import pandas as pd

import ast # Para convertir strings en diccionarios

# Cargar el dataset

aircrafts\_data = pd.read\_csv("aircrafts\_data.csv")

# Extraer la columna JSON

aircrafts\_data["model"] = aircrafts\_data["model"].apply(ast.literal\_eval) # Convierte string en diccionario

aircrafts\_data["model\_en"] = aircrafts\_data["model"].apply(lambda x: x.get("en", "")) # Extrae el nombre en inglés

# Eliminar la columna original model

aircrafts\_data.drop(columns=["model"], inplace=True)

Guardar el archivo limpio

aircrafts\_data.to\_csv("aircrafts\_data\_cleaned.csv", index=False)

# Cargar el dataset

airports\_data = pd.read\_csv("airports\_data.csv")

# Extraer la versión en inglés de airport\_name y city

airports\_data["airport\_name"] = airports\_data["airport\_name"].apply(ast.literal\_eval)

airports\_data["airport\_name\_en"] = airports\_data["airport\_name"].apply(lambda x: x.get("en", ""))

airports\_data["city"] = airports\_data["city"].apply(ast.literal\_eval)

airports\_data["city\_en"] = airports\_data["city"].apply(lambda x: x.get("en", ""))

# Separar coordenadas en latitude y longitude

airports\_data[["longitude", "latitude"]] = airports\_data["coordinates"].str.strip("()").str.split(", ", expand=True).astype(float)

# Eliminar columnas innecesarias

airports\_data.drop(columns=["airport\_name", "city", "coordinates"], inplace=True)

# Guardar el archivo limpio

airports\_data.to\_csv("airports\_data\_cleaned.csv", index=False)

# Cargar el dataset

boarding\_passes = pd.read\_csv("boarding\_passes.csv")

# Eliminar duplicados

boarding\_passes.drop\_duplicates(inplace=True)

# Guardar el archivo limpio

boarding\_passes.to\_csv("boarding\_passes\_cleaned.csv", index=False)

# Cargar el dataset

bookings = pd.read\_csv("bookings.csv")

# Convertir book\_date a formato de fecha

bookings["book\_date"] = pd.to\_datetime(bookings["book\_date"])

# Filtrar valores negativos en total\_amount

bookings = bookings[bookings["total\_amount"] >= 0]

# Guardar el archivo limpio

bookings.to\_csv("bookings\_cleaned.csv", index=False)

import pandas as pd

import numpy as np

# Cargar los archivos CSV

flights = pd.read\_csv("/mnt/data/flights.csv")

seats = pd.read\_csv("/mnt/data/seats.csv")

ticket\_flights = pd.read\_csv("/mnt/data/ticket\_flights.csv")

tickets = pd.read\_csv("/mnt/data/tickets.csv")

2. Limpieza del Archivo flights.csv

Problemas detectados:

Algunas columnas de fechas (scheduled\_departure, scheduled\_arrival, actual\_departure, actual\_arrival) estaban en formato de texto.

Valores \N en algunas columnas, lo que indicaba datos faltantes.

Soluciones aplicadas:

Reemplazar \N por NaN para manejar datos faltantes.

Convertir las fechas a formato datetime para facilitar cálculos de tiempo.

python

Copiar

Editar

# Reemplazar valores "\N" por NaN

flights.replace("\\N", np.nan, inplace=True)

# Convertir las columnas de fechas a datetime

date\_columns = ["scheduled\_departure", "scheduled\_arrival", "actual\_departure", "actual\_arrival"]

for col in date\_columns:

flights[col] = pd.to\_datetime(flights[col], errors='coerce')

3. Limpieza del Archivo seats.csv

Problemas detectados:

Posibles duplicados en los asientos.

La columna fare\_conditions tenía valores categóricos que debían ser consistentes.

Soluciones aplicadas:

Eliminar duplicados para evitar registros erróneos.

python

Copiar

Editar

# Eliminar duplicados

seats.drop\_duplicates(inplace=True)

4. Limpieza del Archivo ticket\_flights.csv

Problemas detectados:

La columna amount debía ser numérica para cálculos de precios.

Se debía validar la consistencia de los datos.

Soluciones aplicadas:

Convertir la columna amount a tipo numérico.

python

Copiar

Editar

# Convertir "amount" a numérico

ticket\_flights["amount"] = pd.to\_numeric(ticket\_flights["amount"], errors='coerce')

5. Limpieza del Archivo tickets.csv

Problemas detectados:

Posibles duplicados en ticket\_no y book\_ref.

Soluciones aplicadas:

Eliminar registros duplicados basados en los identificadores clave.

python

Copiar

Editar

# Eliminar duplicados en ticket\_no y book\_ref

tickets.drop\_duplicates(subset=["ticket\_no", "book\_ref"], inplace=True)

6. Guardar los archivos limpios

Después de realizar las limpiezas, guardamos los archivos procesados.

python

Copiar

Editar

# Guardar los archivos limpios

flights.to\_csv("/mnt/data/cleaned\_flights.csv", index=False)

seats.to\_csv("/mnt/data/cleaned\_seats.csv", index=False)

ticket\_flights.to\_csv("/mnt/data/cleaned\_ticket\_flights.csv", index=False)

tickets.to\_csv("/mnt/data/cleaned\_tickets.csv", index=False)