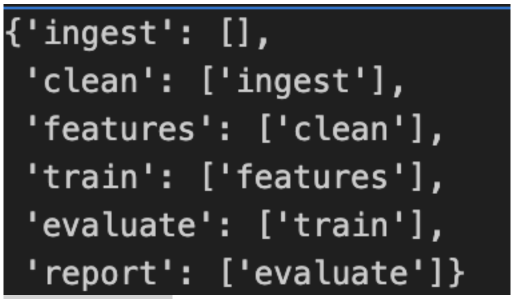
My project orchestrates six jobs: ingest raw data from the source API/database; clean the data no missing variables and noise; engineer featuring to create column I desire for further training; train a baseline model; evaluate with core and subgroup metrics to have a better idea of the data; and generate a stakeholder report to enhance reproducibility.

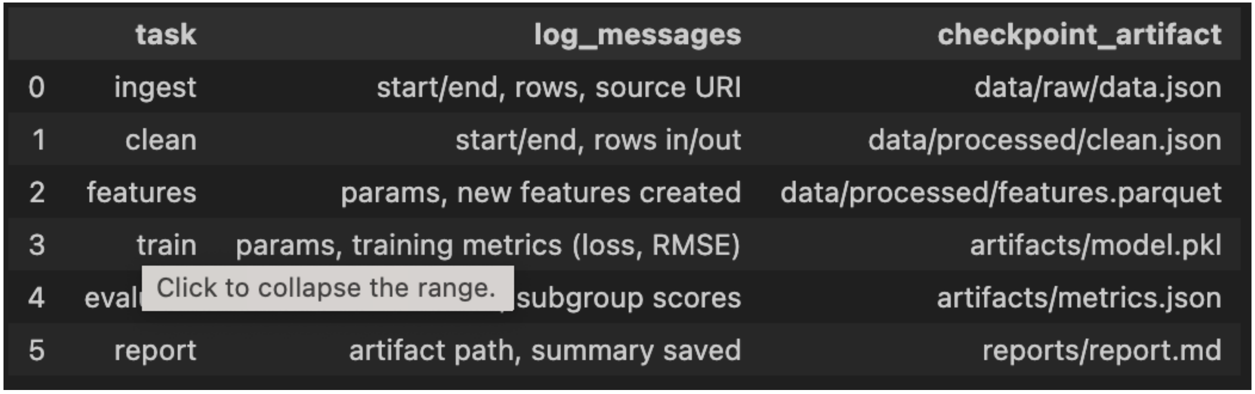
The dependency flow is linear—ingest → clean → features → train → evaluate → report.



```
{'ingest': [],
 'clean': ['ingest'],
 'features': ['clean'],
 'train': ['features'],
 'evaluate': ['train'],
 'report': ['evaluate']}
```

Dag Program

Logging captures per-task start/end timestamps, parameters, row counts in/out, warnings/errors, and key metrics.



	task	log_messages	checkpoint_artifact
0	ingest	start/end, rows, source URI	data/raw/data.json
1	clean	start/end, rows in/out	data/processed/clean.json
2	features	params, new features created	data/processed/features.parquet
3	train	params, training metrics (loss, RMSE)	artifacts/model.pkl
4	eval	, subgroup scores	artifacts/metrics.json
5	report	artifact path, summary saved	reports/report.md

I will automate ingest, clean, features, train, and evaluate (they are frequent and benefit most from repeatability, retries, and scheduling), while keeping the final report semi-manual to allow human interpretation, narrative edits, and audience-specific framing before distribution.