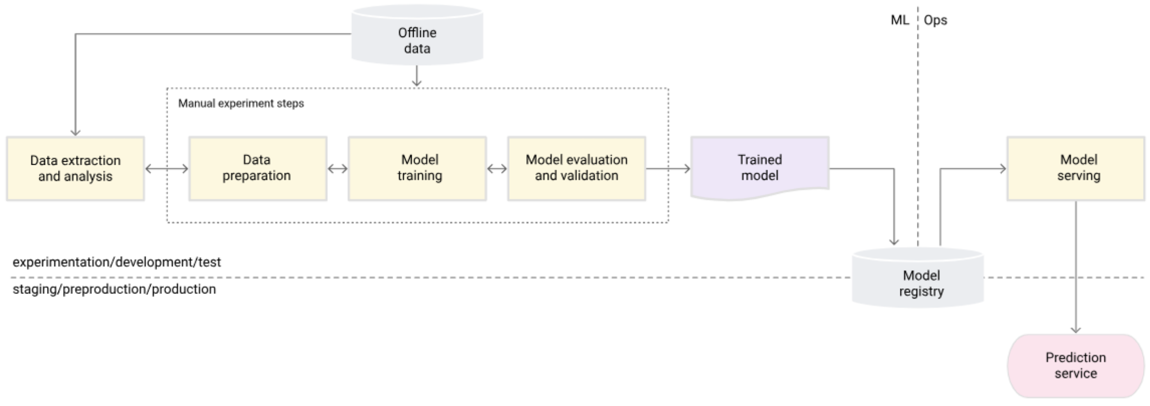
➔ MLops Level 0 orchestration (Everything is manual).

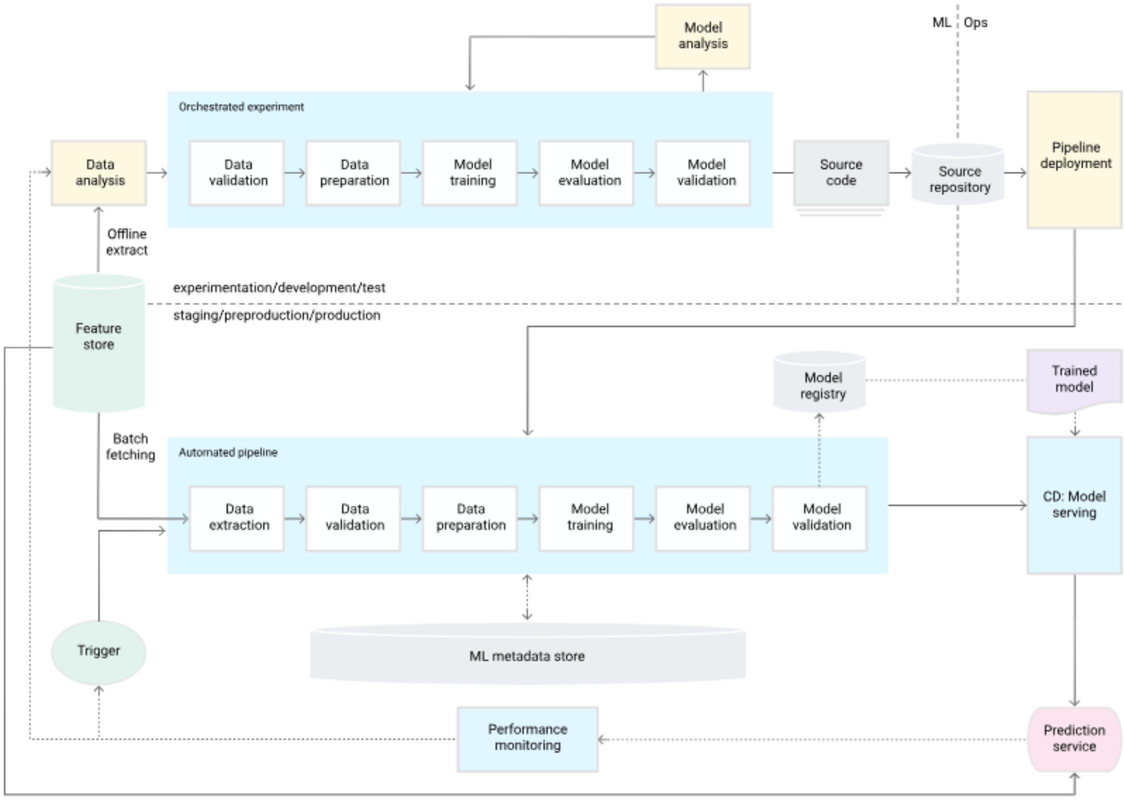


Characteristics

The following list highlights the characteristics of the MLOps level 0 process, as shown in Figure 2:

- Manual, script-driven, and interactive process: Every step is manual, including data analysis, data preparation, model training, and validation. It requires manual execution of each step, and manual transition from one step to another. This process is usually driven by experimental code that is written and executed in notebooks by data scientist interactively, until a workable model is produced.
- Disconnection between ML and operations: The process separates data scientists who create the model and engineers who serve the model as a prediction service. The data scientists' hand over a trained model as an artifact to the engineering team to deploy on their API infrastructure. This handoff can include putting the trained model in a storage location, checking the model object into a code repository, or uploading it to a models registry. Then engineers who deploy the model need to make the required features available in production for low-latency serving, which can lead to *training-serving skew*.
- Infrequent release iterations: The process assumes that your data science team manages a few models that don't change frequently—either changing model implementation or retraining the model with new data. A new model version is deployed only a couple of times per year.
- No CI: Because few implementation changes are assumed, CI is ignored. Usually, testing the code is part of the notebooks or script execution. The scripts and notebooks that implement the experiment steps are source controlled, and they produce artifacts such as trained models, evaluation metrics, and visualizations.
- No CD: Because there aren't frequent model version deployments, CD isn't considered.
- Deployment refers to the prediction service: The process is concerned only with deploying the trained model as a prediction service (for example, a microservice with a REST API), rather than deploying the entire ML system.
- Lack of active performance monitoring: The process doesn't track or log the model predictions and actions, which are required to detect model performance degradation and other model behavioural drifts.

➔ MLops level 1 orchestration (Automating the model training).



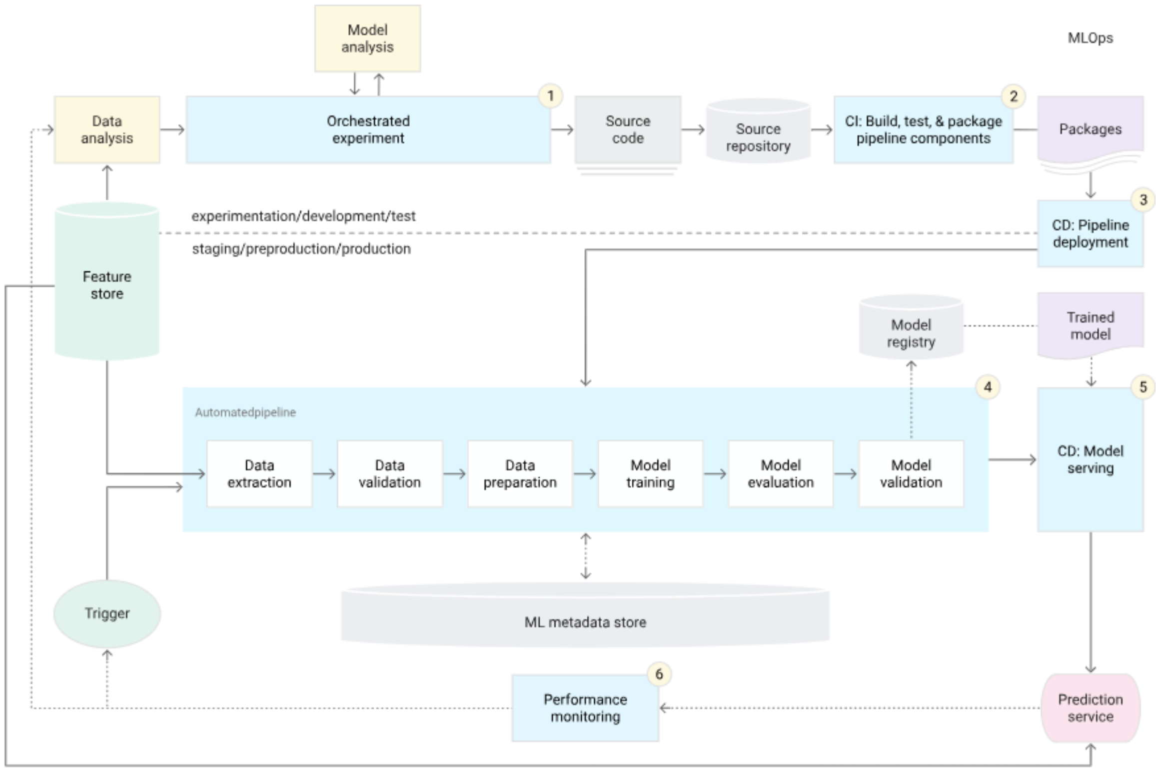
Characteristics

The following list highlights the characteristics of the MLops level 1 setup, as shown in Figure 3:

- Rapid experiment: The steps of the ML experiment are orchestrated. The transition between steps is automated, which leads to rapid iteration of experiments and better readiness to move the whole pipeline to production.
- CT of the model in production: The model is automatically trained in production using fresh data based on live pipeline triggers, which are discussed in the next section.
- Experimental-operational symmetry: The pipeline implementation that is used in the development or experiment environment is used in the preproduction and production environment, which is a key aspect of MLops practice for unifying DevOps.
- Modularized code for components and pipelines: To construct ML pipelines, components need to be reusable, composable, and potentially shareable across ML pipelines. Therefore, while the EDA code can still live in notebooks, the source code for components must be modularized. In addition, components should ideally be containerized to do the following:
  - Decouple the execution environment from the custom code runtime.
  - Make code reproducible between development and production environments.
  - Isolate each component in the pipeline. Components can have their own version of the runtime environment and have different languages and libraries.
- Continuous delivery of models: An ML pipeline in production continuously delivers prediction services to new models that are trained on new data. The model deployment step, which serves the trained and validated model as a prediction service for online predictions, is automated.

- Pipeline deployment: In level 0, you deploy a trained model as a prediction service to production. For level 1, you deploy a whole training pipeline, which automatically and recurrently runs to serve the trained model as the prediction service.

➔ MLops Level 2: CI/CD pipeline automation.



Characteristics/Steps:

1. Development and experimentation: You iteratively try out new ML algorithms and new modeling where the experiment steps are orchestrated. The output of this stage is the source code of the ML pipeline steps that are then pushed to a source repository.
2. Pipeline continuous integration: You build source code and run various tests. The outputs of this stage are pipeline components (packages, executables, and artifacts) to be deployed in a later stage.
3. Pipeline continuous delivery: You deploy the artifacts produced by the CI stage to the target environment. The output of this stage is a deployed pipeline with the new implementation of the model.
4. Automated triggering: The pipeline is automatically executed in production based on a schedule or in response to a trigger. The output of this stage is a trained model that is pushed to the model registry.
5. Model continuous delivery: You serve the trained model as a prediction service for the predictions. The output of this stage is a deployed model prediction service.
6. Monitoring: You collect statistics on the model performance based on live data. The output of this stage is a trigger to execute the pipeline or to execute a new experiment cycle.

The data analysis step is still a manual process for data scientists before the pipeline starts a new iteration of the experiment. The model analysis step is also a manual process.