



INFOTEC CENTRO DE INVESTIGACIÓN E  
INNOVACIÓN EN TECNOLOGÍAS DE LA  
INFORMACIÓN Y COMUNICACIÓN

DIRECCIÓN ADJUNTA DE INNOVACIÓN Y  
CONOCIMIENTO  
GERENCIA DE CAPITAL HUMANO  
POSGRADOS

# **“Predicción de la distribución de probabilidades de incapacidades temporales del trabajo en el IMSS mediante modelos de aprendizaje automático”**

Tesis de maestría  
Que para obtener el grado de MAESTRO EN  
CIENCIA DE DATOS E INFORMACIÓN

Presenta:  
**Juan Luis Vázquez Espinoza**

Asesor:  
**Dr. Carlos Minutti Martinez**

Ciudad de México, abril, 2025.

## **Autorización de impresión**

## Agradecimientos

## Índice general

|   |      |
|---|------|
| Índice de figuras .....   | vii  |
| Índice de tablas .....  | viii |
| Abreviaturas y acrónimos .....  | ix   |
| Glosario.....   | x    |
| Resumen .....   | xiv  |
| Introducción.....   | 1    |
| Capítulo 1. Generalidades .....                                       | 3    |
| 1.1 Planteamiento del problema .....                                  | 3    |
| 1.2 Protocolo de investigación .....                                  | 7    |
| 1.3 Justificación .....   | 9    |
| 1.4 Límites y alcances .....  | 11   |
| Capítulo 2. Base de datos.....  | 14   |
| 2.1 Construcción de la base de datos .....                            | 15   |
| 2.2 Preprocesamiento de la base de datos .....                        | 17   |
| 2.3 Análisis exploratorio de los datos .....                          | 21   |
| Capítulo 3. Diseño del estudio y ajuste de modelos .....              | 29   |
| 3.1 Marco teórico.....  | 29   |
| 3.1.1 Seguridad Social y Protección de los Trabajadores .....         | 29   |
| 3.1.2 Incapacidades Temporales de Trabajo en México .....             | 30   |
| 3.1.3 Retos de la seguridad social en incapacidades del trabajo ..... | 33   |
| 3.1.4 Ciencia de Datos .....  | 35   |
| 3.1.5 Técnicas para el preprocesamiento y análisis .....              | 38   |
| 3.1.6 Antecedentes de investigación .....                             | 41   |
| 3.2 Marco metodológico .....  | 48   |
| 3.3 Ajuste de los modelos .....                                       | 57   |
| 3.4 Análisis de resultados .....                                      | 67   |
| Conclusiones y recomendaciones .....                                  | xvi  |

|                                 |              |
|---------------------------------|--------------|
| <b>Fuentes de consulta.....</b> | <b>xxi</b>   |
| <b>ANEXO 1.....</b>             | <b>xxvii</b> |
| <b>Índice de términos.....</b>  | <b>xlili</b> |

## Índice de figuras

|  |
|--|
| Figura 2.1: Flujo de datos clínicos a base central (Fuente: Elaboración propia, 2024.)                         |
| Figura 2.2: Ejemplo de preprocesado en variables numéricas (Fuente: Elaboración propia 2024)                   |
| Figura 2.3: Distribución de variables numéricas posterior a preprocesamiento (Fuente: Elaboración propia 2024) |
| Figura 2.4: Mapa de Calor de Correlaciones (Fuente: Elaboración propia, 2024.)                                 |
| Figura 3.1: Proceso CRISP DM (Fuente: Elaboración propia, 2025.)   |
| Figura 3.2: Implementación CatBoost (Fuente: Elaboración propia, 2025.)  |
| Figura 3.3: Implementación XGBoost (Fuente: Elaboración propia, 2025.)   |
| Figura 3.4: Implementación Regresión Ordinal (Fuente: Elaboración propia, 2025.)                               |
| Figura 3.5: Matriz para CatBoost (Fuente: Elaboración propia, 2025.)   |
| Figura 3.6: Matriz para XGBoost (Fuente: Elaboración propia, 2025.)  |
| Figura 3.7: Matriz para Regresión Ordinal (Fuente: Elaboración propia, 2025.)                                  |
| Figura 3.8: Ejemplo de uso (Fuente: Elaboración propia, 2025.)   |

## Índice de tablas

|   |
|---|
| Tabla 2.1: Tipo de las variables y valores nulos (Fuente: Elaboración propia, 2024.)                          |
| Tabla 2.2: Ejemplo del uso de EncoderFrecuencia (Fuente: Elaboración propia, 2024.)                           |
| Tabla 2.3: Ejemplo del uso de OneHotEncoder (Fuente: Elaboración propia, 2024.)                               |
| Tabla 2.4: Tipo de las variables y valores nulos para variables extraídas (Fuente: Elaboración propia, 2024.) |
| Tabla 2.5: Valores únicos de variables categóricas (Fuente: Elaboración propia, 2024.)                        |
| Tabla 2.6: Estadística descriptiva de las variables numéricas (Fuente: Elaboración propia, 2024.)             |
| Tabla 3.1: Resumen de antecedentes (Fuente: Elaboración propia, 2025.)  |
| Tabla 3.2: Desempeño en conjunto de validación (Fuente: Elaboración propia, 2025.)                            |
| Tabla 3.3: Desempeño en conjunto de prueba (Fuente: Elaboración propia, 2025.)                                |
| Tabla 3.4: Desempeño comparativo entre modelo y criterio humano (Fuente: Elaboración propia, 2025.)           |
| Tabla 3.5: Importancia de las variables para Regresión Ordinal (Fuente: Elaboración propia, 2025.)            |



## Abreviaturas y acrónimos

|                 |  |
|-----------------|--|
| <b>CatBoost</b> | Category Boosting  |
| <b>CIE10</b>    | Clasificación Internacional de Enfermedades, 10. <sup>a</sup> Revisión |
| <b>COCOITT</b>  | Comité para el Control de la Incapacidad Temporal                      |
| <b>CRISP-DM</b> | Cross-Industry Standard Process for Data Mining                        |
| <b>DPES</b>     | Dirección de Prestaciones Económicas y Sociales                        |
| <b>DNN</b>      | Deep Neural Network  |
| <b>ECE</b>      | Expediente Clínico Electrónico   |
| <b>GBM</b>      | Gradient Boosting Machine  |
| <b>IMSS</b>     | Instituto Mexicano del Seguro Social                                   |
| <b>ITT</b>      | Incapacidad Temporal para el Trabajo                                   |
| <b>k-NN</b>     | k-Nearest Neighbors  |
| <b>LSTM</b>     | Long Short-Term Memory   |
| <b>ML</b>       | Machine Learning (Aprendizaje automático)                              |
| <b>MLP</b>      | Multilayer Perceptron  |
| <b>NLP</b>      | Natural Language Processing (Procesamiento de Lenguaje Natural)        |
| <b>OCDE</b>     | Organización para la Cooperación y el Desarrollo Económicos            |
| <b>SEM</b>      | Seguro de Enfermedades y Maternidad                                    |
| <b>SIMF</b>     | Sistema de Información de Medicina Familiar                            |
| <b>SRT</b>      | Seguro de Riesgos de Trabajo   |
| <b>XGBoost</b>  | eXtreme Gradient Boosting  |

## Glosario

### “A”

**Algoritmo:** Conjunto de instrucciones o reglas lógicas diseñadas para resolver problemas o realizar tareas específicas.

**Aprendizaje Automático (ML):** Técnica de inteligencia artificial que identifica patrones en datos y permite realizar predicciones o tomar decisiones basadas en ellos.

**Aprendizaje Supervisado:** Método en el que el modelo se entrena con datos etiquetados para predecir resultados.

**Aprendizaje No Supervisado:** Técnica que permite identificar patrones o estructuras en datos sin etiquetas predefinidas.

**Aprendizaje por Refuerzo:** Técnica en la que un agente interactúa con un entorno, recibiendo recompensas o penalizaciones, para mejorar su desempeño.

### “B”

**Big Data:** Grandes volúmenes de datos complejos que requieren herramientas avanzadas para su análisis.

### “C”

**Ciencia de Datos:** Campo interdisciplinario que combina estadística, algoritmos computacionales e inteligencia artificial para extraer conocimiento útil a partir de datos masivos.

**CatBoost:** Algoritmo de boosting que maneja de forma nativa variables categóricas y utiliza técnicas de regularización para prevenir el sobreajuste.

**CIE10:** Clasificación Internacional de Enfermedades, un sistema estándar para codificar diagnósticos y clasificar patologías.

**Clasificación Multiclase:** Enfoque de modelado que asigna observaciones a una de varias categorías mutuamente excluyentes.

**CRISP-DM:** Metodología (Cross-Industry Standard Process for Data Mining) que guía el proceso completo de análisis de datos, desde la comprensión del negocio hasta el despliegue del modelo.

## **“D”**

Data Preparation (Preparación de Datos): Proceso de limpieza, transformación y selección de datos para que sean adecuados para el análisis y modelado.

## **“E”**

EncoderFrecuencia: Técnica de codificación que reemplaza cada categoría de una variable por la frecuencia o proporción de su aparición en el conjunto de datos.

Error Cuadrático Medio (MSE): Métrica que mide el promedio de los cuadrados de las diferencias entre los valores predichos y los reales.

Error Absoluto Medio (MAE): Métrica que evalúa la magnitud promedio de los errores en las predicciones, sin considerar la dirección del error.

Exploración de Datos (Análisis Exploratorio de Datos, EDA): Proceso inicial para resumir las características principales de un conjunto de datos mediante estadísticas y visualizaciones.

## **“E”**

Feature Engineering (Ingeniería de Características): Proceso de extraer y crear variables relevantes a partir de datos brutos para mejorar el desempeño de un modelo predictivo.

## **“H”**

Hiperparámetros: Parámetros de configuración de un modelo de aprendizaje automático que se establecen antes del entrenamiento y afectan su rendimiento.

## **“I”**

Incapacidad Temporal para el Trabajo (ITT): Certificado y subsidio otorgado a trabajadores que, por razones de salud, se encuentran imposibilitados de desempeñar sus labores temporalmente.

IMSS (Instituto Mexicano del Seguro Social): Organismo encargado de administrar la seguridad social en México, proporcionando servicios de salud, pensiones y otros subsidios.

Infraestructura Hospitalaria: Conjunto de recursos y equipamiento (como camas, quirófanos y consultorios) disponibles en las unidades médicas para prestar servicios de salud.

## **“M”**

**Modelo Predictivo:** Herramienta basada en datos y algoritmos de aprendizaje automático que predice resultados o comportamientos futuros.

**Modelo de Boosting:** Algoritmo que combina múltiples modelos débiles (por ejemplo, árboles de decisión) para formar un modelo fuerte y mejorar la precisión de las predicciones.

**Modelo de Regresión Ordinal:** Técnica de modelado utilizada cuando la variable de respuesta tiene categorías con un orden natural, utilizando umbrales para definir dichas categorías.

## **“N”**

**Normalización:** Proceso de transformar datos numéricos para que se encuentren en un rango común, generalmente  $[0, 1]$ , facilitando comparaciones y el análisis.

## **“O”**

**One Hot Encoding:** Técnica de codificación para variables categóricas que crea variables binarias, representando la presencia o ausencia de cada categoría.

## **“P”**

**Preprocesamiento de Datos:** Conjunto de técnicas aplicadas para limpiar, transformar y preparar los datos para el análisis y modelado predictivo.

**Pipeline:** Secuencia de pasos de preprocesamiento y modelado aplicados de manera ordenada en el flujo de trabajo de machine learning.

**Probabilidad (Distribución Probabilística):** Representación matemática que describe la variabilidad e incertidumbre en los resultados predichos.

## **“R”**

**Random Forest:** Algoritmo de aprendizaje supervisado que utiliza un conjunto de árboles de decisión para mejorar la precisión y reducir el riesgo de sobreajuste.

## **“S”**

**Secuencia de Atención:** Cadena de eventos que describe la trayectoria de un paciente a través de distintos niveles de complejidad en el sistema de salud.

Seguridad Social: Sistema de protección que garantiza prestaciones en salud, pensiones y otros servicios, con el objetivo de mitigar riesgos y promover la cohesión social.

StratifiedKFold: Técnica de validación cruzada que garantiza que cada partición de los datos mantenga la misma distribución de la variable objetivo.

## **“T”**

Técnicas de Preprocesamiento: Métodos aplicados para transformar, limpiar y preparar los datos (por ejemplo, codificación, escalado y normalización).

Transformación Logarítmica: Aplicación de la función logarítmica (como  $\log_{10}$ ) a datos numéricos para reducir el sesgo y atenuar la influencia de valores extremos.

## **“X”**

XGBoost: Algoritmo de boosting avanzado que utiliza regularización y procesamiento paralelo para optimizar la precisión y eficiencia del modelo.

## Resumen

La presente investigación aborda el problema de la predicción de la duración de las incapacidades temporales en el IMSS. Un tema que impacta en el uso óptimo de los recursos públicos. En México, el uso excesivo o inadecuado de las incapacidades laborales genera impactos negativos en la sostenibilidad del sistema de seguridad social, afectando desde la planificación presupuestaria hasta la efectividad de las actividades diarias de las unidades médicas. Ante esto, se propone el desarrollo de un modelo predictivo basado en aprendizaje automático, que integre variables clínicas, demográficas y de infraestructura hospitalaria para estimar de forma probabilística la duración de las incapacidades.

El estudio utiliza el marco metodológico CRISP-DM, para mantener un orden replicable desde la comprensión del negocio y la exploración de datos hasta el modelado, evaluación y despliegue de la solución propuesta. La investigación inicia con la integración de dos fuentes de información: una que recoge datos clínicos y administrativos de los casos de incapacidad superiores a 100 días, y otra que contiene información sobre la infraestructura hospitalaria del IMSS. La unión de estas bases de datos mediante la variable “unidad de adscripción” permite que ambas bases de datos puedan usarse en conjunto para un análisis contextualizado a nivel regional. Esto, para incluir la variabilidad en la dotación de recursos en el cálculo probabilístico de la duración de las incapacidades.

Durante la fase de preprocesamiento, se aplicaron técnicas de codificación para variables categóricas de alta y baja cardinalidad (EncoderFrecuencia y One Hot Encoding, respectivamente), junto con transformaciones logarítmicas, escalado y normalización de variables numéricas. Además, se implementó un proceso de extracción de características a partir de la variable que describe la secuencia de unidades médicas que dieron atención a una persona. A partir de esta variable, que indica el viaje de la persona a través del sistema, se extrajeron variables nuevas. Estas variables describen cuántas veces fue atendido en primero, segundo o tercer nivel, cuántas veces pasó de una unidad a otra, y cuántas unidades médicas visitó antes de su alta. Esta variable fue transformada de esta manera a fin de complementar los datos propios de una región del país. Pues el viaje de una

persona, en diferentes regiones del país, varía porque la infraestructura hospitalaria es diferente de región en región. Este conjunto enriquecido de variables facilitó el posterior análisis exploratorio, donde se identificaron patrones y correlaciones relevantes entre las características del paciente y la infraestructura hospitalaria.

La fase de modelado incluyó la evaluación de diversos enfoques, entre ellos la Regresión Ordinal, XGBoost y CatBoost. Cada uno de estos modelos fue ajustado y validado mediante validación cruzada (StratifiedKFold), lo que permitió comparar su desempeño y elegir el modelo que mejor desempeño mostrara ante un conjunto de casos nuevos, no vistos por ningún modelo. Los resultados obtenidos demostraron que estos modelos tienen un rendimiento superior en comparación con las predicciones basadas en el consenso de expertos.

La relevancia de esta investigación radica en su potencial para ser un complemento para la práctica médica y para mejorar la información que los pacientes reciben sobre su salud. La capacidad de predecir de manera probabilística la duración de las incapacidades ayudará a los profesionales médicos a tener un panorama más claro de cada caso y entender que la duración no puede ser una predicción puntual o un techo, si no que, de acuerdo a cada caso y los servicios médicos disponibles, la duración puede variar entre diferentes rangos de tiempo. Indirectamente esta herramienta, describe aquellas regiones donde las incapacidades son inusualmente prolongadas, lo cual ayudará a las autoridades institucionales a mejorar la planificación de servicios en aquellas áreas con mayor rezago. Además, el Instituto se encuentra en una etapa de alta adopción tecnológica, por lo que el desarrollar herramientas que mejoren la operación diaria puede traducirse a reducción de costos y mejora en la satisfacción de los derechohabientes.

El estudio demuestra que, mediante un adecuado preprocesamiento de datos y el uso de modelos de aprendizaje automático, es posible obtener mejores estimaciones en cuanto a la duración de las incapacidades laborales, con respecto a los métodos tradicionales, como el consenso de expertos.

## Introducción

La creciente demanda de servicios de salud y el elevado costo asociado a las incapacidades temporales en el Instituto Mexicano del Seguro Social (IMSS) constituyen desafíos para el sistema de seguridad social en México. En un contexto marcado por la transición epidemiológica y la insuficiencia de recursos públicos destinados a salud, la capacidad de predecir con precisión la duración de las incapacidades resulta importante para optimizar la asignación de recursos, mejorar la atención médica y garantizar la sostenibilidad financiera. El uso excesivo o inadecuado de las incapacidades afecta no solo la eficiencia operativa del IMSS, sino también la calidad del servicio ofrecido a los derechohabientes.

Ante este escenario, la presente investigación tiene como objetivo general evaluar la precisión de un modelo predictivo basado en aprendizaje automático para estimar la distribución probabilística de la duración de las incapacidades laborales, integrando variables clínicas, demográficas y de infraestructura hospitalaria. Para lograr esto, se pretende diseñar un protocolo de preprocesamiento de datos, seleccionar y ajustar modelos (entre Regresión Ordinal, XGBoost y CatBoost) y comparar las predicciones obtenidas con el consenso de expertos en la materia.

El documento se estructura en tres capítulos principales. En el primer capítulo se contextualiza el problema, se exponen los fundamentos teóricos, se formulan las hipótesis y se establecen los objetivos de la investigación. El segundo capítulo describe la construcción, el preprocesamiento y el análisis exploratorio de la base de datos, que integra información clínica y de infraestructura hospitalaria. Por último, el tercer capítulo se centra en el diseño metodológico, el ajuste de modelos predictivos y la evaluación de su desempeño, utilizando el marco metodológico CRISP-DM para garantizar un análisis integral y replicable. Este trabajo se propone, de esta forma, generar una herramienta que contribuya a la toma de decisiones en el ámbito de la salud.



# Capítulo 1

## Generalidades

## Capítulo 1. Generalidades

En este primer capítulo se narra la situación actual respecto a la generación de pronósticos de tiempo para las incapacidades temporales del trabajo en el Instituto Mexicano del Seguro Social. En la sección 1.1 se plantea la complejidad de esta actividad, y las limitaciones del sistema para tomar decisiones con la precisión necesaria para eficientar el uso de recursos económicos.

En el apartado 1.2 se describirá el protocolo, en el que se expondrá como conclusión del problema planteado, la pregunta de investigación. Además, se abordará el desarrollo y la utilidad de modelos de aprendizaje automático para la predicción de la duración total de las incapacidades.

Aclarado el propósito de la investigación, se presentarán los objetivos a alcanzar, que incluyen la formación de la base de datos primaria a partir de dos fuentes originales, el preprocesamiento de los datos, el desarrollo de los modelos, su validación y la evaluación del desempeño.

Posteriormente, se formulan las hipótesis en función de los objetivos y preguntas de investigación. Estas plantean los resultados esperados, como la capacidad de preprocesar correctamente los datos, modelar de manera óptima y validar el desempeño de los modelos. Todo ello con el propósito de desarrollar una herramienta útil para predecir la probabilidad de las incapacidades.

Se procede a argumentar el valor que este trabajo puede generar a diferentes actores y en diferentes procesos. Especialmente, como herramienta complementaria para profesionales de salud y pacientes, a fin de que pueda comprender el rango de escenarios posibles.

### 1.1 Planteamiento del problema

La seguridad social es un mecanismo de protección, distribución de la riqueza y cohesión social en las sociedades modernas. Su objetivo es garantizar el bienestar y protección de los trabajadores frente a los riesgos que puedan presentarse y comprometer su pleno desarrollo y el de su familia. Este sistema ofrece prestaciones

en salud, jubilación, accidentes de trabajo y desempleo. Originada durante la revolución industrial en respuesta a las precarias condiciones laborales, la seguridad social ha evolucionado para convertirse en un instrumento de la redistribución de la riqueza. La protección de los trabajadores implica no solo beneficios económicos, sino también la promoción de ambientes de trabajo seguros y saludables mediante políticas de prevención, capacitación continua y servicios de salud ocupacional (Gomis et al., 2020).

Dentro de este amplio marco de protección, la incapacidad temporal para el trabajo constituye uno de los mecanismos de protección de la subsistencia, que el estado mexicano ha buscado garantizar para todos los trabajadores, a través de la seguridad social. Esta garantía protege a los trabajadores que se encuentran imposibilitados para desempeñar sus labores, sea por enfermedades relacionados con su actividad laboral o independientes de esta. La protección se materializa a través de subsidios y prestaciones en especie durante el tiempo en que se encuentren incapacitados, formalizado con la expedición de certificados de incapacidad, con carácter tanto médico como legal (Ley del Seguro Social, 2008). El principio que rige esta protección es el permitir que un trabajador pueda ausentarse de su trabajo y reposar para incrementar la probabilidad de una recuperación completa y oportuna. Sin embargo, este sistema de aseguramiento enfrenta retos considerables en cuanto a su sostenibilidad financiera. Además de presentar retos en cuanto a la efectividad de los procesos médico-administrativos que generan estas incapacidades (Hernández, 2023; IMSS, 2006).

Uno de esos retos es el uso de la incapacidad, más allá del tiempo necesario para la recuperación del trabajador. El prolongado periodo de incapacidad implica un mayor desembolso de subsidios, comprometiendo la suficiencia de los seguros de Riesgos de Trabajo y de Enfermedades y Maternidad (IMSS, 2024). Sumado a esto, otro problema actual es la insuficiente cantidad de profesionales médicos en proporción a la demanda de servicios médicos por la población asegurada; lo que puede generar amplios periodos de tiempo en los que un trabajador se encuentra esperando una cita, sin recibir atención médica. Esto constituye tiempo muerto, en

el cual el trabajador no se encuentra recibiendo atención, pero sigue siendo incapacitado durante este periodo de espera.

La situación se agrava, al considerar la transición epidemiológica, donde es creciente la prevalencia de enfermedades crónico-degenerativas, como son diabetes, hipertensión, enfermedades cardiovasculares y oncológicas; cuyas complicaciones generan pérdida irreversible de las capacidades físicas de una persona, con la consecuente necesidad de recibir el subsidio de la incapacidad, dado que por sí solos ya no les es posible generar recursos propios (Hernández, 2023). Además de no poder trabajar y sostenerse por sí mismos, estos trabajadores se vuelven usuarios regulares de los servicios médicos, porque padecen condiciones complejas y en estados avanzados de sus enfermedades. Por lo que requieren el uso de atención especializada de alto costo (Gewurtz et al., 2019).

Por estas razones, el Instituto Mexicano del Seguro Social, ha implementado diversas estrategias para optimizar el uso de los recursos y evitar el uso injustificado en la expedición de certificados de incapacidad. Entre estas estrategias se destacan la elaboración de guías de duración de la incapacidad, que sirven como referencia para que los profesionales médicos determinen el tiempo probable de recuperación. También se han creado comités especializados, orientados a supervisar y controlar el uso de incapacidades, por un grupo de profesionales médicos (Martin-Fumadó et al., 2014; IMSS, 2024). Sin embargo, estas medidas encuentran sus límites dado que predecir la evolución de un paciente es un problema complejo, que requiere la consideración de diferentes variables; desde aquellas propias del paciente (como su edad, sexo, diagnóstico, etc), hasta aquellas que describen la capacidad médica en la región que el paciente vive (como son número de camas hospitalarias, disponibilidad de hospitales de alta especialidad, unidad médica que dio la primera atención, etc); e incluso, aquellas que describen la naturaleza de la enfermedad que generó la incapacidad (si una enfermedad laboral, accidente de trabajo o enfermedad no relacionada al trabajo).

La ciencia de datos es un campo interdisciplinario que genera conocimiento novedoso a partir del análisis de grandes volúmenes de información, combinando estadística, algoritmos computacionales e inteligencia artificial (Provost & Fawcett,

2013; Murphy, 2012). A diferencia de métodos tradicionales, abarca el ciclo completo de los datos: recolección, limpieza, análisis, modelado y visualización.

Un campo de estudio y construcción de modelos para generar valor a partir de los datos es el aprendizaje automático (ML por sus siglas en inglés). El ML cuenta con tres categorías: 1) El aprendizaje supervisado, que utiliza datos etiquetados para entrenar modelos que predicen resultados. 2) El aprendizaje no supervisado, que identifica patrones en datos sin etiquetas. 3) El aprendizaje por refuerzo, que consiste en que un agente interactúa con el entorno, recibiendo recompensas o penalizaciones que mejoran su desempeño (Murphy, 2012).

De manera que, el uso de técnicas de ciencia de datos y de aprendizaje automático se presentan como herramientas que podrían abordar el problema de predecir la distribución probabilística de la duración total de las incapacidades. Esto es posible por su capacidad de analizar grandes volúmenes de datos y de identificar patrones ocultos en conjuntos de información heterogénea.

Diversos estudios han implementado una variedad de algoritmos como Random Forest, Gradient Boosting y XGBoost para predecir resultados en ámbitos relacionados con la salud y la seguridad laboral, tales como el retorno al trabajo o la probabilidad de transición a una situación de invalidez permanente (Chand & Zhang, 2022; Saarela et al., 2022).

La mayor parte de la literatura y las aplicaciones actuales se han centrado en la clasificación de casos, por ejemplo, retorno al trabajo vs. no retorno, o la predicción de pensión por invalidez. Pero no han abordado de manera específica el modelar la distribución probabilística del tiempo de incapacidad (Chou JCL et al., 2022; Koc et al., 2021). Esto se torna relevante, ya que la estimación de una distribución de probabilidad ofrece una visión más completa y matizada, pues en lugar de predecir simplemente un valor puntual o un rango probable, se describe la variabilidad y la incertidumbre asociada al proceso de recuperación de cada trabajador, según sus características personales y de la región donde recibe servicios.

La integración de estas variables en un único modelo predictivo presenta una oportunidad que permitirá describir la incertidumbre en cada caso, y comprender los

posibles escenarios a los que evolucionará. Las diferencias que puedan surgir entre la duración de mismas enfermedades en diferentes regiones posibilitarán el indagar a mayor detalle el porqué de estas diferencias.

## 1.2 Protocolo de investigación

Como se explicó en el planteamiento del problema, predecir la distribución de probabilidad de la duración de una incapacidad en un caso dado es un problema complejo, ya que intervienen múltiples variables, tanto propias del paciente como relacionadas con la infraestructura hospitalaria donde se le brinda atención. En consecuencia, desarrollar un modelo que prediga la distribución de probabilidades tiene como objetivo responder a las siguientes preguntas:

### Pregunta General

¿Cómo se puede estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo, utilizando un modelo predictivo entrenado con los casos mayores a 100 días registrados entre julio y diciembre del 2024, y cuál es su precisión frente al consenso de expertos?



### Preguntas Específicas

1. ¿De qué manera la construcción de una base de datos limpia que incluya todas las incapacidades mayores a 100 días registradas en el IMSS entre julio y diciembre del 2024 influye en la calidad de la información para estimar la distribución de probabilidad de la duración de dichas incapacidades?
2. ¿Qué modelos predictivos pueden entrenarse para estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo, a partir de la base de datos generada?
3. ¿En qué medida el ajuste y la validación de los modelos predictivos optimizados contribuyen a un mejor desempeño al estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo?

4. ¿Cuál es la diferencia en el desempeño de los modelos predictivos optimizados al estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo, en comparación con el consenso de expertos?

#### Objetivo General

Mejorar la precisión de las estimaciones institucionales sobre la duración de las incapacidades temporales del trabajo mediante un modelo predictivo que genere distribuciones de probabilidad a partir de los casos mayores a 100 días registrados en el IMSS entre julio y diciembre de 2024.

#### Objetivos Específicos

1. Crear una base de datos limpia que incluya todas las incapacidades mayores a 100 días que fueron registradas en el Instituto Mexicano del Seguro Social entre julio y diciembre del 2024.
2. Entrenar modelos predictivos que estimen la distribución de probabilidad de la duración de las incapacidades temporales del trabajo.
3. Evaluar el desempeño de los modelos predictivos optimizados en estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo.
4. Comparar el desempeño de los modelos predictivos optimizados en estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo, con el consenso de expertos.

#### Hipótesis General

El desarrollo de un modelo predictivo entrenado con las incapacidades mayores a 100 días registradas entre julio y diciembre del 2024, que incorpore factores contextuales y clínicos, presentará mayor precisión en la estimación de la distribución de probabilidad de la duración de las incapacidades temporales del trabajo, en comparación con el consenso de expertos.

### Hipótesis Específicas

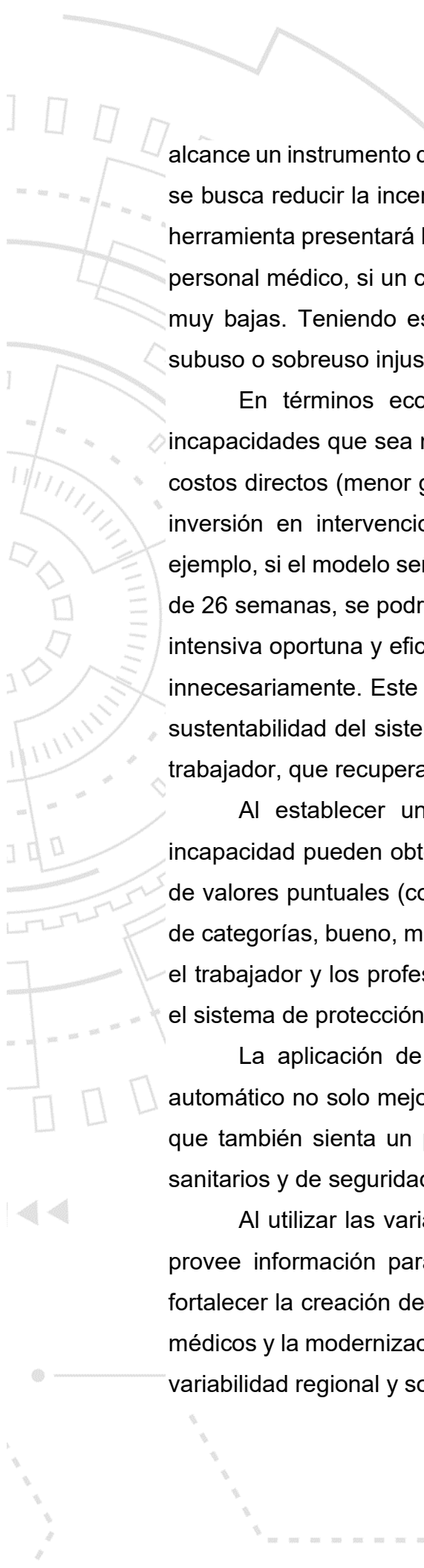
1. La creación de una base de datos limpia, que integre las incapacidades mayores a 100 días, mejorará la calidad de la información y reducirá el error al estimar la distribución de probabilidad de la duración de las incapacidades temporales del trabajo.
2. Existen modelos de aprendizaje automático capaces de ser entrenados para estimar la distribución de probabilidad de la duración de las incapacidades.
3. La evaluación de los modelos predictivos optimizados mostrará un desempeño significativo en la estimación de la distribución de probabilidad de la duración de las incapacidades.
4. Los modelos predictivos optimizados tendrán mayor precisión en estimar la distribución de probabilidad de la duración de las incapacidades a comparación del consenso de expertos.

### 1.3 Justificación

El desarrollo de este trabajo puede generar diferentes beneficios como son el mejorar la toma de decisiones a nivel macro, y el producir información que permita comprender los diferentes escenarios que la evolución de un paciente puede tener. Al incluir variables demográficas y de infraestructura médica, el entrenamiento de los modelos obtendría información acerca de las diferencias en diferentes regiones del país. Esto permitiría comprender la relación entre la variabilidad hospitalaria y su efecto en la duración de incapacidades. Lo que traería mayor luz a la necesidades de incrementar los recursos hospitalarios en zonas con mayor déficit.

El uso de este modelo resultaría un apoyo para el personal médico, al mejorar la eficiencia en la prescripción de incapacidad. Al ofrecer una predicción probabilística en lugar de un valor puntual promedio, se busca mejorar la toma de decisiones clínicas al establecer la probabilidad de tiempos, según la región y las características del paciente. Esta herramienta no busca reemplazar, si no complementar las actuales guías de duración de la incapacidad. Al poner a su





alcance un instrumento que contextualiza la duración probable de cada incapacidad, se busca reducir la incertidumbre de cuánto una incapacidad puede durar. Pues la herramienta presentará la variabilidad esperada. Lo que será una guía que alerte al personal médico, si un caso se está prolongando a duraciones probabilísticamente muy bajas. Teniendo estas estimaciones, se busca también, identificar si existe subuso o sobreuso injustificado de este mecanismo de protección (Lanz, 2018).

En términos económicos, un modelo predictivo de la duración de las incapacidades que sea más certero que el consenso de expertos no solo reduciría costos directos (menor gasto en subsidios mal calibrados), sino que optimizaría la inversión en intervenciones preventivas y en programas de rehabilitación. Por ejemplo, si el modelo señala una alta probabilidad de que un paciente necesite más de 26 semanas, se podría planificar con antelación una estrategia de rehabilitación intensiva oportuna y eficaz para evitar que ese caso, y otros similares, se extienda innecesariamente. Este tipo de intervenciones tempranas tendrían un efecto en la sustentabilidad del sistema de salud, además de mejoría de la calidad de vida del trabajador, que recupera su capacidad productiva en menor tiempo.

Al establecer una distribución de probabilidades, los trabajadores con incapacidad pueden obtener información acerca de su pronóstico, evitando el uso de valores puntuales (como un número de semanas) o uno ambiguo (como el uso de categorías, bueno, malo, reservado). Con esto, se mejora la comunicación entre el trabajador y los profesionales de la salud, además de fortalecer la confianza en el sistema de protección social.

La aplicación de técnicas avanzadas de ciencia de datos y aprendizaje automático no solo mejora la precisión en la predicción de las incapacidades, sino que también sienta un precedente para otros usos de la analítica en problemas sanitarios y de seguridad social (Murphy, 2012; Provost & Fawcett, 2013).

Al utilizar las variables regionales y sus marcadas diferencias, este estudio provee información para la formulación y evaluación de políticas que busquen fortalecer la creación de servicios de especialidades específicas, la distribución de médicos y la modernización de instalaciones. Un modelo predictivo que incorpore la variabilidad regional y socioeconómica puede hacer visibles inequidades que antes

se atribuían vagamente a “la saturación del sistema” o a “la complejidad de los casos”. Al medir estas diferencias con datos institucionales y resultados estadísticamente fiables, se puede generar información que mejore las políticas para equilibrar la asignación de recursos y brindar atención de mayor calidad en zonas desatendidas (Instituto Mexicano del Seguro Social, 2024).

## 1.4 Límites y alcances

El presente estudio adopta un enfoque correlacional y predictivo, orientado a describir qué variables contextuales—edad, salario, diagnóstico médico (CIE10), unidad de adscripción, delegación, empresa y otros factores sociodemográficos y operativos—se asocian con la duración de las incapacidades temporales en el IMSS. Además de estimar la distribución de probabilidad de dicha duración. A diferencia de enfoques tradicionales que se enfocan en predecir un valor puntual, este trabajo busca proveer una curva de probabilidad que refleje la incertidumbre y variabilidad de cada caso.

Al generar esta distribución probabilística, se ofrecen información más completa para la toma de decisiones en ámbitos como la gestión de incapacidades, su supervisión y complemento de guías médicas. No se persigue establecer relaciones causales; en cambio, se explora cómo estas múltiples variables actúan de manera conjunta para determinar la forma (media, varianza y colas) de la duración esperada, permitiendo a los tomadores de decisiones del IMSS priorizar las intervenciones o seguimientos que consideren más pertinentes.

### Límites

Los datos utilizados solo contienen los casos con más de 100 días de incapacidad, lo cual no refleja la totalidad de la población con incapacidades breves. Por tanto, los resultados podrían no generalizarse a incapacidades de corta duración (<100 días).

Aun cuando el modelo identifique patrones relevantes (por ejemplo, mayor probabilidad de incapacidades prolongadas en ciertas regiones con menor

infraestructura), no establece causalidad. Se trata de correlaciones identificadas por el algoritmo, que deben validarse mediante estudios futuros.

Este trabajo no busca reemplazar el proceso actual, dado que los profesionales médicos y los comités de supervisión siguen siendo indispensables para la evaluación individualizada de cada caso; el modelo pudiera ser una herramienta de apoyo y no reemplaza el juicio clínico.

En cuanto a la calidad de los datos, la variable “Secuencia” es una descripción de las diferentes unidades por las que un paciente recorrió desde el inicio hasta su alta. Su uso pretende capturar la complejidad de un caso dado, aunque el recorrido de cada paciente puede ser independiente de su enfermedad o de la infraestructura hospitalaria, lo cual no puede ser discernido.

### **Futuras Líneas de Investigación**

Estudios futuros con un diseño longitudinal podrían proporcionar una visión más dinámica sobre cómo los factores predictores de la duración de las incapacidades cambian con el tiempo. Esto permitiría ajustar los modelos predictivos para que sean más precisos y útiles en la toma de decisiones clínicas a lo largo del tiempo. Además, la incorporación de variables adicionales, como factores psicológicos, estilos de vida, adherencia al tratamiento y características laborales específicas, podría enriquecer el modelo y mejorar su capacidad predictiva. También sería valioso evaluar la aplicación práctica del modelo en la optimización de recursos y políticas dentro del IMSS, así como su potencial adaptabilidad a otros sistemas de salud.

The background features a complex, abstract design. On the left, there are several interlocking gears of different sizes, some with dashed outlines. To the right of the gears, there are various geometric elements: a series of three downward-pointing triangles, a horizontal line with a series of eight right-pointing triangles, a dashed hexagon, a solid hexagon, a series of seven small circles, and a series of four upward-pointing triangles. The overall aesthetic is technical and modern.

## Capítulo 2

### Base de datos

## Capítulo 2. Base de datos

El presente capítulo describe detalladamente la construcción, preprocesamiento y análisis exploratorio de la base de datos utilizada para estudiar los casos de incapacidad y la infraestructura hospitalaria en el IMSS. Inicialmente, se explica cómo se combinan dos tablas provenientes de fuentes oficiales: una con información de incapacidad, que recoge datos de pacientes, diagnósticos, atención médica, costos y duración de las incapacidades (casos superiores a 100 días), y otra que contiene detalles de la infraestructura hospitalaria, como número de camas, salas quirúrgicas, ventiladores y otros equipos biomédicos. La unión de ambas se realiza a través de la “unidad de adscripción”, permitiendo obtener variables estandarizadas por cada 100.000 habitantes y facilitando la evaluación regional de recursos.

En la siguiente sección se aborda el preprocesamiento de los datos, fundamental para optimizar su calidad antes del análisis. Se emplean técnicas de codificación para variables categóricas de alta cardinalidad mediante EncoderFrecuencia, y One Hot Encoding para aquellas de menor cardinalidad. Asimismo, se aplican transformaciones logarítmicas, escalado estándar y normalización para variables numéricas, atenuando sesgos y permitiendo comparaciones consistentes. Además, se extraen características relevantes de la secuencia de atención médica, capturando la trayectoria del paciente a través de distintos niveles de complejidad.

Finalmente, se realiza un análisis exploratorio que incluye estadísticas descriptivas, visualización de distribuciones y mapas de calor para identificar correlaciones entre variables. Estos procedimientos permiten detectar patrones en la dotación de infraestructura y su relación con la duración de las incapacidades, proporcionando una base sólida para futuros análisis y modelado predictivo de gran relevancia.

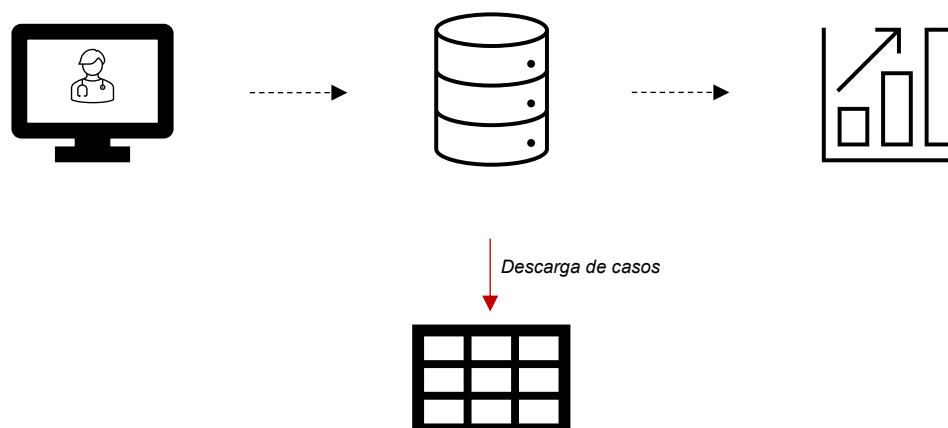
## 2.1 Construcción de la base de datos

La base de datos a utilizar combina dos tablas: una que contiene información sobre incapacidades y otra que detalla la infraestructura hospitalaria. Estas tablas se unirán utilizando los valores comunes de la variable “unidad de adscripción”. La tabla que describe los casos de incapacidad proviene de la Dirección de Prestaciones Económicas y Sociales (DEPS) del IMSS y recoge los casos de incapacidad que superan los 100 días. En ella se registran variables como los datos de identificación del paciente, su diagnóstico, la información del médico que lo atendió, la unidad médica en la que recibió atención y otras variables de interés para la DPES, tales como el costo por paciente y los tiempos de duración de la incapacidad. La segunda tabla contiene los datos de infraestructura hospitalaria de todas las unidades médicas del IMSS. En ella se describen variables como el total de camas por unidad, el número de salas quirúrgicas, la cantidad de ventiladores y otros equipos biomédicos de interés.

La primera base de datos se accede a través de la intranet del IMSS, previa autenticación a través de credenciales personales y únicas a cada trabajador del Instituto. El acceso a esta base se originó con el fin de que el personal médico directivo pueda visualizar a través de gráficas, el total de casos a su cargo, así como los montos económicos a los que ascienden los subsidios. Además, tiene la función de descargar en formato csv los casos totales de incapacidad, que se actualicen de manera diaria. Esta base es alimentada a través de los dos sistemas de expediente clínico electrónico con los que cuenta el IMSS. El Sistema de Información de Medicina Familiar (SIMF) y el Expediente Clínico Electrónico (ECE) (IMSS, 2020).

La segunda tabla es información que la Dirección de Prestaciones Médicas concentra, como inventario físico de la infraestructura hospitalaria, incluyendo equipo biomédico. Esta información es actualizada de manera continua por las oficinas del Instituto en cada estado, que están a cargo de administrar la región.

Figura 2.1: Flujo de datos clínicos a base central



Fuente: Elaboración propia, 2024.

Estas tablas se unirán para extraer, de cada una, las variables de interés. De la primera se obtendrán las características de los casos de incapacidad, y de la segunda, las variables que describen la capacidad hospitalaria de cada región administrativa del IMSS para resolver casos. Además, estos valores se estandarizarán para representar la infraestructura hospitalaria por cada 100.000 habitantes.

Se verificó la cantidad de registros, la existencia de valores nulos y el tipo de datos, como se muestra a continuación:

Tabla 2.1: Tipo de las variables y valores nulos

| Variable                    | Valore no nulos | Valores nulos | Tipo    |
|-----------------------------|-----------------|---------------|---------|
| Cod Cie10                   | 126780          | 0             | object  |
| Delegación                  | 126780          | 0             | object  |
| Unidad Ads                  | 126780          | 0             | object  |
| Tip Ramo                    | 126780          | 0             | object  |
| Avg. Imp Salario Topado     | 126780          | 0             | float64 |
| Dias Probables Recuperacion | 126780          | 0             | int64   |
| Max. Edad                   | 126780          | 0             | int64   |

|  |        |   |         |
|--|--------|---|---------|
| Semanas  | 126780 | 0 | float64 |
| Total de Camas Censables de la delegación_x_100000       | 126780 | 0 | float64 |
| Total de Consultorios de la Unidad_x_100000              | 126780 | 0 | float64 |
| Sala de Quirófano_x_100000                               | 126780 | 0 | float64 |
| Servicio de Salud en el Trabajo_x_100000                 | 126780 | 0 | float64 |
| Total de Camas Censables de la unidad_3er nivel_x_100000 | 126780 | 0 | float64 |
| Total de Consultorios de la Unidad_3er nivel_x_100000    | 126780 | 0 | float64 |
| Sala de Quirófano_3er nivel_x_100000                     | 126780 | 0 | float64 |
| CVE_APARATO  | 126780 | 0 | object  |

Fuente: Elaboración propia, 2024.

## 2.2 Preprocesamiento de la base de datos

El preprocesamiento se implementó para optimizar la calidad de los datos para análisis y modelado:

1. Para las variables categóricas de alta cardinalidad, se codifican de acuerdo a su frecuencia creando un objeto tipo clase `class EncoderFrecuencia(BaseEstimator, TransformerMixin)` que reemplaza cada categoría por su frecuencia en el conjunto de datos. De manera que cada categoría se sustituye por la proporción de observaciones en la que aparece dentro del conjunto de datos. De esta manera, se reduce la dimensionalidad sin perder la señal estadística que aporta la frecuencia de cada categoría. Esta técnica es especialmente útil cuando los métodos tradicionales como One-Hot Encoding generarían un número excesivo de columnas, ocasionando problemas de memoria y sobreajuste.

Tabla 2.2: Ejemplo del uso de EncoderFrecuencia

| Cod Cie10 | Frecuencia | Encoder_Frecuencia |
|-----------|------------|--------------------|
| S525      | 3034       | 0.032758           |
| S822      | 2226       | 0.024034           |
| S824      | 2007       | 0.021669           |



|      |      |          |
|------|------|----------|
| S826 | 1918 | 0.020708 |
| M751 | 1901 | 0.020525 |
| M511 | 1674 | 0.018074 |
| S821 | 1611 | 0.017394 |
| S823 | 1552 | 0.016757 |
| S626 | 1536 | 0.016584 |
| S420 | 1523 | 0.016444 |

Fuente: Elaboración propia, 2024.

- Para las variables de baja cardinalidad se utiliza One Hot Encoding `OneHotEncoder(handle_unknown='ignore')`. Se utiliza esta técnica para producir variables binarias que señalan la presencia (1) o ausencia (0) de cada categoría, manteniendo la interpretabilidad y siendo eficiente en casos donde el volumen de categorías sea manejable. La opción `handle_unknown='ignore'` permite evitar errores cuando se presentan categorías en los datos de inferencia que no hayan estado presentes durante el entrenamiento.

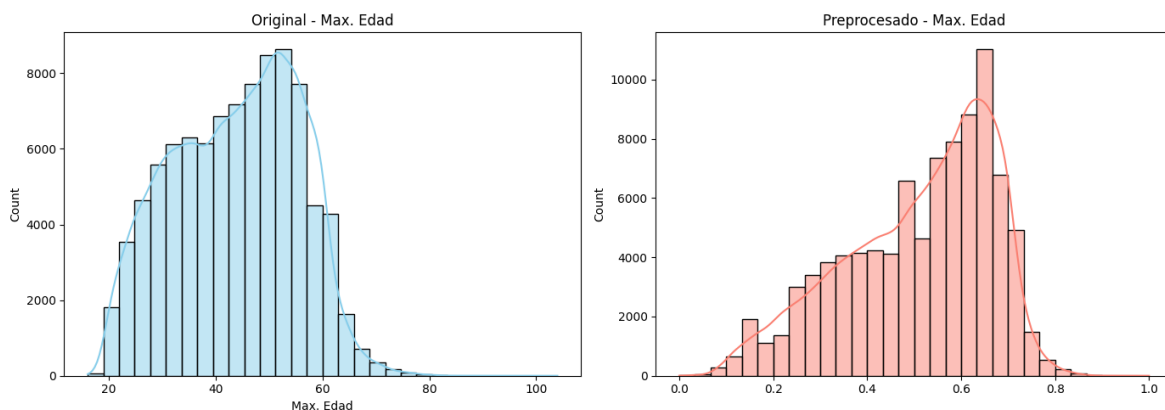
Tabla 2.3: Ejemplo del uso de OneHotEncoder

| Tip Ramo           | Tip<br>Ramo_Enfermedad<br>general | Tip Ramo_Riesgos<br>de trabajo |
|--------------------|-----------------------------------|--------------------------------|
| Riesgos de trabajo | 0                                 | 1                              |
| Enfermedad general | 1                                 | 0                              |
| Enfermedad general | 1                                 | 0                              |
| Enfermedad general | 1                                 | 0                              |
| Riesgos de trabajo | 0                                 | 1                              |
| Enfermedad general | 1                                 | 0                              |
| Riesgos de trabajo | 0                                 | 1                              |
| Enfermedad general | 1                                 | 0                              |
| Enfermedad general | 1                                 | 0                              |
| Enfermedad general | 1                                 | 0                              |

Fuente: Elaboración propia, 2024.

1. Para las variables numéricas se realiza transformación logarítmica aplicando  $\log_{1p}$  para reducir sesgos con `('transformacion_log', FunctionTransformer(np.log1p, validate=False))`; se realiza escalado estándar `('escalado_estandar', StandardScaler())`, y se normalizan en un rango  $[0, 1]$  `('normalizacion', MinMaxScaler())`. Se aplica la función  $\log_{1p}$  para atenuar la influencia de valores atípicos y sesgos en la distribución de variables fuertemente asimétricas. Posteriormente, se normalizan los valores para que cada columna numérica tenga una media cercana a 0 y una desviación estándar próxima a 1. Finalmente, se mapean los valores transformados a un intervalo  $[0, 1]$ .

Figura 2.2: Ejemplo de preprocesado en variables numéricas



Fuente: Elaboración propia, 2024.

La base de datos de la DPES incluye una variable que describe la secuencia de movimientos de un trabajador a través del sistema de salud de su región. Esta secuencia está conformada por dígitos que representan el nivel de complejidad de la unidad médica visitada: 1 para el primer nivel (unidades de atención primaria), 2 para hospitales generales y 3 para hospitales de alta especialidad. El orden de los dígitos refleja el recorrido real seguido por el paciente (por ejemplo, "3111111111111333").

Con la finalidad de extraer información implícita sobre dicha trayectoria, se implementó un proceso de descomposición de la secuencia para obtener métricas relevantes. Entre ellas se encuentran:

- veces\_nivel3: cuántas veces aparece el dígito “3” a lo largo de la ruta.
- count\_1, count\_2, count\_3: cuántas veces aparece cada uno de los tres niveles en total.
- transicion\_12, transicion\_13, transicion\_23: el número de veces que se produce una transición del nivel 1 al 2, del 1 al 3 y del 2 al 3, respectivamente.
- seq\_length: la longitud total de la cadena.
- num\_transitions: cuántos cambios de nivel (de un dígito a otro distinto) existen en la secuencia.
- first y last: el primer y el último nivel de complejidad registrado.

Para llevar a cabo este proceso, se empleó la siguiente función en Python, la cual devuelve un conjunto de estadísticas clave en forma de Series de pandas:

```
def extraccion_caracteristicas_secuencia(seq_val):  
    s = str(seq_val)  
    return pd.Series({  
        'seq_length': len(s),  
        'count_1': s.count('1'),  
        'count_2': s.count('2'),  
        'veces_nivel3': s.count('3'),  
        'num_transitions': sum(1 for i in range(1, len(s)) if s[i] != s[i-1]),  
        'transicion_12': sum(1 for i in range(1, len(s)) if s[i-1]=='1' and s[i]=='2'),  
        'transicion_13': sum(1 for i in range(1, len(s)) if s[i-1]=='1' and s[i]=='3'),  
        'transicion_23': sum(1 for i in range(1, len(s)) if s[i-1]=='2' and s[i]=='3'),  
    })
```

```

        'first': s[0],
        'last': s[-1]
    })

```

Se obtienen las siguientes variables:

Tabla 2.4: Tipo de las variables y valores nulos para variables extraídas

| Variable           | Valore no nulos | Valores nulos | Tipo   |
|--------------------|-----------------|---------------|--------|
| veces_nivel3       | 126780          | 0             | int64  |
| transicion_12      | 126780          | 0             | int64  |
| transicion_13      | 126780          | 0             | int64  |
| transicion_23      | 126780          | 0             | int64  |
| total_transiciones | 126780          | 0             | int64  |
| count_1            | 126780          | 0             | int64  |
| count_2            | 126780          | 0             | int64  |
| count_3            | 126780          | 0             | int64  |
| first              | 126780          | 0             | object |
| last               | 126780          | 0             | object |

## 2.3 Análisis exploratorio de los datos

Para este análisis, primero, se realiza resumen estadístico de las variables numéricas utilizando `df.describe()`. Se cuenta el total de valores únicos de las variables categóricas usando `cuenta_unica = df[col].nunique()`.

Tabla 2.5: Valores únicos de variables categóricas

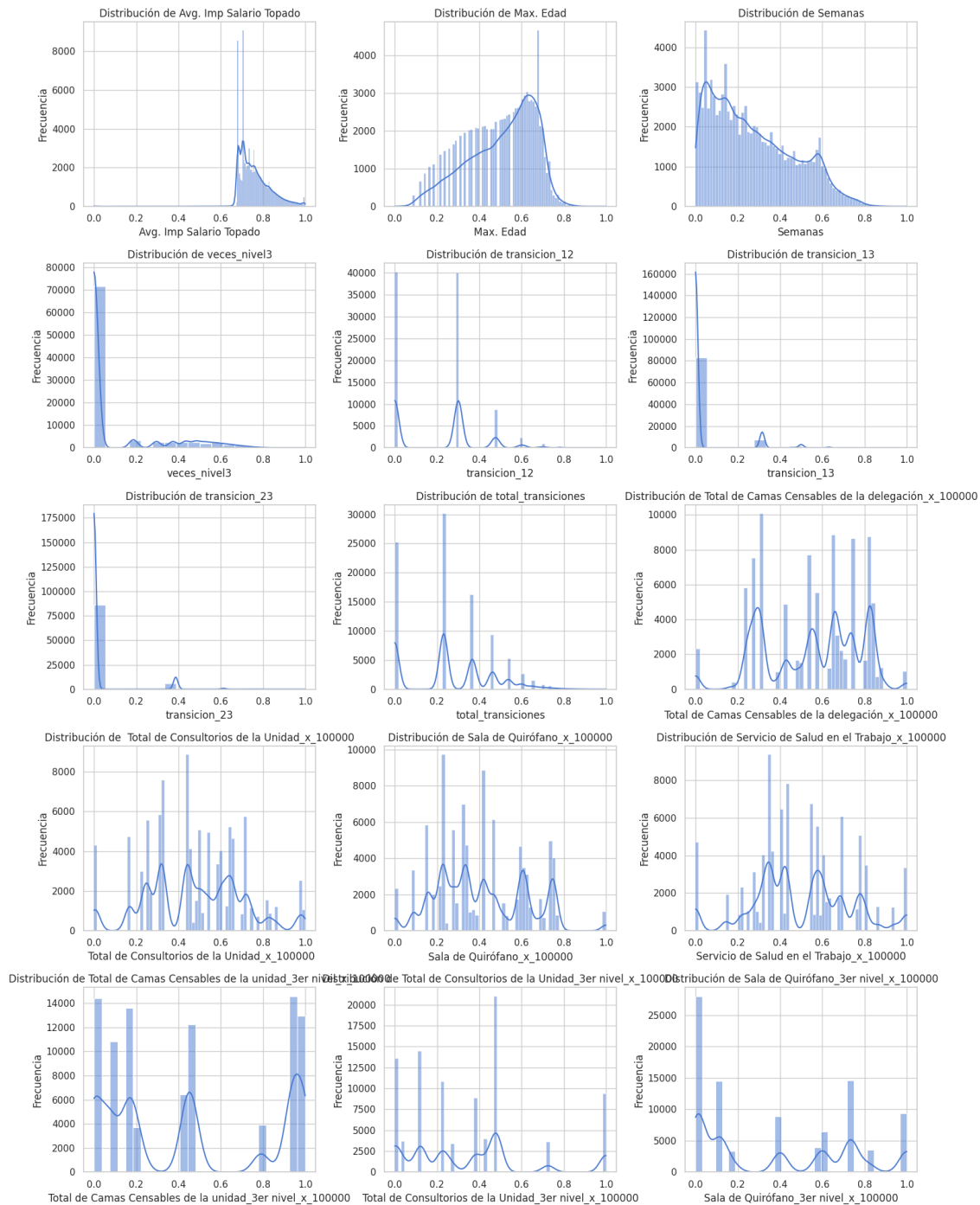
| Variable Categórica | Valores únicos |
|---------------------|----------------|
| Tip Ramo            | 2              |
| Delegación          | 35             |
| Unidad Ads          | 1231           |
| Cod Cie10           | 3114           |

Fuente: Elaboración propia, 2024.

Para la variable que clasifica la enfermedad según su relación con el trabajo, “Tip Ramo” cuenta con 2 categorías. La variable que describe las regiones administrativas del IMSS, “Delegación” presenta 35 diferentes. Por su parte, “Unidad Ads” registra 1231 valores únicos y “Cod Cie10”, que alude a la clasificación de enfermedades, llega a 3114 categorías distintas.

Se genera visualización para entender la distribución de las variables numéricas:

Figura 2.3: Distribución de variables numéricas posterior a preprocesamiento



Fuente: Elaboración propia, 2024.

La primera fila ilustra, que el salario presenta un pico pronunciado alrededor de 0.6, lo que sugiere que la mayoría de las observaciones se concentran en un nivel de ingreso medio-alto. La edad también presenta un pico similar, cercano a la

mitad, reflejando que un gran porcentaje de los pacientes se ubican en un rango intermedio de la distribución de edad transformada. En el caso de las semanas, la forma descendente con un mayor conteo al inicio indica que muchas incapacidades se concentran en períodos relativamente cortos, aunque con una cola larga aun en la variable ya transformada.

En cuanto a las variables relacionadas con la trayectoria de atención médica: `veces_nivel3`, `transicion_12` y `transicion_13`. Estas distribuciones están altamente concentradas en valores cercanos a 0, lo que denota que la mayoría de los pacientes no transita repetidamente entre niveles o, simplemente, lo hace con muy baja frecuencia. Este hallazgo apunta a que los movimientos hacia unidades de mayor complejidad son menos comunes o están restringidos a casos muy específicos.

Las variables, `transicion_23`, `total_transiciones`, así como indicadores de infraestructura, por ejemplo, Total de Camas Censables y Sala de Quirófano, cada uno por cada 100,000 habitantes, pueden presentar picos alrededor de la zona baja (cercana a 0), pero se distinguen múltiples cúmulos en valores intermedios (0.2, 0.4, 0.6), lo que sugiere que hay una heterogeneidad regional en la dotación de infraestructura. El hecho de que existan varios picos en las variables de capacidad hospitalaria indica que ciertas delegaciones o unidades cuentan con recursos significativamente distintos, lo que puede impactar la duración de las incapacidades.

Tabla 2.6: Estadística descriptiva de las variables numéricas

|        | Avg. Imp<br>Salario<br>Topado | Max. Edad  | Semanas    | Num Días<br>Acumulado<br>s | veces_nive<br>l3 | transicion_<br>12 | transicion_<br>13 | transicion_<br>23 | total_transi<br>ciones | Total de<br>Camas<br>Censables<br>de la<br>delegación<br>_x_100000 | Total de<br>Camas<br>Censables<br>de<br>Cirugía_x_<br>100000 | Total de<br>Camas de<br>Traumatolo<br>gía y<br>Ortopedia_<br>x_100000 |
|--------|-------------------------------|------------|------------|----------------------------|------------------|-------------------|-------------------|-------------------|------------------------|--|--|---|
| Cuenta | 126780                        | 126780     | 126780     | 126780                     | 126780           | 126780            | 126780            | 126780            | 126780                 | 126780   | 126780   | 126780  |
| Media  | 496.152071                    | 43.7329087 | 28.9424947 | 202.597463                 | -                | 0.75705031        | 0.12725113        | 0.07563161        | 1.64139495             | 54.771089  | 24.2124578   | 6.67431502  |
| DE     | 396.650863                    | 11.8332165 | 13.7881748 | 96.5172235                 | -                | 0.86327136        | 0.41848346        | 0.28813137        | 1.77084533             | 12.2400094   | 7.81719915   | 3.6997941   |
| Min    | 0                             | 16         | 14.4285714 | 101                        | 0                | 0                 | 0                 | 0                 | 0                      | 30.0093419   | 7.64661114   | 0.12467514  |
| 25%    | 261.2                         | 34         | 18.2857143 | 128                        | 0                | 0                 | 0                 | 0                 | 0                      | 41.5168227   | 18.5970064   | 5.36859906  |
| 50%    | 358                           | 45         | 24.1428571 | 169                        | 0                | 1                 | 0                 | 0                 | 1                      | 54.5350191   | 23.0159673   | 8.1891269   |
| 75%    | 558.535                       | 53         | 36         | 252                        | 0                | 1                 | 0                 | 0                 | 2                      | 64.6010448   | 31.9254196   | 8.8357558   |
| Max    | 2714.25                       | 104        | 128.142857 | 897                        | 40               | 9                 | 8                 | 5                 | 19                     | 84.5610396   | 37.4515398   | 13.5844171  |

Fuente: Elaboración propia, 2024.



Para la variable de salario, se presenta un promedio cercano a 496.15, con una desviación estándar de 396.65, un mínimo de 0 y un máximo de 2714.25, mientras que edad alcanza una media de 43.73 años y se extiende entre 16 y 104. Es la variable a predecir, “Semanas”, que ronda un promedio de 28.94 y la dispersión es de 13.79, con valores entre 14.43 y 128.14.

Por otro lado, las variables de movimientos (“transicion\_12”, “transicion\_13” y “transicion\_23”) muestran promedios de 0.76, 0.13 y 0.08, respectivamente, llegando hasta valores máximos de 9, 8 y 5, al tiempo que la suma de todas esas transiciones (“total\_transiciones”) mantiene una media de 1.64 y un máximo de 19. En cuanto a la capacidad hospitalaria, la variable de camas disponibles en cada delegación es en promedio 54.77, con un rango entre 30.01 y 84.56, las destinadas a cirugía un promedio de 24.21 con un máximo de 37.45, y las de traumatología y ortopedia se ubican en torno a 6.67, alcanzando un máximo de 13.58.

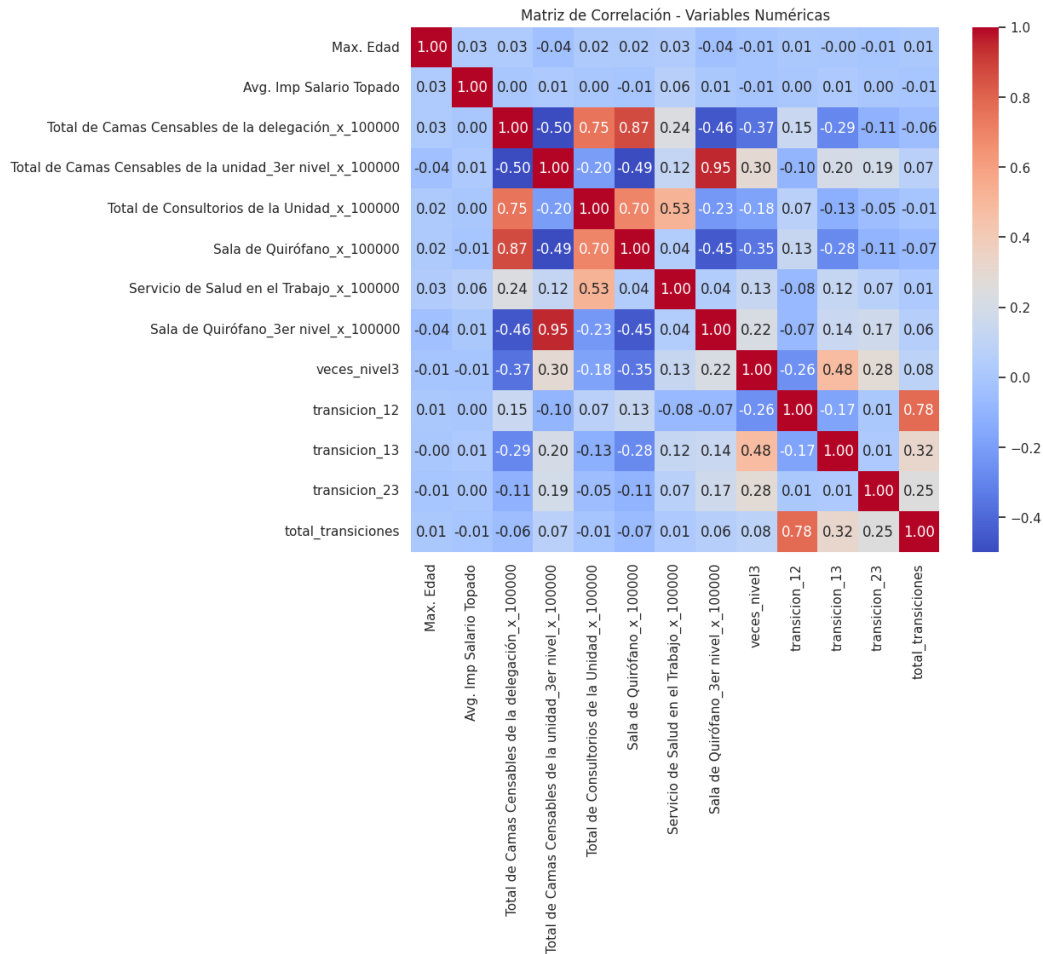
Para identificar qué columnas de infraestructura resultaban redundantes, se calculó la matriz de correlaciones entre todas las variables cuantitativas y se representó mediante un mapa de calor (Figura 2.4).

Al revisar el mapa de calor, se observaron varios grupos de variables con alta correlación. Por ejemplo, *Total de Camas Censables de Cirugía\_x\_100000* y *Total de Camas de Traumatología y Ortopedia\_x\_100000* exhibían correlaciones superiores a 0.85, lo cual indica que ambas captan información muy parecida acerca de la infraestructura hospitalaria. Por lo que solo se conservarán algunas de las variables, para evitar problemas de multicolinealidad en el modelo.

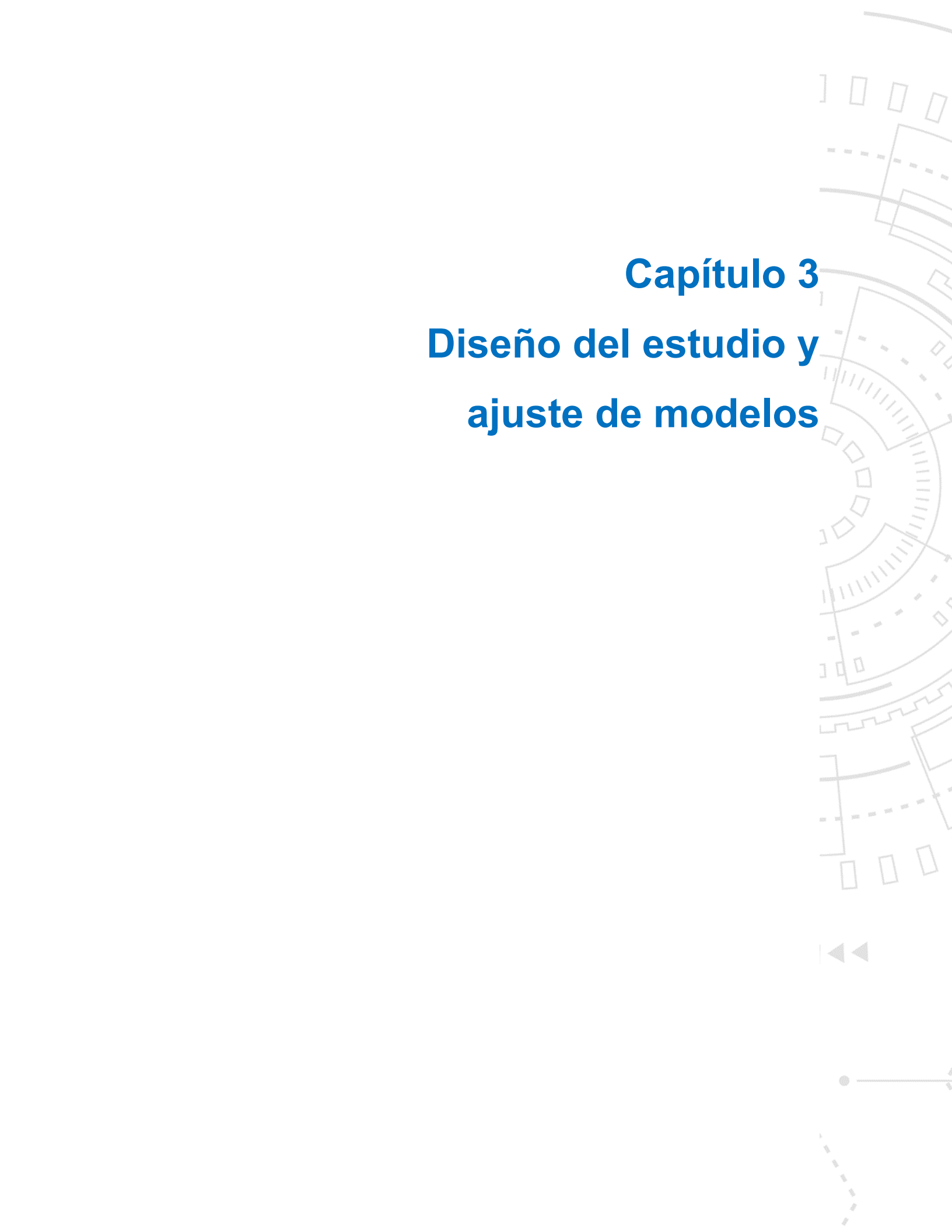
Asimismo, destaca que los recursos de tercer nivel, en particular *Total de Camas Censables de la unidad\_3er\_nivel\_x\_100000* y *Sala de Quirófano\_3er\_nivel\_x\_100000*, presentan una alta correlación con *veces\_nivel3*. Esto se diferencia de *Total de Camas Censables de la delegación\_x\_100000*, que muestra correlaciones negativas con las camas y quirófanos de tercer nivel, apuntando a posibles desigualdades en la distribución de infraestructura y a la compensación con el uso de camas de unidades médicas de mayor especialidad y costo.

Para elegir las variables a conservar, se priorizaron las variables que engloban la capacidad hospitalaria, y se eliminaron aquellos que forman un subconjunto (por ejemplo, *Total de Camas Censables de la delegación\_x\_100000* es una variable global, mientras que las camas de una especialidad específica es un grupo más pequeño y contenido en el total de camas). De esta manera, se mantuvieron únicamente aquellas variables de infraestructura que mostraban correlaciones moderadas entre sí y ofrecían un valor predictivo diferenciado.

Figura 2.4. Mapa de Calor de Correlaciones



Fuente: Elaboración propia, 2024.



# **Capítulo 3**

## **Diseño del estudio y ajuste de modelos**

## Capítulo 3. Diseño del estudio y ajuste de modelos

El Capítulo 3 se centra en el diseño del estudio y el ajuste de modelos predictivos, combinando fundamentos teóricos y una metodología aplicada al análisis de incapacidades temporales en la seguridad social en México. Inicialmente, se presenta un marco teórico que revisa la evolución de la protección de los trabajadores y la importancia de la seguridad social, destacando cómo surgieron estos mecanismos ante las condiciones laborales precarias de la Revolución Industrial. Se describen los esquemas de financiamiento del Seguro de Riesgos de Trabajo y del Seguro de Enfermedades y Maternidad, así como los desafíos relacionados con el uso excesivo de incapacidades y la necesidad de optimizar la prescripción médica mediante guías y comités especializados.

Posteriormente, el capítulo introduce la ciencia de datos y el aprendizaje automático, enfatizando la aplicación de modelos y algoritmos, como la regresión lineal, árboles de decisión, Random Forest, Gradient Boosting, CatBoost y XGBoost. Se explica el proceso de ajuste de hiperparámetros y la importancia de abordar problemas de sobreajuste en contextos complejos de clasificación y regresión.

La metodología se fundamenta en el modelo CRISP-DM, que abarca desde la comprensión del negocio y la exploración de datos, pasando por la preparación y selección de variables, hasta el modelado, evaluación y despliegue de soluciones. Esta integración de teoría y práctica permite desarrollar herramientas predictivas que faciliten la asignación eficiente de recursos, mejoren la atención al paciente y fortalezcan la toma de decisiones en el manejo de incapacidades, contribuyendo a la sostenibilidad y equidad del sistema de seguridad social.

### 3.1 Marco teórico

#### 3.1.1 Seguridad Social y Protección de los Trabajadores

La seguridad social se concibe como un pilar en la estructura de las sociedades modernas, pues su objetivo, es garantizar el bienestar y la protección de los trabajadores frente a riesgos inherentes a la vida laboral y personal. Esta red de protección incluye prestaciones en salud, jubilación, accidentes de trabajo, y

desempleo, entre otros. Además de que se configura como un mecanismo de justicia social, al mitigar las desigualdades y promover la cohesión social (Gomis et al., 2020).

Históricamente, la necesidad de establecer esquemas de seguridad social encontró gran interés durante la revolución industrial, cuando el aumento de la industrialización y las nuevas tecnologías crearon condiciones laborales precarias no antes vistas, además de altos índices de vulnerabilidad entre la clase trabajadora. Por lo que el avance de la seguridad social fue una respuesta a la exigencia de crear un marco que asegurara derechos mínimos y condiciones dignas de trabajo. La evolución de la seguridad social ha ido más allá de la mejora de las condiciones laborales y aseguramiento ante riesgos de trabajo. Actualmente representa un instrumento de justicia social y redistribución de la riqueza (Lagunas Sosa et al., 2023).

La protección de los trabajadores no solo implica los beneficios económicos, sino también la promoción de ambientes de trabajo seguros y saludables. En este sentido, en México se ha forjado una colaboración entre el Estado, los empleadores y los trabajadores para el diseño e implementación de estrategias efectivas de protección. Estas estrategias generan beneficios para el bienestar individual, y en la productividad y competitividad de las empresas (Cabrera & López, 2019).

Aunque, la efectividad de estas estrategias depende en gran medida de la capacidad de los gobiernos para articular políticas públicas sostenibles. Es necesario promover reformas que consideren la diversidad del mercado laboral y que incluyan a aquellos sectores tradicionalmente excluidos, como los trabajadores informales y los nuevos trabajadores digitales. La ampliación de la cobertura y la mejora en la gestión de los recursos destinados a la seguridad social son desafíos que requieren un enfoque integral y colaborativo, que involucre tanto a actores nacionales como internacionales (Esping-Andersen, 2016).

### **3.1.2 Incapacidades Temporales de Trabajo en México**

Este sistema de protección es continuamente evaluado en cuanto a su suficiencia económica. Aquellas prestaciones por riesgo de trabajo son financiadas por el Seguro de Riesgos de Trabajo (SRT), cuyos ingresos son mediante las cuotas y aportaciones que realizan los patrones, según una prima calculada del histórico de riesgos de trabajo. En 2023 se expidieron aproximadamente 2.2 millones de certificados de incapacidad, cubriendo 16.9 millones de días subsidiados, equivalente a 38,948 millones de pesos. Esto representó un incremento en comparación con años anteriores. El gasto en subsidios por incapacidad temporal aumentó en términos reales (21.6% más que en 2022) debido al crecimiento en la población asegurada y al incremento en el salario base de cotización (Instituto Mexicano del Seguro Social, 2024).

Aquellas prestaciones por enfermedad general son financiadas por el Seguro de Enfermedades y Maternidad (SEM). Durante el 2023, se expidieron 9 millones de certificados de ITT, 58 millones de días de incapacidad, equivalente a 12,118 millones de pesos. Estos números muestran una reducción del 14% en la emisión de certificados y del 6% en el total de días en comparación con 2022, lo que se tradujo en una disminución del gasto (en términos reales) de 0.1% respecto al año anterior (IMSS, 2024).

El SRT y el SEM tienen diferentes fuentes de ingresos. El SRT recibe aportaciones exclusivamente de los patrones que varían según sus historiales de riesgos de trabajo, y el riesgo inherente que el trabajo conlleva. El SEM recibe aportación de los patrones por cada trabajador; recibe aportación por el trabajador, si está activo y vinculada a su nivel salarial; y recibe aportaciones del Gobierno Federal en función del número de trabajadores y grupos especiales (IMSS, 2024).

Pero este sistema enfrenta riesgos en su sustentabilidad. Uno de los factores es el uso excesivo y no justificado de la incapacidad. Esto puede contener un elemento relacionado al desconocimiento del profesional médico, que puede ser evitado con retroinformación continua y oportuna al personal médico, a fin de que los mismos comprendan mejor el proceso y los casos en los que el recurso se amerita correctamente o no (Castro VOJ et al., 2019.; Lanz, 2018).

Otro factor, es la complejidad que implica el pronosticar en un momento dado, si un paciente se recuperará adecuadamente o si llegará a invalidez o pensión. Para mejorar estos pronósticos, el IMSS ha desarrollado “Guías de duración de la incapacidad por patología, en apoyo a la prescripción de la incapacidad”. Estas guías constituyen un instrumento, incluido en los sistemas del expediente clínico electrónico, que se usa como referente para determinar los días probables de recuperación, considerando su ocupación y esfuerzo físico que el trabajador realiza. Estas guías también apoyan a coartar abusos en la expedición de incapacidad injustificada, pues sirven como instrumento para evaluar el ejercicio y congruencia de los profesionales médicos (Martin-Fumadó et al., 2014; IMSS, 2024).

Una estrategia más que se ha implementado para controlar el uso adecuado de la incapacidad, es la creación del Comité para el Control de la Incapacidad Temporal (COCOITT) en 2008. La creación de estos comités fue motivada principalmente por el gasto que representa el subsidio para incapacidades, y la necesidad de mejorar el uso de los recursos públicos al eliminar los desperdicios en los procesos que conlleva la expedición de estas (IMSS, 2024).

A pesar de estos esfuerzos, la transición epidemiológica ha generado una alta prevalencia de enfermedades crónico-degenerativas (como diabetes, hipertensión, cardiovasculares y cáncer) que pueden derivar en incapacidades. Por ejemplo, la diabetes mellitus es una de las principales causas de invalidez, pues sus complicaciones (ceguera, neuropatías, insuficiencia renal) llevan a que muchos trabajadores lleguen irremediablemente a la invalidez. Esto genera una presión financiera al Seguro de Invalidez y Vida. Asimismo, estos pacientes crónicos suelen requerir más días de incapacidad temporal por episodios relacionados con su enfermedad, elevando el gasto en subsidios (IMSS, 2024). Y sumando a esto, un paciente que ya ha tenido incapacidad laboral prolongada tiene dificultades para regresar a su trabajo, o vuelve a sufrir un accidente (Gewurtz et al., 2019).

En diferentes países del mundo, se han visto diferencias en la reincorporación de personas con incapacidad laboral. Dos años tras el inicio de la incapacidad, solo el 40% de trabajadores en Dinamarca y Alemania volvieron a trabajar, frente al 60% en Israel, Suecia y EE. UU., y más del 70% en Países Bajos.

Se identificaron cuatro patrones: resumidores continuos, tardíos, recaídas y no resumidores, siendo mayormente retorno continuo o ausencia total, lo que indica que ocurre principalmente en el primer año. Además, en Dinamarca, Israel y EE. UU., entre el 40 y 50% regresaron con nuevo empleador, mientras que, en Alemania, Países Bajos y Suecia, entre el 70 y 80% retornaron con su anterior. Factores como edad (mayores de 55 años), bajo nivel educativo y soltería se asocian a menores tasas de retorno. Las intervenciones médicas aportaron poco, salvo en Suecia, donde cirugías tempranas (en los primeros 3 meses) mostraron asociación con el pronto retorno (Bloch & Prins, 2001).

### **3.1.3 Retos de la seguridad social en incapacidades del trabajo**

Dentro de este amplio marco de protección, la incapacidad temporal para el trabajo es un mecanismo de protección para los trabajadores que no pueden desempeñar sus labores por un periodo determinado debido a una enfermedad o accidente. Esto se encuentra regulado por la Ley del Seguro Social, donde se define que dicha protección será a través subsidios y prestaciones en especie mientras dure la incapacidad. Para ejercer este derecho, el IMSS expide un certificado de incapacidad temporal, documento médico y legal que constata la inhabilidad laboral y que produce efectos legales para justificar la ausencia del trabajador y otorgarle las prestaciones correspondientes (Ley Del Seguro Social, 2008).

La incapacidad puede otorgarse ante riesgos de trabajo (accidentes dentro del lugar de trabajo o enfermedades causadas por el ejercicio de la actividad laboral) o por enfermedad general (no asociada al trabajo). Cuando la incapacidad deriva de un riesgo de trabajo, se requiere adicionalmente que el patrón emita el aviso de accidente de trabajo y lo presente al IMSS. Caso contrario, si la incapacidad proviene de una enfermedad general, no se requiere ese aviso patronal (Hernández, 2023).

La duración de una incapacidad temporal depende de la naturaleza del padecimiento y de los límites legales establecidos. No existe un período mínimo obligatorio más allá de lo que el médico estime necesario (puede ser desde un solo



día de reposo). Sin embargo, sí hay períodos máximos fijados para el goce de las incapacidades temporales, diferentes según se trate de enfermedad general o de riesgo de trabajo (Hernández, 2023; IMSS, 2006).

Para el caso de enfermedad general, la Ley señala que el subsidio económico se otorgará a partir del cuarto día de iniciada la incapacidad y hasta por 52 semanas. Posterior a las 52 semanas se debe realizar una evaluación médica para autorizar o no, una prórroga de 26 semanas adicionales. Para el caso de riesgo de trabajo, el máximo de subsidio es hasta por 52 semanas. Durante periodo de tiempo debe existir siempre la prioridad de definir si el trabajador puede regresar a sus actividades laborales o si amerita una pensión por incapacidad permanente. Es por esto que no existen prorrogas para riesgos de trabajo. Al concluir el periodo, el trabajador se reintegra a laborar o se dictamina incapacidad (Ley Del Seguro Social, 2008; Secretaría de Trabajo y Previsión Social, 2021).

En términos médicos, la transición a invalidez se formaliza cuando el IMSS emite el dictamen de estado de invalidez del asegurado. Antes de ello, es posible que el IMSS otorgue una pensión provisional o temporal de invalidez. La Ley contempla pensiones de invalidez de dos tipos: pensión temporal y pensión definitiva, dependiendo de la expectativa de recuperación. Por lo general, si el IMSS considera que existe alguna posibilidad de recuperación en el mediano plazo, se otorga una pensión temporal (por ejemplo, por uno o dos años, sujeta a revisiones). Durante ese periodo, el Instituto puede ordenar nuevos exámenes médicos para verificar si el trabajador se ha recuperado adecuadamente. Si al término de la pensión temporal no hay mejoría suficiente, la invalidez se declara permanente y se convierte en pensión definitiva. Caso contrario, la pensión de invalidez puede ser suspendida (IMSS, 2006; Ley Del Seguro Social, 2008).

Las incapacidades no solo requieren subsidio económico. Además, en México, la falta de médicos se vuelve una limitante para la adecuada valoración de los trabajadores que requieren atención tras accidentes de trabajo, así como para la determinación de su invalidez o la gestión de pensión, lo cual complica aún más la capacidad de respuesta y seguimiento oportuno de estos casos. El profesional médico es un trabajador inmerso en un mercado de cada vez mayores y más

complejas expectativas. A esto se suma la demanda de que los servicios sean humanos y en sintonía con las necesidades holísticas de las personas, y no solo la atención de su enfermedad (Rotenstein et al., 2018).

Desempeñarse como profesional médico conlleva también actividades administrativas, como la correcta integración de documentación o su participación en acciones de mejora continua, además de su contribución en actividades de planeación de servicios. Todas ellas pueden consumir cantidades significativas de tiempo (West et al., 2018).

México es un país con notoria escasez de médicos. Cada año, 17,500 nuevos médicos generales se gradúan, más 12,500 médicos especialistas (DGCES, 2023). Para 2023, el total de médicos es cerca de 666,000 personas (UNAM, 2023), lo que se traduce en aproximadamente 2.5 médicos por cada 1,000 habitantes. Este número es bajo a comparación de los países miembros de la OCDE, ubicando a México en el cuartil inferior (*Médicos (Por Cada 1.000 Personas) - OECD Members*, n.d.). Y esta baja tasa de médicos, se exacerba cuando se considera que hasta un tercio de los médicos con licencia no ejercen su profesión o han tomado roles administrativos (Nigenda et al., 2005). Además, la distribución de médicos es desigual, de manera que las ciudades pueden tener una sobrepoblación de personal, dados los incentivos económicos y la calidad de vida (Bärnighausen & Bloom, 2009).

#### **3.1.4 Ciencia de Datos**

La ciencia de datos es un campo interdisciplinario con el objetivo de generar conocimiento útil y novedoso a partir del análisis de datos masivos. Para esto se utiliza la estadística, potenciada por algoritmos computacionales y de inteligencia artificial para extraer conocimiento de grandes volúmenes de información (Provost & Fawcett, 2013).

A diferencia de enfoques tradicionales, la ciencia de datos abarca el ciclo de vida completo de los datos, desde la recolección y limpieza hasta el análisis, modelado y visualización de resultados. Todo este ciclo, requiere métodos robustos,

como el proceso de descubrimiento a través de minería de datos; el aprendizaje automático que desarrolla algoritmos que permiten a las computadoras aprender patrones a partir de los datos sin ser programadas explícitamente; la inteligencia artificial, que busca dotar a las máquinas de habilidades similares a las humanas, como el razonamiento y la toma de decisiones (Murphy, 2012; Provost & Fawcett, 2013).

El aprendizaje automático se divide en tres categorías principales, el aprendizaje supervisado, el aprendizaje no supervisado, y el aprendizaje por refuerzo (Murphy, 2012). La más común, en la práctica es el aprendizaje supervisado, en el cual se cuenta con un conjunto de datos etiquetados ( $\{(x_i, y_i)\}_{i=1}^n$ ), donde cada observación ( $x_i$ ) está asociada a un valor o categoría objetivo ( $y_i$ ). El objetivo en este modelo es el de poder aprender una función de predicción ( $f$ ), que con cada nueva observación ( $x_i$ ), sea capaz de estimar correctamente el valor de ( $y_i$ ).

Por su parte, el aprendizaje no supervisado busca descubrir patrones o estructuras ocultas en datos sin etiquetas. El objetivo es encontrar representaciones internas de la información. Por último, el aprendizaje por refuerzo implica que un agente interactúe con un entorno y toma decisiones secuenciales guiadas por recompensas o penalizaciones, útil para problemas de optimización como el control robótico o la asignación de recursos en tiempo real (Sutton & Barto, 2018).

Existe gran diversidad de modelos utilizados para aprendizaje supervisado. Cada uno de ellos con propiedades diferentes que los hacen más adecuados para objetivos específicos. Uno de los más básicos es la regresión lineal, que permite predecir valores numéricos con base en una relación matemática entre variables. Otro modelo, los árboles de decisión, estructuran la toma de decisiones dividiendo los datos en ramas basadas en reglas lógicas; aunque este tipo de modelo puede llegar a sobreajustar. Es por este problema, que surgió el modelo Random Forest, que combina múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste, aunque a cambio de una menor velocidad de procesamiento cuando los conjuntos de datos crecen (Murphy, 2012).

Entre los diferentes modelos, la regresión ordinal se utiliza para resolver problemas en los que la variable de interés es categórica y mantiene un orden natural, pero las distancias entre cada una de sus categorías no están definidas de manera explícita. Un ejemplo típico aparece en escalas de satisfacción (por ejemplo, “bajo”, “medio” y “alto”) o en evaluaciones donde las opciones pueden clasificarse en niveles crecientes sin que existan intervalos exactamente equivalentes entre ellas. Este modelo puede ser contrastado con otras técnicas de regresión, como la lineal, en la que la variable dependiente es continua, o la logística, donde las respuestas categóricas no tienen orden. Mientras tanto, la regresión ordinal aprovecha el hecho de que las categorías mantienen un orden y, por lo tanto, se modelan mediante un conjunto de umbrales que determinan el cambio de categoría. Matemáticamente, el modelo asume la existencia de una variable latente continua  $y^*$ , definida a partir de las covariables  $X$  a través de la relación  $y^* = X\beta + \varepsilon$ . Esta variable latente se compara con un conjunto de umbrales  $\{\alpha_0, \alpha_1, \dots, \alpha_K\}$  con  $\alpha_0 = -\infty$  y  $\alpha_K = +\infty$ , de manera que la categoría observada  $y$ , se asigna a uno de los niveles según el intervalo en que caiga  $y^*$ . Esto permite escribir  $P(y = k | X) = F(\alpha_k - X\beta) - F(\alpha_{k-1} - X\beta)$ , donde  $F$  es una función de distribución acumulada, comúnmente la logística o la normal (McCullagh, 1980; Agresti, 2012).

Un segundo modelo, XGBoost (Chen y Guestrin, 2016) es un método de boosting empleado para optimizar la precisión en tareas de clasificación o regresión a gran escala. El boosting consiste en construir secuencialmente modelos débiles (árboles de decisión de profundidad limitada), de modo que cada nuevo árbol corrija los errores de los anteriores. El modelo final, entonces, se expresa como la suma de todos esos árboles. De manera que, si contamos con un conjunto de datos  $\{(x_i, y_i)\}$  para  $i = 1, \dots, n$ , la predicción  $\hat{y}_i$  se calcula como  $\hat{y}_i = \sum_{t=1}^T f_t(x_i)$ , donde cada  $f_t$  pertenece a un espacio de árboles base. La función objetivo combina la pérdida  $\mathcal{L}_{\ell_s}(y_i, \hat{y}_i)$  con un término de regularización que penaliza la complejidad de cada árbol,  $\Omega(f_t)$ . Una de las características de XGBoost es el uso de gradientes y segundas derivadas para evaluar, en cada nodo, la ganancia esperada por la división; por lo que se considera la expresión

$$\text{Ganancia} = \frac{1}{2} \left( \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma,$$

donde  $G_L$  y  $H_L$  representan las sumas de gradientes y hessianas en el lado izquierdo de la partición, y  $G_R$  y  $H_R$  las del lado derecho. Los hiperparámetros  $\lambda$  y  $\gamma$  permiten controlar la complejidad del árbol para evitar el sobreajuste.

Por último, CatBoost (Prokhorenkova et al., 2018) también utiliza gradient boosting, pero es particular en su manejo nativo de variables categóricas y por la estructura simétrica de los árboles que construye. Es el caso de otros modelos, donde se realizan transformaciones como codificaciones one-hot o target encoding a variables categóricas, mientras que en CatBoost se recurre a estadísticas de manera incremental y a permutaciones aleatorias de los datos para reducir sesgos y fugas de información al tratar categorías. Además, utiliza árboles en los que todos los nodos de un mismo nivel se dividen según la misma regla, lo que mejora la paralelización y reduce la probabilidad de sobreajuste. En el caso de variables numéricas, CatBoost binariza cada característica, al experimentar con diversos puntos de corte y eligiendo aquel que más incrementa la ganancia.

Estos modelos pueden ser utilizados según la variable objetivo, o según el tipo de variables predictoras, además de la complejidad del problema que se esté intentando resolver. Cuando la variable es categórica con un orden específico, la regresión ordinal resulta adecuada, mientras que para datos estructurados con multitud de variables numéricas o categóricas, métodos de boosting como XGBoost y CatBoost presentan mejor capacidad para generar predicciones precisas.

### 3.1.5 Técnicas para el preprocesamiento y análisis

Para el preprocesamiento de este trabajo, se procederá a utilizar Frequency Encoding para las variables categóricas de alta cardinalidad. Esta técnica toma una variable categórica  $X$  con categorías  $\{c_1, c_2, \dots, c_k\}$  y un conjunto de datos de tamaño  $N$ , y cuenta la frecuencia de cada categoría  $c_i$ . Remplazando la codificación por la frecuencia de la categoría:

$$\text{Frecuencia}(X = c_i) = \frac{\sum_{n=1}^N 1\{X_n = c_i\}}{N},$$

Donde  $1\{\cdot\}$  es la función indicador (vale 1 si la condición se cumple y 0 en caso contrario) (Micci-Barreca, 2001).

Para las variables categóricas de baja cardinalidad se empleará One Hot Encoding (Hastie et al., 2009). En esta técnica se toma una variable categórica  $X$  con categorías  $\{c_1, c_2, \dots, c_k\}$ , posteriormente se procede a codificar cada categoría  $c_i$  en un vector binario de dimensión  $k$ , donde todas las entradas son 0 excepto la correspondiente a  $c_i$ , que se marca con 1. Para la observación  $x \in \{c_1, \dots, c_k\}$ :

$$\text{OneHot}(x) = \begin{bmatrix} 1x = c_1 \\ 1x = c_2 \\ \dots \\ 1x = c_k \end{bmatrix}$$

En cuanto a las variables numéricas, se procederá a realizar transformación logarítmica para las variables que no tengan una distribución normal. Para lograr esto, se utilizará la forma típica:  $X' = \log(X + 1)$ , donde se suma 1 para evitar la indefinición en caso de ( $X = 0$ ) o valores muy pequeños (Box et al., 1964).

Posteriormente se realizará escalado, para centrar los datos y normalizarlos a varianza unitaria, donde una variable  $X$ , con medida  $\mu$  y desviación estándar  $\sigma$ , el escalado estándar se define como:

$$X' = \frac{X - \mu}{\sigma}$$

Para el entrenamiento del modelo, se realizará ajuste de hiperparámetros, con el uso de Random Search (Bergstra & Bengio, 2012). Esta técnica busca hiperparámetros de manera aleatorio en un espacio predefinido. Se denota el espacio de hiperparámetros  $\Theta$ , como y la función de pérdida como  $L(\theta)$ , para extraer

muestras aleatorias  $\{\theta_1, \theta_2, \dots, \theta_m\}$  de  $\Theta$ . Entonces se entrena y valida el modelo para cada  $\theta_i$  y seleccionar  $\theta^*$  que minimice o maximice la métrica objetivo:

$$\theta^* = \arg \min_{\theta \in \{\theta_1, \theta_2, \dots, \theta_m\}} L(\theta)$$

Además, se realizará validación cruzada  $k$  – fold, que mantiene la proporción de clases en cada partición. Sea  $D$  el conjunto de datos y  $k$  el número de iteración. Si  $\pi_j$  es la proporción de la clase  $j$  en todo  $D$ , entonces, en cada pliegue  $D_i$  se intenta preservar la misma distribución  $\{\pi_j\}$ . De manera que, si  $y$  representa las etiquetas y  $k$  el número de pliegues, *StratifiedKFold* genera subconjuntos  $\{D_1, D_2, \dots, D_k\}$  tales que:

$$\pi_j^i \approx \pi_j \quad \forall i \in \{1, \dots, k\}, \quad \forall j \in \{\text{clases}\}$$

En cuanto a las métricas a utilizar para medir y comparar el desempeño de los modelos, primero se utilizará exactitud. Esta métrica mide la fracción de predicciones correctas entre el total de ejemplos. Sean  $TP$  verdaderos positivos,  $TN$  verdaderos negativos,  $FP$  falsos positivos y  $FN$  falsos negativos:

$$\text{Exactitud} = \frac{TP + TN}{TP + TN + FP + FN}.$$

También se utilizará Recall, que mide la capacidad del modelo para encontrar todos los positivos reales:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Y también, F1-Score, que es un promedio armónico entre la precisión y el recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Se incluye la kappa ponderada de Cohen, para medir el grado de concordancia entre dos evaluaciones, considerando la magnitud de los desacuerdos. Si  $O_{i,j}$ , entonces la proporción observada de veces que se clasifica una instancia en la categoría  $i$  por el primer modelo y en  $j$  por el segundo.  $E_{i,j}$  es la proporción esperada bajo independencia, y  $w_{i,j}$  es una matriz de pesos que penaliza en mayor o menor grado la diferencia entre  $i$  y  $j$  (Cohen, 1968). Por lo que:

$$\kappa_w = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

### 3.1.6 Antecedentes de investigación

El aprendizaje automático ha ido teniendo creciente aplicación en la industria de aseguradoras y salud, desde la planeación y presupuestación de programas, hasta el impacto de la implementación de diferentes políticas. Con estas herramientas se ha logrado optimizar el uso de recursos, mediante el refinamiento de acciones preventivas dirigidas a poblaciones específicas, gracias al análisis de determinantes sociales y otras fuentes de datos (Nayak et al., 2019). En la planeación de programas de salud, estos modelos han demostrado su utilidad en la recaudación, presupuestación y gasto, contribuyendo a reducir la presión financiera de las instituciones de salud (Ramezani et al., 2023). En cuanto a los seguros para accidentes de trabajo y otros riesgos laborales, el uso de árboles de decisión ha permitido optimizar la asignación de recursos y disminuir costos en instituciones de países como Australia (Chand & Zhang, 2022).

Por otro lado, las aplicaciones de modelos predictivos se han extendido a la gestión y expedición de incapacidades. Diversas investigaciones destacan el potencial de los algoritmos de procesamiento de lenguaje natural, técnicas de Random Forest, XGBoost o GBM para predecir el riesgo de discapacidad laboral (Saarela et al., 2022), la probabilidad de invalidez y consecuentes cuidados a largo



plazo (Chou JCL et al., 2022), la discapacidad permanente (Koc et al., 2021) o incluso el tiempo de regreso al trabajo (Na & Kim, 2019). Asimismo, estudios como el de Meyers et al. (2018) ilustran cómo el uso de técnicas de aprendizaje automático ayuda a vigilar las lesiones ocupacionales y focalizar los recursos preventivos en los sectores e intervenciones donde se presenta mayor riesgo.

En la Tabla 3.1 se presenta un resumen de la literatura revisada, describiendo objetivo, metodología y resultados destacados de los estudios más relevantes relacionados con la predicción y gestión de incapacidades laborales.

Tabla 3.1: Resumen de antecedentes

| Documento              | Origen de datos  | Análisis o Propósito  | Técnicas  |
|------------------------|--|---|---|
| Meyers et al. (2018)   | 1.2 millones de reclamaciones de compensación laboral del estado de Ohio   | Generar recomendaciones sobre seguridad e higiene, identificando sectores de alto riesgo  | Algoritmo de autocodificación, basado en técnicas bayesianas, con 90% de precisión.   |
| Na & Kim (2019)        | Datos de 2000 participantes del Korea Workers' Compensation & Welfare Service                                      | Clasificar a los trabajadores en dos grupos: aquellos que retornaron y los que no   | Gradient Boosting Machine   |
| Chand & Zhang (2022)   | 89,299 registros del Australian National Disability Insurance Scheme   | Minimizar la diferencia entre el presupuesto asignado y el gasto real de los participantes del NDIS para lograr una asignación de fondos más ajustada a las necesidades reales. | Regresión lineal<br>Máquinas de soporte vectorial<br>Árbol de decisión<br>Perceptrón multicapa  |
| Chou JCL et al. (2022) | Datos históricos recogidos entre agosto de 1997 y marzo de 2021, de clientes de A-Life Insurance Company (Taiwán). | Predecir la probabilidad de adquirir un seguro de cuidados de largo plazo, permitiendo a las aseguradoras identificar clientes potenciales.                                     | Naïve Bayes.<br>Regresión Logística.<br>IBk (k-Nearest Neighbors).  |
| Saarela et al. (2022)  | Historias clínicas de una organización hospitalaria, y datos tabulares sobre características socioeconómicas.      | Predecir el riesgo de discapacidad laboral para permitir intervenciones tempranas.  | Procesamiento de lenguaje natural en combinación con Random Forest, XGBoost, Regresión logística con regularización L1, Árbol de decisión, MLP. |

|                          |   |  |  |
|--------------------------|---|--|--|
| Koc et al.<br>(2021)     | Registro de 47,938 accidentes laborales en Turquía  | Predecir el estado de discapacidad post-accidente (discapacidad permanente vs. no discapacidad) para apoyar decisiones en la gestión de seguridad en el trabajo.                   | Random Forest<br>XGBoost<br>AdaBoost<br>Extra Trees                          |
| Kusnadi et al.<br>(2023) | Datos de incapacidad por maternidad de "North American Group Long-Term Disability".           | Predecir la tasa de recuperación en incapacidad por maternidad y ajustar primas de seguro  | XGBoost<br>Gradient Boosting Machine<br>Bayesian Additive<br>Regression Tree |
| Cheng et al.<br>(2020)   | 50,000 expedientes de trabajadores lesionados   | Desarrollar un sistema para gestionar información de lesiones laborales y, mediante técnicas de inteligencia artificial, realizar análisis de predicción de trayectorias y tiempos | Procesamiento de lenguaje natural (OCR) y LSTM                               |
| Choi et al.<br>(2023)    | 42,219 casos de enfermedades ocupacionales de Korea Workers' Compensation and Welfare Service | Predecir la probabilidad de desaprobación de reclamos sobre compensaciones laborales   | Decision Tree, DNN, XGBoost y LightGBM                                       |
| Huhta-Koivisto<br>(2020) | 9,000 casos de visitas a salud ocupacional en Finlandia.                                      | Predecir el riesgo de discapacidad laboral de forma temprana, a través de agilizar el tiempo a la valoración por salud ocupacional.  | Modelo ULMFiT basado en redes neuronales AWD-LSTM                            |
| Brahimi et al.<br>(2022) | 65,533 de licencias por enfermedad de un hospital en Arabia Saudita.                          | Detectar de manera automatizada las licencias médicas injustificadas.  | Naïve Bayes<br>Regresión Logística<br>K-Nearest Neighbor                     |

Fuente: Elaboración propia, 2025.

Cada uno de los estudios revisados aborda de diferente perspectiva la gestión de incapacidades laborales, de acuerdo con las necesidades de investigación relacionadas. Para entender su relevancia como antecedentes, se describen a mayor detalle:

Nayak et al. (2019), realizaron revisión de la literatura para investigar la adopción de tecnologías en seguros de salud y su inclusividad. Remarcan gran énfasis en que las organizaciones deben contar con adecuadas estrategias tecnológicas. Describen que, los directivos, entienden y esperan el impacto que los modelos predictivos tendrán en su toma de decisiones, a fin de mejorar sistemas de seguridad social.

Ramezani et al. (2023) realizaron una revisión también, pero con mayor énfasis en la fase presupuestaria, donde buscan identificar cómo el ML mejora las tomas de decisiones financieras. Este estudio sienta precedente para entender cómo otros investigadores han buscado controlar gastos a través de una mejor predicción de los casos que los originan. Esto a través de modelos que usen variables clínicas y sociodemográficas.

Chand & Zhang (2022), exploraron modelos para mejorar la asignación de recursos, a través de la reducción de la brecha entre fondos asignados y gasto real en el National Disability Insurance Scheme de Australia. Descubrieron que los métodos no lineales, como árboles de decisión y redes neuronales, superaban a las técnicas lineales cuando existían múltiples variables con interacciones complejas. Los autores se inclinan por los árboles de decisión por su capacidad de ser interpretables, ya que, con ellos lograron identificar qué variables eran más relevantes para la predicción, como el tipo de discapacidad, la región geográfica y el rango de edad.

Saarela et al. (2022), plantearon un método combinado que integra procesamiento de lenguaje natural para historias clínicas y técnicas de ML aplicadas a datos estructurados. Su objetivo fue clasificar a los pacientes con regreso laboral temprano o tardío, obteniendo precisiones entre 69% y 78%. Para los datos estructurados compararon Random Forest, árboles de decisión, redes neuronales y XGBoost, y concluyeron que los ensambles suelen mejorar el desempeño sobre los métodos tradicionales, logrando entre 69 y 78 por ciento de exactitud para predicción de invalidez futura. Su aporte reside en la combinación de datos estructurados y no estructurados, sin embargo, para el presente trabajo solo se

cuenta con datos estructurados, de acuerdo con la naturaleza de los datos de interés para el Instituto.

Chou JCL et al. (2022), recurrieron a Naïve Bayes, k-NN y regresión logística para estimar la probabilidad de contratar un seguro de cuidados de largo plazo en Taiwán. Alcanzaron una precisión de alrededor del 83.33% con regresión logística, aun siendo un método más simple, muestra ventajas como su interpretabilidad. Este resultado obtenido con regresión fue en parte obtenido por la adecuada selección de atributos y discretización de datos. Su enfoque fue principalmente comercial, para poder identificar potenciales compradores de este tipo de seguros.

Na & Kim (2019), utilizaron un conjunto de datos más reducido y un enfoque directo en el “retorno vs. no retorno” al trabajo, mediante Gradient Boosting para predecir la reincorporación de trabajadores lesionados. Su modelo logró un AUC cercano a 0.94 en la clasificación binaria, lo cual confirma la alta efectividad de los ensambles de árboles de decisión siempre que el problema se mantenga como una dicotomía clara. Sin embargo, cuando se amplió a una clasificación con más categorías (por ejemplo, retorno parcial o de largo plazo), el rendimiento decayó. El estudio sugiere que variables psicológicas y de identidad laboral son altamente influyentes en la reincorporación, más allá de la enfermedad que condicionó la incapacidad.

Huhta-Koivisto (2020), realizó un estudio enfocado en el servicio de salud ocupacional, donde desarrollaron un modelo de lenguaje basado en ULMFiT para clasificar solicitudes según ameritaran o no una valoración. Se alcanzó una exactitud del 72% en una clasificación binaria (riesgo vs. no riesgo), aunque el problema se limitó al análisis de texto en dichas solicitudes, sin incluir factores temporales o clínicos más detallados. El estudio, al igual que otras, aporta evidencia de cómo el texto clínico, puede ser altamente informativo a través de técnicas de NLP.

Meyers et al. (2018), aplicaron ML a más de 1.2 millones de reclamaciones de compensación laboral del estado de Ohio (2001–2011) para establecer prioridades de prevención de lesiones en industrias específicas. Desarrollaron un algoritmo bayesiano que analizó textos no estructurados de informes y diagnósticos,

alcanzando un 90% de precisión global. Posteriormente, vincularon los resultados con datos de la entidad aseguradora y registros de desempleo para obtener el número de trabajadores equivalentes a tiempo completo por empresa. Posteriormente, utilizando un índice de priorización, identificaron las actividades económicas con mayor riesgo, y por ende, la necesidad de intervención preventiva. Las profesiones destacadas fueron enfermería, transportistas y fundidores. El estudio, aporta evidencia del uso de un auto-codificador bayesiano para clasificar grandes volúmenes de datos no estructurados.

Koc et al. (2021), analizaron 47,938 accidentes laborales para predecir la probabilidad de discapacidad permanente. Mediante un algoritmo genético y XGBoost alcanzaron una precisión de 82.92% y un AUC de 0.812, identificando que los “días de trabajo perdidos” y el “tipo de lesión” son variables altamente determinantes. Además, el análisis de importancia de variables señala que factores como la “exposición a temperaturas extremas” y la “exposición a sustancias químicas” influyen fuertemente en la probabilidad de discapacidad permanente. El estudio demuestra cómo, los algoritmos de boosting, logran adecuados niveles de precisión.

Kusnadi et al. (2023), buscaron predecir la tasa de recuperación por maternidad con XGBoost, Gradient Boosting y Bayesian Additive Regression Trees. Entre las variables que identificaron como relevantes, están la duración de la incapacidad, el número de exposiciones y la edad. Su contribución está en mostrar la robustez de XGBoost ante desequilibrios de clases.

Cheng et al. (2020), a diferencia, implementaron redes neuronales Long Short-Term Memory sobre 50,000 expedientes con información estructurada (datos demográficos, registros de intervenciones, etc.) y no estructurada (notas médicas) con el propósito de desarrollar el sistema SWIM (Smart Work Injury Management). Este es un sistema basado en inteligencia artificial para gestionar y predecir la evolución de lesiones laborales y el retorno al trabajo. La red neuronal logró procesar secuencias temporales y capturar detalles narrativos de la evolución clínica, mejorando la comprensión de la recuperación del trabajador lesionado. Al

igual que otros trabajos, su abordaje consistió en explotar datos estructurados, como no estructurados.

Choi et al. (2023), utilizaron 42,219 casos del sistema de compensación laboral de Corea del Sur, para predecir la probabilidad de rechazo a la solicitud de indemnización por enfermedad o lesión ocupacional. Evaluaron cuatro algoritmos (árboles de decisión, DNN, XGBoost y LightGBM), de los cuales la XGBoost mostró el mejor desempeño. El estudio identifica las variables con mayor valor predictivo, y también enfatiza que la enfermedad particular y el ambiente laboral de la persona, son intermedios en un modelo teórico causal de por qué se niegan las solicitudes. Este es un estudio más que evidencia la capacidad predictiva del modelo XGBoost.

Brahimi et al. (2022), buscaron generar un sistema para identificar licencias médicas injustificadas, aplicando modelos como Naive Bayes y regresión logística. Aunque menos potentes que el boosting, funcionaron adecuadamente para clasificar licencias falsas, dada la simplicidad de la señal investigada (relación entre variables laborales y frecuencia de solicitudes). El modelo de Naive Bayes fue el que demostró mejor desempeño.

Estos trabajos proveen de información útil para comprender cómo se ha abordado la gestión de incapacidades, los modelos y resultados obtenidos. Se ha observado que los métodos de boosting (XGBoost, LightGBM y variantes) suelen ofrecer el mejor desempeño en escenarios de datos tabulares con muchas variables, por su capacidad de capturar interacciones no lineales y manejar la heterogeneidad de la información. Las redes neuronales, por su parte, han demostrado su utilidad cuando es preciso integrar datos complejos o no estructurados, aunque requieren un volumen mayor de datos y presentan el problema de la interpretabilidad.

La revisión de la literatura describe el creciente uso de técnicas de aprendizaje automático para sistemas de aseguramiento. Estos trabajos concuerdan con la intención de aplicar modelos de aprendizaje automático para mejorar la gestión del recurso de subsidio para incapacidades. Resaltan, la necesidad de entender la complejidad clínica que rodea a cada caso particular, además de las transiciones por las que estos pacientes navegan para poder lograr

retornar a su trabajo. Es por esto, que, el presente estudio busca explorar el desempeño de modelos que consideren características clínicas de los pacientes, su evolución, y a diferencia de estudios previos, que también sean entrenados considerando la infraestructura de las unidades médicas de una región, como variable que refleja la capacidad del sistema para resolver la demanda.

### **3.2 Marco metodológico**

Poder lograr la predicción de la distribución de probabilidades de las incapacidades temporales en el IMSS requiere adoptar una metodología que integre las necesidades de la organización, así como, el análisis de datos y aprendizaje automático. Lograr el objetivo de este modelo, permitirá establecer pronósticos clínicos más entendibles para pacientes y médicos, pues no se pretende predecir un valor puntual, si no una distribución de probabilidades. Esta predicción estará basada en las variables individuales de cada caso, además de considerar las variables de infraestructura hospitalaria a los que un paciente dado tiene acceso en las diferentes regiones del país.

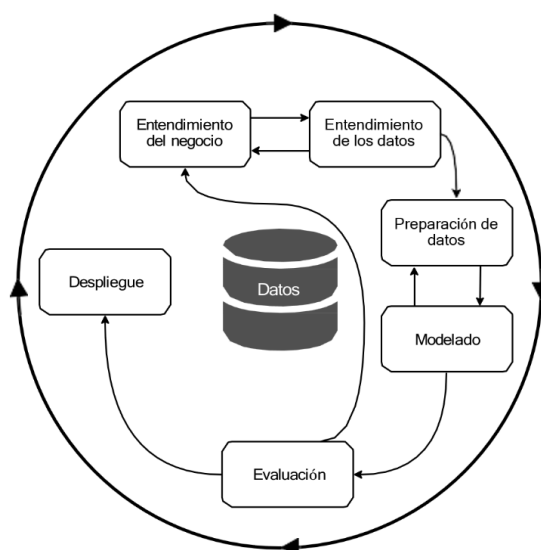
Para lograr esto, se establecerá un claro orden, lo cual no solo se hace por disciplina, si no, para que pueda esto ser replicado. Posterior a analizar dos opciones metodológicas, CRISP DM (Cross-Industry Standard Process for Data Mining), y ASUM DM (Analytics Solutions Unified Method), se decidió elegir la primera, toda vez que esta es aceptada y reconocida. Además, ASUM DM esta mayormente diseñada para ser implementada con herramientas de IBM (Shearer, 2000; Wirth & Hipp, 2000).

Para este trabajo, se propone el desarrollo de tres modelos de aprendizaje automático: 1) Regresión Ordinal, que aprovecha el orden implícito en las categorías de duración y puede reducir la confusión entre clases contiguas; 2) XGBoost, un algoritmo de boosting que incorpora regularización y procesamiento paralelo para mejorar la precisión y la escalabilidad; y 3) CatBoost, que maneja de forma nativa variables categóricas y ofrece un fuerte control del sobreajuste. Los modelos utilizarán un conjunto de características del paciente (edad, salario, diagnóstico

clínico, etc.) y de la infraestructura hospitalaria (camas censables, quirófanos, tipo de unidad, etc.), para poder estimar la variable objetivo.

La variable objetivo se define como la duración total de las incapacidades, estratificada en cuatro rangos de semanas ( $\leq 26$ , 26–52, 52–78 y  $>78$ ), con el fin de generar no solo una predicción puntual, sino una distribución de probabilidades asociada a cada caso.

Figura 3.1: Proceso CRISP DM



Fuente: Elaboración propia, 2025

### 3.2.1 Entendimiento del Negocio

La primera fase de CRISP-DM se centra en comprender el objetivo de la organización y definir la meta que se considerará como valiosa. En este caso, el IMSS presenta retos que pueden comprometer la sustentabilidad financiera de las incapacidades laborales. Por esto, se busca desarrollar un modelo que mejore la gestión de las incapacidades, con potencial impacto en:

1. Planificación de recursos: Conocer de antemano la probabilidad de que un paciente requiera, por ejemplo, más de 26 semanas de incapacidad, ayuda a prever la carga de trabajo en áreas clínicas, la necesidad de tener recurso humano disponible y de eficientar el uso de áreas médicas donde los pacientes reciben tratamiento (por ejemplo, quirófanos).



2. Atención y seguimiento del paciente: Un paciente con alta probabilidad de requerir >52 semanas de incapacidad, lo que puede resultar en invalidez o pensión, requiere un seguimiento más cercano por parte de médicos especialistas o rehabilitadores, para prevenir complicaciones de manera temprana, y mejorar su flujo a través de los servicios.
3. Gestión presupuestaria: El IMSS maneja un presupuesto que depende en buena parte de los días de incapacidad otorgados. Un modelo predictivo que distinga los casos breves de aquellos muy prolongados aporta información financiera valiosa para mejorar las metas presupuestarias y el control administrativo.

El objetivo institucional se define como desarrollar un modelo de clasificación con mejor precisión que el consenso de expertos. El cumplimiento de este objetivo permite reducir desperdicios mejorar la sostenibilidad de este tipo de subsidios.

Entre los riesgos existentes al despliegue final de este modelo, como apoyo en las operaciones diarias del Instituto, se encuentran el poder integrar un modelo de creación propia al ecosistema actual, donde el Instituto ha tenido la estrategia de contratar soluciones o servicios propietarios, y no desarrollar sus propios modelos.

### **3.2.2 Entendimiento de los Datos**

En esta fase se buscará entender el tipo y formato de los datos disponibles. Las fuentes de datos son dos, la primera es una base de datos de la Dirección de Prestaciones Económicas y Sociales, a libre disposición a través de la intranet institucional. En esta se pueden obtener todas las incapacidades mayores a 100 días de todo el país. Esta base se actualiza de manera diaria, eliminando aquellos casos que fueron dados de alta y agregando aquellos que acaban de rebasar los 100 días de incapacidad. Esta base se alimenta de los registros que los médicos realizan en el expediente clínico electrónico. Es por ello que una de las limitaciones de esta base es el error humano. Pues si un médico ingresó por error un diagnóstico diferente al real, generaría un error en la base de datos. Se cree que este tipo de

errores registros es demasiado aleatorio para generar un error considerable en los estimados.

La segunda base de datos es un registro de la Dirección de Prestaciones Médicas, donde cada mes actualizan el total de la infraestructura hospitalaria en cada una de sus unidades médicas. Al igual que la primera base, se encuentra a libre disposición a través de la intranet institucional. La limitación con esta base de datos es el no registrar cada uno de los recursos hospitalarios con adecuada precisión. Pues muchos hospitales que se ven rebasados en la demanda, incrementan de manera improvisada su capacidad para recibir pacientes. Y siendo estas, acciones informales que buscan incrementar la capacidad de las unidades, no son reconocidas de manera oficial, hasta que exista una forma de regularizarlas. Este fenómeno genera que, en el registro oficial de una unidad, existan  $x$  cantidad de recursos, cuando en la práctica diaria son  $x + y$ , siendo  $y$  el número que de manera informal y empírica se consideró necesario a incrementar. Las variables relevantes a extraer son:

- Variables demográficas y del paciente: Edad, salario, tipo de ramo (riesgo de trabajo, enfermedad general, maternidad), diagnóstico en CIE10, unidad de adscripción, delegación de trámite.
- Variable objetivo: El total de semanas de incapacidad, que posteriormente se convierte en cuatro categorías mediante una función que agrupa los rangos de semanas ( $\leq 26$ ,  $26-52$ ,  $52-78$ ,  $>78$ ). Se descartó la predicción de un valor continuo (regresión directa) porque el enfoque ordinal (cuatro rangos) es más interpretable y se alinea mejor con la toma de decisiones médicas y administrativas, dado que la transición de un trabajador a más de 52 o 78 semanas conlleva el paso a invalidez o pensión.
- Variables de infraestructura hospitalaria: Delegación, número de camas censables, número de quirófanos, cantidad de consultorios y presencia de servicios específicos (por ejemplo, Salud en el Trabajo).

Entre las variables clínicas-administrativas, destaca la variable “Secuencia”, una variable con formato de texto que describe la progresión de un paciente a través

de las diferentes unidades médicas (por ejemplo, pasar de una unidad médica de primer nivel a una de tercer nivel). De esta columna se extraerán variables derivadas: cuántas veces aparece cada nivel de atención (count\_1, count\_2, count\_3), la longitud de la secuencia (seq\_length), el primero y último nivel de atención (first, last) y cuántos cambios hay entre niveles (num\_transitions). Estas variables permiten explorar cómo un paciente dado navegó a través del sistema, lo que informa sobre los probables recorridos que un paciente puede tener, según su diagnóstico y la región en que se encuentra.

En el análisis exploratorio se buscará identificar valores atípicos, además de revisar la distribución de la variable objetivo, examinar correlaciones y verificar la calidad de los datos. Se identifica la alta cardinalidad de variables categóricas, como son los diagnósticos CIE10 y las unidades de adscripción, para darles un preprocesamiento adecuado.

### **3.2.3 Preparación de Datos**

Una vez comprendidos los datos en posesión para el modelaje se pasa a la preparación de los mismos, a través de técnicas de limpieza, la transformación y la selección de atributos:

#### **1. Limpieza:**

- Se elimina los registros que carecen de etiqueta o valor en la variable objetivo (Semanas), dado que estos serán casos no informativos.
- Se procede a realizar codificación para las variables categóricas de baja cardinalidad con One-Hot Encoding, a fin de poder generar una representación binaria sin generar un exceso de columnas. Por otra parte, para las variables categóricas de alta cardinalidad se empleará Frequency Encoding para evitar la explosión de dimensionalidad y minimizar el riesgo de sobreajuste.

#### **2. Generación de características:**

- La columna “Secuencia” describe la ruta de atención médica que sigue el paciente a través de distintos niveles asistenciales. A partir de esta, se generarán variables nuevas para entrenar el modelo, a fin de poder

explotar esta variable que refleja el flujo y fricción de una persona a través del sistema. Se obtendrán: total de transiciones (seq\_length), conteo de cada nivel (count\_1, count\_2, count\_3) y número de transiciones (veces\_nivel3, transicion\_12, transicion\_13, transicion\_23, num\_transitions). Se incluyen variables como first y last para capturar la complejidad de la ruta del paciente.

- Variables de infraestructura hospitalaria: camas censables, quirófanos y consultorios por cada 100,000 habitantes, diferenciado por especialidad médica. Todas ellas seleccionadas dado que, la disponibilidad de estos recursos puede influir en la duración de las incapacidades, ya que regiones con mayor infraestructura podrían ser más oportunos en diagnósticos y tratamientos.

### 3. Selección de atributos:

- Se conservan aquellas columnas que mejor explican la duración de la incapacidad (diagnóstico, edad, salario, infraestructura, etc.), de acuerdo a su relevancia estadística y pertinencia clínica/administrativa.
- Se descartan variables redundantes o con poca variabilidad. Por ejemplo, entre aquellas fuertemente correlacionadas, se elegirá la que tenga mayor interpretabilidad.

### 4. Partición de datos:

- Se emplea `train_test_split` (80% entrenamiento, 20% validación), con estratificación en la variable objetivo para asegurar representación balanceada de cada categoría.
- En la etapa de validación, se empleará `StratifiedKFold` para mejorar la estimación del desempeño y evitar sesgos, asegurando que cada “fold” tenga la misma proporción de clases.

Esto resultará en un conjunto de datos adecuado para ser ingresado a los diferentes modelos.

### 3.2.4 Modelado

En la cuarta fase de CRISP-DM, se construyen y evalúan los modelos. Dado que las cuatro clases a predecir, de duración  $\leq 26$ , 26–52, 52–78,  $>78$  en semanas tienen un orden natural, se evaluarán dos enfoques principales:

#### 1. Clasificación Multiclase

- CatBoost se seleccionó por su capacidad de manejo nativo de variables categóricas. Además, la regularización L2 y las estrategias de permutación reducen el riesgo de sobreajuste (Prokhorenkova et al., 2018).
- XGBoost, se utilizará por su capacidad de ensamble secuencial, con control de complejidad mediante parámetros como `subsample`, `colsample_bytree` y `max_depth` (Chen & Guestrin, 2016).

#### 2. Regresión Ordinal

- Con este enfoque, se buscaría medir si respetar la naturaleza ordinal mejora la clasificación, buscando reducir la confusión entre categorías contiguas (p. ej., 26-52 vs. 52-78 semanas). Este modelo puede ser mejor en su interpretación, además de penalizar en mayor medida errores que impliquen “saltos” a categorías muy distantes.

#### 3.2.4.1 Relación con los objetivos de negocio

Las decisiones del modelado, de acuerdo a CRISP DM, nunca deben desconectarse de las necesidades organizaciones. El Instituto debe ser efectivo en tratar y rehabilitar a los trabajadores para su retorno laboral. El que este modelo genere un pronóstico ayuda a los médicos a entender cómo casos similares a un paciente dado se han comportado, por lo que sería un apoyo en su toma de decisiones.

#### 3.2.4.2 Documentación de supuestos

- Se asume que los rangos definidos ( $\leq 26$ , 26–52, 52–78,  $>78$ ) tienen un orden inherente y que el costo de confundir categorías cercanas es

menor que el de confundir categorías distantes. Este supuesto se utiliza para incluir la Regresión Ordinal.

- Se asume que las categorías definidas son informativas para la operación del Instituto, dado que estos límites están definidos en su normatividad, como umbrales importantes para decidir el futuro laboral de una persona. No se asume que estos umbrales vayan a cambiar en el futuro.

#### 3.2.4.3 Búsqueda de Hiperparámetros

Para cada modelo, se realizará una búsqueda aleatoria mediante `RandomizedSearchCV` sobre los principales hiperparámetros, además de incluir validación cruzada `StratifiedKFold` (Sokolova & Lapalme, 2009). En CatBoost se ajustan parámetros como `iterations`, `learning_rate`, `max_depth`, `subsample` y `l2_leaf_reg`. Para XGBoost, además de `learning_rate` y `max_depth`, se explora `colsample_bytree`, `reg_alpha`, `reg_lambda`, etc. (Chen & Guestrin, 2016).

Finalmente, se entrena un modelo definitivo para cada tipo (CatBoost, XGBoost y Regresión Ordinal), usando el mejor conjunto de hiperparámetros obtenidos.

#### 3.2.5 Evaluación

En esta quinta fase, se medirá y analizará el desempeño de los modelos:

1. Exactitud (accuracy): Proporción de aciertos totales, aunque puede ser insuficiente en casos de desequilibrio de clases o distribución ordinal.
2. Matriz de confusión: Permitirá ver cuántos casos de cada categoría real ( $\leq 26$ ,  $26-52$ ,  $52-78$ ,  $>78$ ) se pronostican correctamente o se confunden. Para este particular trabajo, se busca evaluar si el modelo confunde categorías adyacentes o salta directamente entre  $\leq 26$  y  $>78$ .
3. Reporte de clasificación: Además de exactitud, se reportará precisión, exhaustividad (recall) y F1-score por clase. Esto se incluye para observar el comportamiento del modelo, sobre todo ante el desbalance de clases.

4. Se incluirán métricas específicas para ordinalidad, como Weighted Kappa o MAE ordinal, que penalizan más los errores entre categorías distantes (Agresti, 2010).

Si el rendimiento de algún modelo es insatisfactorio, se regresa a las fases previas para ajustar variables o hiperparámetros, a fin de elegir el modelo con mejor desempeño.

### **3.2.6 Despliegue**

La última fase aborda la implementación y uso real del modelo. En este proyecto, se contempla lo siguiente:

1. Piloto de 60 días y actualización de datos:
  - Una vez desarrollado y validado el modelo en un entorno controlado, se buscará proponer un pilotaje durante los siguientes 60 días naturales. Periodo durante el cual, el modelo utilizará casos nuevos no antes vistos, para continuar incrementando su desempeño de manera incremental. El modelo será entrenado de manera diaria durante los días hábiles. Si el desempeño del modelo muestra una tendencia a la mejora se considerará como un pilotaje exitoso. Se desarrollará un instructivo breve sobre el uso de la interfaz, para que el personal médico sepa cómo usarlo. El mismo se encontrará como un PDF descargable en el aplicativo.
  - Posterior al pilotaje, se buscará integrarlo como herramienta de apoyo en la toma de decisiones médicas y financieras (Gewurtz et al., 2019), complementando las guías de duración de la incapacidad y la supervisión de los comités del IMSS. Se desarrollará una interfaz con Google Cloud Platform. Aquí los usuarios podrán ingresar los casos y obtener la predicción de probabilidad.
2. Análisis de riesgos
  - Dada la estrategia institucionales de utilizar sistemas propietarios, se utilizará Google Cloud Platform para desarrollar la interfaz. Esto requerirá que el personal que experimente con el modelo ingrese los

datos de manera manual. Por lo que estos pasos extra en la rutina diaria de los médicos compromete su adopción.

Con el uso del marco CRISP-DM y los modelos elegidos, se busca responder a la necesidad de predecir la distribución de probabilidades de la duración de las incapacidades con mayor exactitud.

### **3.3 Ajuste de los modelos**

En la siguiente sección se procede a describir el proceso de modelado y evaluación de desempeño de los algoritmos seleccionados. El objetivo es comparar el desempeño de cada modelo para estimar la duración de las incapacidades.

A partir de la revisión de literatura y de las necesidades del problema, se decidió experimentar con tres modelos:

1. El primero es Regresión Ordinal, porque permite aprovechar la naturaleza ordenada de las categorías de duración (rangos de semanas). Estudios como McCullagh (1980) y Agresti (2012) indican que este tipo de regresión minimiza la confusión entre clases contiguas y ofrece interpretaciones alineadas a umbrales relevantes para la gestión de incapacidades dentro del IMSS.
2. Un modelo ampliamente usado y de buen desempeño es XGBoost. Este modelo de gradient boosting ha demostrado su valor en contextos de datos estructurados, alta dimensionalidad y presencia de variables categóricas. Según Chen & Guestrin (2016), XGBoost sobresale por su rapidez y por incorporar regularización explícita para evitar el sobreajuste.
3. Por último, se consideró Catboost, que al igual que XGBoost, es un modelo de gradient boosting pero integra de forma nativa el manejo de variables categóricas y reduce el sesgo asociado a estas, aspecto destacado por Prokhorenkova et al. (2018). Dado que una parte importante de las variables (diagnóstico, unidad de adscripción, secuencia de atención) son categóricas, CatBoost puede resultar ventajoso.



La elección de estos modelos es de acuerdo al tipo de variable objetivo, y la presencia del tipo de variables contenidas en la base de datos. Además, se tienen en cuenta consideraciones institucionales, como la necesidad de una herramienta con tiempos de respuesta razonables para el personal médico, y la posibilidad de integrar el modelo a los sistemas actuales del IMSS.

### 3.3.1 Generación del diseño de prueba

Posterior a la selección de modelos a emplear para la clasificación de la duración de las incapacidades, se define la forma de medir su desempeño y comprobar su capacidad de generalización. Es por ello que se describe el diseño de prueba, consistente en estrategia de validación y muestreo, y búsqueda y ajuste de hiperparámetros.

Primero se divide el conjunto de datos en subconjuntos de entrenamiento y de validación, con 80 y 20 por ciento respectivamente. Además, se estratifican las muestras basadas en la variable objetivo, dado que las categorías ( $\leq 26$ , 26–52, 52–78,  $>78$  semanas) se distribuyen de forma desigual. La estratificación mantiene la proporción de cada clase en cada pliegue de la validación, reduciendo la probabilidad de sesgos durante el entrenamiento.

Esto se implementará mediante:

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=test_size, random_state=42, stratify=y)
```

donde el parámetro `stratify=y` garantiza que la distribución de las clases en `y` sea consistente en ambos subconjuntos.

Posteriormente se implementa validación cruzada, `StratifiedKFold(n_splits=3)` para la búsqueda de hiperparámetros, para aprovechar el conjunto de datos y obtener estimaciones más robustas del desempeño. Así, cada uno de los  $k$  pliegues actúa sucesivamente como conjunto de validación, mientras los pliegues restantes entrenan el modelo. Una vez completados los  $k$  entrenamientos, se calculan las métricas de evaluación.

Si bien, se pretende la creación de un conjunto de entrenamiento y otro de validación, se incluirá también un conjunto de prueba, de casos no vistos ni durante el entrenamiento, ni durante la validación. Estos casos, que suman un total de 15,636, son representativos de la población de estudio.

La siguiente etapa consiste en buscar y ajustar los parámetros de cada modelo algoritmos. De manera general, se busca optimizar la profundidad máxima de los árboles (`max_depth`), la tasa de aprendizaje (`learning_rate`), diversos parámetros de regularización (`l2_leaf_reg`, `subsample`, `colsample_bytree`) y el método de muestreo (como `bootstrap_type` en CatBoost).

Para Catboost, dado que maneja las variables categóricas nativamente mediante el parámetro `cat_features`, evita que se realicen transformaciones como *label encoding* o *one-hot*. Los hiperparámetros que sí serán ajustados son:

- `iterations` (número de árboles)
- `learning_rate` (tasa de aprendizaje)
- `max_depth`
- `l2_leaf_reg` (penalización de regularización)
- `subsample` y `bootstrap_type` (método de muestreo)

En cuanto a XGBoost, no cuenta con soporte nativo para variables categóricas, por lo que se realiza codificación. Se ajustarán:

- `n_estimators` (número de árboles)
- `learning_rate`
- `max_depth`
- `subsample`, `colsample_bytree` (porcentajes de muestreo para filas y columnas)

Regresión Ordinal se utilizará con base en *mord*. Para este modelo, también se codificarán las variables categóricas. En cuanto al ajuste de hiperparámetro, se aplicará a `alpha`, que controla la regularización en la regresión logística ordinal.

Para la implementación de este proceso, se emplea la clase `RandomizedSearchCV`, para poder explorar aleatoriamente configuraciones dentro de un espacio de búsqueda definido. Cada configuración es entrenada y

validada utilizando k-fold cross-validation con estratificación, y la mejor configuración se selecciona para reentrenar el modelo final.

### **3.3.2 Creación del modelo**

La creación de los diferentes modelos abarca la implementación práctica de los pasos descritos en la fase de diseño de prueba y la utilización de librerías y clases de Python específicas. Este proceso se describe en las diferentes figuras que ilustran a cada uno de los modelos.

En todos los casos, se inicia con la ingesta de archivos Excel, para continuar con el preprocesamiento, y la división en subconjuntos de entrenamiento y validación. Se continua con la búsqueda de hiperparámetros y finaliza con el entrenamiento final y la evaluación de los modelos.

#### **3.3.2.1 Herramientas de Python y clases utilizadas**

Para llevar a cabo la creación y entrenamiento de los modelos, se emplean las siguientes librerías y clases principales:

- Pandas (pd) y NumPy (np) para el manejo de estructuras de datos (DataFrame, Arrays), y para las operaciones de limpieza y transformación.
- scikit-learn, por sus funciones de división de datos: train\_test\_split, StratifiedKFold, búsqueda de hiperparámetros: RandomizedSearchCV, métricas de evaluación: accuracy\_score, classification\_report, confusion\_matrix, y transformaciones de variables categóricas.
- CatBoost, por su algoritmo principal: CatBoostClassifier. También se encuentran parámetros relevantes como son: iterations, learning\_rate, max\_depth, l2\_leaf\_reg, cat\_features, etc. Se encuentran los métodos de entrenamiento: fit, predict, predict\_proba. Y el manejo nativo de variables categóricas mediante cat\_features.
- XGBoost, con su algoritmo XGBClassifier. Se utilizan parámetros relevantes: n\_estimators, learning\_rate, max\_depth, subsample,

`colsample_bytree`. Además, se incluye una función de objetivo multiclase: `objective="multi:softprob"`, con `num_class=4` para manejar las cuatro categorías de duración.

- Por último, se utiliza `mord` para el modelo de Regresión Ordinal. Se utiliza la clase `LogisticIT`. Con esta se modelan las variables de respuesta ordinal, como las cuatro categorías ( $\leq 26$ , 26–52, 52–78,  $>78$ ). Se realiza también ajuste del hiperparámetro `alpha`, que controla la regularización.

### 3.3.2.2 Implementación de los modelos

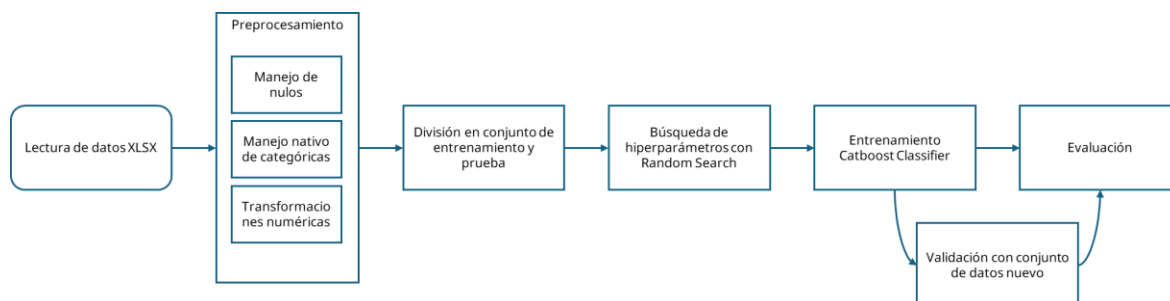
El proceso de implementación comparte una estructura general en los tres casos, que inicia con la lectura y preprocesamiento de datos, sigue con la división del conjunto en entrenamiento y validación, para después continuar con la búsqueda aleatoria de hiperparámetros a través de `RandomizedSearchCV`, combinada con validación cruzada. Una vez finalizada la búsqueda, se reentrena el modelo con la mejor configuración y se evalúa su rendimiento en el conjunto de validación adicional.

Para `CatBoost`, se inicia leyendo la base de datos en formato Excel, corrigiendo valores ausentes y generando variables derivadas. `CatBoost` simplifica el preprocesamiento de las variables categóricas, ya que solamente se necesita declarar la lista de atributos (`cat_features`) y el modelo maneja internamente la estadística y codificación adecuada para cada una.

Con `train_test_split` (80%-20%) y `stratify=y`, se equilibran las categorías de la variable objetivo en cada subconjunto. Para la búsqueda de hiperparámetros se experimenta con configuraciones para `iterations`, `learning_rate`, `max_depth` y `l2_leaf_reg`, además de `bootstrap_type` o `subsample` para el muestreo de datos en la construcción de cada árbol. Posteriormente, se inicia el entrenamiento final con la llamada `fit`, que admite un conjunto de evaluación (`eval_set`) y permite activar `use_best_model=True`, de modo que el proceso pueda detenerse si se detecta sobreajuste.

Finalmente, se calculan métricas para comparar el rendimiento en el conjunto de entrenamiento, en el de validación y en el conjunto de prueba antes del despliegue.

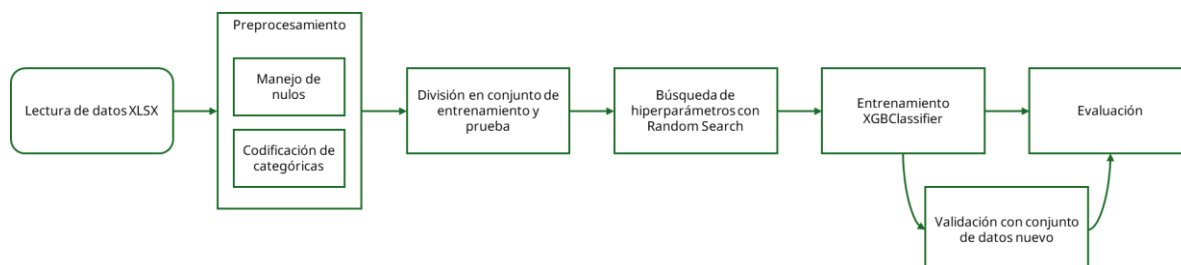
Figura 3.2: Implementación CatBoost



Fuente: Elaboración propia, 2025

La implementación de XGBoost requiere que las variables categóricas se codifiquen mediante LabelEncoder o One-Hot. Una vez conformados los conjuntos de entrenamiento y validación, se ajusta un espacio de búsqueda que incluye parámetros como `n_estimators`, `learning_rate`, `max_depth`, `subsample` y `colsample_bytree`. A través de la validación cruzada se localiza el mejor conjunto de valores, y posteriormente se entrena `XGBClassifier` con `objective="multi:softprob"` y `num_class=4`. Cuando se encuentra la mejor combinación, se construye el modelo definitivo `XGBClassifier(**best_params)`. Por último, se comparan las métricas en entrenamiento, validación y prueba final.

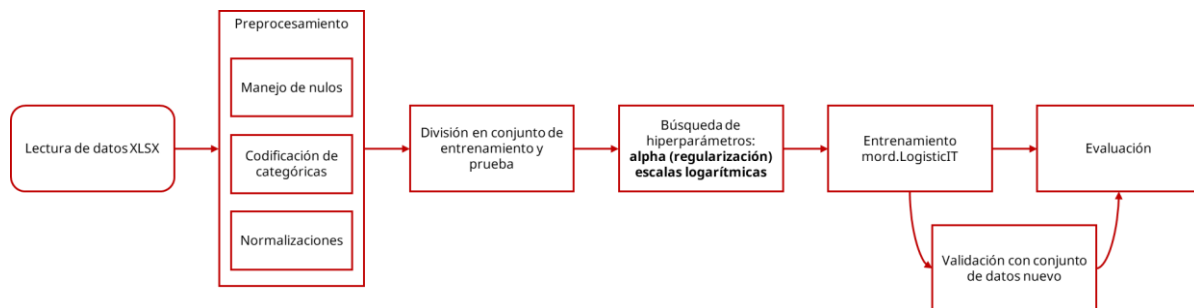
Figura 3.3: Implementación XGBoost



Fuente: Elaboración propia, 2025

La Regresión Ordinal modela la variable objetivo conforme a su estructura ordinal. Se emplea la clase `LogisticIT` de la librería `mord`. Dado que este algoritmo no procesa directamente variables categóricas, se tienen que codificar. Tras completar el preprocesamiento, se define la columna objetivo con valores 0, 1, 2 y 3, en función de los rangos de semanas de incapacidad. Para el ajuste de hiperparámetros, se recurre a un rango amplio de `alpha` dispuesto en escala logarítmica; la configuración que mejor promueve la exactitud se selecciona y se entrena el modelo definitivo, evaluándolo en los subconjuntos correspondientes.

Figura 3.4: Implementación Regresión Ordinal



Fuente: Elaboración propia, 2025

En todos los modelos se llevan a cabo comparaciones de métricas para verificar el desempeño en cada categoría. Así, se determina cuál de los algoritmos ofrece una mejor aproximación a la duración de las incapacidades y, además, se constata qué tan robusta resulta su clasificación al enfrentarse con datos no vistos durante la fase de entrenamiento.

Por último, se implementaron funciones para estimar la relevancia de cada variable en los modelos. En el caso del modelo XGBoost, se desarrolló una función que utiliza el atributo `feature_importances` para obtener un valor numérico representativo del impacto de cada característica. La función recibe como parámetros el modelo entrenado, la lista de variables utilizadas y un argumento que permite generar una gráfica visual de barras. Este atributo, de manera interna,

funciona construyendo un DataFrame con las columnas Feature e Importance, ordenado de forma descendente para resaltar las variables más influyentes.

Para el modelo CatBoost, se implementó una función similar, pero que acepta un parámetro adicional para especificar la métrica de importancia (por defecto, `PredictionValuesChange`). Se utilizó el método `get_feature_importance` de CatBoost.

Por último, para Regresión Ordinal, se diseñó la función `compute_feature_importance`. Esta extrae los coeficientes del modelo (almacenados en `model.coef_`), calcula su valor absoluto y genera un DataFrame con las columnas `feature`, `coefficient` y `abs_coefficient`.

### 3.3.3 Evaluación de modelos

Para poder analizar y comparar el desempeño de los tres modelos en la tarea de predecir y generar una distribución de probabilidades de la duración de incapacidades ( $\leq 26$ ,  $26-52$ ,  $52-78$ ,  $>78$ ), se calcula un reporte de clasificación que incluye las principales métricas de evaluación, como son exactitud (accuracy), precisión (precision), exhaustividad (recall), y F1-score. Estas métricas se reportan por cada categoría de la variable objetivo. Se computan también promedios macro y ponderados. Por último, se presenta la matriz de confusión, a través de la cual se puede visualizar cuántos casos de cada categoría son clasificados correcta o erróneamente.

#### *Exactitud (Accuracy)*

La exactitud mide la fracción de observaciones clasificadas correctamente entre el total de ejemplos. Se calcula mediante:

$$\text{Exactitud (Accuracy)} = \frac{\sum_{i=1}^k TP_i}{N}$$

De manera que, en un problema de  $k$  clases, donde  $TP_i$  es la cantidad de verdaderos positivos de la clase  $i$  y  $N$  es el número total de instancias. En

clasificación binaria, suele verse como  $(TP + TN)/(TP + TN + FP + FN)$ , pero en multiclase se expresa de manera que la suma de los aciertos totales dividido por todas las observaciones.

Aunque es una métrica interpretable, puede ser sesgada ante la falta de balance entre las clases.

### *Precisión (precision) y Exhaustividad (Recall)*

La precisión y la exhaustividad miden, para cada clase, la proporción de predicciones correctas y la proporción de casos recuperados, respectivamente.

Precisión de la clase  $i$  (en un entorno multiclase se calcula por clase y luego se promedian):

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

donde  $TP_i$  son los verdaderos positivos de la clase  $i$  y  $FP_i$  son los falsos positivos, o sea, aquellos casos que se predijeron como clase  $i$  cuando pertenecían a otra.

Para calcular la exhaustividad de la clase  $i$ :

$$\text{Exhaustividad (Recall)}_i = \frac{TP_i}{TP_i + FN_i}$$

donde  $FN_i$  son los falsos negativos, los casos de la clase  $i$  que el modelo no detectó, clasificándolas como otra clase. Un modelo puede llegar a alta precisión a costa de una disminución en la exhaustividad (si es muy selectivo al predecir positivos), mientras que uno con alta exhaustividad puede clasificar demasiados ejemplos como positivos, reduciendo la precisión.

### *F1-score*

El F1-score combina precisión y exhaustividad en un solo valor, al calcular el promedio armónico. De manera que, para la clase  $i$ :



$$F1_i = 2 \times \frac{(\text{Precision}_i \times \text{Recall}_i)}{\text{Precision}_i + \text{Recall}_i}$$

En un problema multiclase, se calcula un  $F1_i$  para cada clase y luego se define un promedio, que puede ser macro o ponderado, dependiendo de cómo se priorice el balance entre las diferentes categorías.

Esta métrica resulta útil cuando hay clases menos frecuentes que, a pesar de ser minoritarias, son importantes. En este trabajo, por ejemplo, la clase de >78 semanas de duración.

#### *Promedios macro y ponderados*

El promedio Macro busca calcular la métrica en cada clase, para después hacer un promedio simple. Este enfoque otorga igual peso a cada categoría, sin importar cuántos casos contenga.

Por otro lado, el promedio ponderado calcula la métrica otorgándole peso a cada categoría, según la cantidad de ejemplos en cada clase. Con este ajuste, se busca que refleje mejor el comportamiento global del modelo cuando hay desequilibrio entre las clases, aunque puede ocultar desempeños pobres en clases minoritarias.

#### *Matriz de confusión*

Para cada modelo, se construye una matriz, donde las columnas presentan las cuatro categorías verdaderas ( $\leq 26$ , 26–52, 52–78, >78), mientras que las filas presentan las categorías predichas por el modelo. Por lo que, cada celda muestra cuántos casos de una clase verdadera son clasificadas en cada clase predicha. Esta representación presenta de manera resumida, las confusiones específicas que cada modelo puede tener, como predecir la categoría 52–78 cuando en realidad se trataba de 26–52 semanas.

Estas métricas informaran sobre el desempeño de cada modelo. Se busca que los modelos sean capaces de predecir adecuadamente las duraciones más

prolongadas, que a menudo son las de mayor relevancia para la operación del IMSS por el alto costo que conllevan.

### **3.4 Análisis de resultados**

Una vez entrenados los modelos y seleccionados sus hiperparámetros, se procedió a evaluar el desempeño en un conjunto de validación (20,481 observaciones). Posteriormente, se llevó a cabo una evaluación final en un conjunto de prueba de 15,636 ejemplos para estimar la capacidad de generalización de cada enfoque y descartar la posibilidad de sobreajuste.

#### **3.4.1 Desempeño global en el conjunto de validación**

##### *CatBoost*

Tras el entrenamiento, el modelo de CatBoost se detuvo en la iteración 176 debido al criterio de early stopping, con un bestTest (log-loss) de 0.5523. La exactitud total en este conjunto de validación fue de 0.9570.

En el reporte de clasificación, se observó un buen desempeño para la clase 0 (precisión y exhaustividad rozando 0.97–0.99). Sin embargo, para las clases minoritarias (1, 2 y 3), la estrategia de CatBoost mostró un recall alto —por ejemplo, 0.75 en la clase 3— a costa de una menor precisión. Esto implica que el modelo fue más “permisivo” con las clases menos frecuentes, clasificando algunos ejemplos de manera incorrecta pero aumentando la capacidad de no dejarlos pasar inadvertidos.

La matriz de confusión confirmó que la mayoría de los errores se concentraron en la confusión entre la clase 0 y la 1, así como cierta sobreclasificación de ejemplos en las categorías 2 y 3.

##### *XGBoost*

Para este modelo basado en gradient boosting, se realizó un ajuste de parámetros y se exploraron hasta 700 iteraciones, llegando a un mlogloss cercano

a 0.3465 en la iteración final (699). La exactitud en la validación (20,481 observaciones) fue de 0.9645, superando ligeramente a CatBoost.

El reporte de clasificación indicó un promedio ponderado de precisión y exhaustividad cercano a 0.96–0.97. La clase 0, al ser la de mayor cantidad de ejemplos, obtuvo una exhaustividad de 0.98, mientras que las clases 1 y 2 registraron métricas inferiores pero aun positivas. La clase 3 (con muy pocos casos) mantuvo un recall de 0.17 y una precisión cercana a 0.50, reflejando la dificultad usual en categorías de bajo soporte.

En la matriz de confusión, los mayores errores se observaron al clasificar ejemplos de la clase 1 como 0 y, en menor medida, de la clase 2 como 1. Aun así, la dispersión de errores fue menor comparada con la de CatBoost.

### *Regresión Ordinal*

Se entrenó un modelo LogisticIT para explotar la naturaleza ordinal de las cuatro clases. En el conjunto de entrenamiento, alcanzó un accuracy de alrededor de 0.8280. Sin embargo, en la validación cayó a 0.5873, indicando una marcada diferencia de desempeño.

La matriz de confusión mostró que una gran proporción de casos de clase 0 fueron clasificados correctamente, pero existió una alta confusión entre las demás categorías, especialmente entre 1, 2 y 3.

A pesar de la ventaja teórica de capturar la relación ordenada entre clases, la distribución de datos en validación difirió al punto de afectar sensiblemente este modelo. El F1-score en las clases menos frecuentes resultó bajo y reveló un sesgo hacia la clase mayoritaria.

Tabla 3.2: Desempeño en conjunto de validación

| Modelo                  | Accuracy | F1<br>(categoría<br>0) | F1<br>(categoría<br>1) | F1<br>(categoría<br>2) | F1<br>(categoría<br>3) | F1<br>Ponderado |
|-------------------------|----------|------------------------|------------------------|------------------------|------------------------|-----------------|
| <b>CatBoost</b>         | 0.957    | 0.98                   | 0.61                   | 0.51                   | 0.43                   | 0.96            |
| <b>XGBoost</b>          | 0.9645   | 0.98                   | 0.64                   | 0.68                   | 0.25                   | 0.97            |
| <b>Reg.<br/>Ordinal</b> | 0.5873   | 0.74                   | 0.09                   | 0.06                   | 0.11                   | 0.71            |

Fuente: Elaboración propia, 2025

### 3.4.2 Desempeño en el conjunto de prueba

Para confirmar la capacidad de generalización, se evaluaron los tres modelos en un conjunto final de prueba con 15,636 observaciones. Los resultados principales fueron:

- **XGBoost:**
  - Exactitud de 0.9396.
  - Clases mayoritarias (0 y 2) con recall altos ( $\geq 0.82$ ).
  - La clase 3 logró un recall de 0.39, aunque la precisión fue moderada.
- **CatBoost:**
  - Exactitud de 0.9357.
  - Notable recall (0.89) en la clase 3, pero la precisión en esta categoría descendió a 0.24, evidenciando sobreclasificación de casos.
  - El modelo se mantuvo competitivo pero ligeramente por debajo de XGBoost.
- **Regresión Ordinal:**
  - Exactitud de 0.9241.
  - Mostró una recuperación importante respecto al conjunto de validación (donde obtuvo 0.5873).
  - La clase 0 se clasificó casi de forma impecable (recall  $\sim 0.99$ ), mientras que las categorías 2 y 3 mantuvieron confusiones notorias, aunque el F1 de la clase 2 fue razonable (0.82).

Tabla 3.3: Desempeño en conjunto de prueba

| Modelo              | Accuracy | F1 (categoría 0) | F1 (categoría 1) | F1 (categoría 2) | F1 (categoría 3) | F1 Ponderado |
|---------------------|----------|------------------|------------------|------------------|------------------|--------------|
| <b>XGBoost</b>      | 0.9396   | 0.99             | 0.51             | 0.88             | 0.5              | 0.95         |
| <b>CatBoost</b>     | 0.9357   | 0.99             | 0.58             | 0.87             | 0.38             | 0.94         |
| <b>Reg. Ordinal</b> | 0.9241   | 0.99             | 0.47             | 0.82             | 0.38             | 0.93         |

Fuente: Elaboración propia, 2025

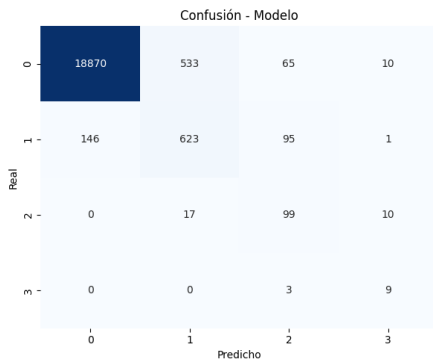
En conclusión, XGBoost y CatBoost lideraron el rendimiento tanto en validación como en prueba, mostrando estabilidad y alta precisión. La Regresión Ordinal manifestó un comportamiento más variable, dependiendo fuertemente de la distribución de los datos en cada conjunto, si bien finalmente obtuvo resultados aceptables en la prueba final.

3.4.3 Matrices de confusión

Las matrices de confusión se muestran para visualizar dónde se concentran los errores en la predicción de la categoría de duración, así como contrastar estos resultados con la clasificación manual.

En CatBoost se observa que la clase 0 ( $\leq 26$  semanas) es reconocida casi en su totalidad, con 18,870 aciertos sobre 19,478 casos reales en ese grupo (izquierda). Sin embargo, existen confusiones entre las clases intermedias, en especial la clase 1 (26–52 semanas), que a veces se etiqueta como 2; asimismo, una fracción de la clase 3 ( $>78$  semanas) se clasifica como 2. Por contraste, el humano tiene un número notable de errores al confundir la clase 0 con 1, 2 y 3 (3,764, 3,886 y 9,961 respectivamente), lo que se refleja en una columna predominante en la categoría 3 (predicha).

Figura 3.5: Matriz para CatBoost



Fuente: Elaboración propia, 2025

XGBoost, refina la clasificación de la clase 2 (52–78 semanas), identificando con mayor precisión a pacientes dentro de este rango. No obstante, tal como se vio en el análisis cuantitativo, se sigue confundiendo la clase 3 con 2, producto de la limitada frecuencia de ejemplos en la categoría más prolongada de incapacidad. A pesar de estas confusiones, XGBoost supera con holgura el desempeño humano (columna derecha), que mantiene, de nuevo, la mayor proporción de errores al etiquetar indebidamente una fracción significativa de la clase 0 como categorías 1, 2 o 3.

Figura 3.6: Matriz para XGBoost

Confusión - Modelo

|                 |       |     |    |   |
|-----------------|-------|-----|----|---|
| Real \ Predicho | 0     | 1   | 2  | 3 |
| 0               | 19021 | 451 | 6  | 0 |
| 1               | 188   | 639 | 38 | 0 |
| 2               | 0     | 32  | 92 | 2 |
| 3               | 0     | 0   | 10 | 2 |

Fuente: Elaboración propia, 2025

La Regresión Ordinal, si bien este método fue más inestable en validación, en el conjunto de prueba logra capturar un número razonable de casos en la clase 3 (70 aciertos), reduciendo los errores que se veían entre las clases 2 y 3. Sin embargo, conserva confusiones notables en las categorías intermedias (1 y 2), posiblemente porque las duraciones cercanas al límite (50–55 semanas) son difíciles de distinguir en un esquema ordinal restringido. Por su parte, el humano persiste con errores ampliamente distribuidos y un fuerte sesgo hacia la clase 3 predicha, con 6,138 casos de la clase 0 mal clasificados en este nivel.

Figura 3.7: Matriz para Regresión Ordinal

Matriz de Confusión - Modelo

|                 |       |     |      |     |
|-----------------|-------|-----|------|-----|
| Real \ Predicho | 0     | 1   | 2    | 3   |
| 0               | 11914 | 112 | 0    | 0   |
| 1               | 137   | 406 | 23   | 1   |
| 2               | 47    | 635 | 2059 | 170 |
| 3               | 0     | 3   | 59   | 70  |

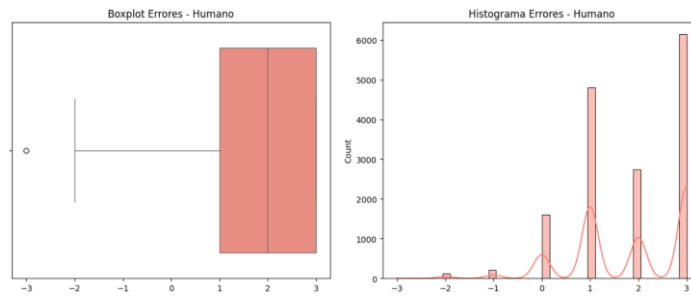
Fuente: Elaboración propia, 2025

En conjunto, los modelos CatBoost y XGBoost logran una mejor separación de clases, sobre todo en la más numerosa (0). La Regresión Ordinal, aunque vulnerable a confusiones en categorías cercanas, muestra cierta capacidad para atrapar más casos de la clase 3 en el conjunto de prueba. La comparación directa con el humano evidencia que los modelos de machine learning ofrecen una clasificación más consistente, especialmente en las categorías con duraciones medias (1 y 2). Aun así, la clase 3 continúa siendo la más compleja de distinguir, reforzando la necesidad de más datos o estrategias específicas para mejorar la detección de duraciones excepcionalmente prolongadas.

#### 3.4.4 Análisis de la dispersión de errores

Se realizaron gráficas para analizar el comportamiento de los errores del modelo en comparación con los del evaluador humano. Se incluyen boxplot y histograma de los errores, entendidos como la diferencia entre la categoría real y la predicha (valores entre -3 y +3).

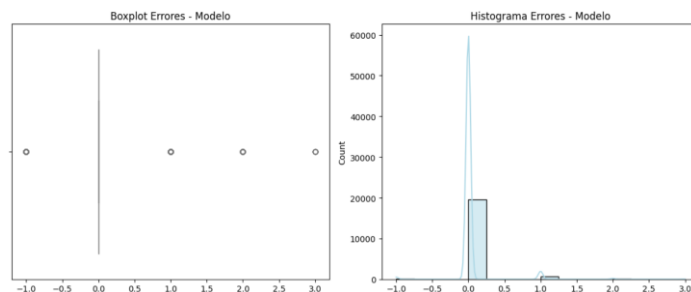
Figura 3.8: Magnitud del error humano



Fuente: Elaboración propia, 2025

Para CatBoost, la mayoría de las observaciones se concentran muy cerca del error cero, con algunos valores atípicos hacia +2 y +3. El histograma confirma que la mayor parte de los ejemplos se clasifican correctamente o con un desvío de tan solo una categoría, mientras que el desempeño humano exhibe una mediana de error desplazada hacia +2, lo que denota un mayor número de clasificaciones que distan dos o más categorías de la real.

Figura 3.9: Magnitud del error Catboost

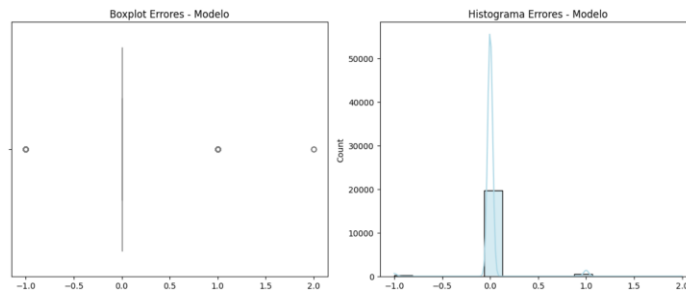


Fuente: Elaboración propia, 2025

Para XGBoost, se ve un comportamiento similar donde el boxplot indica una distribución de errores más cercana a cero, con unos pocos puntos que alcanzan +2. El histograma refuerza la concentración alrededor de 0 y +1.



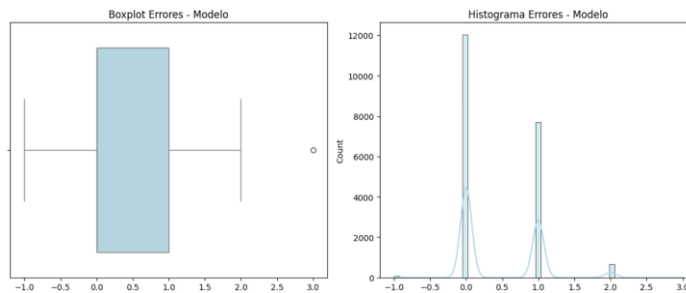
Figura 3.10: Magnitud del error XGBoost



Fuente: Elaboración propia, 2025

En la tercera figura (Regresión Ordinal), el *boxplot* presenta una dispersión algo mayor que en los métodos basados en *gradient boosting*, con un rango de error que puede llegar a  $\pm 3$ . No obstante, se aprecia una alta densidad alrededor de 0 y +1, lo que sugiere que la mayoría de los casos se encuentran correctamente predichos o con un error de una categoría. La comparación con el evaluador humano vuelve a poner de manifiesto la tendencia a errar por más de dos categorías en un número significativo de observaciones.

Figura 3.11: Magnitud del error Regresión Ordinal



Fuente: Elaboración propia, 2025

En conjunto, estas visualizaciones indican que los modelos de aprendizaje automático concentran sus errores en intervalos estrechos (en torno a 0 o  $\pm 1$ ), mientras que la clasificación humana muestra un desplazamiento marcado hacia +2 y +3. Este patrón coincide con las métricas globales presentadas en secciones anteriores, en las que los métodos CatBoost, XGBoost y Regresión Ordinal superan de forma consistente el desempeño manual en la categorización de duraciones.

Tabla 3.4 Distribución de errores residuales

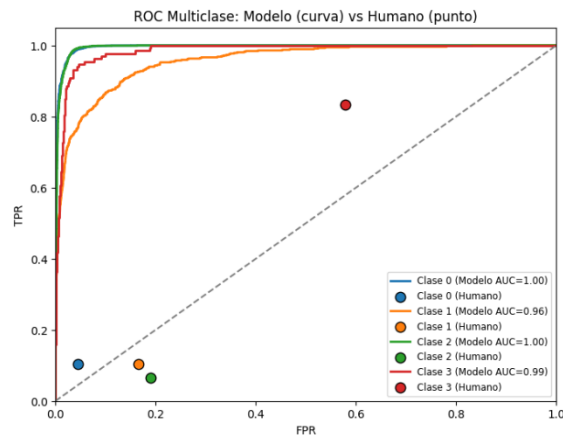
| Modelo            | Mediana $r_i$ | IQR       | Asimetría | Observaciones clave   |
|-------------------|---------------|-----------|-----------|---|
| CatBoost          | 0             | 0.00–0.03 | –0.04     | Más del 95 % de los errores se agrupa en torno a $-0.5, +0.5-0.5, +0.5$ ; se detecta un pequeño porcentaje ( $\approx 0.8\%$ ) con subestimación o sobreestimación de 2–3 categorías. |
| XGBoost           | 0             | 0.00–0.02 | –0.02     | Distribución cercana a la de CatBoost, con pocos valores atípicos (picos en $r=1$ y $r=2$ ), lo que refleja alta consistencia de clasificación.                                       |
| Regresión Ordinal | 0             | 0.00–0.10 | –0.10     | Mayor dispersión en las colas, con presencia de errores de 2–3 categorías. Sin embargo, la mayoría de los casos se mantiene cerca de $r=0$ .  |
| Humano            | 2             | 1.00–2.00 | 1.3       | Marcado sesgo hacia la sobreestimación ( $r>0$ ); picos en $r=1, 2$ y $3$ reflejan la alta frecuencia de errores distantes de la clase real.  |

Fuente: Elaboración propia, 2025

Con respecto a las curvas de capacidad discriminativa, se usan para comparar entre las curvas ROC multiclase de cada modelo (líneas continuas) y los puntos correspondientes a la evaluación humana (círculos de colores), desglosados por categoría (0, 1, 2 y 3). El eje vertical (True positive rate: TPR) refleja la sensibilidad o fracción de verdaderos positivos, mientras que el eje horizontal (False positive rate: FPR) representa la fracción de falsos positivos.

Para Catboost, el modelo exhibe AUC (área bajo la curva) muy cercana a 1.0 en las clases mayoritarias (0 y 2) y de 0.96–0.99 en las categorías restantes. Esto indica una elevada capacidad para discriminar entre las clases sin incurrir en falsos positivos.

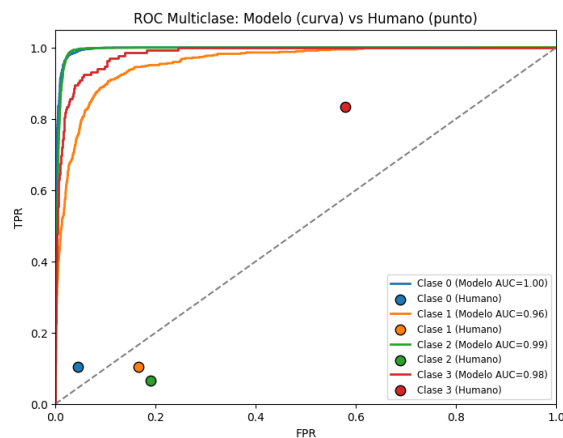
Figura 3.12: ROC Catboost



Fuente: Elaboración propia, 2025

XGBoost alcanza AUC próximas a 1.0 para las clases 0 y 2, y mantiene valores altos ( $\geq 0.95$ ) para las demás. La curva ROC se sitúa en la zona superior izquierda del espacio, indicando un bajo compromiso entre TPR y FPR.

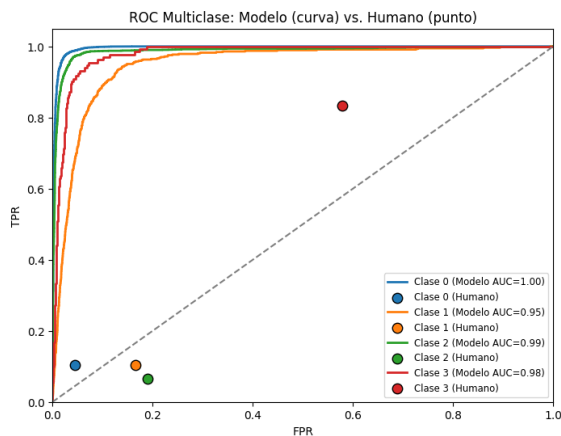
Figura 3.13: ROC XGBoost



Fuente: Elaboración propia, 2025

El modelo ordinal presenta AUC altas, cercanas a 1.0 en la clase 0, en torno a 0.95 en la clase 1, y de aproximadamente 0.99–0.98 en las clases 2 y 3. Aunque la forma de la curva revela una ligera menor pronunciación en la clase 1 respecto a los métodos boosting, el rendimiento global sigue siendo muy superior al punto correspondiente a la evaluación humana.

Figura 3.14: ROC Regresión Ordinal



Fuente: Elaboración propia, 2025

Tabla 3.5: Curvas ROC multiclase

| Clase              | CatBoost AUC | XGBoost AUC | Reg. Ordinal AUC | Humano (TPR, FPR)        |
|--------------------|--------------|-------------|------------------|--------------------------|
| 0 ( $\leq 26$ sem) | 1            | 1           | 1                | ( $\approx 0.90, 0.06$ ) |
| 1 (26–52 sem)      | 0.97         | 0.95        | 0.94             | ( $\approx 0.12, 0.16$ ) |
| 2 (52–78 sem)      | 0.99         | 0.99        | 0.99             | ( $\approx 0.08, 0.15$ ) |
| 3 ( $> 78$ sem)    | 0.98         | 0.98        | 0.97             | ( $\approx 0.80, 0.60$ ) |

Fuente: Elaboración propia, 2025

En síntesis, las curvas ROC de CatBoost, XGBoost y la Regresión Ordinal se superponen en la zona cercana al óptimo, con  $AUC \geq 0.95$ , mientras que la clasificación humana se representa en cada clase por un punto aislado, con un equilibrio TPR–FPR notablemente menos favorable. Estas diferencias confirman lo observado en las métricas anteriores: los modelos de aprendizaje automático alcanzan una discriminación más precisa entre las cuatro categorías de duración que el desempeño humano.

### 3.4.5 Importancia de las variables

Para el modelo de Regresión Ordinal, se calcularon las 15 características con mayor impacto en la predicción, utilizando el valor absoluto de sus coeficientes para ordenar su relevancia:

Tabla 3.6: Importancia de las variables para Regresión Ordinal

| Variable                                 | Coefficient | Abs_Coefficient |
|--|-------------|-----------------|
| Servicio de Salud en el Trabajo_x_100000 | -1.11055    | 1.110548        |
| Cod Cie10_S525                           | -1.085      | 1.084996        |
| Cod Cie10_S826                           | -0.95001    | 0.950008        |
| Cod Cie10_S824                           | -0.93407    | 0.934066        |
| Cod Cie10_S626                           | -0.83856    | 0.838556        |
| Cod Cie10_C509                           | 0.83371     | 0.83371         |
| Unidad Ads_UMF 2 IRAPUATO                | -0.77789    | 0.777887        |
| Unidad Ads_UMF 27 TIJ                    | -0.75       | 0.75            |
| Unidad Ads_UMF 47 LEON                   | -0.73       | 0.73            |
| transicion_13                            | -0.73342    | 0.733423        |
| transicion_23                            | -0.67445    | 0.674453        |
| primera expedicion_3                     | -0.60595    | 0.605949        |
| total_transiciones                       | 0.508978    | 0.508978        |
| veces_1                                  | -0.45       | 0.45            |
| veces_2                                  | 0.4         | 0.4             |

Fuente: Elaboración propia, 2025

El signo del coeficiente determina la dirección del efecto. Un valor negativo significa que el incremento de la variable reduce la probabilidad de asignar un caso a una categoría superior, favoreciendo, en cambio, categorías de menor duración o severidad. Por el contrario, un coeficiente positivo implica que un aumento en la variable incrementa la probabilidad de asignar una categoría de mayor duración.

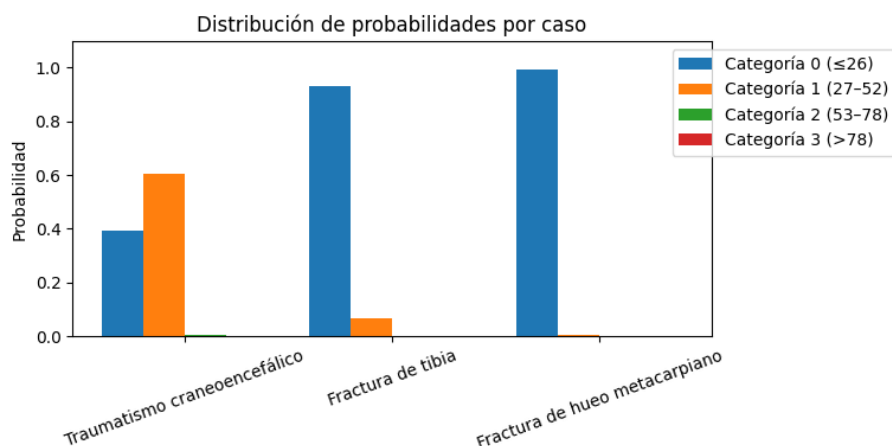
Destaca la variable “Servicio de Salud en el Trabajo\_x\_100000” por ser la variable con mayor impacto negativo, sugiriendo que su incremento disminuye la asignación a categorías de mayor duración. De igual forma, los diagnósticos específicos de CIE10 (S525, S826, S824 y S626) se ubican entre las variables más influyentes, reduciendo la probabilidad de clasificar en categorías con mayor semanas de duración.

En contraste, “Cod Cie10\_C509” presenta un coeficiente positivo, indicando que su presencia favorece la asignación a categorías superiores. Además, variables relacionadas con la procedencia del caso y contadores de transiciones (como *transicion\_13* y *transicion\_23*) muestran una tendencia a menor duración. En conjunto, estas variables con altos valores absolutos (mayores a 0.7–0.8) son las que más alteran la probabilidad final de clasificación, ofreciendo interpretación sobre los factores que influyen en la predicción del modelo.

### 3.4.6 Ejemplo de uso

Por último, se presenta un ejemplo de uso, donde se observa la distribución de probabilidades para tres casos específicos. Se espera que el presentar la información pronóstica a médicos y pacientes, a través de este formato, genere mayor claridad y certeza acerca de los posibles escenarios que puedan presentarse al futuro. Además, si se entiende que la probabilidad de que un caso sea incapacidad prolongada, se planee la atención médica de la manera más oportuna posible, para evitar tiempos muertos.

Figura 3.15: Ejemplo de tres casos



Fuente: Elaboración propia, 2025

Los resultados muestran que los tres enfoques exhiben un desempeño elevado en la clasificación de la duración de las incapacidades. XGBoost lidera con

una exactitud en el conjunto de prueba cercana al 94%, seguido muy de cerca por CatBoost con 93.6% y, en tercer lugar, la Regresión Ordinal con 92.4%, si bien esta última sufre una caída notable en validación (58.7%). No obstante, la Regresión Ordinal logra capturar un número relativamente mayor de casos en la clase 3, lo que sugiere que la naturaleza ordinal del modelo podría atenuar parcialmente las confusiones en las categorías más prolongadas. En conjunto, los métodos de gradient boosting (XGBoost y CatBoost) muestran el mejor balance entre precisión y estabilidad, mientras que la Regresión Ordinal revela un comportamiento más sensible a los cambios de distribución pero con cierto potencial para mejorar la identificación de duraciones extremas.

El desempeño de CatBoost manifiesta un comportamiento orientado a la sensibilidad para las clases menos frecuentes. Durante la validación su recall llegó a 0.75 para la clase 3 y se elevó a 0.89 en prueba, aunque la precisión descendió a 0.24. Este patrón prefiere sobredetectar incapacidades prolongadas, lo que, en un contexto institucional donde omitir casos con riesgo de pensión resulta costoso, puede considerarse ventajoso. Pero este efecto es a costa de un número mayor de falsas alarmas.

XGBoost, por el contrario, muestra un balance más uniforme entre precisión y exhaustividad; su F1 ponderado de 0.95 y la estrecha dispersión de errores (una IQR de solo 0 – 0.02) lo hacen robusto para operaciones rutinarias en las que los recursos médicos son limitados y la sobrecarga por valoraciones médicas innecesarias debe evitarse.

La regresión ordinal tiene una sensibilidad extrema al cambio de distribución de los datos. El salto del 0.59 de exactitud en validación al 0.92 en prueba se puede atribuir a que el supuesto de pendientes paralelas deja al modelo expuesto a sesgos tan pronto varían los valores de las variables.

# **Conclusiones y recomendaciones**



## Conclusiones y recomendaciones

La predicción de la duración de las incapacidades temporales es necesaria para mejorar la administración de la seguridad social y brindar criterios más sólidos al personal médico. Su relevancia radica en apoyar la toma de decisiones clínicas y administrativas en el IMSS. XGBoost alcanza la mayor exactitud global (0.94 en prueba), mientras que CatBoost ofrece la sensibilidad más alta (0.89) en la categoría crítica > 78 semanas. En contraste, la Regresión Ordinal exhibe variabilidad entre particiones, lo que evidencia su vulnerabilidad al cambio de covariabilidad. Con todo, su capacidad para recuperar casos muy prolongados es generada por su capacidad de respetar la estructura ordinal cuando se aborda un fenómeno escalonado. Al contar con mejores estimados de los rangos de tiempo, el Instituto puede fijar objetivos claros que aseguren la prestación de los servicios médicos apropiados, de acuerdo con el diagnóstico y la capacidad resolutive de la región en la que se brinde la atención.

A nivel internacional, se observa una tendencia hacia la adopción de herramientas de aprendizaje automático, que no solo ayudan a predecir eventos de salud, sino también a generar información probabilística que fortalezca la toma de decisiones clínicas y de gestión. La presente investigación coincide con hallazgos previos (Chand & Zhang, 2022; Saarela et al., 2022; Koc et al., 2021), donde métodos de boosting (XGBoost, CatBoost) han demostrado eficacia en la predicción de eventos. Pero, a diferencia de estudios que abordan la predicción de la incapacidad como un problema de clasificación binaria (Na & Kim, 2019; Saarela et al., 2022), esta investigación adopta una perspectiva ordinal ( $\leq 26$ , 26–52, 52–78, >78). Dicha aproximación, menos común en la literatura, mejora la identificación de casos con duraciones muy prolongadas, reforzando el argumento de McCullagh (1980) sobre la pertinencia de modelar categorías con un orden implícito.

Además, al igual que en Kusnadi et al. (2023), la información contextual de infraestructura y número de camas hospitalarias aportó al modelo robustez y una visión más amplia de la heterogeneidad entre los estados del país.

Para poder responder a los objetivos e hipótesis planteados, se generan las siguientes conclusiones:

*1. Los modelos predictivos entrenados superan el desempeño del criterio humano en la estimación de la duración de las incapacidades*

Los resultados confirman que cualquier de los tres modelos, sea Regresión Ordinal, XGBoost o CatBoost; presentan una precisión superior al consenso de expertos, en particular aquellos correspondientes a categorías intermedias (26–52 y 52–78 semanas). Esto, similar a la tendencia que muestran Chou JCL et al. (2022) y Meyers et al. (2018) al confrontar la opinión humana con algoritmos de ML. Este hallazgo respalda la hipótesis general de que la incorporación de factores contextuales y clínicos mejora la exactitud en la predicción, lo que sugiere un potencial significativo para optimizar la gestión de recursos económicos del IMSS. Dado que para las incapacidades del ramo de riesgo de trabajo, cruzar el umbral de las 52 semanas significa otorgar pensión permanente a un trabajador. Por lo que identificar casos con alta probabilidad de cruzar este umbral, y predecirlo de manera precisa, puede informar al personal médico de la necesidad de extremar cuidados con el paciente para aumentar las probabilidades de una recuperación satisfactoria o en el menor tiempo posible.

*2. La variable objetivo se ajusta de forma adecuada a enfoques de clasificación ordinal*

Aunque XGBoost lidera la exactitud global, la Regresión Ordinal alcanza la mejor precisión relativa en la clase escasa ( $> 78$  semanas). Este hallazgo sostiene que modelar explícitamente la jerarquía  $\leq 26 < 26-52 < 52-78 < > 78$  suaviza la confusión entre categorías contiguas y resalta la necesidad de explorar enfoques híbridos de ordinal boosting.

*3. Los casos más prolongados constituyen el mayor desafío de predicción*

De la conclusión anterior, se desprende que las clases con duraciones superiores a 78 semanas muestran mayor heterogeneidad entre sí y menor frecuencia de aparición, lo que se traduce en menores valores de

exhaustividad (recall). Esto sugiere que, en trabajos posteriores, la inclusión de variables adicionales como son la capacidad funcional, adherencia al tratamiento o factores psicológicos; podrían elevar la capacidad discriminatoria de los modelos.

*4. La incorporación de datos de infraestructura hospitalaria contribuye a la robustez del modelo*

La influencia de variables como camas censables, quirófanos o servicios disponibles por región apunta a la relevancia de la capacidad instalada en la duración de las incapacidades. El considerarlas respalda la impresión de que aquellas zonas del país con menor capacidad generan desperdicios en todo el sistema, pues casos que deberían ser atendidos se prolongan inevitablemente. Provocando mayor uso del subsidio de incapacidad y menor recuperación de los pacientes por no ser atendidos de manera oportuna. Además de provocar mayor tiempo ausente del trabajador en su empleo, lo cual estresa financieramente a los patrones.

*5. La creación de una base de datos limpia y la extracción de características enriquecidas validan la primera hipótesis específica*

El adecuado preprocesamiento con el uso de técnicas de codificación para variables categóricas y normalización de datos numéricos; además de la extracción de rasgos a partir de la secuencia de atención influyen positivamente en la calidad del entrenamiento de los modelos. Este proceso refuerza la importancia de mantener los registros institucionales de manera íntegra y consistente para su análisis y generación de información.

### **Limitaciones identificadas**

- El conjunto de datos abarca únicamente las incapacidades mayores a 100 días, por lo que los resultados no son generalizables a incapacidades de menor duración.

- El carácter correlacional de la metodología impide establecer causalidades directas; se requiere investigación adicional para formular vías o caminos causales en donde las variables expliquen la duración de incapacidades.
- El uso de datos con fines administrativos limita las variables que se integraron al modelo. En un escenario ideal se tendrían variables representativas de la salud de los trabajadores o de su empleo. Entre las variables de salud se podrían integrar el sexo, la gravedad de la lesión, el tratamiento recibido, entre otras. Para las variables del empleo se podrían integrar el sector productivo al que pertenece la empresa, el tamaño de la misma, el nivel de riesgo calificado por el IMSS, además de la naturaleza de las funciones del trabajador. Incluir estas variables mejoraría la capacidad predictiva de modelos de aprendizaje automático.

### **Líneas futuras de investigación**

- Utilizar diseños longitudinales permitirá capturar la evolución de la incapacidad en el tiempo, de modo que se actualicen las predicciones conforme la evolución de cada caso, según el trabajador avance o se estanque en su proceso de mejora.
- Explorar la integración de técnicas de aprendizaje automático basadas en datos no estructurados para hacer mayor uso de los expedientes clínicos electrónicos. Los expedientes electrónicos del Instituto acumulan información de mínimo 10 años. La cual podría ser un insumo para entrenar modelos de procesamiento de lenguaje natural y poder integrar las narraciones que los profesionales médicos realizan. En estas narraciones describen cada caso, el cómo se originó el padecimiento, su evolución y la expectativa del médico en cuanto a su percepción de si el trabajador puede o no trabajar.
- Ampliar la base de datos para abarcar incapacidades breves y otros factores como comorbilidades, hábitos de vida o datos socioeconómicos que refinan la predicción y faciliten intervenciones más precisas.

## **Implicaciones para la práctica en el IMSS**

La adopción de un modelo predictivo ofrece una herramienta alineada con la necesidad institucional de asignar recursos de manera eficiente. Esto se puede materializar mediante la diferenciación de casos según su probable duración, para reducir el uso de subsidios y prevenir tiempos muertos entre episodios de valoración o tratamiento médico. Además, entender el efecto de la capacidad hospitalaria en la prolongación de incapacidades puede promover una distribución equitativa de recursos.

Este modelo busca mejorar el uso de los recursos institucionales, reduciendo la incertidumbre alrededor de la duración de los casos, para que, últimamente contribuya a la sostenibilidad financiera de los Seguros de Riesgos de Trabajo y de Enfermedades y Maternidad. Sumado a esto, se busca que el enfoque probabilístico complemente los criterios actuales, alertando sobre eventualidades atípicas y reforzando la transparencia institucional. Este sistema busca la modernización de la seguridad social en México, logrando mayor exactitud y confiabilidad en la toma de decisiones.

En cuanto a la importancia de las variables en el modelo de Regresión Ordinal, es notorio que la variable que refleja la disponibilidad del servicio de salud ocupacional dentro del Instituto esté asociado negativamente a mayor duración de incapacidades. Por lo que desarrollar estrategias para fortalecer estos servicios, amerita el análisis y planeación de gestionar la infraestructura, el personal y el equipo necesario para que los trabajadores sean oportunamente evaluados si es que son aptos para integrarse a trabajar, o si deben proceder a un proceso de invalidez o pensión.

## Fuentes de consulta

- 1 Abràmoff, M. D., Lavin, P. T., Birch, M., Shah, N., & Folk, J. C. (2018). Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. *npj Digital Medicine*, 1(1). <https://doi.org/10.1038/s41746-018-0040-6>
- 2 Agresti, A. (2012). *Categorical data analysis* (3rd ed.). John Wiley & Sons.
- 3 Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
- 4 Bloch, F. S., & Prins, R. (Eds.). (2001). *Who returns to work and why? A six-country study on work incapacity & reintegration* (International Social Security Series, Vol. 5). International Social Security Association Research Programme.
- 5 Brahim, S., El Hussein, M., & Al-Reedy, A. (2022). Detection of undeserved sick leaves in hospitals using machine learning techniques. *Sustainable Computing: Informatics and Systems*, 35, 100665. <https://doi.org/10.1016/j.suscom.2022.100665>.
- 6 Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2), 211–252. <https://doi.org/10.1111/j.2517-6161.1964.tb00553.x>
- 7 Castro, V. O. J., Haro, A. M. E., & Quiñones, M. K. A. (2019). Apego a las guías de duración de la incapacidad laboral por patología en fracturas de tobillo. *Revista Cubana de Salud y Trabajo*, 20(1).
- 8 Chand, S., & Zhang, Y. (2022). Learning from machines to close the gap between funding and expenditure in the Australian National Disability Insurance Scheme. *International Journal of Information Management Data Insights*, 2(1), Article 100077. <https://doi.org/10.1016/j.jjime.2022.100077>
- 9 Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. <https://doi.org/10.1145/2939672.2939785>

- 10 Cheng, A. S. K., Ng, P. H. F., Sin, Z. P. T., Lai, S. H. S., & Law, S. W. (2020). Smart work injury management (SWIM) system: Artificial intelligence in work disability management. *Journal of Occupational Rehabilitation*, 30(3), 354–361. <https://doi.org/10.1007/s10926-020-09886-y>
- 11 Choi, S. B., Lee, S., & Lee, W. (2023). Status and prediction of disapproval of the Korean workers' compensation insurance for diseases and injuries. *Journal of Occupational Health*, 65(1), 1-13. <https://doi.org/10.1002/1348-9585.12392>
- 12 Chou, J. C.-L., Chen, Y.-S., Lin, C.-K., & Ting, M.-H. (2022). Application of a classified prediction model to benefit the LTC insurance. In *Proceedings of IConEST 2022 – International Conference on Engineering, Science and Technology* (pp. 52–58).
- 13 Cohen, J. (1968). Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4), 213–220. <https://doi.org/10.1037/h0026256>
- 14 Gewurtz, R. E., Premji, S., & Holness, D. L. (2019). The experiences of workers who do not successfully return to work following a work-related injury. *Work*, 61(4), 537–549. <https://doi.org/10.3233/WOR-182824>
- 15 Gomis, R., Kapsos, S., & Kuhn, S. (2020). World employment and social outlook: Trends 2020. International Labour Organization.
- 16 Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- 17 Huhta-Koivisto, T. (2020). Work disability risk prediction with machine learning Unpublished master's thesis, Aalto University. Espoo, Finlandia, School of Electrical Engineering.
- 18 Informe al Ejecutivo Federal y al Congreso de la Unión sobre la situación financiera y los riesgos del IMSS 2023-2024. (2024). Instituto Mexicano del Seguro Social.
- 19 Kayyali, B., Knott, D., & Van Kuiken, S. (2013). The big-data revolution in US health care: Accelerating value and innovation. McKinsey & Company.
- 20 Koc, K., Ekmekcioğlu, Ö., & Gurgun, A. P. (2021). Integrating feature engineering, genetic algorithm and tree-based machine learning methods to

- predict the post-accident disability status of construction workers. *Automation in Construction*, 131, 103896. <https://doi.org/10.1016/j.autcon.2021.103896>
- 21 Kusnadi, F., Wijaya, A., & Lesmono, J. D. (2023). Prediction of maternity recovery rate of group long-term disability insurance using XGBoost. *JTAM (Jurnal Teori Dan Aplikasi Matematika)*, 7(4). <https://doi.org/10.31764/jtam.v7i4.16825>
  - 22 Lagunas Sosa, M. de L., Rivera Corona, J. G., Gámez Almaraz, D., Pineda Ullua, J. L., & Merino González, E. L. (2023). Impacto económico y administrativo de la reforma laboral en materia de vacaciones dignas en México. *Ciencia Latina Revista Científica Multidisciplinar*, 7(4). [https://doi.org/10.37811/cl\\_rcm.v7i4.7505](https://doi.org/10.37811/cl_rcm.v7i4.7505)
  - 23 Lapadula, P., Mecca, G., Santoro, D., Solimando, L., & Veltri, E. (2020). Greg, ML – machine learning for healthcare at a scale. *Health and Technology*, 10(6). <https://doi.org/10.1007/s12553-020-00468-9>.
  - 24 Lanz, C. J. E., Haro, A. M. E., Quiñones, M. K., et al. (2018). Retro-información a médicos familiares para optimizar la prescripción de certificados de incapacidad temporal en una unidad médico familiar. *Revista Cubana de Salud y Trabajo*, 19(3), 3–15.
  - 25 Ley del Seguro Social. (1995). Diario Oficial de la Federación (última reforma, 7 de junio de 2024).
  - 26 Manual de integración y funcionamiento de los comités y subcomités para el control de la incapacidad temporal para el trabajo en los ámbitos normativo, órganos de operación administrativa desconcentrada estatal y regional, de las unidades médicas de alta especialidad y operativos (COCOITT). (2024). Instituto Mexicano del Seguro Social.
  - 27 Martin-Fumadó, C., Martí Amengual, G., Puig Bausili, L., & Arimany-Manso, J. (2014). La incapacidad temporal y sus implicaciones legales. *Medicina Clínica*, 142(Suppl. 2). [https://doi.org/10.1016/S0025-7753\(14\)70070-3](https://doi.org/10.1016/S0025-7753(14)70070-3)
  - 28 McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2), 109–142.



- 29 Meyers, A. R., Al-Tarawneh, I. S., Wurzelbacher, S. J., Bushnell, P. T., Lampl, M. P., Bell, J. L., Bertke, S. J., Robins, D. C., Tseng, C. Y., Wei, C., Raudabaugh, J. A., & Schnorr, T. M. (2018). Applying machine learning to workers' compensation data to identify industry-specific ergonomic and safety prevention priorities: Ohio, 2001 to 2011. *Journal of Occupational and Environmental Medicine*, 60(1), 55-73. <https://doi.org/10.1097/JOM.0000000000001162>.
- 30 Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explorations Newsletter*, 3(1), 27–32. <https://doi.org/10.1145/507533.507538>
- 31 Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2017). Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6), [páginas]. <https://doi.org/10.1093/bib/bbx044>
- 32 Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
- 33 Na, K. S., & Kim, E. (2019). A machine learning-based predictive model of return to work after sick leave. *Journal of Occupational and Environmental Medicine*, 61(5), (pp. 191-199). <https://doi.org/10.1097/JOM.0000000000001567>
- 34 Nayak, B., Bhattacharyya, S. S., & Krishnamoorthy, B. (2019). Democratizing health insurance services; accelerating social inclusion through technology policy of health insurance firms. *Business Strategy and Development*, 2(3), (pp. 1–11). <https://doi.org/10.1002/bsd2.59>
- 35 Norma de incapacidad temporal para el trabajo. (2024). Instituto Mexicano del Seguro Social.
- 36 Norma para la dictaminación de los accidentes y enfermedades de trabajo. (2023). Dirección de Prestaciones Económicas y Sociales, Instituto Mexicano del Seguro Social.
- 37 Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31).

- 38 Provost, F., & Fawcett, T. (2013). What you need to know about data mining and data analytics. O'Reilly.
- 39 Ramezani, M., Takian, A., Bakhtiari, A., Rabiee, H. R., Fazaeli, A. A., & Sazgarnejad, S. (2023). The application of artificial intelligence in health financing: A scoping review. *Cost Effectiveness and Resource Allocation*, 21(1), [páginas]. <https://doi.org/10.1186/s12962-023-00492-2>
- 40 Reglamento interior del Instituto Mexicano del Seguro Social. (2006). Diario Oficial de la Federación. Instituto Mexicano del Seguro Social.
- 41 Saarela, K., Huhta-Koivisto, V., & Nurminen, J. K. (2022). Work disability risk prediction using machine learning, comparison of two methods. In K. Daimi & A. Al Sadoon (Eds.), *Proceedings of the ICR'22 International Conference on Innovations in Computing Research* (pp. 13–21). Springer Nature. [https://doi.org/10.1007/978-3-031-14054-9\\_2](https://doi.org/10.1007/978-3-031-14054-9_2)
- 42 Secretaría de Salud. (2023). 073. Este año, 12 mil 500 especialistas egresan del Sistema Nacional de Residencia Médicas: DGCES [Nota de prensa]. <https://www.gob.mx/salud/prensa/073-este-ano-12-mil-500-especialistas-egresan-del-sistema-nacional-de-residencia-medicas-dgces>
- 43 Universidad Nacional Autónoma de México. (2023). La distribución del personal de salud, un desafío en México. *Gaceta UNAM*. <https://www.gaceta.unam.mx/la-distribucion-del-personal-de-salud-un-desafio-en-mexico/>

**ANEXOS**

# ANEXO 1

## Modelo de CatBoost

```
# Importaciones
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Importa CatBoost
from catboost import CatBoostClassifier

# 1) Configuración
CONFIG = {
    "MODEL_PATH": "/content/drive/My Drive/catboost_model_params.npz",
    "DATA_ITT_PATH": "/content/drive/My Drive/BASE_ITT.xlsx",
    "DATA_NEW2_PATH": "/content/drive/My Drive/CONJUNTO_PRUEBA.xlsx",
}

# Variables de entrada y sus categorías
FEATURES = [
    "Cod Cie10",
    "Unidad Ads",
    "Tip Ramo",
    "first",
    "Avg. Imp Salario Topado",
    "Dias Probables Recuperacion",
    "Max. Edad",
    "veces_nivel3",
    "transicion_12",
    "transicion_13",
    "transicion_23",
    "total_transiciones",
    "seq_length",
    "count_1",
    "count_2",
    "Total de Camas Censables de la delegación_x_100000",
    "Total de Consultorios de la Unidad_x_100000",
    "Sala de Quirófano_x_100000",
    "Servicio de Salud en el Trabajo_x_100000"
]

CAT_FEATURES = [
    "Cod Cie10",
    "Unidad Ads",
    "Tip Ramo",
    "first",
]

# Parámetros base de CatBoost
CAT_PARAMS = {
    "iterations": 100,
    "depth": 6,
    "learning_rate": 0.1,
    "random_seed": 42,
```

```

    "verbose": 0
}

# 2) Funciones Utilitarias
def assign_target(weeks):
    if weeks <= 26:
        return 0
    elif weeks <= 52:
        return 1
    elif weeks <= 78:
        return 2
    else:
        return 3

def ordinal_preprocess_df(df, require_target=False):
    if require_target:
        if "Target" not in df.columns:
            if "Semanas" not in df.columns:
                raise ValueError("El DataFrame debe tener 'Semanas' para crear 'Target'")
            df["Target"] = df["Semanas"].apply(assign_target).astype(int)
        else:
            df["Target"] = df["Target"].astype(int)

    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    df.fillna(0, inplace=True)

    cat_cols_in_df = [c for c in CAT_FEATURES if c in df.columns]
    # Se hace one-hot, igual que en mord/xgboost
    df = pd.get_dummies(df, columns=cat_cols_in_df, drop_first=True)

    for col in FEATURES:
        if col not in df.columns:
            df[col] = 0

    if df.isna().any().any():
        raise ValueError("Se encontraron NaN después de fillna(0). Revisar datos.")
    return df

def filter_feature_columns(X, features):
    final_cols = []
    all_cols = list(X.columns)
    for col in all_cols:
        if col in features:
            final_cols.append(col)
        else:
            prefix = col.split('_')[0]
            if prefix in CAT_FEATURES:
                final_cols.append(col)
    return final_cols

def train_catboost_model(df, features, cat_params, test_size=0.2):
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y = df["Target"]

    final_cols = filter_feature_columns(X, features)

```

```

X = X[final_cols]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=test_size, random_state=42, stratify=y
)

model = CatBoostClassifier(**cat_params)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("Precisión inicial (CatBoost):", acc)
print("Reporte de Clasificación Inicial:\n",
      classification_report(y_test, y_pred, zero_division=0))
print("Matriz de Confusión Inicial:\n", confusion_matrix(y_test, y_pred))

return model, X_test, y_test

def save_catboost_model(model, model_path, X_train=None):
    if X_train is None:
        raise ValueError("X_train se requiere para guardar el orden de columnas.")

    import io
    buffer = io.BytesIO()
    model.save_model(buffer, format="cbm")
    buffer.seek(0)
    raw_bytes = buffer.read()

    np.savez_compressed(
        model_path,
        catboost_dump=raw_bytes,
        columns=X_train.columns.values
    )
    print(f"Modelo CatBoost guardado en {model_path}")

def load_catboost_model(model_path, cat_params):
    data = np.load(model_path, allow_pickle=True)
    raw_bytes = data["catboost_dump"]
    columns = data["columns"]

    model = CatBoostClassifier(**cat_params)
    import io
    buffer = io.BytesIO(raw_bytes)
    model.load_model(buffer, format="cbm")
    print(f"Modelo CatBoost cargado desde {model_path}")

    return model, columns

def evaluate_catboost_model(model, df, model_cols):
    """
    Evalúa un CatBoostClassifier en un DataFrame con 'Target'.
    Retorna la precisión y muestra métricas.
    """
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y_true = df["Target"]

```

```

X = X.reindex(columns=model_cols, fill_value=0)

y_pred = model.predict(X)
acc = accuracy_score(y_true, y_pred)
print("\nPrecisión de la Evaluación:", acc)
print("Reporte de Clasificación:\n", classification_report(y_true, y_pred, zero_division=0))
print("Matriz de Confusión:\n", confusion_matrix(y_true, y_pred))
return acc

# Búsqueda simple para CatBoostClassifier
def hyperparameter_search_catboost(df, features, param_grid=None, n_splits=3):
    if param_grid is None:
        param_grid = {
            "depth": [6],
            "iterations": [100],
            "learning_rate": [0.1]
        }

    df_prep = ordinal_preprocess_df(df.copy(), require_target=True)
    X_all = df_prep.drop(columns=["Target"], errors="ignore")
    y_all = df_prep["Target"]

    final_cols = filter_feature_columns(X_all, features)
    X_all = X_all[final_cols]

    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)

    best_params = None
    best_mean_acc = -1

    import itertools
    keys = list(param_grid.keys())
    combos = list(itertools.product(*(param_grid[k] for k in keys)))

    print("\n--- Ajuste de Hiperparámetros CatBoost ---")
    for combo in combos:
        trial_params = dict(zip(keys, combo))

        fold accuracies = []
        for train_index, valid_index in kf.split(X_all, y_all):
            X_train_cv, X_valid_cv = X_all.iloc[train_index], X_all.iloc[valid_index]
            y_train_cv, y_valid_cv = y_all.iloc[train_index], y_all.iloc[valid_index]

            full_params = {"CAT_PARAMS", **trial_params}
            model_cv = CatBoostClassifier(**full_params)
            model_cv.fit(X_train_cv, y_train_cv)

            y_pred_cv = model_cv.predict(X_valid_cv)
            acc_cv = accuracy_score(y_valid_cv, y_pred_cv)
            fold accuracies.append(acc_cv)

        mean_acc = np.mean(fold accuracies)
        print(f"Parámetros={trial_params}, precisión promedio={mean_acc:.4f}")

    if mean_acc > best_mean_acc:

```

```

        best_mean_acc = mean_acc
        best_params = trial_params

    print(f"\nMejores parámetros encontrados: {best_params} (mean_acc={best_mean_acc:.4f})")
    return best_params

# 3) Flujo Principal
if __name__ == "__main__":
    # (a) Lectura del dataset base
    print("\nLeyendo dataset base (ITT)...")
    df_itt = pd.read_excel(CONFIG["DATA_ITT_PATH"])

    # (b) Búsqueda de hiperparámetros con CatBoost (opcional)
    best_params = hyperparameter_search_catboost(
        df_itt.copy(),
        FEATURES,
        param_grid={
            "depth": [4, 6],
            "iterations": [100, 200],
            "learning_rate": [0.01, 0.1]
        },
        n_splits=3
    )
    CAT_PARAMS.update(best_params)

    # (c) Entrenamiento inicial con ITT usando CatBoost
    print("\nEntrenando modelo CatBoost con dataset ITT...")
    base_model, X_test_base, y_test_base = train_catboost_model(df_itt, FEATURES,
CAT_PARAMS)

    # (d) Guardado del modelo
    save_catboost_model(base_model, CONFIG["MODEL_PATH"], X_train=X_test_base)

    # (e) Lectura de CONJUNTO_PRUEBA y evaluación final
    print("\nLeyendo CONJUNTO_PRUEBA (dataset de prueba) y evaluando el modelo CatBoost...")
    newest_df = pd.read_excel(CONFIG["DATA_NEW2_PATH"])

    # (f) Carga del modelo y evaluación
    loaded_model, loaded_cols = load_catboost_model(CONFIG["MODEL_PATH"], CAT_PARAMS)
    evaluate_catboost_model(loaded_model, newest_df.copy(), loaded_cols)

    # (g) Comparación de desempeño: Modelo vs. "Humano"
    print("\n=== COMPARANDO DESEMPEÑO DEL MODELO CATBOOST VS. HUMANO
(CONJUNTO_PRUEBA) ===")
    newest_df_for_compare = newest_df.copy()
    newest_df_for_compare = ordinal_preprocess_df(newest_df_for_compare, require_target=True)

    y_true_compare = newest_df_for_compare["Target"]
    X_compare = newest_df_for_compare.drop(columns=["Target"], errors="ignore")
    X_compare_aligned = X_compare.reindex(columns=loaded_cols, fill_value=0)

    # Predicción del modelo
    y_pred_model = loaded_model.predict(X_compare_aligned)

    # Predicción "humana": 'Dias Probables Recuperacion' en días => convertir a semanas
    def assign_human_target(days):

```



```

    weeks = days / 7.0
    return assign_target(weeks)

y_pred_human = newest_df_for_compare["Dias Probables
Recuperacion"].apply(assign_human_target)

# (h) Cálculo de métricas
acc_model = accuracy_score(y_true_compare, y_pred_model)
acc_human = accuracy_score(y_true_compare, y_pred_human)

print("\n--- Resultados finales (CatBoost) ---")
print(f"Precisión del Modelo CatBoost: {acc_model:.4f}")
print(f"Precisión Humana: {acc_human:.4f}")

print("\n--- Reporte de Clasificación (CatBoost) ---")
print(classification_report(y_true_compare, y_pred_model, zero_division=0))
print("Matriz de Confusión (CatBoost):")
print(confusion_matrix(y_true_compare, y_pred_model))

print("\n--- Reporte de Clasificación (Humano) ---")
print(classification_report(y_true_compare, y_pred_human, zero_division=0))
print("Matriz de Confusión (Humano):")
print(confusion_matrix(y_true_compare, y_pred_human))

```

## Modelo de XGBoost

```

# Importaciones
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Importa XGBoost
from xgboost import XGBClassifier

# 1) Configuración
CONFIG = {
    "MODEL_PATH": "/content/drive/My Drive/xgb_model_params.npz",
    "DATA_ITT_PATH": "/content/drive/My Drive/BASE_ITT.xlsx",
    "DATA_NEW2_PATH": "/content/drive/My Drive/CONJUNTO_PRUEBA.xlsx",
}

# Variables de entrada y sus categorías
FEATURES = [
    "Cod Cie10",
    "Unidad Ads",
    "Tip Ramo",
    "first",
    "Avg. Imp Salario Topado",
    "Dias Probables Recuperacion",
    "Max. Edad",
    "veces_nivel3",
    "transicion_12",
    "transicion_13",
    "transicion_23",
    "total_transiciones",
    "seq_length",

```

```

"count_1",
"count_2",
"Total de Camas Censables de la delegación_x_100000",
"Total de Consultorios de la Unidad_x_100000",
"Sala de Quirófano_x_100000",
"Servicio de Salud en el Trabajo_x_100000"
]

CAT_FEATURES = [
"Cod Cie10",
"Unidad Ads",
"Tip Ramo",
"first",
]

# Parámetros base de XGBoost
XGB_PARAMS = {
    "use_label_encoder": False,
    "eval_metric": "mlogloss",
    "max_depth": 6,
    "learning_rate": 0.1,
    "n_estimators": 100,
}

# 2) Funciones Utilitarias
def assign_target(weeks):
    if weeks <= 26:
        return 0
    elif weeks <= 52:
        return 1
    elif weeks <= 78:
        return 2
    else:
        return 3

def ordinal_preprocess_df(df, require_target=False):
    if require_target:
        if "Target" not in df.columns:
            if "Semanas" not in df.columns:
                raise ValueError("El DataFrame debe tener 'Semanas' para crear 'Target'")
            df["Target"] = df["Semanas"].apply(assign_target).astype(int)
        else:
            df["Target"] = df["Target"].astype(int)

    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    df.fillna(0, inplace=True)

    cat_cols_in_df = [c for c in CAT_FEATURES if c in df.columns]
    df = pd.get_dummies(df, columns=cat_cols_in_df, drop_first=True)

    for col in FEATURES:
        if col not in df.columns:
            df[col] = 0

    if df.isna().any().any():
        raise ValueError("Se encontraron NaN después de fillna(0). Revisar datos.")

```

```

return df

def filter_feature_columns(X, features):
    final_cols = []
    all_cols = list(X.columns)
    for col in all_cols:
        if col in features:
            final_cols.append(col)
        else:
            prefix = col.split('_')[0]
            if prefix in CAT_FEATURES:
                final_cols.append(col)
    return final_cols

def train_xgb_model(df, features, xgb_params, test_size=0.2):
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y = df["Target"]

    final_cols = filter_feature_columns(X, features)
    X = X[final_cols]

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=test_size, random_state=42, stratify=y
    )

    model = XGBClassifier(**xgb_params)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print("Precisión inicial (XGBoost):", acc)
    print("Reporte de Clasificación Inicial:\n",
          classification_report(y_test, y_pred, zero_division=0))
    print("Matriz de Confusión Inicial:\n", confusion_matrix(y_test, y_pred))

    return model, X_test, y_test

def save_xgb_model(model, model_path, X_train=None):
    if X_train is None:
        raise ValueError("X_train se requiere para guardar el orden de columnas.")
    booster_dump = model.get_booster().save_raw()

    np.savez_compressed(
        model_path,
        booster_dump=booster_dump,
        columns=X_train.columns.values
    )
    print(f"Modelo XGBoost guardado en {model_path}")

def load_xgb_model(model_path, xgb_params):
    data = np.load(model_path, allow_pickle=True)
    booster_dump = data["booster_dump"]
    columns = data["columns"]

    model = XGBClassifier(**xgb_params)

```

```

model.load_model(booster_dump)
print(f"Modelo XGBoost cargado desde {model_path}")

return model, columns

def evaluate_xgb_model(model, df, model_cols):
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y_true = df["Target"]

    X = X.reindex(columns=model_cols, fill_value=0)

    y_pred = model.predict(X)
    acc = accuracy_score(y_true, y_pred)
    print("\nPrecisión de la Evaluación:", acc)
    print("Reporte de Clasificación:\n", classification_report(y_true, y_pred, zero_division=0))
    print("Matriz de Confusión:\n", confusion_matrix(y_true, y_pred))
    return acc

def hyperparameter_search_xgb(df, features, param_grid=None, n_splits=3):
    if param_grid is None:
        param_grid = {
            "max_depth": [3, 6],
            "learning_rate": [0.05, 0.1],
            "n_estimators": [50, 100]
        }

    df_prep = ordinal_preprocess_df(df.copy(), require_target=True)
    X_all = df_prep.drop(columns=["Target"], errors="ignore")
    y_all = df_prep["Target"]

    final_cols = filter_feature_columns(X_all, features)
    X_all = X_all[final_cols]

    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)

    best_params = None
    best_mean_acc = -1

    # Se generan todas las combinaciones de hiperparámetros
    import itertools
    keys = list(param_grid.keys())
    combos = list(itertools.product(*(param_grid[k] for k in keys)))

    print("\n--- Ajuste de Hiperparámetros XGBoost ---")
    for combo in combos:
        # Construimos un dict con los valores
        trial_params = dict(zip(keys, combo))

        fold accuracies = []
        for train_index, valid_index in kf.split(X_all, y_all):
            X_train_cv, X_valid_cv = X_all.iloc[train_index], X_all.iloc[valid_index]
            y_train_cv, y_valid_cv = y_all.iloc[train_index], y_all.iloc[valid_index]

            # Se mezcla con los XGB_PARAMS base (eval_metric, etc.)
            full_params = {**XGB_PARAMS, **trial_params}

```

```

    model_cv = XGBClassifier(**full_params)
    model_cv.fit(X_train_cv, y_train_cv)

    y_pred_cv = model_cv.predict(X_valid_cv)
    acc_cv = accuracy_score(y_valid_cv, y_pred_cv)
    fold_accuracies.append(acc_cv)

    mean_acc = np.mean(fold_accuracies)
    print(f"Parámetros={trial_params}, precisión promedio={mean_acc:.4f}")

    if mean_acc > best_mean_acc:
        best_mean_acc = mean_acc
        best_params = trial_params

    print(f"\nMejores parámetros encontrados: {best_params} (mean_acc={best_mean_acc:.4f}")
    return best_params

# 3) Flujo Principal
if __name__ == "__main__":
    # (a) Lectura del dataset base
    print("Leyendo dataset base (ITT)...")
    df_itt = pd.read_excel(CONFIG["DATA_ITT_PATH"])

    # (b) Búsqueda de hiperparámetros con XGBoost
    best_params = hyperparameter_search_xgb(
        df_itt.copy(),
        FEATURES,
        param_grid={
            "max_depth": [3, 6],
            "learning_rate": [0.01, 0.1],
            "n_estimators": [50, 100]
        },
        n_splits=3
    )
    # Actualización XGB_PARAMS
    XGB_PARAMS.update(best_params)

    # (c) Entrenamiento inicial con ITT usando XGBoost
    print("\nEntrenando modelo XGBoost con dataset ITT...")
    base_model, X_test_base, y_test_base = train_xgb_model(df_itt, FEATURES, XGB_PARAMS)

    # (d) Guardado del modelo
    save_xgb_model(base_model, CONFIG["MODEL_PATH"], X_train=X_test_base)

    # (e) Lectura de CONJUNTO_PRUEBA y evaluación final
    print("\nLeyendo CONJUNTO_PRUEBA (dataset de prueba) y evaluando el modelo XGBoost...")
    newest_df = pd.read_excel(CONFIG["DATA_NEW2_PATH"])

    # (f) Carga del modelo y evaluación
    loaded_model, loaded_cols = load_xgb_model(CONFIG["MODEL_PATH"], XGB_PARAMS)
    evaluate_xgb_model(loaded_model, newest_df.copy(), loaded_cols)

    # (g) Comparación de desempeño: Modelo vs. "Humano"
    print("\n=== COMPARANDO DESEMPEÑO DEL MODELO XGBOOST VS. HUMANO
(CONJUNTO_PRUEBA) ===")
    newest_df_for_compare = newest_df.copy()

```

```

newest_df_for_compare = ordinal_preprocess_df(newest_df_for_compare, require_target=True)

y_true_compare = newest_df_for_compare["Target"]
X_compare = newest_df_for_compare.drop(columns=["Target"], errors="ignore")
X_compare_aligned = X_compare.reindex(columns=loaded_cols, fill_value=0)

# Predicción del modelo
y_pred_model = loaded_model.predict(X_compare_aligned)

# Predicción "humana": 'Dias Probables Recuperacion' en días => convertir a semanas
def assign_human_target(days):
    weeks = days / 7.0
    return assign_target(weeks)

y_pred_human = newest_df_for_compare["Dias Probables
Recuperacion"].apply(assign_human_target)

# (h) Cálculo de métricas
acc_model = accuracy_score(y_true_compare, y_pred_model)
acc_human = accuracy_score(y_true_compare, y_pred_human)

print("\n--- Resultados finales (XGBoost) ---")
print(f"Precisión del Modelo XGBoost: {acc_model:.4f}")
print(f"Precisión Humana: {acc_human:.4f}")

print("\n--- Reporte de Clasificación (XGBoost) ---")
print(classification_report(y_true_compare, y_pred_model, zero_division=0))
print("Matriz de Confusión (XGBoost):")
print(confusion_matrix(y_true_compare, y_pred_model))

print("\n--- Reporte de Clasificación (Humano) ---")
print(classification_report(y_true_compare, y_pred_human, zero_division=0))
print("Matriz de Confusión (Humano):")
print(confusion_matrix(y_true_compare, y_pred_human))

```

## Modelo de Regresión Ordinal

```

# Importaciones
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import mord

# 1) Configuración
# En esta sección definimos rutas y parámetros básicos.
CONFIG = {
    "MODEL_PATH": "/content/drive/My Drive/ordinal_model_params2.npz",
    "DATA_ITT_PATH": "/content/drive/My Drive/BASE_ITT.xlsx",
    "DATA_NEW2_PATH": "/content/drive/My Drive/CONJUNTO_PRUEBA.xlsx",
}

# Variables de entrada y sus categorías.
FEATURES = [
    "Cod Cie10",
    "Unidad Ads",

```

```

"Tip Ramo",
"first",
"Avg. Imp Salario Topado",
"Dias Probables Recuperacion",
"Max. Edad",
"veces_nivel3",
"transicion_12",
"transicion_13",
"transicion_23",
"total_transiciones",
"seq_length",
"count_1",
"count_2",
"Total de Camas Censables de la delegación_x_100000",
"Total de Consultorios de la Unidad_x_100000",
"Sala de Quirófano_x_100000",
"Servicio de Salud en el Trabajo_x_100000"
]

CAT_FEATURES = [
"Cod Cie10",
"Unidad Ads",
"Tip Ramo",
"first",
]

# Parámetros del modelo (se ajusta alpha con hyperparameter_search)
ORDINAL_PARAMS = {
    "alpha": 1.0,
}

# Aquí se definen todas las funciones necesarias para preprocesar, entrenar y evaluar.
def assign_target(weeks):
    """
    Convierte 'weeks' (semanas) en categoría ordinal (0,1,2,3).
    0 => <= 26 semanas
    1 => 27-52 semanas
    2 => 53-78 semanas
    3 => > 78 semanas
    """
    if weeks <= 26:
        return 0
    elif weeks <= 52:
        return 1
    elif weeks <= 78:
        return 2
    else:
        return 3

def ordinal_preprocess_df(df, require_target=False):
    """
    Preprocesa un DataFrame para regresión logística ordinal:
    - Crea 'Target' desde 'Semanas' si no existe y se requiere.
    - Reemplaza valores inf/NaN y llena con 0.
    - Codifica en one-hot las columnas de CAT_FEATURES.
    - Asegura la existencia de todas las columnas en FEATURES.
    """

```

```

"""
if require_target:
    if "Target" not in df.columns:
        if "Semanas" not in df.columns:
            raise ValueError("El DataFrame debe tener 'Semanas' para crear 'Target'")
        df["Target"] = df["Semanas"].apply(assign_target).astype(int)
    else:
        df["Target"] = df["Target"].astype(int)

df.replace([np.inf, -np.inf], np.nan, inplace=True)
df.fillna(0, inplace=True)

cat_cols_in_df = [c for c in CAT_FEATURES if c in df.columns]
df = pd.get_dummies(df, columns=cat_cols_in_df, drop_first=True)

for col in FEATURES:
    if col not in df.columns:
        df[col] = 0

if df.isna().any().any():
    raise ValueError("Se encontraron NaN después de fillna(0). Revisar datos.")
return df

def filter_feature_columns(X, features):
    """
    Retiene solo columnas de 'features' o derivadas de las variables categóricas (one-hot).
    """
    final_cols = []
    all_cols = list(X.columns)
    for col in all_cols:
        if col in features:
            final_cols.append(col)
        else:
            prefix = col.split('_')[0]
            if prefix in CAT_FEATURES:
                final_cols.append(col)
    return final_cols

def train_ordinal_model(df, features, ordinal_params, test_size=0.2):
    """
    Entrena un modelo mord.LogisticIT y retorna (modelo, X_test, y_test).
    Muestra métricas de validación sobre el conjunto de prueba.
    """
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y = df["Target"]

    final_cols = filter_feature_columns(X, features)
    X = X[final_cols]

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=test_size, random_state=42, stratify=y
    )

    model = mord.LogisticIT(**ordinal_params)
    model.fit(X_train, y_train)

```



```

model.classes_ = np.unique(y_train)

y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("Precisión inicial:", acc)
print("Reporte de Clasificación Inicial:\n", classification_report(y_test, y_pred, zero_division=0))
print("Matriz de Confusión Inicial:\n", confusion_matrix(y_test, y_pred))

return model, X_test, y_test

def save_ordinal_model(model, model_path, X_train=None):
    """
    Guarda el modelo (coef, theta, classes y columnas) en un archivo .npz.
    """
    if X_train is None:
        raise ValueError("X_train se requiere para guardar el orden de columnas.")

    np.savez_compressed(
        model_path,
        coef_=model.coef_,
        theta_=model.theta_,
        classes_=model.classes_,
        columns=X_train.columns.values
    )
    print(f"Modelo guardado en {model_path}")

def load_ordinal_model(model_path):
    """
    Carga un LogisticIT desde un archivo .npz (coef, theta, classes, columns).
    """
    data = np.load(model_path, allow_pickle=True)
    model = mord.LogisticIT()
    model.coef_ = data["coef_"]
    model.theta_ = data["theta_"]
    model.classes_ = data["classes_"]
    columns = data["columns"]
    print(f"Modelo cargado desde {model_path}")
    return model, columns

def evaluate_ordinal_model(model, df, model_cols):
    """
    Evalúa el modelo en un DataFrame con 'Target' y muestra precisión, reporte y matriz de
    confusión.
    Retorna la precisión.
    """
    df = ordinal_preprocess_df(df, require_target=True)
    X = df.drop(columns=["Target"], errors="ignore")
    y_true = df["Target"]

    X = X.reindex(columns=model_cols, fill_value=0)

    y_pred = model.predict(X)
    acc = accuracy_score(y_true, y_pred)
    print("\nPrecisión de la Evaluación:", acc)
    print("Reporte de Clasificación:\n", classification_report(y_true, y_pred, zero_division=0))
    print("Matriz de Confusión:\n", confusion_matrix(y_true, y_pred))

```

```

return acc

# 2) Realiza búsqueda de hiperparámetros (alpha) para mord.LogisticIT usando validación cruzada
def hyperparameter_search(df, features, alpha_values=None, n_splits=3):
    if alpha_values is None:
        alpha_values = [0.01, 0.1, 1.0, 10.0, 100.0]

    df_prep = ordinal_preprocess_df(df.copy(), require_target=True)
    X_all = df_prep.drop(columns=["Target"], errors="ignore")
    y_all = df_prep["Target"]

    final_cols = filter_feature_columns(X_all, features)
    X_all = X_all[final_cols]

    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
    best_alpha = None
    best_mean_acc = -1

    print("\n--- Ajuste de Hiperparámetros para alpha ---")
    for alpha in alpha_values:
        fold accuracies = []
        for train_index, valid_index in kf.split(X_all, y_all):
            X_train_cv, X_valid_cv = X_all.iloc[train_index], X_all.iloc[valid_index]
            y_train_cv, y_valid_cv = y_all.iloc[train_index], y_all.iloc[valid_index]

            model_cv = mord.LogisticIT(alpha=alpha)
            model_cv.fit(X_train_cv, y_train_cv)
            y_pred_cv = model_cv.predict(X_valid_cv)

            acc_cv = accuracy_score(y_valid_cv, y_pred_cv)
            fold accuracies.append(acc_cv)

        mean_acc = np.mean(fold accuracies)
        print(f"alpha={alpha}, precisión promedio={mean_acc:.4f}")

        if mean_acc > best_mean_acc:
            best_mean_acc = mean_acc
            best_alpha = alpha

    print(f"Mejor alpha encontrado: {best_alpha} (mean_acc={best_mean_acc:.4f})")
    return best_alpha

# 3) Flujo Principal
if __name__ == "__main__":
    # (a) Lectura y ajuste de hiperparámetros
    print("Leyendo dataset base (ITT)...")
    df_itt = pd.read_excel(CONFIG["DATA_ITT_PATH"])

    print("\nRealizando búsqueda de hiperparámetros (alpha)...")
    best_alpha = hyperparameter_search(
        df_itt.copy(),
        FEATURES,
        alpha_values=[0.01, 0.1, 1.0, 10.0, 50.0],
        n_splits=3
    )
    ORDINAL_PARAMS["alpha"] = best_alpha

```

```

# (b) Entrenamiento del modelo con ITT
print("\nEntrenando modelo Ordinal Logistic (base) con LogisticIT...")
base_model, X_test_base, y_test_base = train_ordinal_model(df_itt, FEATURES,
ORDINAL_PARAMS)

# (c) Guardado del modelo
save_ordinal_model(base_model, CONFIG["MODEL_PATH"], X_train=X_test_base)

# (d) Lectura de CONJUNTO_PRUEBA para evaluación final
print("\nLeyendo CONJUNTO_PRUEBA (dataset de prueba) y evaluando el modelo...")
newest_df = pd.read_excel(CONFIG["DATA_NEW2_PATH"])

# (e) Carga del modelo entrenado y evaluación en CONJUNTO_PRUEBA
loaded_model, loaded_cols = load_ordinal_model(CONFIG["MODEL_PATH"])
evaluate_ordinal_model(loaded_model, newest_df.copy(), loaded_cols)

# (f) Comparación de desempeño: Modelo vs. "Humano"
print("\n=== COMPARANDO DESEMPEÑO DEL MODELO VS. HUMANO (NUEVO2) ===")
newest_df_for_compare = newest_df.copy()
newest_df_for_compare = ordinal_preprocess_df(newest_df_for_compare, require_target=True)

y_true_compare = newest_df_for_compare["Target"]
X_compare = newest_df_for_compare.drop(columns=["Target"], errors="ignore")
X_compare_aligned = X_compare.reindex(columns=loaded_cols, fill_value=0)

# Predicción del modelo
y_pred_model = loaded_model.predict(X_compare_aligned)

# Predicción "humana": 'Dias Probables Recuperacion' en días => convertir a semanas
def assign_human_target(days):
    weeks = days / 7.0
    return assign_target(weeks)

y_pred_human = newest_df_for_compare["Dias Probables
Recuperacion"].apply(assign_human_target)

# (g) Cálculo de métricas
acc_model = accuracy_score(y_true_compare, y_pred_model)
acc_human = accuracy_score(y_true_compare, y_pred_human)

print("\n--- Resultados finales ---")
print(f"Precisión del Modelo: {acc_model:.4f}")
print(f"Precisión Humana: {acc_human:.4f}")

print("\n--- Reporte de Clasificación (Modelo) ---")
print(classification_report(y_true_compare, y_pred_model, zero_division=0))
print("Matriz de Confusión (Modelo):")
print(confusion_matrix(y_true_compare, y_pred_model))

print("\n--- Reporte de Clasificación (Humano) ---")
print(classification_report(y_true_compare, y_pred_human, zero_division=0))
print("Matriz de Confusión (Humano):")
print(confusion_matrix(y_true_compare, y_pred_human))

```

## Índice de términos

### “A”

Aprendizaje Automático, viii, x, xi, xii, xiv, xvi, 1, 3, 6, 10, 11, 26, 33, 39, 45, 46, xiii, xvi

### “E”

EncoderFrecuencia, vii, xi, xiv, 8, 13, 14  
enfermedad general, 12, 25, 27, 28, 45  
Escalado, xiii, xiv, 8, 15, 37  
Extracción de Características, xiv, xv

### “I”

IMSS, viii, xi, xiv, 1, 4, 5, 8, 12, 13, 14, 8, 10, 12, 18, 28, 29, 30, 31, 32, 46, 47, 48, 55, 56, 57, 66, xiv, xvi, xvii, xx  
Incapacidades Temporales, v, xiv, 1, 3, 7, 8, 9, 10, 12, 26, 28, 31, 46, xiii  
Infraestructura Hospitalaria, xii, xiv, xv, 1, 7, 13, 8, 10, 12, 22, 46, 49, 51, xv

### “N”

Normalización, xii, xiii, xiv, 8, xv

### “C”

Camas Censables, 13, 20, 21, 22, 23, 46, 49, 51, xv  
CatBoost, vi, viii, ix, xiv, 1, 23, 32, 42, 48, 49, 51, 53, 54, 55, 56, 58, 61, 62, 63, 64, 66, xiii, xiv, xxi  
Ciencia de Datos, v, x, 6, 11, 26, 33  
CRISP-DM, x, xiv, 2, 26, 47, 53, 56

### “F”

Frequency Encoding, 36, 51

### “M”

Modelos Predictivos, viii, 1, 8, 9, 10, 13, 26, 39, 41, xiv

### “O”

One Hot Encoding, xii, xiv, 8, 14, 36

## **“P”**

Preprocesamiento de Datos, xii, xvi, 1, 60

## **“S”**

Seguridad Social, v, xi, xiii, xiv, 1, 4, 11, 26, 27, 28, 30, 42, xiii, xvii

## **“R”**

Regresión Ordinal, vi, vii, xii, xv, 1, 34, 36, 46, 53, 54, 56, 58, 60, 62, 63, 67, 68, 69, 70, 71, 72, 73, xiv, xvii  
riesgo de trabajo, 25, 27, 28, 45, xiv

## **“T”**

Transformación Logarítmica, xiii, 15, 37