Klynt

Klynt AG
(CHE-423.370.875), Obere Weidstrasse 3,
6343 Rotkreuz, Switzerland

Re: Internship Technical Assessment — Document Classification & Structuring

Context

Klynt is building an AI-powered platform to automate sell-side M&A execution. One interesting challenge is data room preparation — taking a messy folder of documents and turning it into a well-organized, professionally structured data room.

A PE buyer navigating a chaotic data room with files like "Report_Final_v2.pdf", "scan0042.pdf", and "Document (3).docx" scattered everywhere loses confidence instantly. How do you automatically understand what a document is and where it belongs? That's what we want you to explore.

Why This Challenge?

This problem is particularly interesting because there is no single right answer.

You could approach it with filename pattern recognition, metadata extraction (dates, authors, file properties), content analysis (OCR, text extraction, NLP), LLM-powered semantic classification, or hybrid approaches.

Each approach has trade-offs in accuracy, speed, cost, and handling edge cases. We want to see how you think, what choices you make, and how you justify them.

A Note on AI-Assisted Coding

Let's be real: everyone uses LLMs to code nowadays, and so do we.

Using AI tools is absolutely welcome — what matters is how you use them. Blindly copy-pasting ChatGPT output is easy to spot. Using AI as a smart pair-programmer to accelerate your work while understanding every line you ship? That's a skill we value.

Your Mission

Build a Proof of Concept (POC) for a Document Classification Agent that analyzes a set of files and proposes an organized structure.

Input: A messy folder of documents (PDFs, Word docs, spreadsheets, etc.)

Output: A classification report with: document type, suggested folder, confidence level, and any flags.

Key challenges to address:

Document Understanding: How do you figure out what a document actually is? A file named "scan0042.pdf" could be anything — a contract, an invoice, a financial statement. How deep do you go?

Classification Strategy: What taxonomy do you use? (e.g., Financial, Legal, Commercial, HR, Technical). How do you handle documents that fit multiple categories?

Confidence & Edge Cases: How do you handle uncertainty? What about documents you can't classify?

Deliverables

1. GitHub Repository containing a clean, well-structured codebase, a README with setup instructions and architecture overview, and sample test files.

2. Working POC that ingests a folder of documents, analyzes them (your choice: metadata, content, or both), classifies each document, and outputs a structured report.

3. Brief Technical Note (in README or separate doc) explaining your approach and why you chose it, trade-offs you considered, what signals you used for classification (and why), limitations and potential improvements, and time spent.

Bonus: Simple API or web interface to interact with your classifier.

Evaluation Criteria

Problem Approach: How you break down the problem, explore options, and make trade-offs.
Code Quality: Clean OOP design, separation of concerns, readability.
Git Practices: Meaningful commits, clear history.
Testing: Show us your tests. Unit tests, edge cases — we want to see how you ensure your code actually works.
Documentation: Clear README, architecture decisions explained.
Pragmatism: A working POC is better than a perfect but incomplete solution.

Submission

Share your GitHub repository link. Ensure the repo is PRIVATE and invite Sknuter. Deadline: 72h.

Questions?

Reach out at augustin@klynt.ai if anything is unclear.

Good luck!

The Klynt Team