GenAI-Powered Data Engineering Agent Workflow

1. Format Detection and Schema Inference

Overview

The agent scans data to determine file formats and infer schemas. It first generates synthetic data using LLMs and then analyzes file structures.

Steps

- Initialize LLM

```
llm = init_chat_model(
    "us.anthropic.claude-3-5-haiku-20241022-v1:0",
    model_provider="bedrock_converse",
    region_name="us-east-1",
    client=bedrock_client
)
```

- Create Tools
    - `@tool def generate_synthetic_data()`
    - `@tool def save_file_to_parquet()`
    - `@tool def upload_file_to_s3()`
- Write Prompt
- Create Agent

```
agent = create_react_agent(llm, prompt, tools)
```

- Detect File Formats
    - Use filename extensions to identify format (JSON, CSV, XML, etc.).
    - Use content-based analysis for more accurate detection.
        - Identify column names and datatypes in CSV.
        - Analyze structural aspects of JSON/XML.
- Create Detection Tools
    - `@tool def detect_file_formats()`
    - `@tool def detect_schema()`
- Enhance Accuracy
    - Incorporate RAG/in-context learning if needed.

2. Code Generation

Overview

The agent generates Python code to read, clean, and transform data into standardized formats.

Steps

- Generate Code to Read Files
    - Based on file type detected in the previous step.
- Validate Data with Pydantic
    - Ensure only valid data is processed.
- Standardization to Parquet
    - Convert data into Parquet format.
- Data Cleaning & Transformation
    - Generate transformation logic based on detected schema.
- Save Transformed Data
    - Store data in AWS S3 or locally based on file size.
- Enhance Accuracy
    - Utilize RAG/in-context learning for schema improvements.

3. Code Execution

Overview

The system automates running the generated code, ensuring proper execution flow.

Steps

- Guide Multi-Agent Execution
    - Ensure all agents run at the correct steps.
- Agents Involved
    - Data Generation Agent – Generates synthetic data.
    - Format Detection Agent – Identifies file formats.

  - Schema Inference Agent – Extracts schema from files.
  - Pydantic/Parquet Agent – Validates and standardizes data.
  - S3 File Storage Agent – Saves data to AWS S3.
  - Code Execution Agent – Runs generated Python code.

4. Testing Data

Overview

To evaluate the system, we use synthetic and real-world datasets.

Data Types

- CSV, JSON, XML, Parquet, Avro, GeoJSON, YAML.

Real-World Data

- Additional datasets may be used to validate performance.

5. Evaluation Metrics

Goals

To ensure high performance and accuracy, we evaluate:

- File Type Detection Accuracy
  - Compare agent predictions vs actual file types.
- Code Accuracy
  - Validate with unit tests or LLM-based evaluation.
- Performance Metrics
  - Measure execution time, token usage, and efficiency.
- Scalability Testing
  - Assess system performance on datasets of varying sizes.
- Consistency Across Formats
  - Test the same dataset in multiple formats to ensure uniform results.

Summary

This workflow enables a fully automated multi-agent data engineering system that detects, processes, and transforms structured data. By leveraging LLM-based automation, schema inference, and dynamic code execution, the system enhances data engineering efficiency, standardization, and accuracy.