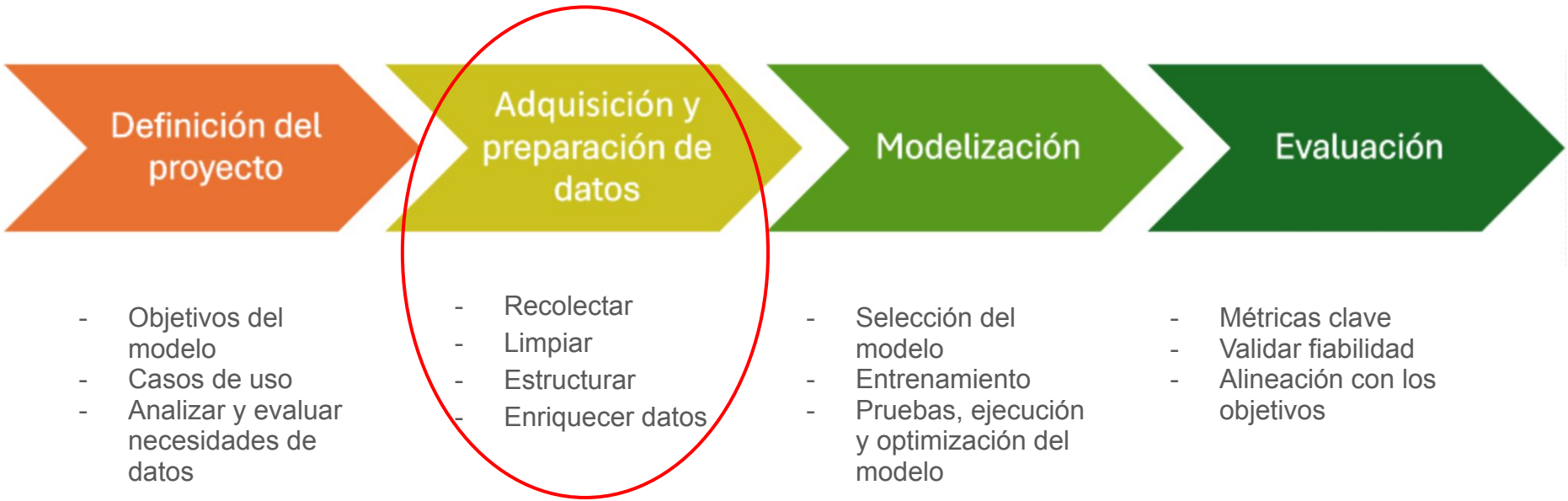
Procesos ETL

Tema 6: Limpieza, transformación y normalización de datos. Parte II

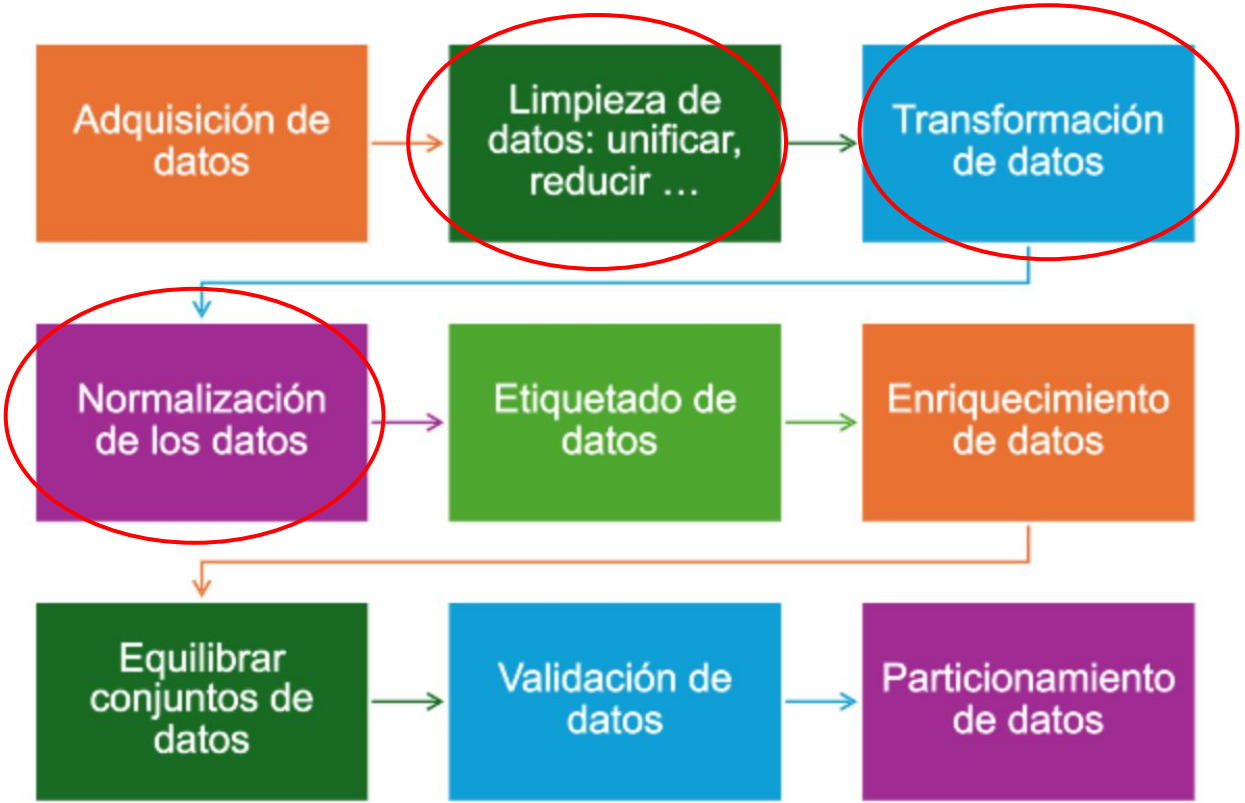
Adquisición y preparación de datos

Universitat d'Alacant
Universidad de Alicante

Ciclo de vida del proyecto IA



Preparación de datos



Limpieza, transformación y normalización

Limpieza de datos

Proceso esencial que elimina errores y anomalías en *raw data* con el objetivo de asegurar la **calidad** y **fiabilidad** de los datos, aspecto crítico para optimizar el rendimiento y la precisión de los modelos de IA.

Forma parte de la calidad de los datos y de la gobernanza de datos.

¿Dónde resolver los problemas con la calidad de los datos? ¿En la fuente?

➔ Diseñar transformaciones de limpieza de datos reutilizables

Implica entre otras tareas:

- **Corregir de errores, eliminar datos incorrectos o no válidos**
- **Gestionar los valores que faltan**
- **Eliminar o minimizar la redundancia de datos**

Corregir errores

Eliminar las inconsistencias, imprecisiones y errores tipográficos para asegurar la **calidad** y **uniformidad** de la información.

Errores tipográficos: fallos de escritura en la introducción de datos

"...cancionero catalán de la universidad de zAragoza"

"Edició digital a partir de l'edició de Mariano Baselga y Ramírez, Zaragoza, Cecilio Gasca, 18963".

"El espejo de la veredad"

&#0    

"Edición digital basada ↲↲↲en la de ParíS, Sociedad de Ediciones Literarias y Artísticas, [s.a.]."

Corregir errores

Errores de formato: diferentes representaciones de la misma información

aprox 1581-1639 → aproximadamente 1581-1639

Siglo XVII →S. 17º

1609-12-9 →1609-12-09 (formato YYYY-MM-DD)

1810/09/08 →1810-09-08 (formato YYYY-MM-DD)

02/12/2000 15-00-24 →2000-12-02 (formato YYYY-MM-DD)

12 de diciembre 1925 →1925-12-12 (formato YYYY-MM-DD)

chi / zho, zho, zh, chino, chinés →chi

spa, es, esp, español → es

mapa, material cartográfico (impreso) →cartográfico

EE.UU., USA, E.E.U.U., EUA → EE.UU.

Corregir errores

Inconsistencias: datos contradictorios, incorrectos o no válidos para el dominio

"Edició digital a partir de l'edició de Mariano Baselga y Ramírez, zAragoza, Cecilio Gasca, 18963".

Fechas de un autor (nacimiento-fallecimiento) 1733-1921 → vivió 188 años??

Fechas de un autor (nacimiento-fallecimiento) 1560-1536 → invertidas, error tipográfico??

Errores en códigos de ISBN o ISSN 978-84-690-2311-02 → ISBN incorrecto

El mismo autor aparece como: "Federico García Lorca", "Lorca F.", "García, Federico"

"Presentación contextualizada de cartas enviadas por Gómez de Baquero a Miguel de Unamuno, además de una dirigida a Ortega y Gasset. La correspondencia entre aquellos dos autores, iniciada por Unamuno en 1896, permite el seguimiento de la evolución…"

Ejemplar -5 páginas

Valores faltantes

Los valores faltantes es un problema común en la adquisición de datos y pueden ser debido a errores, datos sensibles, no suministrados, etc.

Gestionar valores que faltan por **reemplazo, interpolación o eliminación**.

Registros bibliográficos sin título <dc:title>null</dc:title> → se registra el error y se descarta registro

Registros bibliográficos sin autor <dc:creator>null</dc:creator> → se deja vacío o normaliza a “no dado” para el análisis, no normaliza a “Anónimo”

Género del autor o autora faltante → normalizar a “Desconocido”

Formato de una obra <dc:format></dc:format> → normalizar a “sin definir”

Redundancia de datos

El Objetivo principal es es garantizar datos únicos.

Reducir el tamaño del dataset y optimizar el entrenamiento del modelo.

Duplicados completas → eliminar duplicado

Dos registros bibliográficos con mismo título y mismo autor pero uno es está en PDF y otro en EPUB → fusionar y consolidar los campos más completos reflejando la relación.

El mismo autor aparece con distintos nombre: "Azorín" y “José Martínez Ruiz” → reflejar alternativas válidas que identifican al autor.

Materia o tema de un libro que proviene de diferentes fuentes Universidades -- España -- Siglo-19 y Universidades -- España -- S-XIX → normalizar y eliminar duplicados

Estrategias y técnicas

- Uso de expresiones regulares para encontrar y reemplazar formatos específicos.
- Limpieza para eliminar espacios en blanco, caracteres especiales, puntuación no deseada o caracteres extraños.
- Algoritmos para la comparación de cadenas (*string matching*), se identifican errores tipográficos, duplicados.
- Validación con datos del dominio →Mapeo de datos a vocabularios controlados.
- Estandarización aplicando un formato único o valor de referencia.

Dirección en “bruto”

"Avda. de la Estación, s/n, entresuelo C - 03003 (ALICANTE)"

¿Qué tareas o pasos pensáis que deberíamos hacer para “limpiar y normalizar”?

Ejemplo

1- Normalizar nombres

"Avda. de la Estación, s/n, entresuelo C - 03003 (ALICANTE)"

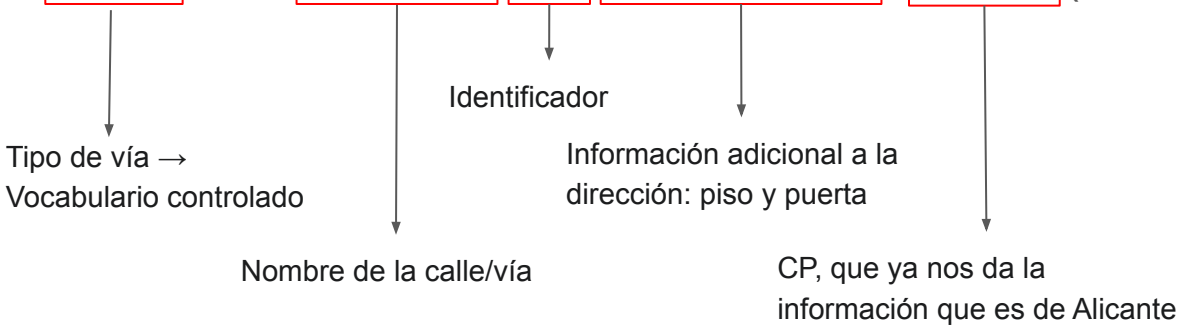
La abreviatura Avda. debe ser estandarizado a "**Avenida**"

Se debe determinar que hacemos con "s/n" podría convertirse en "sin número" o eliminarlo si no es necesario

Ejemplo

2- Separar en los distintos componentes

"Avda. de la Estación, s/n, entresuelo C - 03003 (ALICANTE)"



Ejemplo

3- Eliminar redundancias

El CP ya indica que es Alicante. El nombre de la localidad “(ALICANTE)" es redundante y se podría eliminar o estandarizar.

¿Es correcta la dirección postal?

Comprobar datos por ejemplo consultado bases de conocimiento o APIs disponibles.

- Postcode www.postcode.eu/es www.postcode.eu/es/products/validate-api
- Cartociudad datos.gob.es/es/aplicaciones/direcciones-postales-de-cartociudad-espana plataforma.idee.es/cnig-api

Este proceso de limpieza y normalización asegura que los datos sean coherentes, precisos y fáciles de usar:

Tipo Vía	Nombre Vía	Número	Piso/Puerta	Código Postal	Localidad
Avenida	Estación	s/n	Entresuelo C	03003	Alicante

Limpieza de datos

- No hay un único paso de limpieza de datos
- El proceso de depuración de datos comienza en el mismo paso de Extracción
  - Muchos de los pasos de entrada contienen funciones para leer los datos en un formato específico
- Generalmente es mejor leer los datos sin procesar “tal cual” están en origen (*raw data*) y aplicar en el ETL los pasos de limpieza necesarios
  - Por ejemplo en el paso Table input se puede aplicar cierta limpieza de datos desde las sentencias SQL. → Se dificulta la auditoría
- Se facilita la definición de flujos de error o procesamiento condicional para redirigir los datos que no satisfacen los requisitos.

Transformación y normalización

El objetivo de la **transformación** es preparar los datos para que puedan ser utilizarlos directamente.

Las transformaciones pueden incluir tareas como:

- Seleccionar campos, filtrar filas, ordenar, agrupar y aplicar funciones de agregación
- Realizar fusión de varias fuentes de datos
- *Feature Engineering*, transformación y creación de nuevas características a partir de datos existentes para mejorar la precisión y el rendimiento de los modelos
- Transformar datos no estructurados

Transformación y normalización

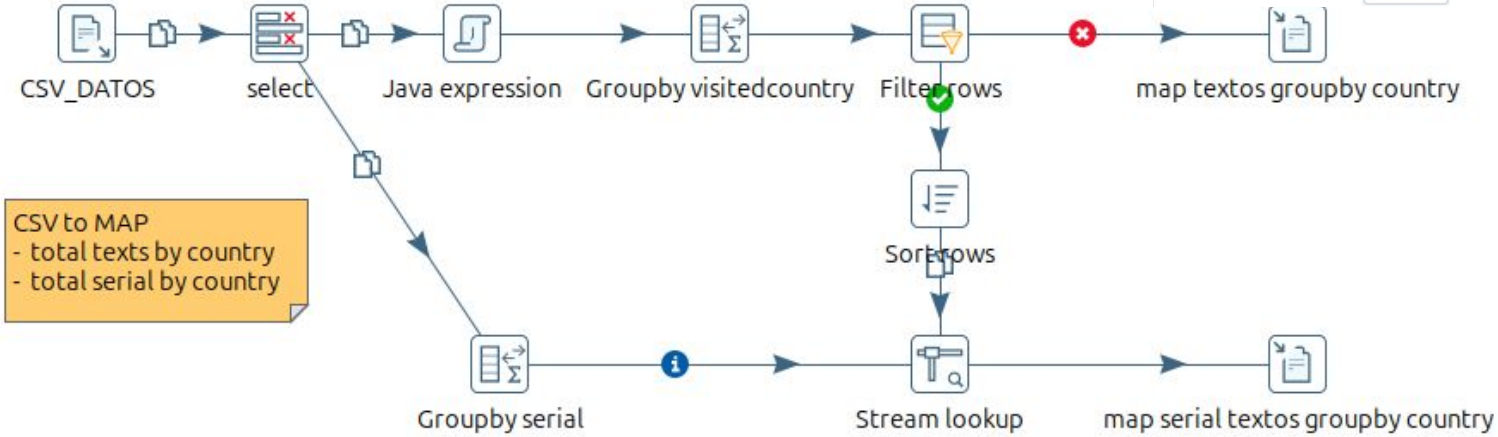
Ejemplo de transformación con **selección** de campos, **filtrado**, **ordenación**, **agrupación** y **agregación**

Examine preview data

Rows of step: Groupby serial (27 rows)

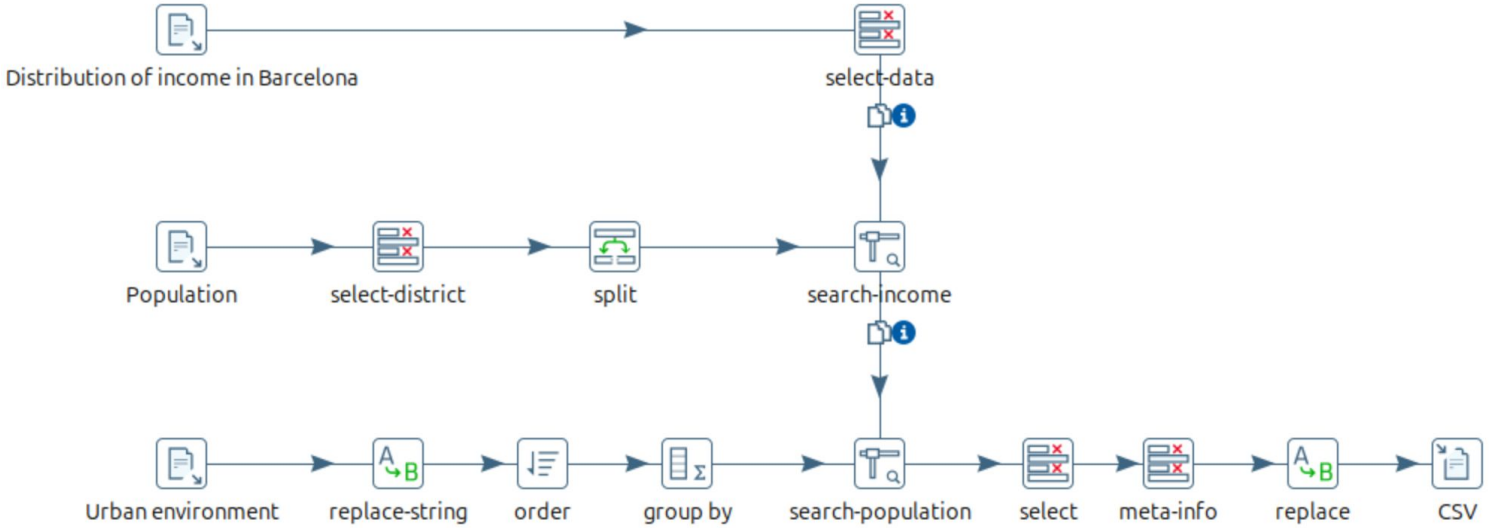
	PAÍSES_VISITADOS	countTextos	countSerial
1	<null>	4	2
2	Paraguay	21	13
3	Santo Domingo	7	5
4	varios	261	26
5	Martinica	5	4
6	Guyana Francesa	63	3
7	Argentina	182	26
8	Chile	119	12
9	Perú	94	14
10	Bolivia	26	7

Close



Transformación y normalización

Ejemplo de **fusión** de varias fuentes de datos



Feature Engineering

Proceso de transformar los datos sin procesar en nuevas variables, características o *features*, para mejorar el rendimiento de los modelos.

Representan **aspectos esenciales** de los datos, **relevantes** para el problema planteado.

Las características son variables clave de los datos y pueden ser textuales, numéricas o categóricas.

¿Por qué es importante?

- Mejorar la precisión y la eficiencia
- Mejora la interpretabilidad del modelo
- Puede reducir el sobreajuste

Feature Engineering

- **Creación** de nuevas características: combinar, transformar, extraer o derivar nuevas, por ejemplo *Splitting*
- **Codificación** de variables categóricas, por ejemplo *One-Hot Encoding*
- **Transformación** variables numéricas continuas en categorías, *Binning*
- **Enriquecimiento** de datos
- **Escalado** de datos
- Tratamiento de los ***Outliers***

Feature Engineering

Ejemplo de **creación de nuevas características** mediante **transformación** y **combinación** de datos

```
//MENCION DE PUBLICACION ORIGINAL
var lpo = place_of_publication+ publisher + editor + date_created;
if(place_of_publication != null)
    lpo = place_of_publication
if(publisher != null)
    lpo = lpo + ", " + publisher;

if(date_created != null)
    lpo = lpo + ", " + date_created;

//Detalles físicos mapeo directo de campos
//dimensions extension digital_file_characteristics
//var characteristicsAux = digital_file_characteristics
```

Feature Engineering

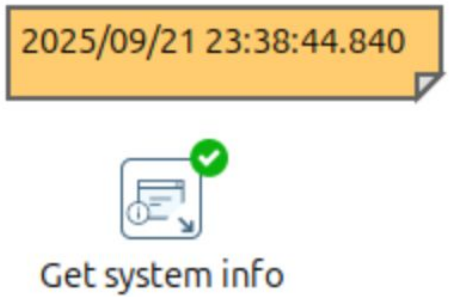
Ejemplo de **creación de nuevas características** mediante **extracción**

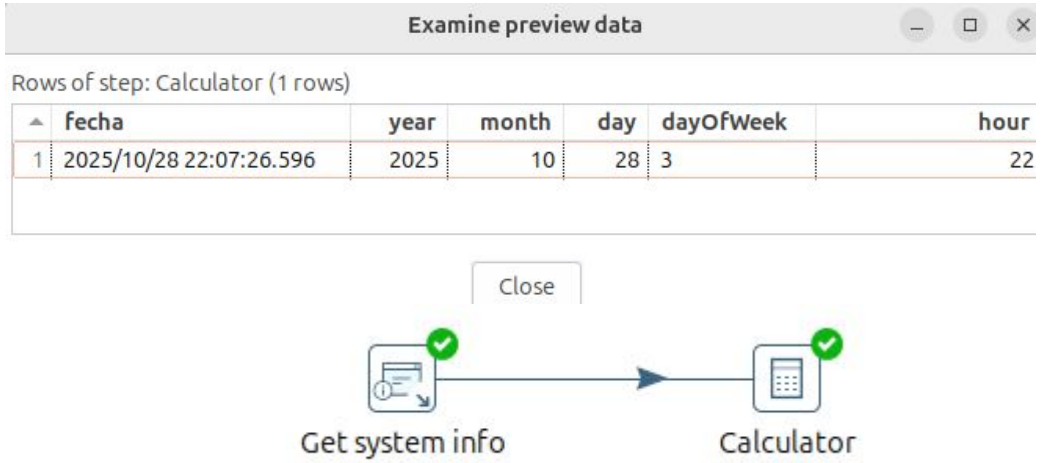
```
import pandas as pd

# fechas datetime
data = pd.DataFrame({'Fecha': pd.to_datetime(['2025-09-08', '2024-06-15', '2025-09-20'])})

data['year'] = data['Fecha'].dt.year
data['month'] = data['Fecha'].dt.month
data['day'] = data['Fecha'].dt.day
data['day_week'] = data['Fecha'].dt.dayofweek  #Lunes=0, Domingo=6
print(data)
```

	Fecha	year	month	day	day_week
0	2025-09-08	2025	9	8	0
1	2024-06-15	2024	6	15	5
2	2025-09-20	2025	9	20	5

Limpieza, transformación y normalización

Ejemplo de **creación de nuevas características** mediante **extracción**





Feature Engineering

Ejemplo de c**reación de nuevas características** mediante **splitting**

"Avda. de la Estación, s/n, entresuelo C - 03003 (ALICANTE)"

Tipo Vía	**Nombre Vía**	**Número**	**Piso/Puerta**	**Código Postal**	**Localidad**
Avenida	Estación	s/n	Entresuelo C	03003	Alicante

Feature Engineering

Ejemplo de **creación de nuevas características** mediante ***splitting***

```
import pandas as pd

data = {'Dirección completa': [
'Alonso Cano 26, Alicante, 03012', 'Calle Sierpes 56, Sevilla, 41004']}
df = pd.DataFrame(data)

df[['Calle', 'Ciudad', 'CP']] = df['Dirección completa'].str.extract(
    r'([\w\s]+\b[0-9]+\b),\s([\w\s]+),\s(\b\d{5}\b)')

print(df)
```

```
                Dirección completa             Calle    Ciudad     CP
0   Alonso Cano 26, Alicante, 03012    Alonso Cano 26  Alicante  03012
1  Calle Sierpes 56, Sevilla, 41004  Calle Sierpes 56   Sevilla  41004
```

Feature Engineering

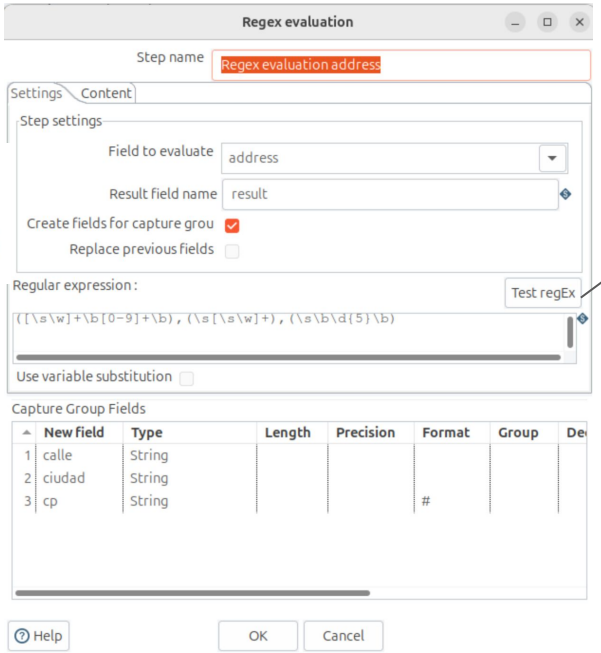
Ejemplo de **creación de nuevas características** mediante ***splitting***

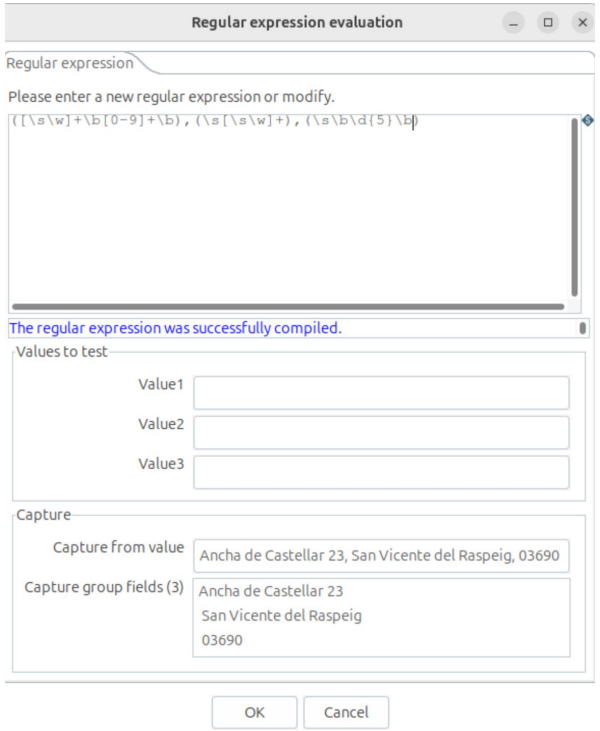
```
import pandas as pd

data = {'Full_Address': [
    '123 Elm St, Springfield, 12345', '456 Oak Rd, Shelbyville, 67890']}
df = pd.DataFrame(data)

df[['Street', 'City', 'Zipcode']] = df['Full_Address'].str.extract(
    r'([0-9]+\s[\w\s]+),\s([\w\s]+),\s(\d+)')

print(df)
```

```
                  Full_Address      Street         City Zipcode
0  123 Elm St, Springfield, 12345  123 Elm St  Springfield   12345
1  456 Oak Rd, Shelbyville, 67890  456 Oak Rd  Shelbyville   67890
```

Feature Engineering

Ejemplo de transformación y creación de nuevas características - **Splitting**





Feature Engineering

$a Poesía infantil y juvenil $x Historia y crítica

$a Sordos $x Enseñanza $x Obras anteriores a 1800

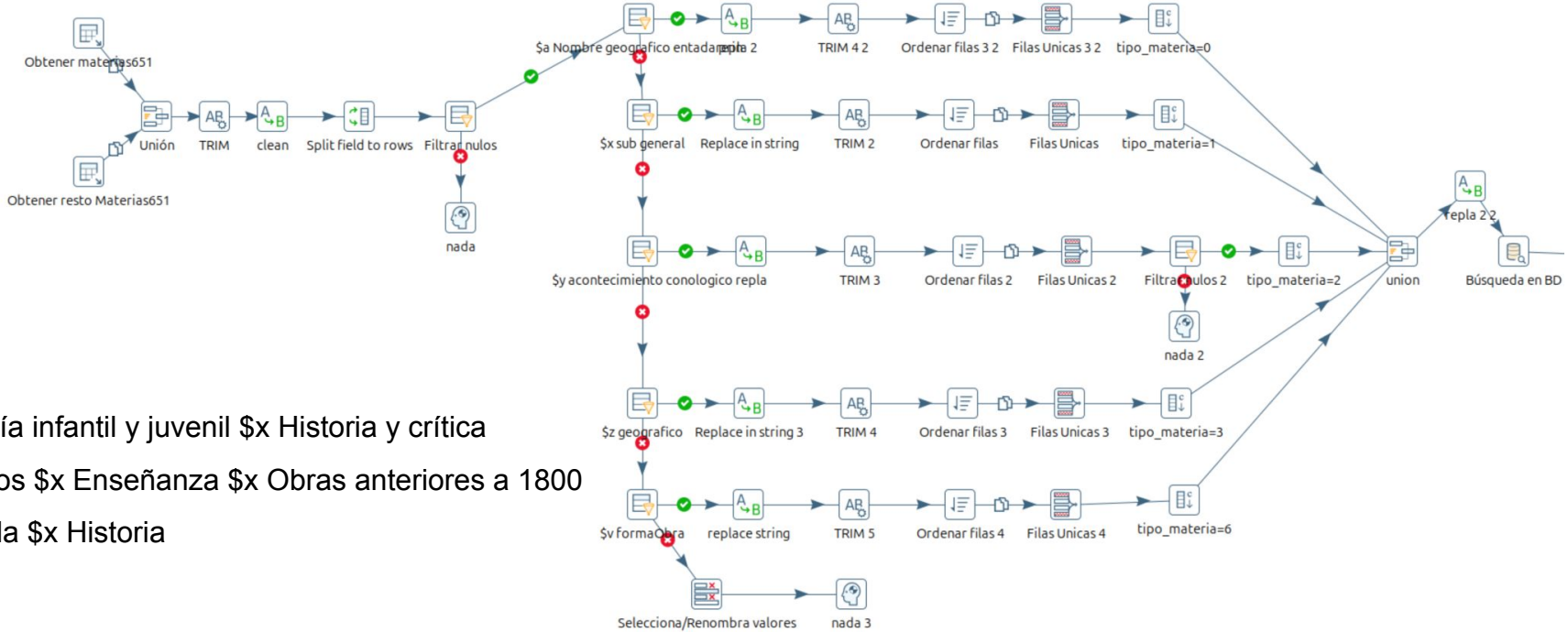
$a Sevilla $x Historia

LC Control Number: sh 85083794

HEADING: Mercury

000 00558cz a2200217n 450
001 4734282
005 19900221112154.6
008 860211i| anannbabn |a ana
035 __ |a (DLC)sh 85083794
906 __ |t 8528 |u fk03 |v 0
010 __ |a sh 85083794
040 __ |a DLC |c DLC |d DLC
053 _0 |a QD181.H6 |c Chemistry
053 _0 |a TA480.M4 |c Engineering materials
053 _0 |a TN271.M4 |c Prospecting
053 _0 |a TP245.M5 |c Chemical technology
150 __ |a Mercury
450 __ |a Hydrargyrum
450 __ |a Quicksilver
550 __ |w g |a Liquid metals
953 __ |a xx00 |b fg07

Feature Engineering



$a Poesía infantil y juvenil $x Historia y crítica

$a Sordos $x Enseñanza $x Obras anteriores a 1800

$a Sevilla $x Historia

Feature Engineering

Codificación de variables categóricas

One-Hot Encoding

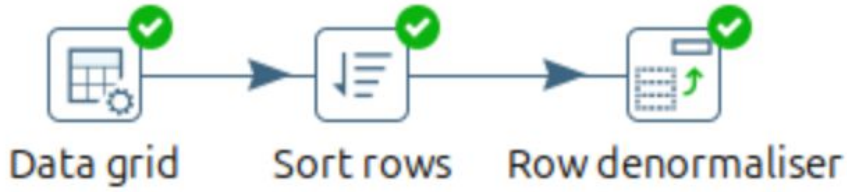
Convierte las variables categóricas en indicadores binarios

[15]:
```
import pandas as pd

data = {'ContentType': ['audio', 'video', 'graphic', 'text', 'online']}
df = pd.DataFrame(data)

df_encoded = pd.get_dummies(df, columns=['ContentType'], prefix='content')

print(df_encoded)
```

```
   content_audio  content_graphic  content_online  content_text  content_video
0           True            False           False         False          False
1          False            False           False         False           True
2          False             True           False         False          False
3          False            False           False          True          False
4          False            False            True         False          False
```

Feature Engineering

Codificación de variables categóricas



Feature Engineering

Binning, transformación de variables continuas discretas en categorías

Binning

Divide los datos originales en intervalos (bins), y después se reemplazan por un valor general calculado para ese bin Efecto suavizado de datos y además, puede reducir el riesgo de sobreajuste en conjuntos de datos pequeños

```
import pandas as pd

data = {'edad': [23, 45, 18, 34, 67, 50, 21]}
df = pd.DataFrame(data)

bins = [0, 20, 40, 60, 120]
labels = ['0-20', '21-40', '41-60', '+61']

df['Grupo_edad'] = pd.cut(df['edad'], bins=bins, labels=labels, right=False)

print(df)
```

```
   edad Grupo_edad
0    23      21-40
1    45      41-60
2    18       0-20
3    34      21-40
4    67        +61
5    50      41-60
6    21      21-40
```

Feature Engineering

Binning, transformación de variables continuas discretas en categorías

Number range

Step name: Number range

Input field: edad

Output field: range

Default value(if no r: unknown

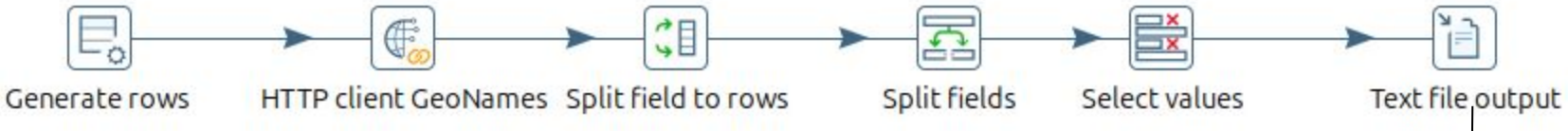
Ranges (min <= x< m:

	Lower Bound	Upper Bound	Value
1		20.0	-20
2	21.0	40.0	21-40
3	41.0	60.0	41-60
4	61.0		+61

	nombre	edad	range
1	Pepe	45	41-60
2	Ana	52	41-60
3	Eva	25	21-40
4	Jorge	23	21-40
5	Alejandro	78	+61
6	Pablo	65	+61

Feature Engineering

Enriquecimiento de datos, conectar con datos externos para aumentar el valor de nuestro conjunto de datos



Rows of step: Select values (255 rows)

	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
1	iso alpha2	iso alpha3	iso numeric	fips code	name	capital	areaInSqKm	population	continent	languages
2	AD	AND	020	AN	Andorra	Andorra la Vella	468.0	77006	EU	ca
3	AE	ARE	784	AE	United Arab Emirates	Abu Dhabi	82880.0	9630959	AS	ar-AE,fa,en,hi,ur
4	AF	AFG	004	AF	Afghanistan	Kabul	647500.0	37172386	AS	fa-AF,ps,uz-AF,tk
5	AG	ATG	028	AC	Antigua and Barbuda	St John's	443.0	96286	NA	en-AG
6	AI	AIA	660	AV	Anguilla	The Valley	102.0	13254	NA	en-AI

Feature Engineering

Enriquecimiento de datos

OpenWeather

https://api.openweathermap.org/data/2.5/weather?q=Zaragoza&APPID=


```
{
  "coord": {
    "lon": -0.8773,
    "lat": 41.6561
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 302.75,
    "feels_like": 301,
    "temp_min": 302.4,
    "temp_max": 303.16,
    "pressure": 1014,
    "humidity": 17,
    "sea_level": 1014,
    "grnd_level": 976
  },
  "visibility": 10000,
  "wind": {
    "speed": 6.69,
    "deg": 310
  },
  "clouds": {
    "all": 0
  },
  "dt": 1757006653,
  "sys": {
    "type": 2,
    "id": 2003310,
    "country": "ES",
    "sunrise": 1756963963,
    "sunset": 1757010766
  },
  "timezone": 7200,
  "id": 3104324,
  "name": "Zaragoza",
  "cod": 200
}
```

Transformación y normalización

El objetivo de la **transformación** es preparar los datos para que puedan ser utilizarlos directamente.

Las transformaciones pueden incluir tareas como:

- Seleccionar campos, filtrar filas, ordenar, agrupar y aplicar funciones de agregación
- Realizar fusión de varias fuentes de datos
- *Feature Engineering*, transformación y creación de nuevas características a partir de datos existentes para mejorar la precisión y el rendimiento de los modelos
- **Transformar datos no estructurados**

Clasificación de los datos según formato y esquema

	Datos Estructurados	Datos No Estructurados	Datos Semiestructurados
Estructura	Modelo de datos predefinido y estricto	Sin modelo de datos predefinido	Estructura más flexible y jerárquica Uso de metadatos
Almacenamiento	Formato tabular Esquema rígido bases de datos relacionales y almacenes de datos*	Formato nativo, bases de datos NoSql, data lakes	Bases de datos NoSQL
Análisis	Fácil y rápido	Más complejo y requieren habilidades y herramientas especializadas	Moderadamente complejo
Ejemplos	Tablas, bases de datos Sql, hojas de cálculo, csv*	Textos, imágenes, videos, multimedia, IoT, social media	JSON, XML
Casos de uso	Machine learning (ML)	PLN Modelos IA generativa RAG (Generación Aumentada por Recuperación) Análisis predictivos	

Transformar datos no estructurados

Datos no estructurados : no tiene un formato predefinido o un modelo de datos fijo ni estandarizado.

PDFs, archivos multimedia (audios, imágenes y vídeos), contenido en redes sociales, logs de servidor, datos IoT, correos, etc.

Según Gartner entre el 80 y el 90 % son datos no estructurados.

Muy valiosos, pero difíciles de procesar directamente con herramientas de análisis tradicionales, especialmente para IA.

La información está ahí, pero hay que 'enseñarle' a los modelos a verla.

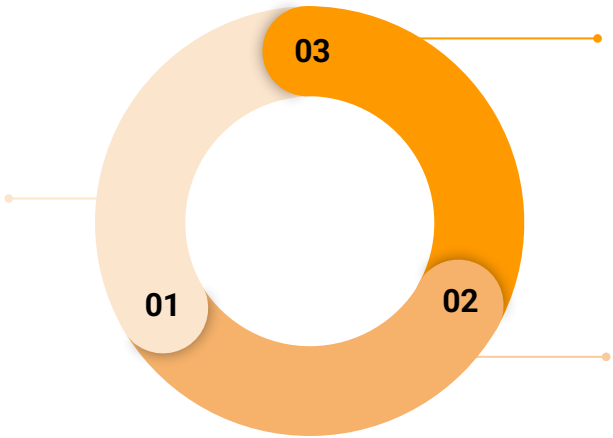
Transformar datos no estructurados

El Desafío: preparar los datos no estructurados para la IA

***Raw data* directamente de la fuente**
Diversas fuentes

Textos **sin procesar,** sin limpiar, imágenes sin etiquetar, audios sin transcripción

Es más complejo que los modelos aprendan de esto



Modelo IA

Se genera conocimiento

Preprocesamiento de los datos

El proceso de transformación: **limpieza y estructuración** que hace que el dato sea adecuado y útil para el entrenamiento de modelos

Transformar datos no estructurados

¿Cómo?

- Preprocesamiento texto, puede implicar tareas como eliminar *stop-words, stemming* y *vectorizing* en los datos de texto para prepararlos para los modelos
- Transformación de Imágenes y video
- Transformación de Audio

El objetivo es convertir el dato en una representación numérica (**vector**), que capture su significado.

Transformar datos no estructurados

```
import spacy

nlp = spacy.load("es_core_news_md")
text = nlp("""Los veía todas las mañanas durante esos días otoñales en que el cielo de Madrid finge sonrisas
    de primavera; sentados frente al borde de las aceras, la larga fila de vendedores ambulantes se extendía
    a lo largo de la calle.""")

# Filter stopwords using spaCy
refinedWords = [token.text for token in text if not token.is_stop]
print("Sin stop words:", refinedWords)
```

```
Sin stop words: ['veía', 'mañanas', 'otoñales', 'cielo', 'Madrid', 'finge', 'sonrisas', '\n    ', 'primavera',
';', 'sentados', 'frente', 'borde', 'aceras', ',', 'larga', 'fila', 'vendedores', 'ambulantes', 'extendía', '\n
', 'calle', '.']
```

```
for word in text:
    print(word.text + ' --> ' + word.lemma_, '[' + word.pos_ + ']')
```

```
Los --> él [PRON]
veía --> ver [VERB]
todas --> todo [DET]
las --> el [DET]
mañanas --> mañana [NOUN]
durante --> durante [ADP]
esos --> ese [DET]
días --> día [NOUN]
otoñales --> otoñal [ADJ]
vendedores --> vendedor [NOUN]
ambulantes --> ambulante [ADJ]
se --> él [PRON]
extendía --> extender [VERB]

 la --> el [DET]
 calle --> calle [NOUN]
 . --> . [PUNCT]
```

Transformar datos no estructurados

```
# Delete Stop-words using nltk

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Text unstructured data
text = "Los veía todas las mañanas durante esos días otoñales en que el cielo de Madrid finge sonrisas de primavera; sentados frente al bor

# Get Spanish stopwords and tokenize
liststopwords = set(stopwords.words('spanish'))
tokens = word_tokenize(text.lower())

# Remove stopwords
refinedTokens = [word for word in tokens if word not in liststopwords]

# print("Original:", tokens)
print(refinedTokens)
```

```
['veía', 'todas', 'mañanas', 'días', 'otoñales', 'cielo', 'madrid', 'finge', 'sonrisas', 'primavera', ';', 'sentados', 'frente', 'borde',
'aceras', ',', 'larga', 'fila', 'vendedores', 'ambulantes', 'extendía', 'largo', 'calle', '.']
```

```
from nltk.stem import PorterStemmer

porterStemmer = PorterStemmer()
stemmed_words = [porterStemmer.stem(word) for word in refinedWords]
print("Stemmed words:", stemmed_words)
```

```
Stemmed words: ['veía', 'mañana', 'otoñal', 'cielo', 'madrid', 'fing', 'sonrisa', 'primavera', ';', 'sentado', 'frent', 'bord', 'acera',
',', 'larga', 'fila', 'vendedor', 'ambulant', 'extendía', 'call', '.']
```

Transformar datos no estructurados

Ejemplo *Vectorizing*

```python
# Bag of Words (BoW)
from sklearn.feature_extraction.text import CountVectorizer

text = ["Los veía todas las mañanas durante esos días otoñales en que el cielo de Madrid finge sonrisas de primavera;",
        "sentados frente al borde de las aceras, la larga fila de vendedores ambulantes se extendía a lo largo de la calle."
       ]

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(text)

print(X.toarray())
print(vectorizer.get_feature_names_out())
```

```
[[0 0 0 0 0 1 2 1 1 1 1 1 0 0 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0 1]
 [1 1 1 1 1 0 3 0 0 0 0 0 1 1 0 1 2 1 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0]]
['aceras' 'al' 'ambulantes' 'borde' 'calle' 'cielo' 'de' 'durante' 'días'
 'el' 'en' 'esos' 'extendía' 'fila' 'finge' 'frente' 'la' 'larga' 'largo'
 'las' 'lo' 'los' 'madrid' 'mañanas' 'otoñales' 'primavera' 'que' 'se'
 'sentados' 'sonrisas' 'todas' 'vendedores' 'veía']
```