ETL Pipeline Project: Agriculture Crop Yield Forecasting

Technical Documentation - By Arsalan

1. Pipeline Design & Architecture

The ETL pipeline integrates data from five sources to produce an enriched dataset:

1. Google Drive: Contains crop_yield.csv for historical crop yield data.

2. Google Sheets: Stores field-level inputs such as irrigation and fertilizer.

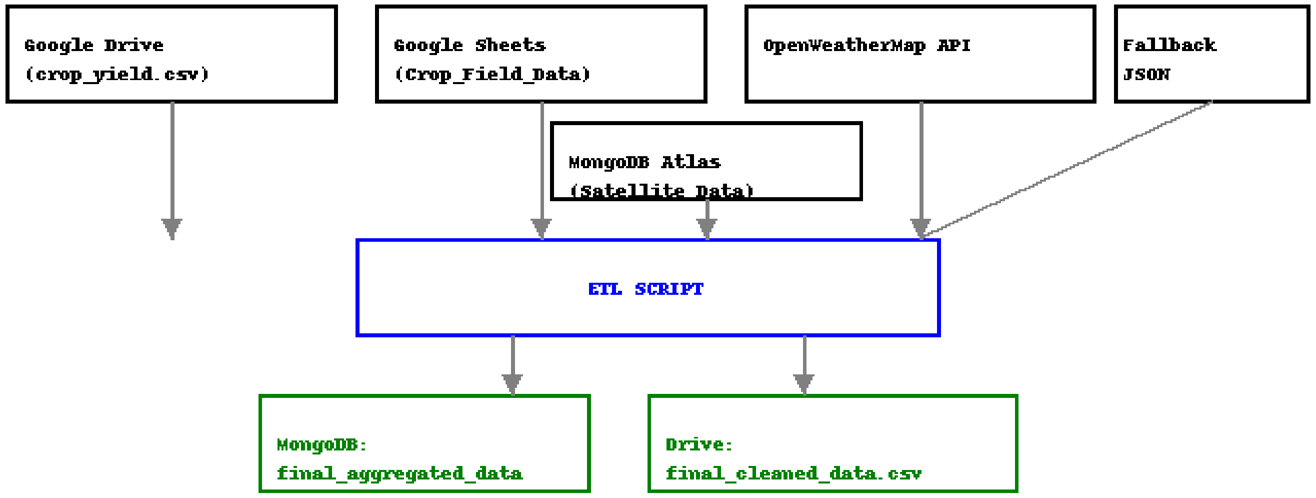
3. OpenWeatherMap API: Provides real-time weather data (temperature, humidity, wind speed).

4. MongoDB Atlas: Stores satellite metrics like NDVI and soil moisture.

5. sample_weather.json: A fallback JSON file used if the real-time weather API fails.

Final data is processed in Python and saved to MongoDB and also written to a CSV file on Google Drive.

Pipeline Design Diagram



2. Technology & Tool Choices

- Python: Rich data libraries and scripting support
- schedule: Task scheduler
- MongoDB Atlas: Flexible cloud NoSQL DB
- GitHub Actions: For CI/CD automation

3. CI/CD Integration

- Runs unit tests using pytest
- Validates output schema
- Executes ETL pipeline on every push to main

4. CI/CD Reliability Benefits

- Reduces manual errors
- Provides quick feedback on code
- Maintains data integrity via schema validation
- Speeds up deployment cycles

5. Data Normalization & Analytical Readiness

The ETL pipeline handles data format and unit inconsistencies by:

- Converting area units from hectares to acres for uniformity
- Mapping categorical fields like 'Irrigation' (Yes/No) to numeric (1/0)
- Merging data from JSON, API, CSV, and MongoDB collections into a single structured DataFrame
- Handling missing values with drop or fallback options

The final DataFrame allows analysts to:

- Visualize crop profitability by region
- Detect weather-related yield patterns
- Model relationships using NDVI, soil moisture, and field inputs

- Conduct time-series or geospatial analysis on agricultural trends

6. Missing Items & Next Steps

- The REST API used for real-time weather data may fail due to key limits or rate limits. In such cases, fallback to sample_weather.json is used.

- CI/CD was implemented using GitHub Actions but has not yet been tested with pull requests (PRs).

- Screenshots of test logs and deployments were not included due to time constraints.

- Future improvements include GitHub PR testing, CI badges, and production deployment triggers.