Architecture Overview

The system uses a **rule-based semantic engine** that examines actual column values (not headers) to determine whether a column contains phone numbers, company names, countries, dates, or generic text.
This enables the pipeline to work on messy, unlabeled datasets.

Input CSV

Raw CSV data is loaded without relying on column names.

Semantic Classifier

This file is processed by the **semantic classifier**.

Parsing Module

The parsing module analyzes **phone and company data**.

Output CSV

The output is generated in a **new CSV format**.

MCP Tools

Finally, tools like **list_files, predict_column,parse_file** enhance user interaction.

Prepared By - Ayushman Pal

User Journey with MCP

File Selection

User selects a CSV to process. The system loads the file without relying on column names.

Column Classification

Each column is analyzed using **rule-based scoring** to determine if it contains **phone numbers, company names, dates, countries, or generic text.**

Best Columns

The highest-confidence **PhoneNumber** and **CompanyName** columns are chosen for parsing.

Values Parsing

Selected columns are normalized **(Phone → Country & Number**, **Company → Name** & **Legal)**, and the finalized structured CSV is produced.

Model Design Highlights

- Hybrid rule-based scoring ensures reliable detection even on small datasets
- Suffix reconstruction algorithm handles broken multi-word legal suffixes from raw legal.txt
- Modular architecture: classifier, parser, MCP server are independent units
- LLM-friendly interface: MCP server provides clean JSON tool responses