Senior Software Engineer Challenge, Seqana

Overview

Design and implement a scalable data pipeline to process and analyze geospatial soil sample data. This task is aimed at showcasing your ability to structure data pipelines and come up with an adapted data infrastructure.

The Task

You'll work with analyzing information from two different sources:

1. A dataset we provide of geospatial points, representing **soil samples** with measurements. This dataset should be processed and stored in the database of your choice. The records should be organized in a **normalized form**.
2. The boundaries of **European countries**, to be obtained from the **Overpass API**. This information should also be processed and stored in the database of your choice.

Out of those two data sources, we would like to create an analytical pipeline. This pipeline should receive a desired number of points as inputs, and executes through the following steps:

- **Associate** soil samples with the country they fall into.
- **Draw a random sample** of N points from each country, where N is an input parameter to your pipeline, and provide in return **summary statistics** for each country.
    - Mean of SOC% measurement values
    - Variance of SOC% measurement values
    - Average fraction of clay in your soil texture
- **Return the results** to the user as an output of your pipeline.

Additional points

- Update your analytical pipeline to sample a **cluster of points**, instead of individual, randomly selected points.
    - Organize the data within each country into clusters, whose size is adjusted for each country to match as closely as possible the number of samples required in input parameters.
    - Sample one cluster for each country in your data structure, and compute the summary statistics for the given input sample size.
    - If needed, suggest ways to retrieve the exact sample size provided by the user consistent with your clustering algorithm.
- Outline how the **system design** could evolve if clients wanted to submit their own geospatial areas to obtain the corresponding summary statistics.

Results

You are free to design the pipeline and architecture using the tools and workflow you see fit (Python-based). Make sure to include rationale for your design decisions, especially related to scalability, modularity, and maintainability.

- A **reproducible initialization script** and associated instructions that sets up the database, loads the corresponding data from the provided dataset as well as the API.
- An executable **data pipeline** executing the spatial operations, and returning the outputs as defined in the previous section.
- An **architectural diagram** and accompanying notes that explain your chosen design

Please create a **Github repository** with your code, and add jobs@seqana.com (seqana-jobs) as a collaborator, once the results are finalized.

Guidelines

- Use lightweight infrastructure tools as needed. Docker is optional but appreciated.
- Make sure your code is easy to read and structured into logical components, with clear instructions for running everything end-to-end.
- You may use AI tools to accelerate development, e.g. for boilerplate code. Make sure you understand and can justify every part of the solution.

Time Limit

- **Time-boxed to 4 hours**. Please note the time spent and document any areas you would prioritize if given more time.