Data Preprocessing

The preprocessing stage involved:

- **Conversion of Checklists:** The checklists were transformed into a JSON format, segmenting each point into individual queries.
- **Text Extraction from PDFs:** Municipal determination documents were downloaded in PDF format and converted into plain text using Python scripts. This ensured that the text was in a consistent format for further analysis.

Integration of the LLM

Integration of the LLM into the workflow was implemented via a Python-based pipeline:

- **Template-Based Querying:** A prompt template was developed to structure queries for the LLM. For each determination, the corresponding checklist was identified, and each point was queried individually using the template.
- **Automated Query Execution:** A Python program was created to loop over each document and checklist point, sending the prompt to the LLM and capturing the responses. The process is parameterized by:
  - Model type (e.g., "gpt-4o-mini" vs. a larger model)
  - Temperature settings (e.g., 0.0, 0.01, 0.5, 1.0) to assess consistency and output quality.
- **Output Processing:** The raw outputs from the LLM, which are often lengthy, are parsed using regular expressions to extract standardized responses (SI/NO/NON PERTINENTE). The results for each document are compiled into a CSV file for further analysis.

Workflow Diagram

A workflow diagram (to be included as Figure X) summarizes the entire process:

1. **Data Collection:** Download PDFs and extract text.
2. **Checklist Selection:** Match documents with their corresponding checklists.
3. **Prompt Generation:** Convert checklists into JSON and generate prompts.
4. **LLM Querying:** Send prompts to the LLM and receive responses.
5. **Response Extraction:** Use regex to parse and standardize responses.
6. **Data Analysis:** Compare LLM results with manually compiled checklists.

Implementation Challenges

During implementation, several challenges were encountered:

- **Text Extraction Issues:** Converting PDFs to clean text sometimes resulted in formatting problems or loss of information.
- **Prompt Engineering:** Designing prompts that reliably guided the LLM was iterative; adjustments were made to ensure clarity and precision in the responses.
- **Regex Limitations:** Extracting the standardized SI/NO/NON PERTINENTE responses from long texts required robust regular expressions, which sometimes needed fine-tuning to accommodate unexpected output variations.
- **Model Variability:** Different temperature settings and model sizes influenced the consistency of outputs, necessitating multiple pilot tests.

Pilot Tests

Before finalizing the experimental setup, several pilot tests were conducted:

- **Hyper-Parameter Tuning:** Experiments with various temperature settings helped determine the optimal balance between creativity and consistency in responses.
- **Validation:** Initial tests compared the LLM's responses with a small set of manually evaluated documents to fine-tune the prompt design and extraction process.
- **Iterative Refinement:** Feedback from pilot tests led to improvements in the prompt template, regex patterns, and overall processing pipeline, ensuring that the final system was robust and reliable.