Level 0 Maturity : Manual Process

→ Manual build, train, line, and deploy model { Jupyter Notebook }

→

Characteristics.

① Manual | Script Driven | Interactive process.

② Disconnection between ML and operations teams.

③ Infrequent release of models | no CD of models.

④ No CI. or CI is ignored.

⑤ Deployment ⇒ prediction service

✗ No ML pipeline deployed.
✓ No Cont. Training.

⑥ Lack of monitoring of performance.

Challenges : -

① Time consuming

② Maintenance cost → high

③ No new ML ideas will pushed easily

④ Prone to bugs.

Possible solution:

(1) Actively monitor your model in production.

(2) frequently (depends on the situation) retrain your model.

(3) frequent experiments with new optimized implementation to produce better models.

for eg: Choose or experiment with latest SOTA

Make it more adaptable / Robust

Level 1 maturity : ML pipeline automation.

Aim

→ perform CT of the model in production by automating ML pipeline ⇒ CD of model in prediction service

→ To automate :—

→ new data ⇒ fresh model based on latest trend

→ data & model validation

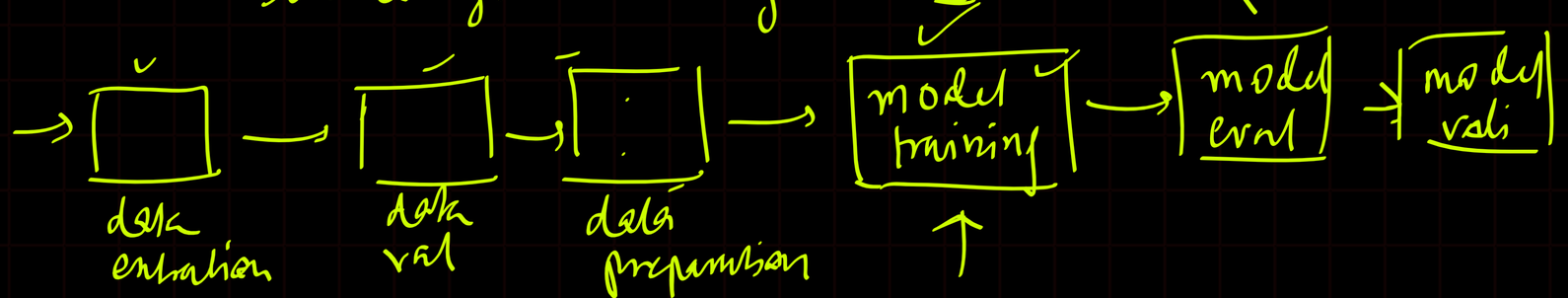
→ pipeline trigger & metadata management

→ Pipeline Orchestration

Characteristics :-

1 Rapid Experiment

because you are using orchestration



2. CT of model in production

3. Experimental - Operational Symmetry.

Experimental Env

Pre Prod | Prod

4. Modularized code components and pipeline :-

→ Reusable

→ Shareable

5. CD of model.

6. Pipeline deployment-

deploy pipeline → deploys model

Frequently used terms :-

① Pipeline Orchestration
ordered execution of components.

earlier failure of M2 project :-
↳ Glue code | code to connect two components of pipeline.

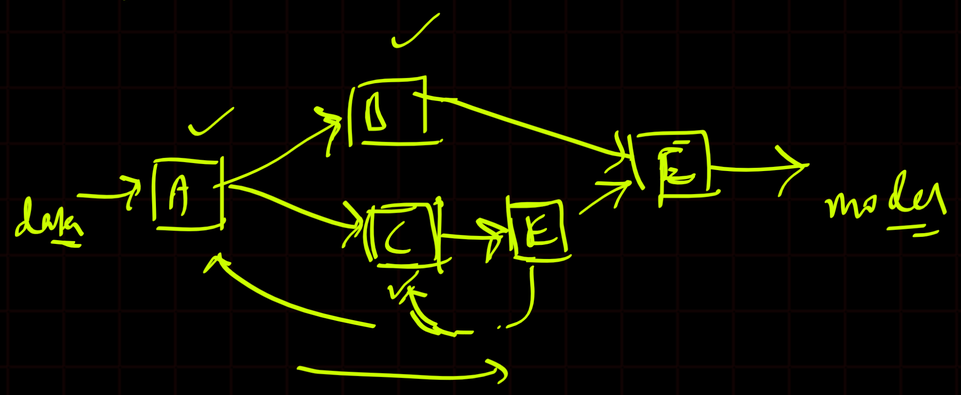
Tools to solve above issue :-
① Apache Beam
② " Airflow → UI
③ Kubeflow (Kubernetes based) } supported by major cloud platforms

Advantage :-
① Standard Orchestration + abstraction
② Supported by many cloud platforms
③ Easy to monitor / debug.

UI

DAGs → Directed Acyclic Graphs.

Graphs →



cyclic → ∞ loop

ML pipelines

DAGs → core component of all your pipeline orchestration tools.

Level 1. → Important components.

pipeline execution

new data / live data → new model version on new data

Requirements to fulfil above execution :-

① Automatic data validation

↳ Decides whether to start retraining

or

stop execution of pipeline. & so for manual investigation {DS}

on following basis :-

Ⓐ Data schema skews :-

↳ data is not in compliance with expected schema or as per DSA → Data sharing agreement

Solution :- → stop pipeline & let DS team investigate.

serve last stable model

or

maintenance time out { worst case }

(5) Data Validation skew :-

When data pattern or statistical properties changes

↓

Trigger retraining.

Requirement

(2) Model Validation

↳ After successful model training on new data

↓

Evaluate & validate your model before pushing it into production.

Offline steps	Online steps
before serving your model	~~model is serving~~ after deployment

offline steps :-

(1) Get evaluation metrics using your trained model on Test data set
to
know its prediction quality.

(2) Compare above metrics with current- model in production

(3) Performance } your model must be consistent-
on regional / class- samples