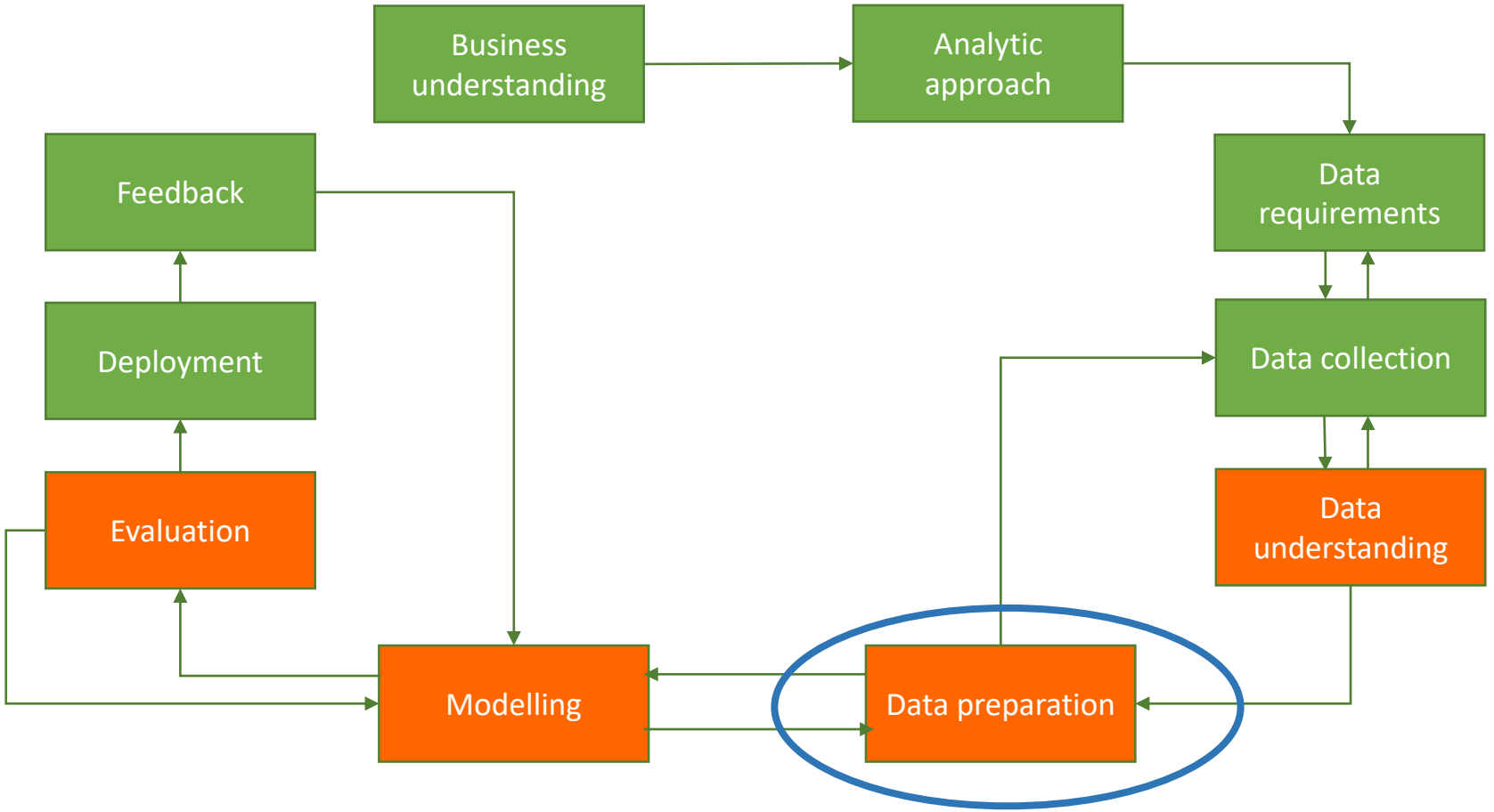
# Preprocesamiento de datos

Inteligencia Artificial - Paola A. Sánchez-Sánchez

# Metodología para el análisis de datos



# Agenda

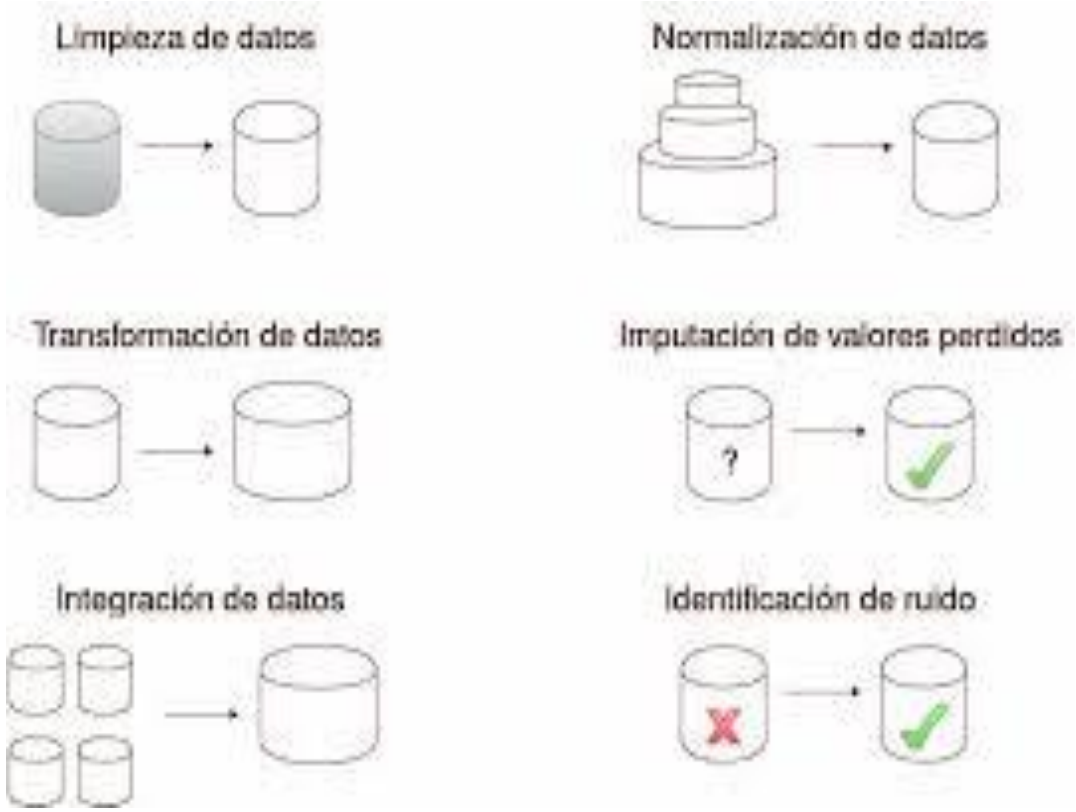
- Preprocesamiento de datos
- Técnicas de preprocesamiento
- Preprocesar con Python

# Preprocesado/preparación de los Datos

Se busca construir el conjunto de datos que se utilizará en el modelado.

Incluye:

- Limpieza de datos (tratar con valores faltantes o no válidos, outliers, eliminar duplicados, eliminar ruido),
- Transformar datos
- Combinar datos de múltiples fuentes (archivos, tablas, plataformas)
- Crear variables explicativas adicionales



# Limpieza de datos

Datos Faltantes

| Country | Age | Salary | Purchased |
|---|---|---|---|
| France | 44 | 72000 | No |
| Spain | 27 | 48000 | Yes |
| Germany | 30 | 54000 | No |
| Spain | 38 | 61000 | No |
| Germany | 40 | | Yes |
| France | 35 | 58000 | Yes |
| Spain | | 52000 | No |
| France | 48 | 79000 | Yes |
| Germany | 50 | 83000 | No |

df.isnull().any()

La limpieza de datos es el proceso de detectar y corregir o eliminar registros corruptos o inexactos de un conjunto de registros, tabla o base de datos y se refiere a la identificación de partes incompletas, incorrectas, inexactas o irrelevantes de los datos, para su posterior sustitución, modificación o eliminación de los datos sucios o poco precisos.

# Limpieza de datos: Faltantes (NaN o None)

1. Puedes eliminar las líneas con los datos si el conjunto de datos es lo suficientemente grande y el porcentaje de valores perdidos es alto, más del 50%, por ejemplo.

2. Puedes rellenar todas las variables nulas con 0, si se trata de valor numéricos.

3. Puedes rellenar los valores perdidos con la media, media o el valor más frecuente de la columna.

4. También puedes decidir rellenar los valores que faltan con cualquier valor que venga directamente después en la misma columna.

# Limpieza de datos en Python


```
df.dropna(axis = 0)
df.dropna(subset = [“NombreColPerdidos”], axis = 0)
df.dropna(thresh=half_count, axis = 1)
```


```
Media = df.[“Age”].mean()
df[“Age”].replace(np.nan, Media)
```

# Transformación de los datos

Transformar/Normalizar /Escalar los datos implica convertir el conjunto de datos en una distribución normal

```
from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)
```

# Manejo de datos categóricos

Modelos de datos solo utilizan valores numéricos, tipo de datos flotantes o enteros. Los conjuntos de datos a menudo contienen datos categóricos, por lo tanto se hace necesario transformarlo en numérico.

En la mayoría de los casos, los valores categóricos son discretos y puedes ser codificados como variables ficticias, asignando un número para cada categoría.

```
from sklearn.preprocessing import OneHotEncoder

onehotencoder = OneHotEncoder(datos_categoricos = [0])
X = onehotencoder.fit_transform(X).toarray()
```