# Preparación y modelado de datos para algoritmos de machine learning

**Breve descripción:**

Este componente aborda las técnicas y metodologías para la preparación y modelado de datos en contextos de machine learning. Explora desde la construcción inicial de datasets hasta la preparación final para el entrenamiento de modelos, incluyendo el tratamiento de sesgos y estrategias de segmentación. Proporciona herramientas fundamentales para garantizar la calidad y efectividad de los datos en proyectos de IA.

**Noviembre 2024**

## Tabla de contenido

# Introducción

La preparación y modelado de datos constituye una fase determinante en el desarrollo de soluciones de machine learning. Los mejores algoritmos pueden fallar si los datos no están adecuadamente preparados, y las decisiones tomadas durante esta fase impactan directamente en el rendimiento y la confiabilidad de los modelos resultantes.

Este componente formativo aborda sistemáticamente las diferentes etapas de la preparación de datos, desde la construcción inicial de datasets hasta su preparación final para el entrenamiento de modelos. Se exploran técnicas avanzadas para el tratamiento de sesgos, estrategias efectivas de segmentación y métodos de preparación que optimizan el rendimiento de los modelos.

A lo largo del componente, se examinarán prácticas probadas y metodologías actuales que permiten transformar datos crudos en conjuntos de entrenamiento de alta calidad. El énfasis en la calidad de los datos y la eliminación de sesgos refleja la creciente importancia de la equidad y la responsabilidad en el desarrollo de sistemas de IA.

La combinación de fundamentos teóricos con aplicaciones prácticas proporciona las herramientas necesarias para abordar los desafíos reales en la preparación de datos para machine learning. Como dice un principio fundamental en ciencia de datos: "La calidad de un modelo nunca puede superar la calidad de los datos que lo alimentan".

¡Bienvenido a este viaje por las técnicas avanzadas de preparación de datos!

**Video 1.** Preparación y modelado de datos para algoritmos de machine learning


**Enlace de reproducción del video**

| **Síntesis del video: Preparación y modelado de datos para algoritmos de machine learning** |
| --- |
| La preparación efectiva de datos representa la base del éxito en cualquier proyecto de machine learning. Este componente te guiará a través de las técnicas y metodologías esenciales para transformar datos crudos en conjuntos óptimos para el entrenamiento de modelos.<br><br>Comenzaremos explorando la construcción de datasets, donde aprenderás a definir requerimientos claros y establecer controles de calidad robustos. La calidad en esta etapa inicial determina el potencial de todo el proyecto. |

El tratamiento de sesgos emerge como un aspecto fundamental. Descubrirás cómo identificar y corregir diferentes tipos de sesgos que podrían afectar la equidad y efectividad de tus modelos.

La segmentación de datos, incluyendo técnicas de validación cruzada y estrategias de muestreo, te permitirá evaluar adecuadamente el rendimiento de tus modelos y asegurar su capacidad de generalización.

Finalmente, abordaremos las técnicas específicas de preparación para modelos, incluyendo escalamiento, normalización y selección de características, elementos clave para optimizar el rendimiento de los algoritmos de machine learning.

Las tendencias actuales apuntan hacia una automatización creciente de estos procesos, pero el criterio experto sigue siendo esencial para tomar decisiones informadas en cada etapa.

Este componente te proporcionará las herramientas necesarias para preparar datos de manera efectiva, maximizando el potencial de tus modelos de machine learning.

¡Bienvenido al mundo de la preparación avanzada de datos!

# 1. Construcción de datasets

La construcción de datasets es el primer paso para cualquier proyecto de aprendizaje automático. Un dataset bien diseñado y construido es la base sobre la cual se desarrollan modelos robustos y precisos. En este capítulo, se abordará la importancia de los requerimientos y el diseño del dataset, se discutirán diversas técnicas de recolección de datos, y se analizarán los procesos necesarios para garantizar la calidad de los datos obtenidos. Estos elementos son clave para maximizar el valor de los modelos de machine learning, ya que cualquier error en esta fase puede tener consecuencias significativas en las etapas posteriores.

En el ámbito del Machine learning, la construcción de datasets robustos y representativos es fundamental para el éxito de cualquier proyecto. Un dataset, en esencia, es una colección organizada de datos que sirve como materia prima para entrenar y evaluar modelos de Inteligencia Artificial. Este capítulo explorará los aspectos clave en la construcción de datasets, desde la definición de requerimientos hasta las técnicas de recolección y control de calidad, con el objetivo de proporcionar al lector las herramientas necesarias para crear bases sólidas para sus proyectos de Machine learning.

## 1.1. Requerimientos y diseño

Antes de construir un dataset, es esencial establecer los requerimientos y diseñarlo adecuadamente. Esta fase consiste en definir cuál es el objetivo del dataset, qué tipo de información se necesita recolectar, y cuáles serán las características más relevantes. Para diseñar un buen dataset, se debe considerar el tipo de datos que se van a manejar: datos estructurados, semiestructurados, o no estructurados. Cada tipo tiene implicaciones diferentes en cuanto a su manipulación y almacenamiento.

Antes de iniciar la recolección de datos, es esencial definir con precisión los objetivos del proyecto de Machine learning. ¿Qué problema se busca resolver? ¿Qué tipo de modelo se utilizará? ¿Qué preguntas se intentarán responder con los datos? Las respuestas a estas interrogantes guiarán la definición de los requerimientos del dataset.

Aspectos por considerar en esta etapa:

- **Identificación de variables**

  Determinar las variables relevantes para el problema, incluyendo variables predictoras y variables objetivo.

- **Tipo de datos**

  Definir el tipo de datos que se recolectará (numérico, categórico, texto, imágenes, etc.) y su formato (estructurado, no estructurado o semiestructurado).

- **Tamaño del dataset**

  Estimar el tamaño de la muestra necesaria para obtener resultados significativos, considerando la complejidad del problema y el algoritmo de Machine learning a utilizar.

- **Fuente de datos**

  Identificar las fuentes de donde se obtendrán los datos, ya sean bases de datos existentes, APIs, web scraping, encuestas, sensores, etc.

Además, el diseño del dataset incluye determinar la fuente de los datos: ¿provendrán de archivos CSV, bases de datos relacionales, o APIs externas? Definir cómo se capturarán los datos ayuda a garantizar la consistencia y disponibilidad necesaria para el modelado. En esta fase también se debe planificar cómo se realizará

la transformación de datos y si se necesitan datos adicionales para enriquecer el conjunto. La selección de estas estrategias impactará directamente en la calidad del dataset y, por ende, en el rendimiento del modelo.

Un diseño bien definido del dataset asegurará la calidad de los datos y facilitará las etapas posteriores del proceso de Machine learning.

## 1.2. Técnicas de recolección

Una vez que se ha diseñado el dataset, se debe proceder a la recolección de datos. Existen varias técnicas de recolección que se utilizan dependiendo del contexto y del tipo de datos que se necesiten. Algunas de las más comunes incluyen:

- **Web scraping**

  Utilizado para recolectar datos de sitios web de manera automatizada. Este enfoque es útil para la recolección de datos no estructurados.

- **Consultas a bases de datos**

  Utilizar consultas SQL o APIs de bases de datos relacionales permite la recolección de datos estructurados. Esta técnica es eficiente para acceder a información ya organizada.

- **Formularios de entrada**

  La recolección de datos directamente desde usuarios mediante formularios es particularmente útil para obtener información específica y controlada.

- **APIs**

  Acceder a datos a través de interfaces de programación de aplicaciones (APIs) proporcionadas por diferentes plataformas y servicios.

- **Encuestas**

  Diseñar y administrar cuestionarios para recopilar información directamente de los individuos.

- **Sensores**

  Utilizar dispositivos que capturan datos del entorno, como temperatura, humedad, ubicación, etc.

Cada una de estas técnicas tiene sus ventajas y desventajas, y la elección dependerá de los objetivos del proyecto y de los recursos disponibles. Para facilitar la comprensión de las distintas técnicas, se presenta la siguiente tabla que resume algunas de sus principales características:

**Tabla 1.** Técnicas de recolección de datos

| Técnica de Recolección | Tipo de Datos | Ventajas | Desventajas | Herramientas |
|---|---|---|---|---|
| Web Scraping. | No estructurados. | Gran cantidad de datos. | Problemas legales y de calidad. | Beautiful Soup, Scrapy, Selenium. |
| Consultas a Bases de Datos. | Estructurados. | Rápido y consistente. | Limitado por la estructura. | SQL, R, Python. |
| Formularios de Entrada. | Estructurados. | Datos específicos y controlados. | Requiere participación activa. | Google Forms, Survey Monkey. |
| APIs. | Estructurados. | Acceso eficiente a datos. | Dependencia del proveedor. | Librerías de cliente para APIs en R y Python. |
| Sensores. | Estructurados. | Datos en tiempo real. | Requiere infraestructura física. | Dispositivos IoT, Arduino, Raspberry Pi. |

Fuente. OIT, 2024.

Esta tabla muestra las distintas opciones de recolección disponibles, lo que permite elegir la más adecuada según las necesidades del proyecto.

## 1.3. Control de calidad

Una vez recolectados los datos, es fundamental realizar un control de calidad para asegurar que los mismos sean adecuados para el entrenamiento de modelos de machine learning. La calidad de los datos tiene un impacto directo en el rendimiento de los modelos, por lo que cualquier error o inconsistencias pueden resultar en predicciones erróneas.

El control de calidad implica la identificación y eliminación de datos faltantes, valores duplicados o fuera de rango, y errores tipográficos. Además, es importante verificar la consistencia de los datos, asegurándose de que las unidades de medida sean correctas y uniformes. Otro aspecto importante del control de calidad es la detección de outliers, ya que estos pueden afectar negativamente la capacidad del modelo para generalizar los patrones.

Algunas tareas comunes en el control de calidad de datos incluyen:

- **Limpieza de datos**

  Eliminar o corregir datos erróneos, faltantes o duplicados.

- **Validación de datos**

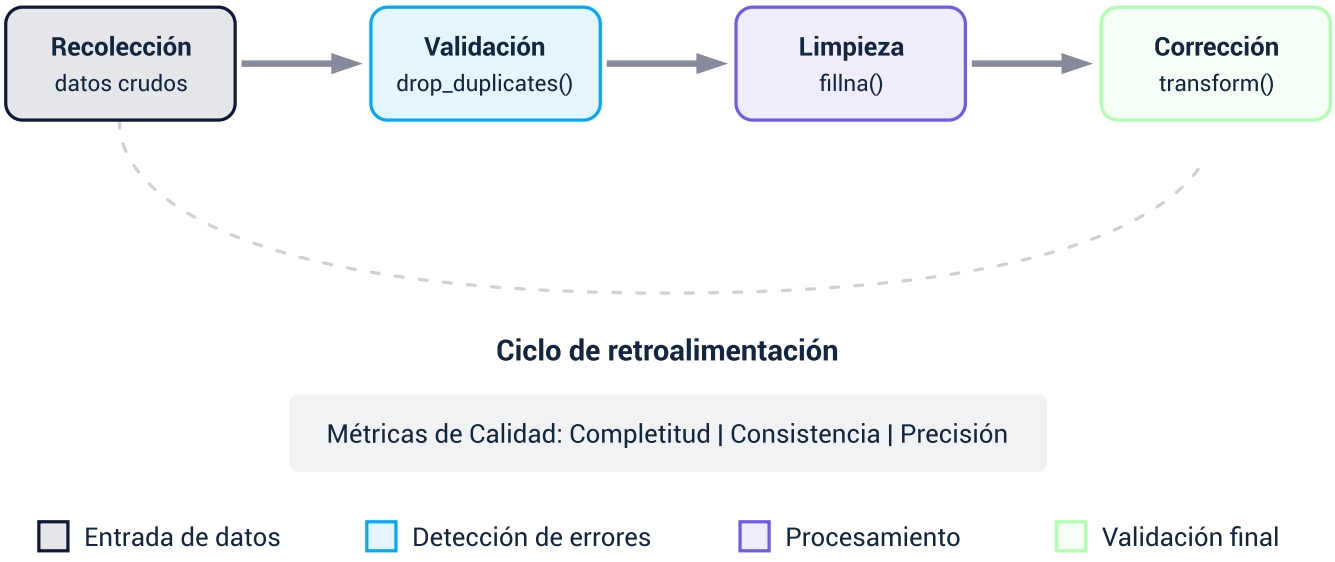
  Verificar que los datos cumplan con los criterios de validez definidos en la etapa de diseño.

- **Transformación de datos**

  Convertir los datos a un formato adecuado para el análisis, como la codificación de variables categóricas o la normalización de datos numéricos.

Para la ejecución del control de calidad, se pueden utilizar herramientas y bibliotecas como Pandas en Python, que permiten realizar operaciones de limpieza y transformación de manera eficiente. Por ejemplo, la eliminación de datos duplicados se realiza con funciones como drop_duplicates(), mientras que los datos faltantes pueden ser gestionados con fillna() para imputar valores. A continuación, se presenta un ejemplo visual que ilustra cómo es el proceso de control de calidad en un dataset, destacando las etapas desde la detección hasta la corrección de los errores:

**Figura 1.** Flujo de control de calidad de datos




Fuente. OIT, 2024.

La figura muestra un diagrama de flujo que empieza con la recolección de datos, seguido de pasos como validación, limpieza, corrección y, finalmente, aprobación de los datos para su uso en el modelado. Este proceso cíclico asegura que los datos finales sean óptimos para el desarrollo del modelo.

El monitoreo continuo de la calidad de datos debe incluir tanto verificaciones automatizadas como revisiones manuales periódicas. Las métricas de calidad deben definirse claramente y monitorearse de manera consistente. Algunos ejemplos comunes incluyen la completitud de los datos, la consistencia entre diferentes fuentes, y la adherencia a los formatos esperados.

Al concluir este capítulo, se debe enfatizar que la construcción del dataset no es solo una tarea técnica, sino una fase crítica que determina el éxito de los modelos de machine learning. Un enfoque riguroso en el diseño, la recolección y el control de calidad de los datos resultará en un conjunto de datos que pueda ser confiablemente usado para entrenamiento y predicción, minimizando así los riesgos de error en etapas posteriores.

# 2. Tratamiento de sesgos

El tratamiento de sesgos en los datasets es una etapa determinante para garantizar que los modelos de machine learning sean equitativos y útiles en la práctica. Los sesgos en los datos pueden llevar a modelos que discriminan contra ciertos grupos o generan resultados inexactos, lo cual tiene un impacto negativo en la calidad del aprendizaje automático y en su aplicabilidad. En este capítulo, se abordarán los diferentes tipos de sesgos que pueden surgir, las técnicas para detectarlos y los métodos disponibles para corregirlos. Estos elementos son esenciales para garantizar que los modelos se comporten de manera ética y precisa.

## 2.1. Tipos de sesgos

El sesgo en los datasets se refiere a una tendencia sistemática que afecta la calidad de los datos y, por ende, la capacidad de los modelos para generalizar. A continuación, se presentan algunos de los tipos más comunes de sesgos:

- **Sesgo de selección**

  Ocurre cuando el conjunto de datos utilizado no representa adecuadamente a la población de interés. Esto puede deberse a un muestreo deficiente o a la falta de diversidad en los datos recolectados.

- **Sesgo de medición**

  Este tipo de sesgo se da cuando existen errores en la forma en que se mide o se recolectan los datos. Puede ser causado por herramientas de medición defectuosas o inconsistencias en la recolección.

- **Sesgo de confirmación**

  Surge cuando los datos recolectados favorecen una hipótesis preexistente. Esto suele ocurrir cuando se tiene una predisposición hacia ciertos resultados durante la etapa de recolección.

- **Sesgo implícito**

  A menudo surge debido a creencias y suposiciones inconscientes que afectan el proceso de recolección y etiquetado de los datos.

- **Sesgo histórico**

  Refleja prejuicios o desigualdades presentes en los datos históricos. Por ejemplo, si entrenamos un modelo de selección de personal con datos históricos de una industria tradicionalmente dominada por un género, el modelo podría perpetuar estos patrones discriminatorios.

- **Sesgo de exclusión**

  Se presenta al omitir variables relevantes en el análisis, lo que puede llevar a conclusiones erróneas.

- **Sesgo algorítmico**

  Se refiere a los sesgos introducidos por el propio algoritmo de machine learning, que puede amplificar los sesgos presentes en los datos de entrenamiento.

Comprender y detectar estos tipos de sesgos es decisivo para evitar que los modelos de machine learning produzcan resultados inexactos. La presencia de sesgos puede impactar negativamente la utilidad del modelo y disminuir su capacidad para generalizar a nuevos datos.

## 2.2. Técnicas de detección

La detección de sesgos en los datos es necesaria para garantizar que los modelos desarrollados sean útiles. Algunas de las técnicas para detectar sesgos en los datasets incluyen:

- **Análisis descriptivo**

  Consiste en explorar las estadísticas básicas de las variables del dataset, como la media, la desviación estándar y las distribuciones, para detectar cualquier irregularidad o patrón inusual.

- **Visualización de datos**

  Herramientas de visualización, como gráficos de barras, diagramas de dispersión y mapas de calor, pueden ser útiles para identificar patrones de sesgo en los datos. Por ejemplo, si una clase está desproporcionadamente representada, esto podría indicar un sesgo de selección.

- **Pruebas de hipótesis**

  Realizar pruebas estadísticas puede ayudar a identificar si ciertas diferencias en los datos son significativas y podrían deberse a un sesgo subyacente.

- **Evaluación de representatividad**

  Comparar la distribución del dataset con la población general puede ayudar a identificar si ciertos grupos están sub o sobre-representados.

- **Análisis de correlación**

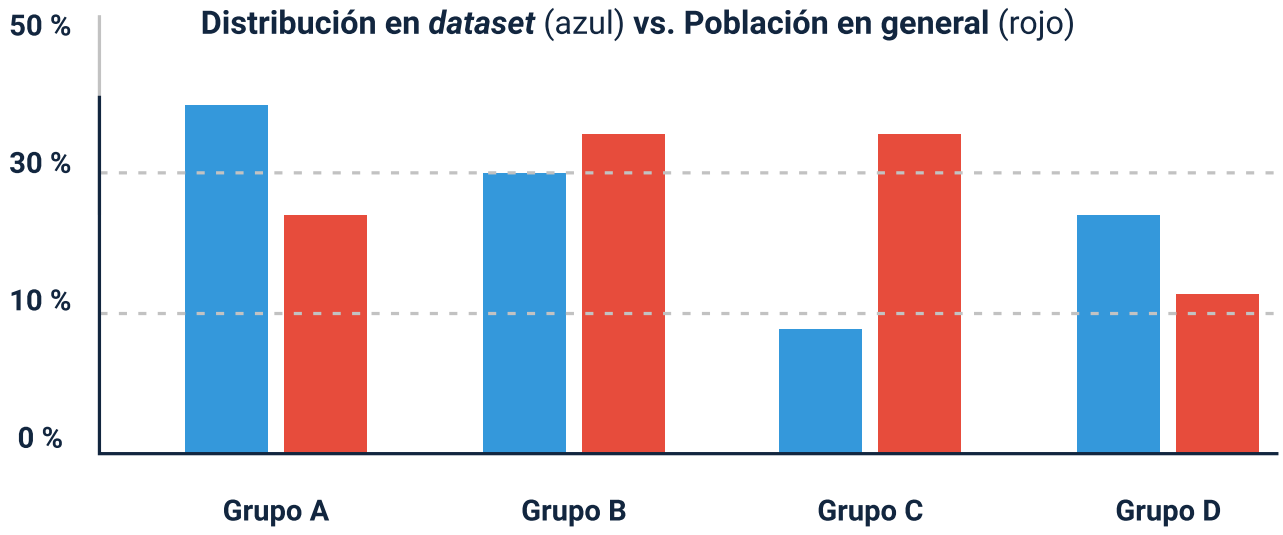
  Examinar la correlación entre las variables para identificar relaciones que puedan indicar la presencia de sesgos.

- **Validación cruzada**

  Utilizar validación cruzada y analizar el rendimiento del modelo para detectar si existe un sobreajuste en ciertos grupos.

Estas técnicas permiten identificar áreas donde los sesgos podrían estar presentes y proporcionan una base para el análisis y la corrección de estos. Para ilustrar cómo se puede detectar visualmente el sesgo de selección, consideremos el siguiente gráfico que compara las distribuciones entre el dataset y la población general:

**Figura 2.** Ejemplo de visualización para detección de sesgo de selección




Fuente. OIT, 2024.

Como se puede observar en el gráfico, existen discrepancias notables entre la distribución de grupos en el dataset (barras azules) y la población general (barras rojas). El Grupo C muestra la mayor disparidad, estando significativamente subrepresentado en el dataset en comparación con su presencia en la población general. Por otro lado, el Grupo A está sobrerrepresentado. Estas diferencias son

indicativas de un sesgo de selección que necesita ser abordado mediante técnicas de remuestreo o ponderación para asegurar que el dataset sea verdaderamente representativo.

## 2.3. Métodos de corrección

Una vez identificados los sesgos, se deben implementar métodos de corrección para garantizar que el modelo sea justo y representativo. Algunos de los métodos más comunes incluyen:

- **Recolección adicional de datos**

  En caso de que se detecte un sesgo de selección, se pueden recolectar más datos para garantizar que todas las clases o grupos estén debidamente representados.

- **Reponderación**

  Este método implica asignar diferentes pesos a las observaciones para corregir los desequilibrios en la representación de clases. Por ejemplo, los datos de una clase poco representada podrían tener un mayor peso durante el entrenamiento del modelo.

- **Sobremuestreo y submuestreo**

  El sobremuestreo consiste en replicar instancias de clases minoritarias, mientras que el submuestreo implica reducir la cantidad de datos de clases mayoritarias. Ambos métodos ayudan a equilibrar el dataset y minimizar el sesgo.

- **Normalización de características**

  En el caso del sesgo de medición, se pueden aplicar técnicas de normalización y estandarización para garantizar que las características

estén en escalas comparables, reduciendo así el impacto de mediciones incorrectas.

- **Algoritmos robustecidos frente a sesgos**

  Modificar los algoritmos de aprendizaje para que sean más robustos frente a sesgos, incorporando términos de regularización que penalicen decisiones sesgadas o modificando las funciones de pérdida.

- **Post-procesamiento de predicciones**

  Ajustar las predicciones del modelo para reducir disparidades identificadas, como calibrar probabilidades o ajustar umbrales de decisión para diferentes grupos.

- **Ponderación de casos**

  Ajustar la importancia relativa de diferentes grupos dentro del conjunto de datos, lo cual permite que el modelo no sobrevalore clases mayoritarias.

La corrección de sesgos debe realizarse con cautela, ya que métodos demasiado agresivos pueden introducir nuevos sesgos o distorsionar relaciones importantes en los datos. El objetivo no es eliminar completamente las diferencias en los datos, sino asegurar que estas diferencias reflejen variaciones reales y no prejuicios sistemáticos.

A continuación, se presenta una tabla que resume los métodos de corrección de sesgos según el tipo de sesgo identificado:

**Tabla 2.** Sesgos y métodos de corrección

| Tipo de sesgo | Método de corrección | Descripción |
|---|---|---|
| Sesgo de selección. | Recolección adicional de datos. | Recolectar más datos para representar a todos los grupos. |

| Tipo de sesgo | Método de corrección | Descripción |
|---|---|---|
| Sesgo de medición. | Normalización de características. | Ajustar las mediciones para reducir errores. |
| Sesgo de confirmación. | Reponderación. | Ajustar los pesos de las observaciones. |
| Sesgo implícito. | Sobremuestreo y submuestreo. | Balancear las clases para minimizar desigualdades. |
| Sesgo histórico. | Algoritmos robustecidos frente a sesgos. | Modificar los algoritmos para reducir el impacto de los sesgos históricos. |
| Sesgo de exclusión. | Recopilación de variables relevantes. | Incluir variables que permitan explicar mejor los datos. |

Fuente. OIT, 2024.

El tratamiento adecuado de los sesgos es esencial para garantizar que los modelos de machine learning sean equitativos y confiables. Sin un tratamiento adecuado, los modelos pueden perpetuar injusticias y proporcionar resultados que no sean útiles en la práctica.

Al concluir este capítulo, se debe enfatizar que el tratamiento de sesgos no solo mejora la precisión del modelo, sino que también es una responsabilidad ética. Las personas formadas en machine learning deben tener en cuenta estos aspectos para construir modelos que sean justos y que proporcionen valor real en contextos del mundo real.

# 3. Segmentación de datos

La segmentación de datos es un paso clave en el proceso de preparación de datasets para el aprendizaje automático. Dividir los datos en diferentes subconjuntos permite evaluar el rendimiento del modelo de manera efectiva y garantizar su capacidad para generalizar a datos no vistos. En este capítulo, se abordarán los conceptos de conjuntos de entrenamiento, prueba y validación, la importancia de la validación cruzada, y las estrategias de muestreo que aseguran una adecuada distribución de los datos.

## 3.1. Conjuntos de entrenamiento y prueba

La correcta división del dataset en conjuntos de entrenamiento y prueba es esencial para evaluar adecuadamente el rendimiento del modelo de machine learning. En general, el dataset se divide en dos o más subconjuntos:

- **Conjunto de entrenamiento**

  Es la porción del dataset que se utiliza para entrenar el modelo. El objetivo es proporcionar al modelo suficientes ejemplos para aprender patrones relevantes y generalizables.

- **Conjunto de prueba**

  Este conjunto se mantiene separado del proceso de entrenamiento y se utiliza para evaluar la capacidad del modelo para generalizar a nuevos datos. El conjunto de prueba permite medir el rendimiento real del modelo y evitar el sobreajuste.

Una buena práctica es destinar aproximadamente un 70-80 % de los datos para el entrenamiento y el 20-30 % restante para la prueba. Sin embargo, estas proporciones pueden ajustarse según el tamaño del dataset y la naturaleza del problema.

Es importante asegurarse de que tanto el conjunto de entrenamiento como el conjunto de prueba sean representativos de la población de datos, evitando así que el modelo aprenda patrones específicos del conjunto de entrenamiento que no se replican en el conjunto de prueba.

La división de datos en conjuntos de entrenamiento y prueba representa el primer paso en la evaluación sistemática de modelos de machine learning. Esta separación permite simular cómo se comportará el modelo ante datos que nunca ha visto, proporcionando una estimación más realista de su rendimiento en el mundo real. Para comprender mejor las implicaciones de diferentes estrategias de división de datos, consideremos las siguientes proporciones comúnmente utilizadas y sus casos de uso:

**Tabla 3.** Proporciones empleadas en entrenamiento y pruebas

| Estrategia de división | Proporción (Train/Test) | Casos de uso | Consideraciones especiales |
|---|---|---|---|
| Clásica. | 80/20. | Datasets grandes (>10,000 muestras). | Balance entre representatividad y evaluación. |
| Conservadora. | 90/10. | Datasets muy grandes (>100,000 muestras). | Maximiza datos de entrenamiento. |
| Equilibrada. | 70/30. | Datasets medianos (1,000-10,000 muestras). | Mayor confianza en la evaluación. |

| Estrategia de división | Proporción (Train/Test) | Casos de uso | Consideraciones especiales |
|---|---|---|---|
| Proporcional. | 60/40. | Datasets pequeños (<1,000 muestras). | Evita sobreajuste en muestras limitadas. |
| Específica del dominio. | Variable. | Casos con restricciones temporales o secuenciales. | Respeta la estructura temporal de los datos. |

Fuente. OIT, 2024.

La elección de la proporción adecuada depende no solo del tamaño del dataset, sino también de factores como la complejidad del problema, la variabilidad en los datos y los requisitos específicos del proyecto. En algunos casos, puede ser necesario ajustar estas proporciones para garantizar que ambos conjuntos contengan muestras representativas de todas las clases o categorías importantes.

## 3.2. Validación cruzada

La validación cruzada es una técnica que se utiliza para evaluar el rendimiento de un modelo de aprendizaje automático de manera más robusta. Consiste en dividir el dataset en múltiples subconjuntos o "folds" y entrenar el modelo varias veces, cada vez utilizando uno de los subconjuntos como conjunto de prueba y los demás como conjunto de entrenamiento.

- **K-Fold Cross-Validation**

  Esta es una de las formas más comunes de validación cruzada. El dataset se divide en "k" partes (folds) y se entrena el modelo "k" veces, cada vez utilizando un fold diferente como conjunto de prueba y los restantes como conjunto de entrenamiento. El rendimiento se promedia sobre todas las

iteraciones, proporcionando una medida más precisa de la capacidad del modelo para generalizar.

- **Leave-One-Out Cross-Validation (LOOCV)**

  En este enfoque, cada observación del dataset se utiliza una vez como conjunto de prueba, mientras que el resto se utiliza para el entrenamiento. Aunque es una técnica muy precisa, puede ser computacionalmente costosa para datasets grandes.

La validación cruzada permite reducir la varianza en la estimación del rendimiento del modelo y asegurar que se está utilizando toda la información disponible en el dataset de manera eficiente. Esto es especialmente útil cuando se trabaja con datasets de tamaño limitado.

## 3.3. Estrategias de muestreo

Las estrategias de muestreo son fundamentales para garantizar que los conjuntos de entrenamiento, prueba y validación sean representativos de la población general. Algunas de las estrategias más comunes incluyen:

- **Muestreo aleatorio simple**

  Cada observación tiene la misma probabilidad de ser seleccionada. Esta es la técnica más básica y es adecuada cuando el dataset es lo suficientemente grande y no existen problemas de desequilibrio entre clases.

- **Muestreo estratificado**

  Se asegura de que la distribución de clases o categorías en los subconjuntos sea similar a la del dataset original. Esto es especialmente

importante cuando se trabaja con datasets desbalanceados, ya que garantiza que todas las clases estén representadas adecuadamente en cada subconjunto.

- **Muestreo con reemplazo**

  En este tipo de muestreo, una observación seleccionada se devuelve al dataset y puede ser seleccionada nuevamente. Aunque no es muy común en la segmentación de datasets para machine learning, puede ser útil en ciertos enfoques como el bootstrap.

A continuación, se presenta una tabla que resume las principales estrategias de muestreo y sus características:

**Tabla 4.** Estrategias de muestreo

| Estrategia de Muestreo | Descripción | Ventajas | Desventajas |
|---|---|---|---|
| Muestreo aleatorio simple. | Cada observación tiene la misma probabilidad de ser seleccionada. | Fácil de implementar y comprender. | No garantiza representatividad en datasets desbalanceados. |
| Muestreo estratificado. | Asegura que la distribución de clases sea similar en cada subconjunto. | Útil para datasets desbalanceados. | Puede ser complejo de implementar. |
| Muestreo con reemplazo. | Las observaciones seleccionadas se devuelven al dataset. | Útil para enfoques de bootstrap. | Puede introducir duplicados innecesarios. |

Fuente. OIT, 2024.

Estas estrategias permiten asegurar una correcta división del dataset y garantizar que los modelos entrenados tengan una buena capacidad para generalizar a datos no vistos, evitando problemas de sobreajuste o subajuste.

Al concluir este capítulo, se debe resaltar que una segmentación adecuada de los datos es fundamental para evaluar correctamente el rendimiento de los modelos de machine learning. Dividir los datos de manera estratégica permite obtener una visión precisa de cómo se comportará el modelo en situaciones reales y garantiza que se está maximizando el valor del dataset disponible. La validación cruzada y el uso adecuado de estrategias de muestreo son prácticas recomendadas para asegurar que los resultados obtenidos sean confiables y representativos del comportamiento esperado del modelo en producción.

# 4. Preparación para modelos

La fase final en el procesamiento de datos antes del entrenamiento de modelos de machine learning requiere una serie de transformaciones específicas que optimicen el rendimiento del algoritmo. Esta etapa resulta determinante para el éxito del modelo, pues los algoritmos de aprendizaje automático son sensibles a la escala y formato de los datos de entrada.

## 4.1. Escalamiento y normalización

El escalamiento y normalización de datos constituyen transformaciones matemáticas que ajustan los valores numéricos a rangos específicos, facilitando el proceso de aprendizaje del modelo. Estas técnicas cobran especial relevancia cuando las variables presentan escalas muy diferentes entre sí.

La normalización min-max ajusta los valores a un rango específico, típicamente entre 0 y 1, preservando las relaciones entre los datos originales. Por otro lado, la estandarización (o normalización Z-score) transforma los datos para que tengan media cero y desviación estándar unitaria. La elección entre estas técnicas depende del algoritmo y la naturaleza de los datos. La siguiente tabla presenta una comparación detallada de las principales técnicas de escalamiento y sus casos de uso:

**Tabla 5.** Técnicas de escalamiento

| Técnica | Fórmula | Rango resultante | Casos de uso recomendados | Consideraciones |
|---|---|---|---|---|
| Min-Max. | (x - min)/(max - min). | [0,1] | Redes neuronales, algoritmos basados en distancias. | Sensible a outliers. |

| Técnica | Fórmula | Rango resultante | Casos de uso recomendados | Consideraciones |
|---|---|---|---|---|
| Z-Score. | (x - media)/desv.est. | [-∞,∞] | Regresión lineal, SVM. | Asume distribución normal. |
| Robust Scaler. | (x - mediana)/IQR. | Variable. | Datos con outliers significativos. | Más robusto a valores extremos. |
| Log Transform. | log(x) | [0,∞] | Datos con distribución sesgada. | Solo para valores positivos. |

Fuente. OIT, 2024.

## 4.2. Codificación de variables

La codificación de variables categóricas representa un paso esencial para convertir datos cualitativos en formatos numéricos que los algoritmos puedan procesar. Esta transformación debe realizarse cuidadosamente para preservar la información semántica contenida en las categorías originales.

La codificación one-hot transforma cada categoría en una columna binaria, evitando la imposición de un orden artificial entre categorías. Sin embargo, puede generar matrices dispersas cuando el número de categorías es elevado. La codificación ordinal, por su parte, asigna números enteros a cada categoría y resulta más apropiada cuando existe un orden natural entre las categorías.

En casos de variables categóricas con alta cardinalidad (muchas categorías únicas), técnicas más avanzadas como target encoding o feature hashing pueden ofrecer alternativas más eficientes. Estas técnicas deben aplicarse con precaución para evitar el sobreajuste, especialmente en conjuntos de datos pequeños.

## 4.3. Selección de características

La selección de características implica identificar el subconjunto más relevante de variables para el modelo, reduciendo la dimensionalidad del problema sin perder información significativa. Este proceso no solo mejora el rendimiento computacional sino que también puede aumentar la capacidad de generalización del modelo.

Los métodos de selección de características se pueden clasificar en tres categorías principales:

- **Métodos de filtro**

  Evalúan las características de manera independiente del algoritmo de aprendizaje, utilizando métricas estadísticas como correlación o información mutua. Estos métodos son computacionalmente eficientes pero pueden pasar por alto interacciones complejas entre variables.

- **Métodos wrapper**

  Utilizan el rendimiento del modelo como criterio de selección, evaluando diferentes subconjuntos de características. Aunque más precisos, resultan computacionalmente intensivos para datasets con muchas variables.

- **Métodos embebidos**

  Realizan la selección de características como parte del proceso de entrenamiento del modelo, como ocurre con la regularización Lasso o los árboles de decisión.

Estas estrategias permiten asegurar una correcta división del dataset y garantizar que los modelos entrenados tengan una buena capacidad para generalizar a datos no vistos, evitando problemas de sobreajuste o subajuste.

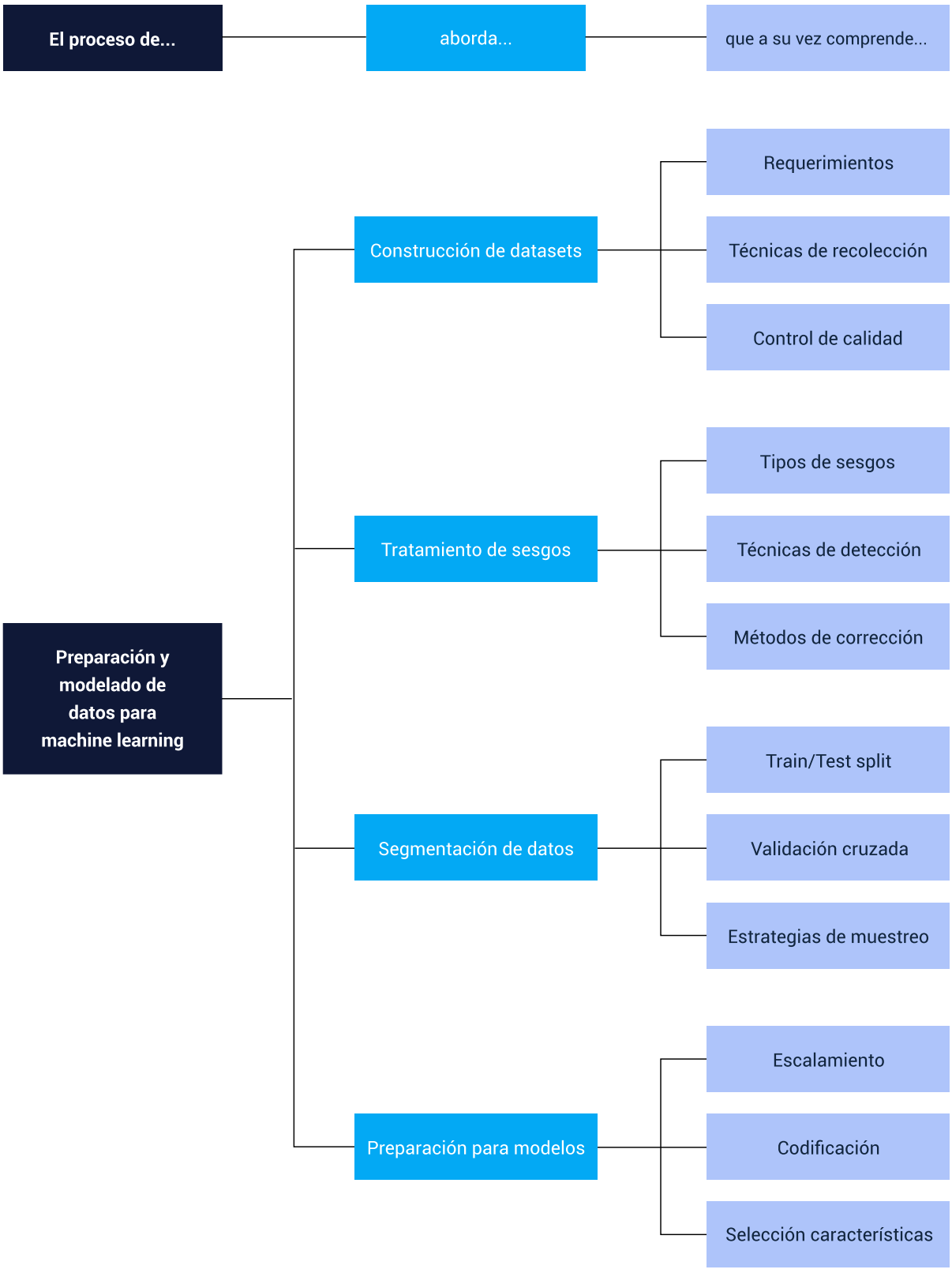
La validación cruzada juega un papel fundamental en la selección de características, ayudando a identificar aquellas que contribuyen consistentemente al rendimiento del modelo a través de diferentes subconjuntos de datos. Este proceso iterativo asegura que las características seleccionadas sean verdaderamente robustas y no dependen de particularidades de una partición específica de los datos.

## Síntesis

El diagrama representa la estructura integral del componente formativo sobre preparación y modelado de datos para machine learning. Partiendo del concepto central de preparación y modelado, se ramifica en cuatro áreas esenciales: construcción de datasets, tratamiento de sesgos, segmentación de datos y preparación para modelos. Cada área incorpora subtemas específicos que constituyen los elementos fundamentales para una preparación efectiva de datos en contextos de IA.

Esta organización ilustra el flujo natural del proceso de preparación de datos, desde su recolección inicial hasta su optimización final para el entrenamiento de modelos. La interrelación entre las diferentes áreas muestra cómo cada etapa construye sobre la anterior, creando un proceso cohesivo que asegura la calidad y efectividad de los datos para machine learning.

El diagrama funciona como una hoja de ruta visual para comprender la estructura y alcance del componente, permitiendo al aprendiz visualizar rápidamente la progresión del aprendizaje y las conexiones entre los diferentes temas. Se sugiere utilizarlo como referencia para organizar el estudio y comprender la integración de los diversos aspectos de la preparación de datos para machine learning.




Fuente. OIT, 2024.

## Material complementario

| Tema | Referencia | Tipo de material | Enlace del recurso |
|---|---|---|---|
| El algoritmo ideal | Ecosistema de Recursos Educativos Digitales SENA. (2023c, octubre 10). El algoritmo ideal. | Video | https://www.youtube.com/watch?v=ZgkwSKyGpnY |
| ¿Qué es Machine learning? | Ecosistema de Recursos Educativos Digitales SENA. (2020, 13 septiembre). ¿Qué es Machine learning? | Video | https://www.youtube.com/watch?v=J9w6KquPKbE |
| Introducción al Machine learning | Ecosistema de Recursos Educativos Digitales SENA. (2023a, septiembre 15). Introducción al machine learning. | Video | https://www.youtube.com/watch?v=xwjQmGJ3q0I |
| Machine learning con Python | Ecosistema de Recursos Educativos Digitales SENA. (2023c, octubre 10). Machine learning con Python. | Video | https://www.youtube.com/watch?v=noMy4-zjR9Q |
| Modelos y metodologías de analítica | Ecosistema de Recursos Educativos Digitales SENA. (2023e, marzo 27). Modelos y metodologías de analítica. | Video | https://www.youtube.com/watch?v=96pohadjEWE |
| Análisis exploratorio de datos | Limpiar datos de Excel, CSV, PDF y Hojas de cálculo de Google con el intérprete de datos. (s. f.). Tableau. | Portal web | https://help.tableau.com/current/pro/desktop/es-es/data_interpreter.htm |
| Definición de pandas | Ecosistema de Recursos Educativos Digitales SENA. (2023b, septiembre 20). Definición de pandas. | Video | https://www.youtube.com/watch?v=W48LYsToQHQ |

## Glosario

**Bias (Sesgo)**: desviación sistemática en los datos que puede llevar a resultados injustos o poco representativos en los modelos de machine learning.

**Cross-validation**: técnica de evaluación que divide los datos en múltiples subconjuntos para validar el rendimiento del modelo de manera más robusta.

**Data splitting**: proceso de dividir un conjunto de datos en subconjuntos para entrenamiento, validación y prueba de modelos.

**Dataset**: conjunto estructurado de datos organizados para su uso en entrenamiento y evaluación de modelos de machine learning.

**Feature engineering**: proceso de selección, creación y transformación de variables para optimizar el rendimiento de modelos de machine learning.

**Feature scaling**: proceso de normalizar o estandarizar variables numéricas para que estén en rangos comparables.

**Hold-out set**: conjunto de datos reservado para la evaluación final del modelo, que no se utiliza durante el entrenamiento.

**Imbalanced data**: situación donde las clases en un conjunto de datos no están representadas equitativamente.

**Missing values**: datos faltantes en un conjunto de datos que requieren tratamiento especial antes del modelado.

**Normalización**: técnica para ajustar valores numéricos a una escala común, típicamente entre 0 y 1.

**One-hot encoding**: técnica para convertir variables categóricas en formato binario para su uso en modelos de machine learning.

**Outliers**: valores atípicos que se desvían significativamente del patrón general de los datos.

**Sampling strategy**: método utilizado para seleccionar subconjuntos representativos de datos para entrenamiento y validación.

**Selection bias**: sesgo que ocurre cuando la selección de datos no es aleatoria o representativa de la población objetivo.

**Standardization**: proceso de transformar variables para que tengan media cero y desviación estándar unitaria.

**Test set**: conjunto de datos utilizado para evaluar el rendimiento final del modelo entrenado.

**Training set**: conjunto de datos utilizado para entrenar el modelo de machine learning.

**Validation set**: conjunto de datos utilizado para ajustar hiperparámetros y evaluar el modelo durante el desarrollo.

**Variable encoding**: proceso de convertir variables categóricas en formato numérico para su uso en modelos.

**Z-score**: medida estadística que indica cuántas desviaciones estándar se aleja un valor de la media.

## Referencias bibliográficas


Alfonso Jaramillo, M. C. (2019). Consideraciones para el diseño de instrumentos de recolección de datos. Recuperado de https://acei.co/wp-content/uploads/2019/10/Webinar-ACEI-Dise%C3%B1o-intrumentos-para-recolecci%C3%B3n-de-datos.pdf

ATLAS.ti. (s.f.). Guía exhaustiva sobre el sesgo en la investigación. Recuperado de https://atlasti.com/es/guias/guia-investigacion-cualitativa-parte-1/sesgo

Cepal. (2020). Recomendaciones para eliminar el sesgo de selección en las encuestas de hogares en el contexto de la pandemia de COVID-19. Recuperado de https://repositorio.cepal.org/bitstream/handle/11362/45552/1/S2000316_es.pdf

Emanuelli, P. B. (s.f.). Selección de técnicas: Técnicas de recolección de datos. Criterios para la selección de técnicas de recolección de datos. Recuperado de https://www.academia.edu/30424818/CAP_1_SELECCI%C3%93N_DE_T%C3%89CNICAS_T%C3%89CNICAS_DE_RECOLECCI%C3%93N_DE_DATOS_CRITERIOS_PARA_LA_SELECCI%C3%93N_DE_T%C3%89CNICAS_DE_RECOLECCI%C3%93N_DE_DATOS

Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC). (2020). Norma Técnica de la Calidad del Proceso Estadístico (NTC PE 1000:2020). Recuperado de https://www.funcionpublica.gov.co/documents/34645357/0/NTC%2BPE%2B1000-2020%2B%281%29.pdf/35f0fc27-cba3-d396-4dc2-639f33969199?t=1635181030263

Meléndez, J. (2019). Técnicas de recolección de datos: ejemplos y guía paso a paso. Recuperado de https://reisdigital.es/datos-e-informacion/tecnicas-de-recoleccion-de-datos-ejemplos/

Ministerio de Ciencia, Tecnología e Innovación de Colombia. (2021). Guía para la gestión de datos de investigación. Recuperado de https://minciencias.gov.co/sites/default/files/upload/noticias/guia_gestion_de_datos_redcolombianadeic_1.pdf

Ministerio de Ciencia, Tecnología e Innovación de Colombia. (s.f.). Diseño de un Plan de Gestión de Datos de Investigación (PGDI). Recuperado de https://red-documentacion.minciencias.gov.co/Gestion_Datos_Investigacion/Gu%C3%ADa-PGDI

Restrepo, D., & Restrepo, M. (2004). Sesgos en diseños analíticos. Revista Colombiana de Psiquiatría, 33(3), 290-297. Recuperado de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0034-74502004000300007

## Créditos

Elaborado por:


Organización Internacional del Trabajo