Data Engineer Homework

BrokerChooser is an affiliate, users arrive on our site and convert to our partners' site. We want to connect our user data with brokers' data to analyze the user journey end-to-end. You are provided with multiple datasets:

Datasets

1. **brokerchooser_conversions.csv:** This is the dataset of our users' conversions.

- id: conversion ID
- session_id: sessions' ID
- country_name: country's name
- is_mobile: if the conversion happened on mobile, it is =1, otherwise 0
- ui_element: the UI element type
- created_at: timestamp
- measurement_category: page category where it happened. I give a page category mapping to read the meaning of these categories.

2. **broker_data.csv:** A dataset representing data received from one of our partners.

- timestamp: timestamp
- country_residency: The country that the users selected during the registration
- ip_country: The country by IP. It could differ from the residency.
- important_score: a score that shows the importance of the users. Higher scores mean higher quality users, meaning deposit more, and trade more.

3. **page_category_mapping.csv:** Mapping of the *measurement_category* attribute in brokerchooser_conversions.csv

Note: These data are synthetic, but represent a real-time situation for BrokerChooser.

Tasks

1. Your task is to create a data processing pipeline in Python that matches and normalizes the datasets. The output should be a cleaned, unified dataset that is ready for analysis. You must ensure that it is scalable to work with multiple batches of data.
2. Draw an ideal pipeline and write recommendations on how the ETL pipeline could be more accurate & automatized. What other tools would you use in this task?
3. Do an exploratory analysis of the matched data and bring insights and action steps to the product of BrokerChooser.

Objectives

1. **Data Ingestion:**

- Design a process to load the datasets into your pipeline. Consider how the data will be ingested for both one-time processing and multiple batches.

2. **Data Normalization:**

   Normalize the datasets to ensure consistency in data types, formats, and units. Consider any discrepancies that may arise between the internal and broker data.

3. **Data Matching:**

- Implement a matching mechanism between the “Our Data” and “Broker Data” datasets. This could involve joining on specific keys, using fuzzy matching.

4. **Data Processing Pipeline:**

- Design and implement a data processing pipeline that:
  - Ingests and processes data in batches.
  - Matches and normalizes the data.
  - Handles exceptions and errors.
  - Outputs a clean, unified dataset ready for further analysis.

Deliverables

1. **Code Implementation:**

- Provide a well-documented script that implements the data processing pipeline. Preferably provide a docker too.
- Ensure the code is well-organized, readable, modular, and follows best practices for Python development.
- Include the code used for exploratory analysis.

2. **Written tasks (PDF/slides):**

- Outline the ideal pipeline and provide recommendations for its implementation.
- Present insights and action steps for BrokerChooser’s product.