LLMs for Automated Data Cleaning: Improving Data Quality with AI Assistance

Pavithra Purushothaman
Matrikel Nummer: 1535949
fd0001571

September 30, 2025

Abstract

When I work with data, I usually spend more time cleaning it than actually analyzing it. Fixing missing values, duplicates and inconsistent formats always slows me down. I wanted to see if a language model could help me with this process, not by automatically changing my data, but by giving me advice on what might be wrong. So I built a small tool that checks a dataset with pandas, then creates a summary of the whole dataset and sends a random sample to a local model to get suggestions. The tool only reports issues, and I stay in control of the cleaning decisions. I tested it on both synthetic and real data. Pandas always caught the obvious problems, while the model often pointed out things like inconsistent formatting or suspicious values. Everything runs locally, which keeps the data private. It's not a perfect solution, but it convinced me that AI can be a useful helper for data cleaning.

1 Introduction

For me, data cleaning is always the hardest and most boring part of a project. A file that looks fine at first usually hides problems: different date formats, numbers saved as text, extra spaces or repeated rows. It takes a lot of time and attention to find and fix everything.

At the same time, language models have become good at spotting patterns and explaining them. I started wondering if I could use one to speed up the cleaning process. My idea was simple: let the model suggest issues and fixes, but don't let it directly change the data. That way, I get help but still keep control.

To try this out, I built a small prototype. The workflow is: upload a CSV, run some quick checks with pandas. I also added a dataset summary step, so the model sees column statistics (like missing values, unique counts, and ranges) along with a sample of rows. The results are shown in a simple Gradio interface. This setup gave me a way to compare what pandas found with what the model suggested.

2 What Others Have Done

There are many existing tools that help with data cleaning. Excel has data validation and libraries like `pandas-profiling` can quickly detect missing values, duplicates or basic type errors. These are useful, but they usually stop at the obvious issues.

More advanced approaches try to automate transformations with AI. Some tools even let you describe what you want in plain English and the model generates code to clean the data. The problem is that if the AI makes a mistake, it might silently change values in a way we won't notice.

That's exactly what I wanted to avoid. My approach is lighter: the AI doesn't touch the data, it only gives suggestions. I think this is a safer way to use it, because I can review the output and decide what to apply.

3 How My Tool Works

The tool works in three steps:

1. First, it reads the CSV and runs baseline checks with pandas. This includes looking for missing values and duplicate rows.

2. Second, it builds a compact summary of all columns and sends that, plus up to 50 random rows, to the local model to a local AI model with a clear prompt: list possible issues, suggest cleaning steps, and add any notes.

3. Finally, it shows both the pandas report and the model's response in the web interface. I can also generate a "cleaned" version of the file where only safe fixes are applied, like trimming spaces or removing duplicates.

Everything runs locally with Ollama and the Gemma 2B model, so the data never leaves my machine. I built the interface with Gradio, which made it quick to put together.

3.1 Technical Architecture

Figure 1 shows how the components connect.

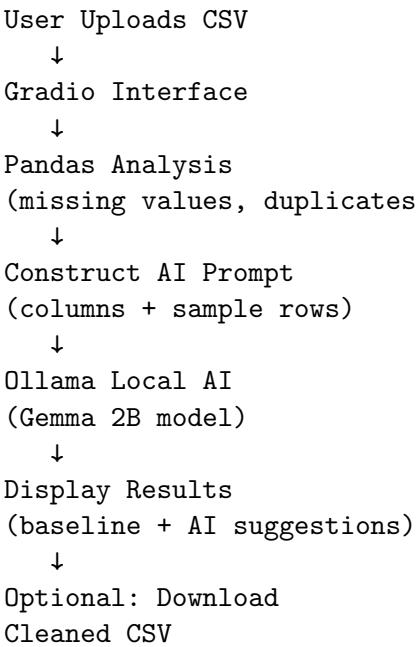

Figure 1: Flow from file upload to results

4 How I Tested It

I tested the tool with two types of datasets:

- A synthetic dataset of 20,000 rows where I manually added problems like inconsistent dates, numbers saved as text, missing values, and duplicates.
- A smaller real dataset to see how it performs on actual messy data.

For each run, I checked how long it took, what pandas reported, what the model suggested, and whether those suggestions made sense. I also looked at the automatically cleaned output to see what got fixed.

The Gemma 2B model is small, so it runs fast on a laptop. It's not the smartest model available, but it was good enough for this purpose.

5 What I Found

The baseline checks were reliable for obvious problems like missing values and duplicates. The model often added extra insight, like spotting inconsistent formats, outliers or numeric columns saved as strings.

For example, in my sample dataset, the model pointed out that a price column looked like text because of dollar signs and that dates were mixed between two formats. These are exactly the kinds of details that usually take me longer to notice.

The tool usually responded within 10–20 seconds, which felt fast enough. The first run required downloading the model, but after that it worked offline. The safe cleaning step handled simple fixes correctly and avoided risky changes.

5.1 Example Output

Listing 1: Sample Output

```
--- Baseline ---

Pandas baseline detected: Missing values found, Duplicate rows found

--- LLM Suggestions ---

**1) Possible data quality issues:**
- Date column has inconsistent formats
- Price column stored as text with dollar signs
- Some product names have extra spaces
- Quantity column has very large values

**2) Cleaning steps:**
- Standardize dates
- Remove dollar signs and convert price to numeric
- Trim whitespace from product names
- Review high quantity values

**3) Additional notes:**
- Dataset summary + random sample (up to 50 rows) provided to the model, full dataset
    coverage is summarized via stats
```

```
- Domain-specific checks may be needed

_Time taken: 3.2s_
```

6 What This Means

From my tests, I think AI can be a good assistant for data cleaning, but only when it stays in an advisory role. If the model directly changed the data, I wouldn't trust it. But as a helper that points out possible issues, it saves time and gives me ideas for what to check.

The tool is not perfect. The model doesn't see every row directly, but it does get a full dataset summary plus a representative random sample (up to 50 rows). This way it can suggest column-level issues while still being limited by its context size. It also doesn't understand the meaning of the data, so it can't flag domain-specific errors. But even with these limits, it still gave useful hints.

7 Technical Details

The main tools I used were:

- **Pandas** for reading files and baseline checks
- **Gradio** for the web interface
- **Ollama** to run the model locally
- Standard Python libraries for handling files and timing

The AI prompt was designed to keep answers short and practical. It asked for three sections (issues, steps, notes) and explicitly told the model not to make things up or include code.

The cleaning routine is conservative. It trims text, tries to convert obvious numbers and dates, and removes duplicates. If something is uncertain, it leaves it unchanged. This keeps the output safe and predictable.

If the AI fails or times out, the tool still shows the baseline results, so it's still useful.

7.1 Code Structure

Listing 2: Main Analysis Function

```python
def analyze_file(file):
    df = pd.read_csv(file.name)
    issues = []
    if df.isnull().sum().sum() > 0:
        issues.append("Missing values found")
    if df.duplicated().sum() > 0:
        issues.append("Duplicate rows found")

    prompt = f"""
    Dataset summary: [stats for each column]
    Random sample of up to 50 rows:
    {sample_df.to_string(index=False)}
```

```
    Write your answer in these sections:
    **1) Possible data quality issues**
    **2) Cleaning steps**
    **3) Additional notes**
    """

    llm_response = query_ollama(prompt, model="gemma:2b")
    return baseline_report, llm_response
```

8 How to Reproduce My Results

To run the tool, you need:

- Python 3.7 or newer
- pandas, gradio, and other required libraries
- Ollama installed locally
- The Gemma 2B model (or another model) downloaded

The steps are in the README. Once installed, you can upload any CSV and test it. The results are deterministic for the cleaning step, so running it again on the same file will give the same output.

9 Conclusion

This project showed me that AI can be helpful for data cleaning if it is used carefully. My prototype combines simple pandas checks with model suggestions, giving me both reliable detection of obvious problems and hints about trickier issues.

The tool is not complete, but it works well enough to show the potential. I plan to keep experimenting with better models, more cleaning steps, and maybe ways to let users give feedback. For now, it already makes data cleaning less painful and keeps me in control of the process.

10 Reflection

Working on this project showed me both the strengths and weaknesses of using language models for data cleaning. On the positive side, the model was able to spot issues that are hard to catch with simple rules like inconsistent formatting or suspicious values. It often gave me concrete suggestions that matched what I would have done manually. This confirmed my idea that AI can act as a useful assistant.

At the same time, the project also made the limitations very clear. The model sometimes missed issues that I expected it to catch, especially if they did not appear in the sample rows. It also cannot understand the meaning of the data, so it might suggest changes that look correct technically but do not make sense in context. For example, it cannot know whether a very high number is a real outlier or just a valid value.

Another lesson was about prompt design. I learned that if the instructions are too vague, the model will produce long or speculative answers. By making the prompt specific and structured, I could reduce this problem and get more practical results.

Overall, I see the tool as a step in the right direction. It is not something that replaces human effort, but it makes the first stage of cleaning easier for us. For future work, I would like to test larger models, integrate more systematic validation checks and experiment with ways for the tool to learn from user feedback. This would bring it closer to a reliable assistant that adapts to real world needs.

References

[1] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. Detecting data errors: Where are we and what needs to be done? *Proceedings of the VLDB Endowment*, 9(12), 2016.

[2] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD)*, pages 1723–1726, 2017.

[3] Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–33, 1996.

[1, 2, 3]

A The AI Prompt

Listing 3: AI Prompt Template

```
You are a data cleaning assistant.

Dataset summary (entire file):
{dataset_summary}

Random sample of rows (up to 50):
{sample_df.to_string(index=False)}

Write your answer in exactly 3 sections:
**1) Possible data quality issues**
- short bullet points
**2) Cleaning steps**
- practical actions
**3) Additional notes**
- limits or cautions

Rules:
- Do not invent columns or values
- Do not include code
- Keep it short and clear
```

B What the Automatic Cleaner Does

1. Trim spaces in text columns
2. Try converting numeric looking columns to numbers
3. Try parsing date like columns as dates
4. Remove duplicate rows
5. Save the cleaned result as a new CSV

If a step is uncertain, the cleaner leaves the data unchanged.

C Step by Step Results

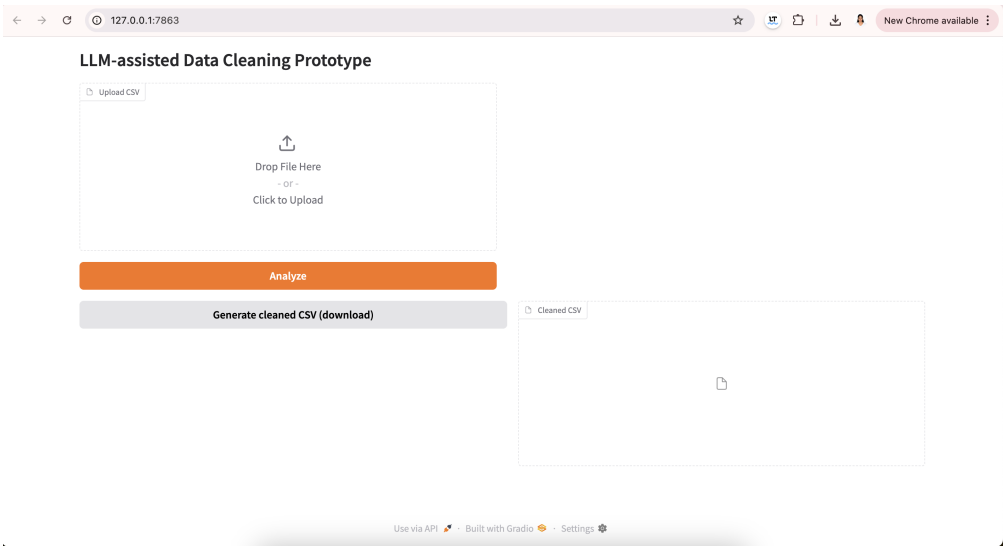

Figure 2: Step 1: Upload a CSV file

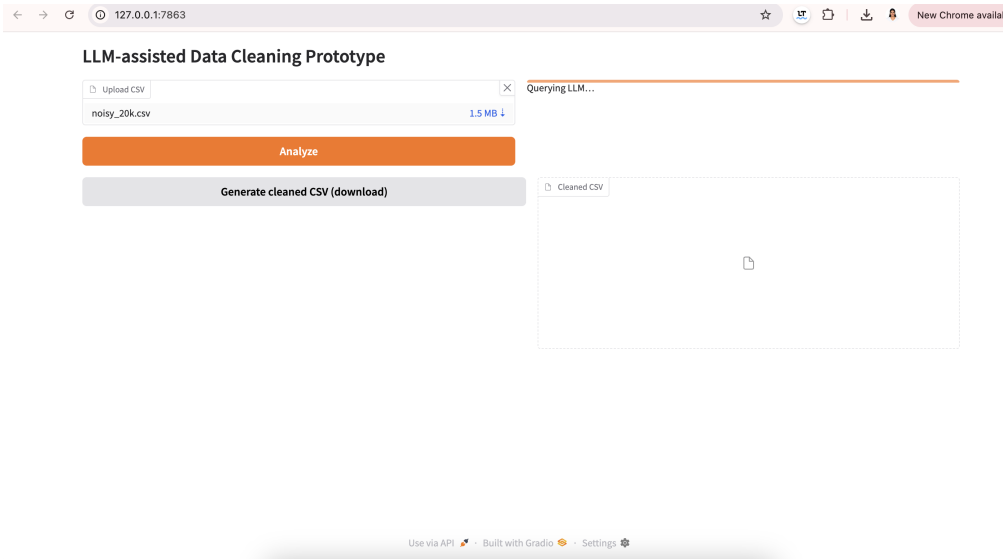

Figure 3: Step 2: Pandas + LLM analysis

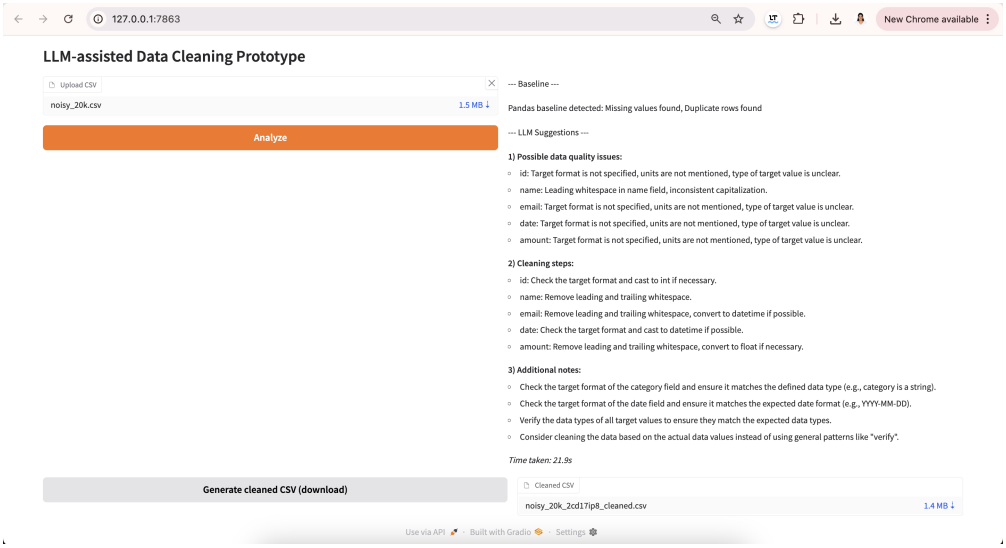

Figure 4: Step 3: Analysis Result and cleaned CSV