ESQUEMA GENERAL DEL PROCESO DE TRATAMIENTO DE DATOS EN APRENDIZAJE AUTOMÁTICO

1. Definición del problema y objetivo

Qué es:

Antes de abrir un dataset, hay que tener claro *qué se quiere predecir o analizar*.

Qué se hace aquí:

- Identificar la pregunta o hipótesis.
- Determinar el tipo de problema (clasificación, regresión, agrupamiento...).
- Saber qué variables son las relevantes y cuál es la variable objetivo (*target*).

2. Carga y exploración inicial de los datos

Qué es:

El primer contacto con los datos: leerlos, entender su estructura y su contenido.

Qué se hace aquí:

- Cargar el dataset (CSV, Excel, SQL, etc.).
- Ver las primeras filas (head()), dimensiones, tipos de datos y resumen estadístico (describe()).
- Detectar problemas evidentes (nulos, tipos incorrectos, escalas diferentes).
- **Comenzar la visualización inicial**: histogramas, diagramas de dispersión, conteos de categorías, etc.

3. Limpieza de datos

Qué es:

Corregir o eliminar los errores, inconsistencias o valores faltantes.

Qué se hace aquí:

- Tratamiento de **valores nulos** (relleno, eliminación o imputación).

- Detección y tratamiento de **outliers** (valores extremos).
- Corrección de tipos de datos o etiquetas mal escritas.

(Visualizaciones útiles: boxplots, histogramas, gráficos de dispersión.)

4. Transformación y preparación

Qué es:

Adaptar los datos para que los algoritmos de machine learning puedan procesarlos correctamente.

Qué se hace aquí:

- **Estandarización o normalización**: poner las variables en la misma escala (útil para distancias o correlaciones).
- **Codificación de variables categóricas** (one-hot encoding, label encoding).
- **Creación de nuevas variables (feature engineering)**.

5. Análisis exploratorio de datos (EDA – Exploratory Data Analysis)

Qué es:

Buscar patrones, relaciones y correlaciones entre variables.

Qué se hace aquí:

- Estudiar correlaciones numéricas (matriz de correlación, **mapa de calor o heatmap**).
- Comparar distribuciones entre grupos.
- Identificar tendencias y posibles relaciones causa-efecto.

(Aquí se usan mucho diagramas de dispersión, pairplots, heatmaps, violin plots, etc.)

6. División del dataset

Qué es:

Separar los datos en conjuntos de entrenamiento y prueba para evitar sobreajuste.

Qué se hace aquí:

- Dividir en train y test (por ejemplo, 80% / 20%).
- A veces también un conjunto de validación (o validación cruzada).

7. Modelado

Qué es:

Aplicar algoritmos de machine learning sobre los datos preparados.

Qué se hace aquí:

- Selección de modelo (regresión lineal, árbol de decisión, redes neuronales...).
- Entrenamiento del modelo con los datos de entrenamiento.
- Ajuste de hiperparámetros.

8. Evaluación del modelo

Qué es:

Medir qué tan bien funciona el modelo.

Qué se hace aquí:

- Comparar predicciones con valores reales.
- Usar métricas (precisión, recall, F1-score, RMSE, R^2...).
- Verificar si hay sobreajuste (train ≠ test).

(Visualizaciones útiles: matrices de confusión, curvas ROC, gráficos de error.)

9. Conclusiones y comunicación de resultados

Qué es:

Interpretar los hallazgos y comunicar las conclusiones de forma clara y visual.

Qué se hace aquí:

- Explicar qué variables son más influyentes.
- Mostrar resultados en gráficos, dashboards o informes.
- Traducir los resultados técnicos a lenguaje comprensible para la audiencia.

10. Iteración y mejora continua

Qué es:

El ciclo no termina: con los resultados obtenidos se vuelven a revisar los pasos anteriores.

Qué se hace aquí:

- Añadir nuevos datos, eliminar ruido, probar otras transformaciones o modelos.
- Mejorar precisión, interpretabilidad o rendimiento.

1. DEFINICIÓN DEL PROBLEMA Y OBJETIVO

Qué es y por qué es importante

Esta es la **fase de planteamiento del proyecto**.

Antes de tocar datos o código, debemos entender **qué queremos conseguir** y **qué tipo de problema estamos resolviendo**.

En machine learning, los algoritmos no "piensan": **aprenden patrones a partir de datos**. Por eso, si el problema está mal definido, aunque el modelo funcione matemáticamente, **las conclusiones serán inútiles o engañosas**.

Objetivo de esta fase:

Traducir una pregunta del mundo real (por ejemplo, "¿Qué hace que una casa sea cara?") a una **pregunta analítica y medible** ("¿Podemos predecir el precio de una vivienda a partir de sus características físicas y de ubicación?").

Elementos principales de esta etapa

1. **Contexto del problema**
    a. Entender el sector y el tema en cuestión (por ejemplo, mercado inmobiliario, salud, educación…).
    b. Identificar **quién necesita el análisis** y **para qué se usará** (empresa, investigación, administración pública…).
    c. Conocer **qué variables** pueden influir y **qué datos** están disponibles.

Ejemplo:

En un proyecto inmobiliario, una agencia quiere estimar el **precio probable** de una vivienda según su tamaño, ubicación y características.

Contexto: el mercado cambia por zonas, la distancia al centro y la antigüedad del edificio afectan el valor.

2. **Formulación de la pregunta analítica**
   Convertir la necesidad en una pregunta precisa y medible.

Ejemplos:

- **a.** *"¿Qué factores influyen más en el precio de una vivienda?"* → **análisis exploratorio y correlacional.**
- **b.** *"¿Podemos predecir el precio de una vivienda a partir de sus características?"* → **problema de regresión.**
- **c.** *"¿Podemos clasificar viviendas según su rango de precio (bajo, medio, alto)?"* → **problema de clasificación.**

3. **Identificación de la variable objetivo (*target*) y las variables predictoras**
    a. **Variable objetivo:** lo que queremos predecir o explicar (por ejemplo, Precio).
    b. **Variables predictoras:** las características que usamos para hacer esa predicción (Tamaño_m2, Habitaciones, DistanciaCentro_km, etc.).
    c. Distinguir entre **variables numéricas** (continuas o discretas) y **categóricas** (como Zona, Garaje).

Ejemplo aplicado al dataset inmobiliario:

d. Precio → *variable objetivo (target)*.
e. Tamaño_m2, Habitaciones, Baños, Antigüedad, Zona, Garaje, Ascensor → *variables predictoras (features)*.

4. **Definir el tipo de problema de aprendizaje automático**
   Según el objetivo y la naturaleza del target:

Tipo de problema	**Variable objetivo**	**Ejemplo**	**Algoritmos típicos**
Regresión	Numérica continua	Predecir el precio de una vivienda	Regresión lineal, árboles de regresión
Clasificación	Categórica	Clasificar viviendas como "baratas", "medias", "caras"	Árboles de decisión, regresión logística, SVM
Clustering (agrupamiento)	Sin variable objetivo	Agrupar viviendas por similitud	K-Means, DBSCAN
Asociación o patrones	Reglas entre variables	"Si tiene garaje y 3 habitaciones → precio alto"	Reglas de asociación (Apriori)

5. **Definición de métricas de éxito**
   a. **Regresión:** error cuadrático medio (RMSE), error absoluto medio (MAE), coeficiente de determinación (R^2).
   b. **Clasificación:** precisión, recall, F1-score, matriz de confusión.
   c. **Clustering:** coeficiente de silueta, inercia, etc.

Ejemplo práctico:
Si queremos predecir el precio de las viviendas, usaremos R^2 y RMSE.
Si queremos clasificar en rangos de precio (bajo/medio/alto), usaríamos precisión y recall.

6. **Hipótesis iniciales (intuiciones)**
   a. Formular hipótesis que luego se validarán con los datos.
   b. Ejemplo: "Las viviendas con garaje y ascensor tienden a ser más caras."
   c. "Las viviendas más alejadas del centro son más baratas."

Errores comunes en esta fase

- Empezar a modelar sin saber qué se quiere predecir.
- No tener variable objetivo definida.
- No considerar el contexto (por ejemplo, el impacto de la ubicación real).
- Intentar responder muchas preguntas a la vez con un solo modelo.
- No definir cómo se medirá el éxito.

Buenas prácticas

- Redactar una **pregunta principal** y **2–3 preguntas secundarias**.
- Hacer un **diagrama del flujo lógico** entre variables.
- Verificar si los datos disponibles pueden responder realmente a la pregunta.
- Documentar todas las decisiones (ideal para enseñar cómo se estructura un notebook de análisis).

2. CARGA Y EXPLORACIÓN INICIAL DE LOS DATOS

Objetivo

El propósito aquí es **conocer el dataset**: su estructura, tamaño, tipo de variables y posibles irregularidades.

Es el momento de hacerse las primeras preguntas:

“¿Qué datos tengo?”, “¿De qué tipo son?”, “¿Hay valores vacíos o extremos?”, “¿Cómo se distribuyen?”.

Nada se corrige todavía — solo **se observa** y **se documenta**.

1. Carga de datos

Usamos librerías como pandas para leer el dataset desde un archivo (CSV, Excel, SQL, etc.).

```
import pandas as pd
df = pd.read_csv("datos_inmobiliarios_simplificado.csv")
```

Inspección básica:

```
df.shape                # Dimensiones (filas, columnas)
df.info()               # Tipos de datos y valores nulos
df.head()               # Primeras filas
df.describe()           # Estadísticos de las variables numéricas
```

Con esto se sabe:

- Cuántas variables hay y de qué tipo son.
- Si hay valores nulos o tipos erróneos.
- Qué rangos numéricos tiene cada variable.

2. Clasificación de las variables

Tipo	Ejemplo	Uso
Numéricas continuas	Precio, Tamaño_m2, DistanciaCentro_km	Se usan para análisis estadístico y correlaciones.
Numéricas discretas	Habitaciones, Baños	Conteos o categorías numéricas.
Categóricas	Zona, Garaje, Ascensor	Factores cualitativos que se codificarán más adelante.
Derivadas	Precio_m2	Variables calculadas o auxiliares.

3. Detección preliminar de problemas

Ya en esta etapa conviene detectar:

- **Valores nulos o faltantes**

df.isnull().sum()

- **Duplicados**

df.duplicated().sum()

- **Tipos erróneos** (por ejemplo, números guardados como texto)

df.dtypes

Esto aún no se corrige — solo se anota para el siguiente paso (limpieza de datos).

4. Visualización exploratoria básica

Aquí usamos gráficos sencillos para ver **cómo se distribuye cada variable individualmente**.

a) Distribuciones de variables numéricas

Gráfico: Histograma

Objetivo: Ver si hay sesgos, concentraciones o valores extremos.

```
df["Precio"].hist(bins=30)
plt.title("Distribución del precio de las viviendas")
```

b) Conteo de categorías

Gráfico: Countplot

Objetivo: Ver si las categorías están equilibradas.

```
sns.countplot(x="Zona", data=df)
```

Si hay zonas con muy pocos datos, podrían tener menos peso estadístico.

c) Comparaciones básicas

Gráfico: Boxplot

Objetivo: Comparar una variable numérica frente a una categórica (ej. Precio por Zona).

```
sns.boxplot(x="Zona", y="Precio", data=df)
```

Aquí ya pueden verse valores anómalos (puntos fuera del rango esperado).

3. LIMPIEZA DE DATOS

Objetivo

Ahora pasamos de *mirar* los datos a *mejorarlos*.

La limpieza tiene como meta **eliminar o corregir imperfecciones** que afectarían el análisis posterior (EDA o modelado).

Se trata de **preparar los datos reales para poder analizarlos con fiabilidad.**

1. Tratamiento de valores nulos

a) Identificación

```
df.isnull().sum()
```

b) Decisión

Existen varias estrategias:

- **Eliminar filas o columnas:** si hay pocos valores faltantes.

```
df = df.dropna(subset=["Garaje"])
```

- **Imputar valores:** reemplazar con una media, mediana o valor más frecuente.

```
df["Antigüedad"].fillna(df["Antigüedad"].median(), inplace=True)
```

- **Mantener nulos temporalmente:** si se desconoce el significado, pero se quieren analizar patrones de ausencia.

c) Visualización de nulos (opcional)

Usar un heatmap de nulos:

```
sns.heatmap(df.isnull(), cbar=False)
```

Permite ver si los nulos se concentran en ciertas variables o grupos.

2. Detección y tratamiento de *outliers* (valores extremos)

a) Detección visual

Boxplot o diagrama de dispersión:

```
sns.boxplot(x=df["Precio"])
sns.scatterplot(x="Tamaño_m2", y="Precio", data=df)
```

Los puntos fuera de los bigotes del boxplot o las nubes dispersas indican posibles outliers.

b) Detección estadística

Usando el rango intercuartílico (IQR):

```
Q1 = df["Precio"].quantile(0.25)
Q3 = df["Precio"].quantile(0.75)
IQR = Q3 - Q1
lim_inf = Q1 - 1.5 * IQR
lim_sup = Q3 + 1.5 * IQR
```

```
outliers = df[(df["Precio"] < lim_inf) | (df["Precio"] > lim_sup)]
```

c) Tratamiento

Opciones:

- **Eliminar** los outliers si son errores claros de registro.
- **Reemplazar** con límites razonables o valores medios.
- **Mantenerlos** si representan casos reales (ej. mansiones legítimamente caras).

3. Corrección de inconsistencias

A veces hay errores de formato o etiquetas mal escritas:

- "si", "Sí", "SI" → se unifican a "Sí".
- Zona con nombres incoherentes ("centro", "Centro", "CENTRO") → se normaliza.

```
df["Zona"] = df["Zona"].str.capitalize()
```

También se revisan **tipos de datos**:

```
df["Antigüedad"] = df["Antigüedad"].astype(float)
```

4. ANÁLISIS EXPLORATORIO DE DATOS (EDA)

Objetivo general

El **EDA** busca **entender la historia que cuentan los datos**, respondiendo a preguntas como:

- ¿Qué tendencias existen?
- ¿Qué variables están relacionadas?

- ¿Existen agrupaciones naturales o diferencias por categorías?
- ¿Hay correlaciones fuertes o posibles causas?

Es el puente entre la *limpieza* y el *modelado*.

Aquí se obtienen **insights** que guían cómo preparar y usar los datos para los algoritmos.

1. Comprender la estructura y distribución de las variables

Antes de buscar relaciones, hay que entender cómo se comporta cada variable por separado (*análisis univariante*).

a) Variables numéricas: forma y dispersión

Objetivo: conocer la forma de la distribución (simétrica, sesgada, multimodal, con valores extremos...).

Gráficos típicos:

- **Histograma**

```
df["Precio"].hist(bins=30)
```

→ Muestra la frecuencia de valores. Si hay colas largas, puede haber sesgo o outliers.

- **KDE plot (Kernel Density Estimation)**

```
sns.kdeplot(df["Tamaño_m2"], fill=True)
```

→ Representa una versión suavizada de la distribución, útil para ver si hay varias "modas" (picos).

- **Boxplot**

```
sns.boxplot(x=df["Precio"])
```

→ Indica mediana, cuartiles y valores atípicos visualmente.

b) Variables categóricas: proporciones y equilibrio

Objetivo: ver cómo se distribuyen las categorías y si hay alguna con muy pocos o muchos casos.

Gráficos típicos:

- **Countplot (barras)**

```
sns.countplot(x="Zona", data=df)
```

→ Muestra cuántos registros hay en cada categoría.

- **Pie chart** (más visual, menos técnico)

```
df["Garaje"].value_counts().plot.pie(autopct='%1.1f%%')
```

2. Analizar relaciones entre variables (bivariado)

Una vez se entiende cada variable, el siguiente paso es **estudiar relaciones entre pares de variables**:

numérica-numérica, categórica-numérica, categórica-categórica.

a) Numérica vs Numérica → correlaciones y relaciones lineales

Ejemplo: Tamaño_m2 vs Precio

Gráficos:

- **Scatter plot (diagrama de dispersión)**

```
sns.scatterplot(x="Tamaño_m2", y="Precio", data=df)
```

→ Permite ver si hay relación lineal o curvilínea.

- **Regplot (con línea de regresión)**

```
sns.regplot(x="Tamaño_m2", y="Precio", data=df, line_kws={"color":"red"})
```

→ Muestra la tendencia central (positiva, negativa o nula).

Interpretación:

Si los puntos forman una nube diagonal ascendente → mayor tamaño implica mayor precio.

b) Categórica vs Numérica → comparar grupos

Ejemplo: Zona o Garaje vs Precio

Gráficos:

- **Boxplot**

```
sns.boxplot(x="Zona", y="Precio", data=df)
```

→ Permite comparar medianas y dispersión de precios por zona.

- **Violin plot**

```
sns.violinplot(x="Garaje", y="Precio", data=df)
```

→ Combina boxplot y densidad, útil para ver diferencias de distribución.

Interpretación:

Si las viviendas con garaje tienen distribuciones desplazadas hacia precios más altos → el garaje influye en el precio.

c) Categórica vs Categórica → dependencia entre categorías

Ejemplo: Garaje vs Ascensor

Tablas de contingencia:

```
pd.crosstab(df["Garaje"], df["Ascensor"], normalize="index")
```

Visualización:

- **Heatmap de frecuencias**

```
sns.heatmap(pd.crosstab(df["Garaje"], df["Ascensor"]), annot=True, cmap="YlGnBu")
```

Interpretación:

Puede verse si la mayoría de pisos con garaje también tienen ascensor (asociación positiva).

3. Estudiar correlaciones entre variables numéricas

a) Matriz de correlación

Qué mide:

Cómo varía una variable numérica respecto a otra (valores entre -1 y 1).

```
corr = df.corr(numeric_only=True)
corr
```

b) Mapa de calor (*heatmap*)

Visualización clave del EDA numérico:

```
sns.heatmap(corr, annot=True, cmap="coolwarm", center=0)
```

Interpretación:

- Colores rojos → correlación positiva fuerte.
- Colores azules → correlación negativa.
- Valores cerca de 0 → sin relación clara.

Ejemplo:

- Precio correlaciona positivamente con Tamaño_m2.
- DistanciaCentro_km correlaciona negativamente con Precio.

Esto permite **identificar variables influyentes** y evitar redundancia (variables muy correlacionadas entre sí).

4. Análisis multivariado

Cuando se combinan **más de dos variables** en el análisis:

Herramientas:

- **Pairplot**

```
sns.pairplot(df[["Precio", "Tamaño_m2", "Antigüedad", "DistanciaCentro_km"]])
```

→ Muestra todas las combinaciones de dispersión entre variables numéricas.

- **Colormaps** en scatterplots:

```
sns.scatterplot(x="Tamaño_m2", y="Precio", hue="Zona", data=df)
```

→ Permite ver cómo influye una categoría en una relación numérica.

- **3D scatter plots** (opcional):

```
from mpl_toolkits.mplot3d import Axes3D
ax = plt.figure().add_subplot(projection='3d')
ax.scatter(df["Tamaño_m2"], df["Precio"], df["DistanciaCentro_km"])
```

Interpretación:

Aquí se observan relaciones más complejas y cómo varios factores interactúan.

5. Detección de patrones o agrupaciones

A veces el EDA revela **segmentos naturales** (por ejemplo, tipos de viviendas).

Métodos:

- **Clustering visual (con PCA o KMeans)** para detectar grupos.
- **Histogramas apilados** para comparar proporciones entre grupos.
- **FacetGrid en Seaborn** para ver la misma variable en distintas categorías:

sns.FacetGrid(df, col="Zona").map_dataframe(sns.histplot, x="Precio")

6. Conclusiones e hipótesis del EDA

El EDA debe terminar con **insights claros** que preparen el camino al modelado:

Observación	Posible conclusión	Acción posterior
Precio ↑ con Tamaño_m2 ↑	Relación lineal	Usar regresión.
DistanciaCentro_km ↑ ⇒ Precio ↓	Relación inversa	Variable predictora relevante.
Zonas con precios muy diferentes	Variable categórica fuerte	Codificar Zona.
Muchos outliers en Precio	Datos extremos	Revisar o tratar.
Garaje afecta al precio	Interacción categórica	Crear variable dummy.

7. Visualizaciones resumen (EDA dashboard)

Al final del EDA, puede hacerse una **síntesis visual** con:

- Histogramas de las principales variables.
- Boxplots comparativos.
- Heatmap de correlaciones.
- Scatterplots principales (Precio vs Tamaño, Distancia, etc.).

Esto puede mostrarse como un **notebook o dashboard de exploración** (por ejemplo, con matplotlib, seaborn, o plotly).