# Técnicas de Limpieza y Preprocesamiento de Datos para Machine Learning

En el universo del Machine Learning, la calidad de los datos es el cimiento de todo modelo exitoso. Antes de que los algoritmos puedan aprender, es crucial preparar la información para asegurar su precisión y relevancia.

# Paso 1: Análisis Exploratorio de Datos (EDA) (EDA)

### Objetivo Fundamental

Comprender la estructura, la calidad y la distribución intrínseca de los datos antes de cualquier intervención de limpieza, revelando patrones ocultos y posibles problemas.

### Herramientas y Ejemplos

Utilice librerías como Pandas en Python (`df.info()`, `df.describe()`) y visualizaciones clave como histogramas y diagramas de caja para identificar valores faltantes, duplicados y anomalías.

### Importancia Crítica

El EDA es una etapa indispensable. Saltarlo puede conducir a modelos con bajo rendimiento, sesgos inherentes o errores que son extremadamente difíciles de diagnosticar y corregir en fases avanzadas.

# Paso 2: Manejo de Valores Faltantes

Los valores faltantes son una realidad en casi todos los conjuntos de datos, y su gestión adecuada es vital para la integridad del modelo. Identifique y aplique la estrategia más apropiada.

### Eliminación

Remueva filas o columnas si la cantidad de valores faltantes es pequeña o si la variable no es crítica. Use con precaución para no perder información valiosa.

### Imputación Simple

Rellene los vacíos con la media, mediana (para datos numéricos) o moda (para categóricos). Adecuado para datos que no presentan un patrón complejo de ausencia.

### Imputación Avanzada

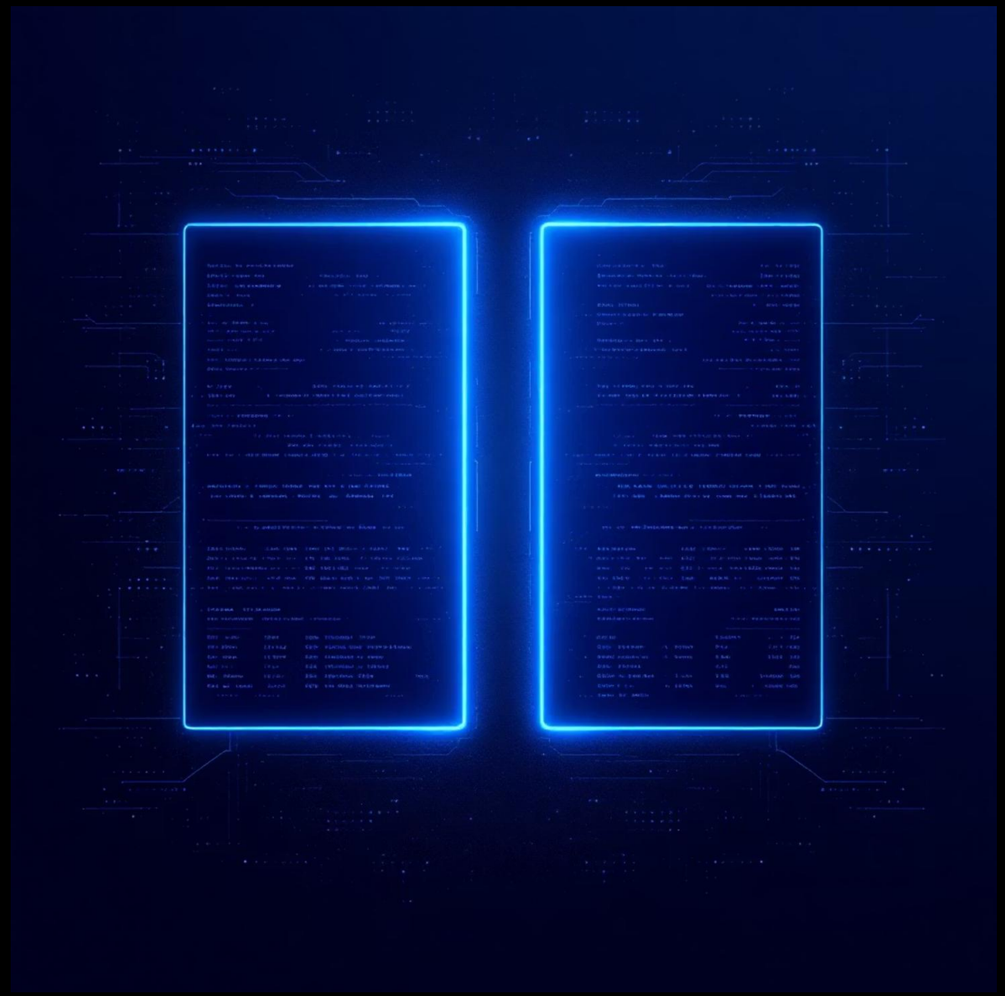
Emplee técnicas sofisticadas como regresión, k-NN o modelos predictivos para estimar los valores faltantes, capturando relaciones más complejas en los datos.

### ¿Cuándo NO Imputar?

Evite la imputación si introducirá sesgos significativos o si la ausencia de un dato es, en sí misma, una información relevante (ej. un campo vacío en una encuesta que indica 'no aplica').

# Paso 3: Eliminación de Duplicados

> ¿Por qué es crucial? Los registros duplicados pueden sesgar las estadísticas del dataset, inflar artificialmente las métricas de rendimiento del modelo y, en última instancia, llevar a conclusiones erróneas o a un sobreajuste.



### Ejemplo Práctico

En Pandas, la función `df.drop_duplicates()` es la herramienta principal para identificar y eliminar filas completamente idénticas, asegurando la unicidad de sus observaciones.

- **Cuándo hacerlo:** Siempre que los duplicados sean un error de entrada de datos o un registro repetido sin significado adicional.
- **Cuándo NO:** Si los registros aparentemente duplicados representan eventos legítimos e independientes (ej. múltiples transacciones del mismo cliente en momentos diferentes).

# Paso 4: Detección y Tratamiento de Valores Atípicos (Outliers)

Los outliers pueden distorsionar el análisis estadístico y afectar negativamente el rendimiento del modelo, pero no siempre deben ser eliminados.

### Visualización

Inicie con **boxplots**, histogramas y scatter plots. Estas herramientas gráficas son excelentes para detectar visualmente valores que se desvían de la norma.

### Análisis Estadístico

Utilice el Rango Intercuartílico (IQR) para definir umbrales: valores fuera de **Q1 - 1.5*IQR** o **Q3 + 1.5*IQR** son considerados atípicos. Métodos como el Z-score también son útiles.

### Opciones de Tratamiento

Puede optar por **eliminar** los outliers, **transformar** la variable (ej. logaritmo) para reducir su impacto, o **mantenerlos** si son representativos de fenómenos importantes.

**Consideración Clave:**

No elimine outliers si son casos reales y relevantes (ej. transacciones fraudulentas, mediciones de eventos raros). Estos pueden ser los datos más importantes para su modelo.

# Paso 5: Codificación de Variables Categóricas

Los algoritmos de Machine Learning requieren datos numéricos. Transformar variables categóricas en un formato numérico es esencial sin perder su significado.

### One-Hot Encoding

Crea nuevas columnas binarias para cada categoría. Ideal para variables **nominales** (sin orden inherente), como 'ciudad' o 'color'. Evita que el modelo asuma una relación ordinal.

Ejemplo: 'Rojo' se convierte en [1, 0, 0], 'Azul' en [0, 1, 0].

### Label Encoding

Asigna un número entero único a cada categoría. Adecuado para variables **ordinales** (con un orden lógico), como 'nivel educativo' (Bajo, Medio, Alto).

Ejemplo: 'Bajo'=1, 'Medio'=2, 'Alto'=3.

**Cuándo no codificar:** Si una variable categórica tiene demasiadas categorías únicas, el One-Hot Encoding puede resultar en un número excesivo de columnas (maldición de la dimensionalidad), lo cual puede ser ineficiente para el modelo. Evalúe la relevancia de la variable antes de codificarla.

# Paso 6: Escalamiento y Normalización de Datos

¿Por qué es necesario? Muchos algoritmos de Machine Learning, especialmente aquellos basados en distancias (SVM, k-NN) o gradientes (Regresión Lineal, Redes Neuronales), son sensibles a las escalas de las variables. Si una característica tiene un rango mucho mayor que otra, podría dominar la función de costo y el entrenamiento.

1

## Min-Max Scaling

Normaliza los datos para que se encuentren dentro de un rango específico, generalmente entre **0 y 1**. Útil cuando se necesita que la distribución de los datos sea compacta.

2

## Estandarización (Z-score)

Transforma los datos para que tengan una **media de 0** y una **desviación estándar de 1**. Es menos sensible a los outliers y es una buena opción por defecto.

# Paso 7: Selección y Reducción de Características

Optimizar el conjunto de características es crucial para la eficiencia del modelo y su interpretabilidad. Menos es más, si se conserva la información esencial.

1

### Eliminación Manual

Identifique y **elimine variables irrelevantes** o aquellas con alta correlación entre sí (multicolinealidad), lo cual puede introducir redundancia y ruido.

2

### Métodos de Filtrado

Utilice pruebas estadísticas (ANOVA, Chi-cuadrado) o métricas de importancia para seleccionar las características más relevantes antes de alimentar el modelo.

3

### Métodos de Envoltura

Evalúe subconjuntos de características entrenando un modelo, seleccionando el conjunto que ofrece el mejor rendimiento. Más costoso computacionalmente.

4

### Métodos Embebidos

Algunos algoritmos, como los basados en árboles o modelos con regularización (Lasso, Ridge), tienen mecanismos **incorporados** para seleccionar características importantes durante el entrenamiento.

5

### Reducción de Dimensionalidad

Técnicas como PCA (Análisis de Componentes Principales) transforman las características originales en un conjunto más pequeño de nuevas variables, conservando la mayor parte de la varianza.

**Cuándo no reducir:** Evite la reducción si al hacerlo se pierde información crítica necesaria para el modelo o si la interpretabilidad de las características originales es primordial para su caso de uso.

# Paso 8: División del Dataset en Entrenamiento y Prueba

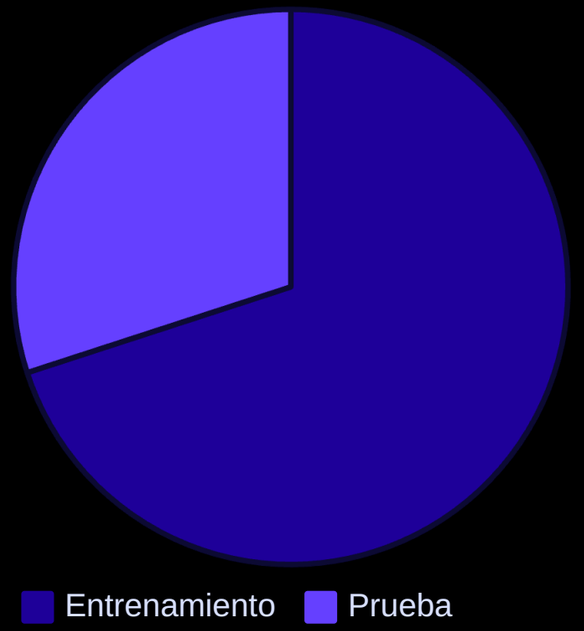
La división correcta del dataset es fundamental para evaluar el verdadero rendimiento del modelo y asegurar su generalización a datos no vistos.

Objetivo: Garantizar que el modelo sea evaluado con datos que **nunca ha "visto"** durante su entrenamiento. Esto previene el sobreajuste y proporciona una métrica de rendimiento realista.

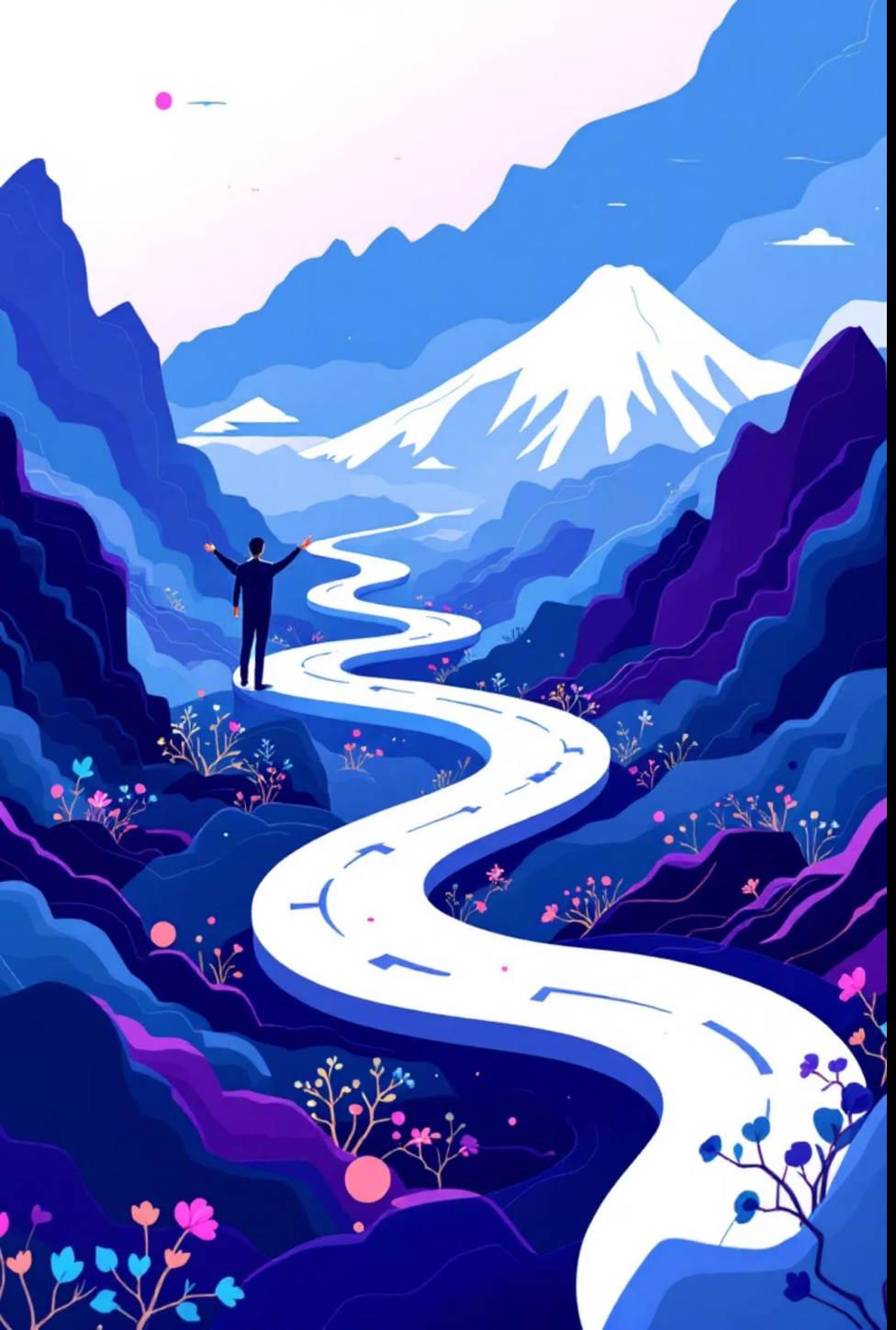
- **Conjunto de Entrenamiento:** Utilizado para que el modelo aprenda patrones y relaciones en los datos.
- **Conjunto de Prueba:** Utilizado para evaluar el rendimiento final del modelo una vez entrenado, simulando datos del mundo real.

**Ejemplo:**

En Scikit-learn, la función `train_test_split` es la herramienta estándar. Divisiones comunes son **70%-30%** o **80%-20%** para entrenamiento y prueba, respectivamente.




> **¡Nunca mezcle!** Es un error crítico permitir que cualquier dato del conjunto de prueba "se filtre" al conjunto de entrenamiento, ya que esto invalidaría la evaluación del modelo y conduciría a un sobreajuste engañoso.



# Conclusión: La Limpieza y Preprocesamiento Son Son Claves para Modelos Exitosos

**Fundamento de Precisión**

Datos limpios y bien preparados son el pilar que mejora la precisión del modelo, minimiza sesgos y evita errores de inferencia.

**Adaptación Constante**

Cada paso debe ajustarse meticulosamente al contexto específico del problema y a la naturaleza intrínseca del conjunto de datos.

**Inversión Rentable**

Invertir tiempo y recursos en un preprocesamiento exhaustivo ahorrará incontables horas en depuración y mejorará drásticamente los resultados finales.

**El Primer Paso Esencial**

Siempre comience con un Análisis Exploratorio de Datos robusto y jamás subestime la importancia crítica de una limpieza meticulosa.