Home Assignment

Introduction

PriceHubble gets a lot of data from different sources about properties for sale/rent, and each source has its own formats and conventions. So it can become messy very quickly!

The data is then processed and exposed internally to all internal teams (Data scientists, Data analysts, Business Intelligence …).

Goal

The goal of this assignment is to design a simple pipeline to process data from a single source, and to get a simplistic overview of the issues we could encounter on a daily basis.

Problem description:

You are given a dataset of offers that was scraped from one website.
However, the given dataset has some issues in it:

- The format of the dataset doesn't entirely fit our internal database schema
- There are some non valid offers that we would like to get rid of

You have to create a pipeline that would enable you to answer all these concerns, in order to have the data processed and conformed with our data format/schema.

UML-like diagrams or pseudo code are acceptable for the system design.

Input data set

File format

One input file is provided in this assessment. It contains properties scrapped in a json format. Each row represents one offer and its corresponding details.

Column	Type	Possible values
id	string	*any*
raw_price	string	*String containing a price*
living_area	float	*any*
property_type	string	house, studio, maisonnette,...
municipality	string	*any*
scraping_date	string	*YYYY-MM-DD*

Example

```
{
    "id": 290,
    "living_area": 45.6,
    "property_type": "house",
    "raw_price": "Loyer: 3741€/mo",
    "scraping_date": "2021-07-21",
}
```

Output data set

File format

The expected output format should follow the data model and specifications defined below:

Column	**Type**	**Possible values**	**Nullable**
id	string	*any*	*False*
price	float	*any*	*False*
living_area	float	*Between 10 and 500*	*False*
property_type	string	apartment / house	False
scraping_date	string	*YYYY-MM-DD*	*False*

Example

id	**price**	**living_area**	**property_type**	**scraping_date**
290	3741	45.6	house	2021-07-21

Questions:

1. Please provide a global design of the ETL pipeline to generate the output data set.

The solution needs to:

- Respect the output format as described above.
- Keep only the offers which meets these requirements:
    - ***price_per_square_meter*** between 500 and 15000,
    - ***property_type*** can only be either apartment or house,
    - ***scraping_date*** has to be in *YYYY-MM-DD* format.

A bit of Software Architecture is welcome where you would present the main objects that you would need to achieve the data transformation part (UML-like, pseudo code). You can ignore everything related to reading the data, saving the data for this part.

We welcome clear explanations in any format for your answers. You are not required to write perfect syntax python code, but just the bare minimum with either pseudo code or docstrings/comments. Here is a short example:

Python

```python
# Simplified example related to data read / write
class InputLoader
      def load(self, paths):
            """ Loads the data from a path and returns a pandas DataFrame. """

      def _get_paths(self):
            """ Returns the list of files to load the data from."""

class OutputLoader
      output_path ## Output path of the data

      def store(self, df):
            """ Stores a DataFrame to a given path. """
```

2. Enumerate the tools that you would be using (python packages, tools, external softwares, etc.).Explanations on why you would be using any of those technologies are welcome.

Unset

```
For example:
If you were to talk about exposing the data internally, you
could be using Big query, Postgres, any known Data
Warehousing technology like Snowflake, Redshift …
You would be using any Data Catalog Technology for internal
discovery: DataHub, DataPlex…
etc.
But you will need to explain why you would choose one over
the others.
```

3. What format would best fit this use case to store the output data ? (csv, xlsx, json …) and why?

4. Now assume that you have a pipeline that has billions of new records/terabytes of data that come on a regular basis. Would you change anything in the previous questions?

5. (Bonus) Knowing that the input is mostly unstructured and humanly inputted, can you think of pipeline steps that could be interesting to add to your current design?