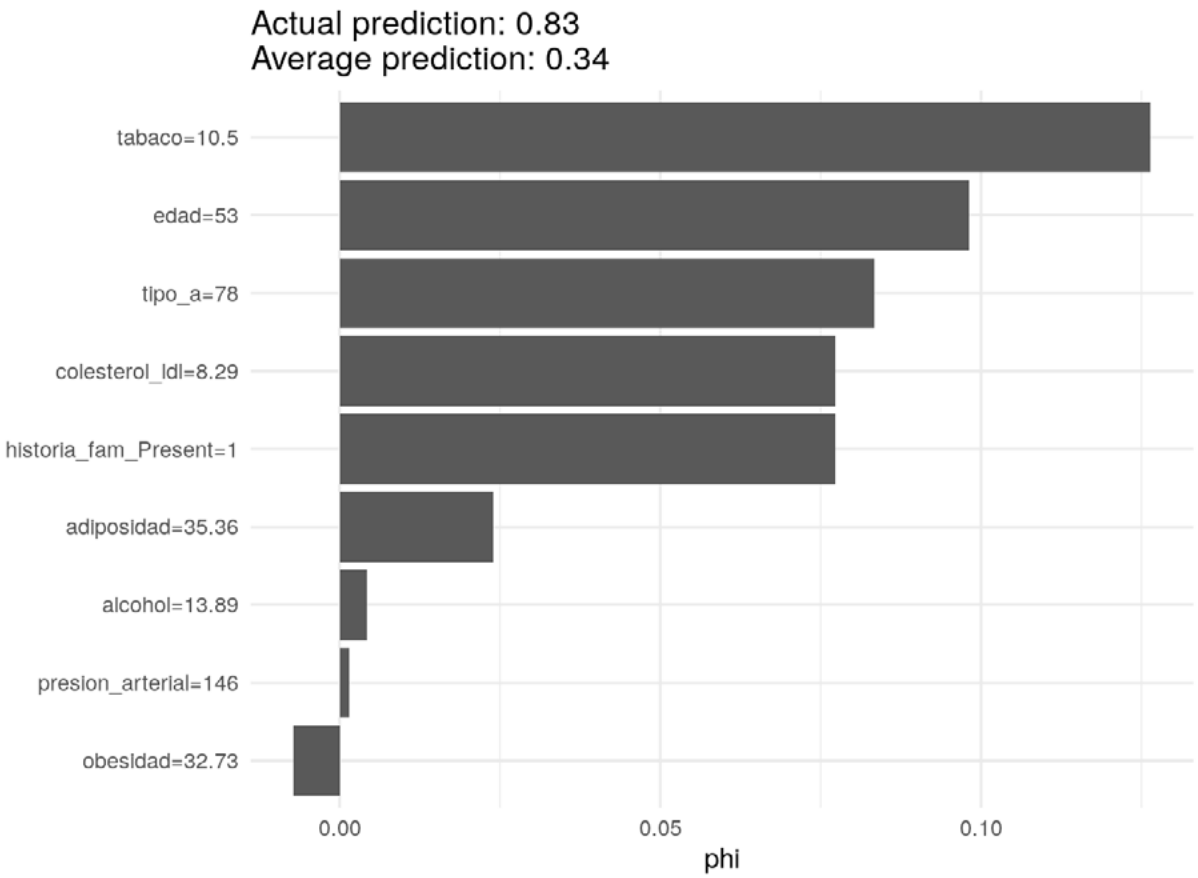
Explicación de predicciones

Para explicar predicciones individuales pueden usarse los valores de Shapley (Molnar, 2019), (Lundberg & Lee, 2017). Estas gráficas indican la contribución asignada de cada atributo a una predicción individual, con la idea de considerar efectos marginales sobre la predicción dependiendo de la presencia o ausencia de otros atributos. Las contribuciones obtenidas suman la diferencia que hay entre la predicción particular y la predicción promedio.

Pueden examinarse también promedios a lo largo de grupos de interés.

Considérese el ejemplo de factores para detectar una enfermedad del corazón (Rossouw, 1983):

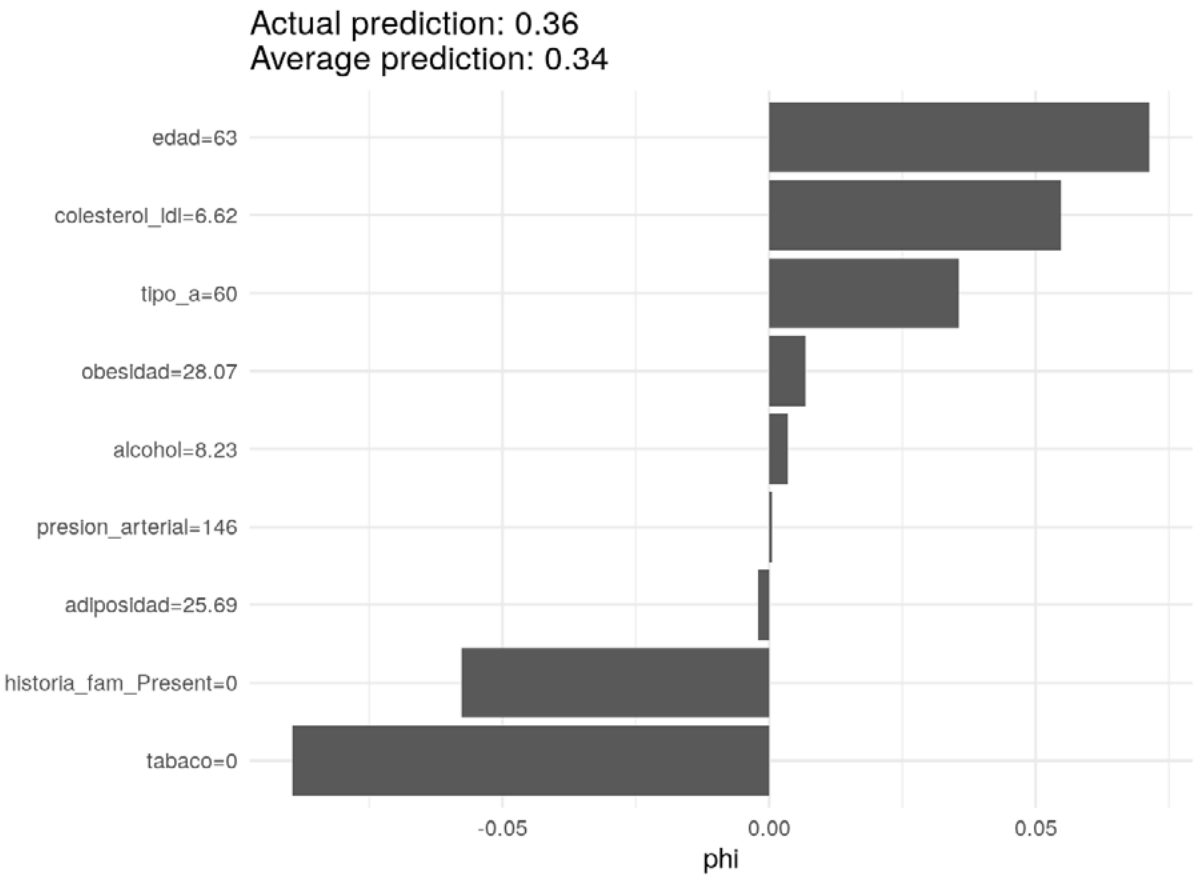
```
modelo_sa <- sa_boosted$fit

sa_entrena_x <- sa_entrena %>% dplyr::select(-enf_coronaria)

predict_fun <- function(object, newdata){

  new_data_x = xgb.DMatrix(data.matrix(newdata), missing = NA)

  results<-predict(modelo_sa, new_data_x)

  return(results)

}

predictor <- Predictor$new(modelo_sa, data = sa_entrena_x, y = sa_entrena$chd



                           type = "prob", predict.function = predict_fun)

# el caso de interés es el caso 15

valores_shapley <- Shapley$new(predictor, x.interest = (sa_entrena_x[15, ]))

valores_shapley$plot() + theme_minimal()
```



En este caso, varias medidas contribuyen positivamente a la probabilidad de enfermedad del corazón, como son uso de tabaco, edad y mediciones de colesterol. Estas contribuciones explican la probabilidad tan alta de este individuo particular.

En contraste, la siguiente persona está cerca del promedio, aumentando positivamente la probabilidad la edad y la medida de colesterol, pero negativamente el no uso de tabaco y ninguna historia familiar de diabetes:

```
# el caso de interés es el caso 24

valores_shapley <- Shapley$new(predictor, x.interest = (sa_entrena_x[24, ]))

valores_shapley$plot() + theme_minimal()
```



Observación: igual que en el modelo y en las gráficas de dependencia parcial que se discutieron anteriormente, estos coeficientes no deben interpretarse de manera causal (por ejemplo: es necesario bajar el colesterol para estos dos individuos). Esta es la información que usa el modelo para construir la predicción a partir de la predicción promedio sobre la población.

Se pueden calcular los valores de Shapley para dos grupos de edad, por ejemplo.

REFERENCIAS

Anna Jobin, M. I. (2019). *The global landscape of AI ethics guidelines.* Springer Science and Business Media LLC.

Athey, S. W. (2018). Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests. *Journal of the American Statistical Association.*

Barocas, S., & Selbst, A. D. (2014). Big Data's Disparate Impact. *SSRN eLibrary.*

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *ArXiv, abs/1607.06520.*

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Buolamwini, J., & Gebru, T. (Feb. 23–24, 2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. En S. A. Friedler, & C. Wilson (Ed.), *Conference Proceedings of the 1st Conference on Fairness, Accountability and Transparency.* 81, 77-91. New York, NY, USA: PMLR.

Buuren, S. v., & Groothuis-Oudshoorn, K. (2011). Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software, 45(*3), 1-67.

Carrillo, A., Cantú, L., & Noriega, A. (2020). *Individual Explanations in Machine.* IADB.

Chawla, N. V. (2002). SMOTE: Synthetic Minority over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357.

drivendata. (2019). *An ethics checklist for data scientists.* Obtenido de https://deon.drivendata.org/

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*(5), 1189-1232.

Fritzler, A. (2015). *An ethical checklist for data science.* Obtenido de http://www.dssgfellowship.org/2015/09/18/an-ethical-checklist-for-data-science/

Gebru, T., Morgenstern, J., Vecchione, B., Wortman, J., Wallach, H., Daumé, H., & Crawford, K. (2018). *Datasheets for Datasets.* Obtenido de https://arxiv.org/pdf/1803.09010.pdf

Gelman, A., & Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models* (1 ed.). Cambridge University Press.

Greene, W. (2003). *Econometric Analysis.* Pearson Education. Obtenido de https://books.google.com.mx/books?id=njAcXDlR5U8C.

Hardt, M. A. (2016). Equality of Opportunity in Supervised Learning. *CoRR, abs/1610.02413.*

Harini Suresh, J. V. (2019). *A Framework for Understanding Unintended Consequences of Machine Learning.* MIT. Obtenido de https://arxiv.org/pdf/1901.10002.pdf

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The Elements of Statistical Learning.* Springer New York Inc.

INEGI. (2014). *Encuesta Nacional de Ingresos y Gastos de los Hogares (Enigh-2014). Diseño Muestral.* Obtenido de http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod_serv/contenidos/espanol/bvinegi/productos/nueva_estruc/702825070359.pdf

James, G. D. (2017). *Data for an Introduction to Statistical Learning with Applications in R.* Obtenido de https://CRAN.R-project.org/package=ISLR.

Kaufman, S., Rosset, S., & Perlich, C. (01 de 2011). Leakage in Data Mining: Formulation, Detection, and Avoidance. *ACM Transactions on Knowledge Discovery from Data,* 6, 556-563.

Kuhn, M. F. (2020). *Rsample: General Resampling Infrastructure*. Obtenido de https://CRAN.R-project.org/package=rsample.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Lackland, D. (06 de 2014). Racial Differences in Hypertension: Implications for High Blood Pressure Management. *The American Journal of the Medical Sciences, 348*(2), 135-138.

Little, R. J., & Rubin, D. B. (2002). *Statistical analysis with missing data.* Wiley.

Lohr, S. L. (2009). *Sampling: Design and Analysis.* Cengage Learning.

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *ArXiv, abs/1705.07874.*

Miller, T. (2019). Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artif. Intell.,* 267, 1-38.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., . . . Gebru, T. (2019). *Model Cards for Model Reporting.* Obtenido de https://arxiv.org/abs/1810.03993

Molnar, C. (2019). *Interpretable Machine Learning.* Obtenido de https://christophm.github.io/interpretable-ml-book/

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science, 366*(6464), 447-453.

OECD (Forthcoming). (s.f.). *Framework for the Classification of AI Systems* . Paris: OECD Publishing.

OECD. (2019c). *Artificial Intelligence in Society.* Paris: OECD Publishing.

OECD. (2021). Good Practice Principles for Data Ethics in the Public Sector.

Pombo, C., Cabrol, M., González, N., & Sánchez, R. (2020). *fAIr LAC: Adopción ética y responsable de la inteligencia artificial en América Latina y el Caribe.* doi:http://dx.doi.org/10.18235/0002169

Prosser, C., & Mellon, J. (2016). *Twitter and Facebook are Not Representative of the General Population: Political Attitudes and Demographics of Social Media Users.* Available at SSRN: https://ssrn.com/abstract=2791625 or http://dx.doi.org/10.2139/ssrn.2791625.

Rossouw, J. E. (1983). Coronary Risk Factor Screening in Three Rural Communities. The Coris Baseline Study. *South African Medical Journal* = Suid-Afrikaanse Tydskrif Vir Geneeskunde, 64(12), 430-436.

Rubin, R. J. (2002). *Statistical Analysis with Missing Data.* Second Edition. John Wiley & Sons, Inc.

Saleiro, P., Kuester, B., Hinkson, L., London, J., Stevens, A., Anisfeld, A., & Ghani, R. (2019). *Aequitas: A Bias and Fairness Audit Toolkit.* Center for Data Science and Public Policy.

Stuart, K. I. (2008). Misunderstandings Among Experimentalists and Observationalists about Causal Inference. *Journal of the Royal Statistical Society, Series A, 171, part 2*, 481-502.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. *ArXiv, abs/1703.01365*.

Suresh, H., & Guttag, J. V. (2019). A Framework for Understanding Unintended Consequences of Machine Learning. *ArXiv, abs/1901.10002*.

Vayena, A. J. (2019). *The global landscape of AI ethics guidelines.* Springer Science and Business Media LLC.

Vaver, J., & Koehler, J. (2011). *Measuring Ad Effectiveness Using Geo Experiments*. Google Inc.

Verma, S., & Rubin, J. (2018). Fairness Definitions Explained. *Conference Proceedings of the International Workshop on Software Fairness* (pp. 1-7). New York, NY, USA: Association for Computing Machinery.

Wachter, S., Mittelstadt, B. D., & Russell, C. (2017). Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *ArXiv, abs/1711.00399*.

Washingtonpost. (04 de 2019). *21 more studies showing racial disparities in the criminal justice system.* Obtenido de https://www.washingtonpost.com/opinions/2019/04/09/more-studies-showing-racial-disparities-criminal-justice-system/

Williams, D. M. (09 de 1981). Racial differences of hemoglobin concentration: measurements of iron, copper, and zinc. *The American Journal of Clinical Nutrition, 34*(9), 1694-1700.

Wilson, J. (2014). *What your IQ score doesn't tell you.* CNN.

BID

Mejorando vidas