/NDURENT

AI Engineer Assessment

Introduction

Company Context	Indurent is a Blackstone portfolio company specialising in commercial real-estate. We have offices in London and Stockport, with more than 200 assets spread across the UK, providing commercial spaces for over 2500 customers. We have a high-support results-focused culture, with a strong focus on curiosity, fairness and collaboration. Our goal is to become the leading multi-let industrial (MLI) business in the UK. A key part of this is investing in our assets and operating platform to enable us to be the most efficient and highest revenue generating company in our sector. We are passionate about revolutionising MLI property to provide great experiences for our customers. We do this by offering our customers high quality business space that is supported by the highest level of customer service so that they can concentrate on running their businesses. Indurent's technology vision is to deliver a world-class technology and data capability that enhances the MLI platform and enables the business to scale. The technology and data teams will contribute to growth and sustainability, while empowering our stakeholders and employees to achieve Indurent's vision of becoming the UK's leading MLI business.
Estimated Time	~2 hours
Goal	Demonstrate your ability to design Python logic for comparing marketed warehouse listings to an existing portfolio using AI techniques.

Background

Our company is building a system to help the Acquisitions team identify new warehouse listings that closely match the existing estate. The aim is to support strategic growth by maintaining a homogeneous portfolio over time.

We ask that you write backend logic to demonstrate:

- Vectorisation of structured data
- Retrieval-Augmented Generation (RAG)

- Prompt Engineering to guide AI analysis

We do not expect a full-stack or production-ready solution. We are interested in your thinking, coding style, and best practices.

Provided Data

You'll work with two Excel files:

1. **Current Portfolio.xlsx** – Existing estate properties
2. **Marketed Warehouses.xlsx** – New listings on the open market

Each file contains attributes such as:

- **Physical**: size, eaves height, number of doors
- **Geospatial**: latitude, longitude (can be used to derive proximity to cities, etc.)
- **Temporal**: build year

Note: The dataset includes real-world data quality issues (missing/inaccurate values).

Your Task

Write Python logic that:

1. Loads and preprocesses the datasets
2. Vectorises the relevant features
3. Applies RAG to enrich or contextualise the data
4. Uses prompt engineering to query an AI model (your choice of API)
5. Returns results based on user-defined prompts

Example Prompts

Your solution should be able to handle prompts like:

1. *"Find the 10 most similar properties in the estate to the newly marketed property."*
2. *"Provide a correlation score for the homogeneity of the marketed property(ies) with the rest of the estate, scoring each by physical characteristics, location, and age separately."*
3. *"Find the closest properties to the marketed property, after excluding any property in the portfolio that is more than 10 miles from a major city."*

Deliverables

Please submit:

- A Python script or notebook that demonstrates your approach
- Comments and docstrings explaining your logic
- (Optional) Output from testing the example prompts
- (Optional) Brief notes on how you'd extend this to a production system that sits behind a Web App.

What We're Looking For

- Clear, modular code
- Thoughtful handling of missing/inaccurate data
- Use of vectorisation and similarity metrics
- Creative use of RAG and prompt engineering
- Understanding of how AI APIs can be integrated
- Best practices in Python (structure, readability, documentation)

It is not required for the code to be successfully executable – we are looking for demonstrations of knowledge, coding style. and application of best practices. If you do end up with a working solution within the allotted time, providing the output from testing along with the example prompts you used would be a bonus.