

Design, develop, deploy and iterate on production-grade ML applications

Prof. Gjorgji Madjarov, PhD

Faculty of Computer Science and Engineering

Ss. Cyril and Methodius University Skopje

Outline

- State of ML, challenges and growth
- ML lifecycle
- MLOps paradigm
- DevOps vs MLOps
- MLOps Maturity Levels
- MLOps: Essential Best Practices

State of Machine Learning

• Today

- 53% of POCs make it into production
 - Average 9 months
- Gartner



Last decade

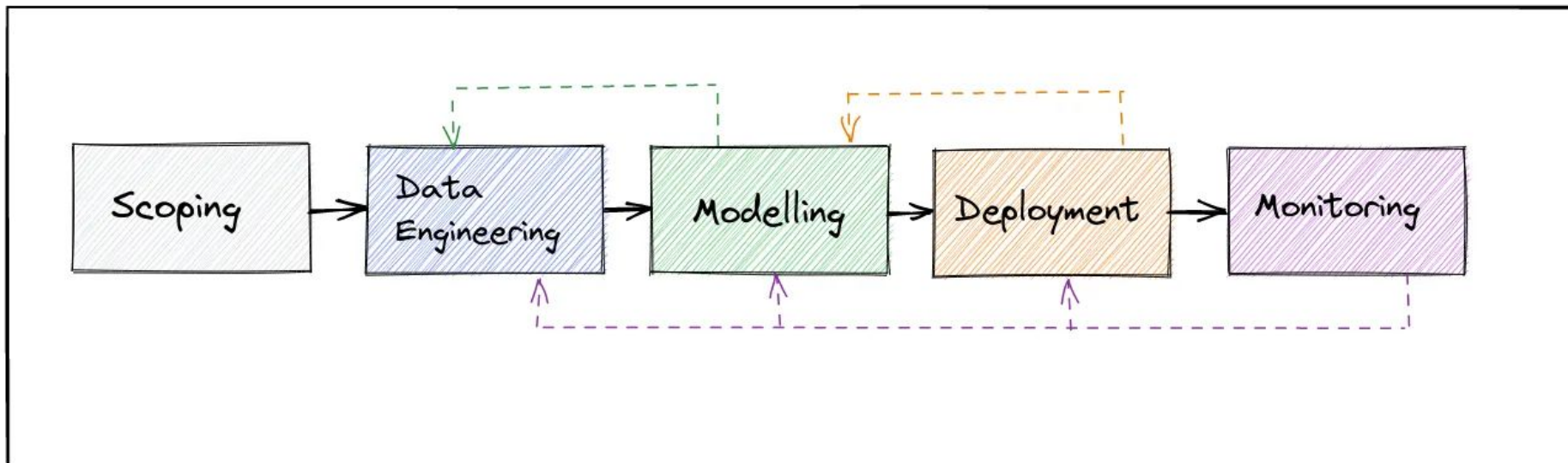
- Focusing mostly on building ML models
- Operationalization was an afterthought



By end of 2024

- 75% of organizations will shift from piloting to operationalizing AI
- Gartner

ML lifecycle



*<https://towardsdatascience.com/a-gentle-introduction-to-mlops-7d64a3e890ff>

Machine Learning Challenges

- ML models rely on a huge amount of data, difficult for a single person to keep track of.
- Difficult to keep track of parameters we tweak in ML models.
 - Small changes can lead to enormous differences in the results.
- We have to keep track of the features the model works with, feature engineering is a separate task that contributes largely to model accuracy.
- Monitoring an ML model isn't like monitoring a deployed software or web app.
- Debugging an ML model is an extremely complicated art
- Models rely on real-world data for predicting, as real-world data changes, so should the model.
 - We have to keep track of new data changes and make sure the model learns accordingly.

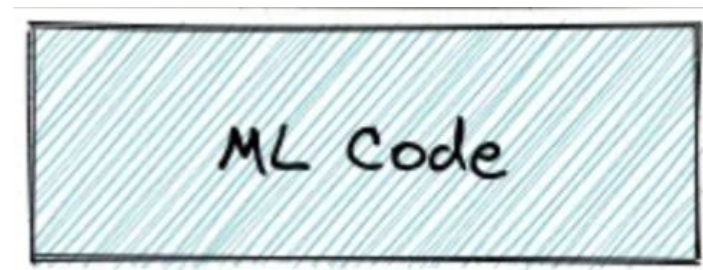
What does it mean for the industry?

- Machine Learning is rapidly becoming a key technology across industries.
- **High level challenges:**
 - Many companies struggle to integrate ML into their existing infrastructure.
 - “...creating a model is only a small part of the whole picture”*
- **Key point:**
 - Focus on managing the entire lifecycle, not just building models.

*Sculley, David, et al. "Hidden technical debt in machine learning systems." Advances in neural information processing systems 28 (2015).

Phase 1: Research/Experiment

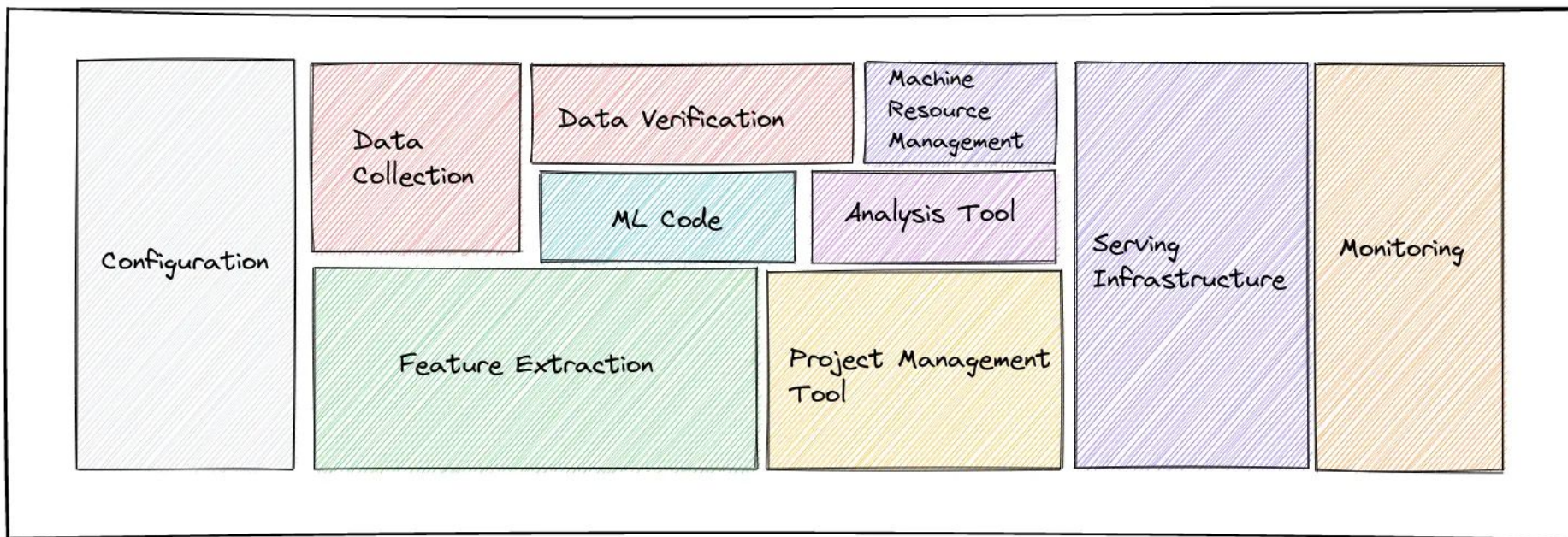
- Question: “Can we use ML to solve this?”
 - “Is it possible to ... ?”
 - “Can we use this data to solve the following problem?”
 - “Surely we must be able to ...”
- Typical scenarios
 - Scientific projects
 - Proof-of-concepts (PoCs)



Phase 2: Operational

- Question: “How do we implement this method at scale?”
 - How do we pipe the data into the model in a timely fashion?
 - How do we collect, store and transform data so models can be retrained consistently?
 - How do we build an A/B testing environment, in order to test future model iterations?
- Typical scenarios
 - After PoC, bringing your ML models to production
 - Migration of existing models into ML platform

Phase 1 + Phase 2



What is MLOps?

- MLOps is a set of practices for:
 - efficiently managing the creation,
 - deployment,
 - monitoring, and
 - maintenance of ML models.
- MLOps bridges the gap between data science and IT operations
 - **Automating and streamlining the ML lifecycle**

DevOps: A Starting Point for MLOps?

- DevOps practices like CI/CD have transformed software deployment by automating key steps.
- DevOps to MLOps: ML models rely heavily on data, adding complexity.
- DevOps focuses on code, MLOps handles both code and data pipelines.

Key DevOps Concepts: CI and CD

- Continuous Integration (CI): Code is continuously integrated into a shared repository and tested.
- Continuous Delivery (CD): Software is frequently built, tested, and deployed, allowing rapid iterations.
- CI/CD ensures ML models are integrated and production-ready.

DevOps vs MLOps

- **MLOps is experimental in nature** - most of the activity of data science teams relates to experimentation. Teams constantly change features of their models to achieve better performance, while also managing an evolving codebase.
- **Hybrid teams** - data science teams include both developers (machine learning engineers) and data scientists or researchers who analyze data and develop models and algorithms.
- **Continuous testing (CT)** - in addition to the regular testing stages of a DevOps pipeline, such as unit tests, functional tests and integration tests, an MLOps pipeline must also continually test the model itself - training it and validating its performance against a known dataset.

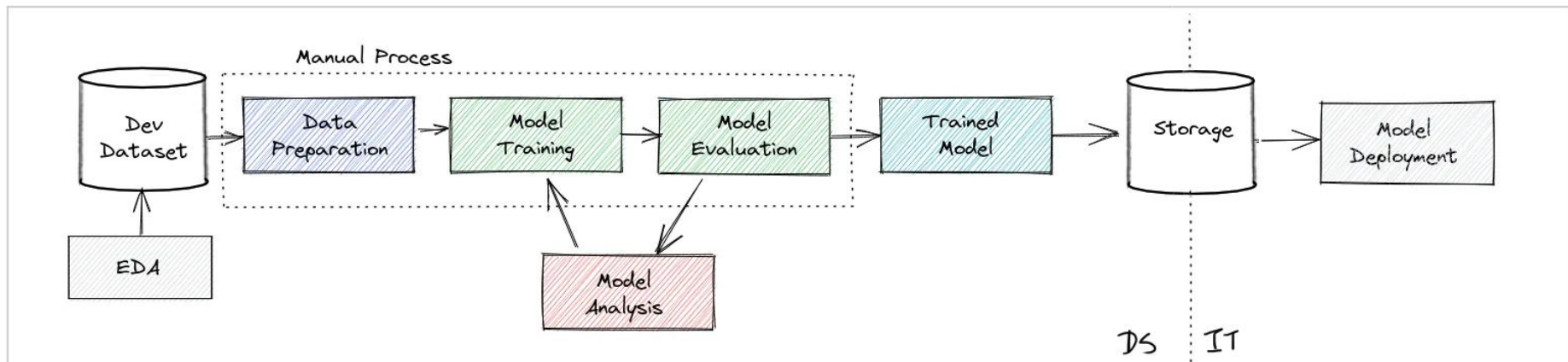
DevOps vs MLOps

- **Automatic retraining** - in most cases, a pre-trained model cannot be used as-is in production. The model needs to be retrained and deployed on an ongoing basis. This requires automating the process data scientists go through to train and validate their models.
- **Performance degradation** - unlike regular software systems, even if a model is working perfectly, performance can degrade over time. This can happen due to unexpected characteristics of data consumed by the model, differences between training and inference pipelines, and unknown biases which can grow with each feedback loop.
- **Data monitoring** - it is not sufficient only to monitor a model as a software system. MLOps teams also need to monitor the data and predictions, to see when the model needs to be refreshed or rolled back.

MLOps Maturity Levels

- Fully Manual Systems: Low automation, high error risk.
- Automatic ML Pipelines (no CI/CD): Some automation, but manual intervention needed.
- Automatic ML Pipelines (with CI/CD): Full automation, enabling rapid iteration.

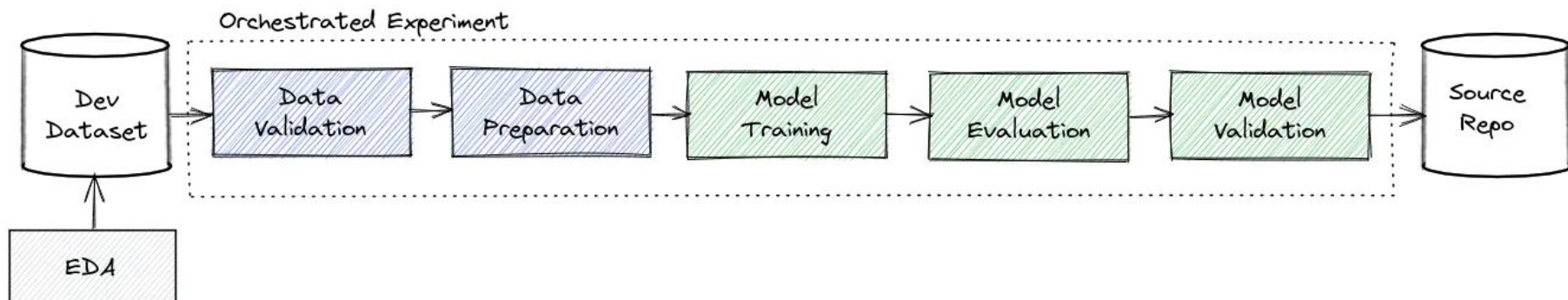
MLOps: Manual Process



MLOps: Manual Process

- At this stage of development, a team can create functional machine learning (ML) or deep learning (DL) models, but the process of deploying them into production is entirely manual. The ML workflow is structured as follows:
- Each step in the workflow, such as data analysis, preparation, model training, and validation, is performed manually or using experimental code within Jupyter Notebooks.
- Data scientists operate independently from engineers, who handle the deployment of the final model as a low-latency prediction service. The data science team provides the trained model to the ML engineers, who are responsible for exposing it via an API. This separation between development and production environments can result in training-serving skew.
- Model releases are infrequent, based on the assumption that once the data science team has completed the model, it can be put into production.
- There is no CI/CD pipeline because the model is not expected to change often. As a result, there is no focus on automating the build process for model code (CI) or automating the deployment of the prediction service (CD).
- Ongoing monitoring of model performance is not implemented, with the assumption that the model will continue to perform consistently as it encounters new data.

MLOps: ML Pipeline Automation

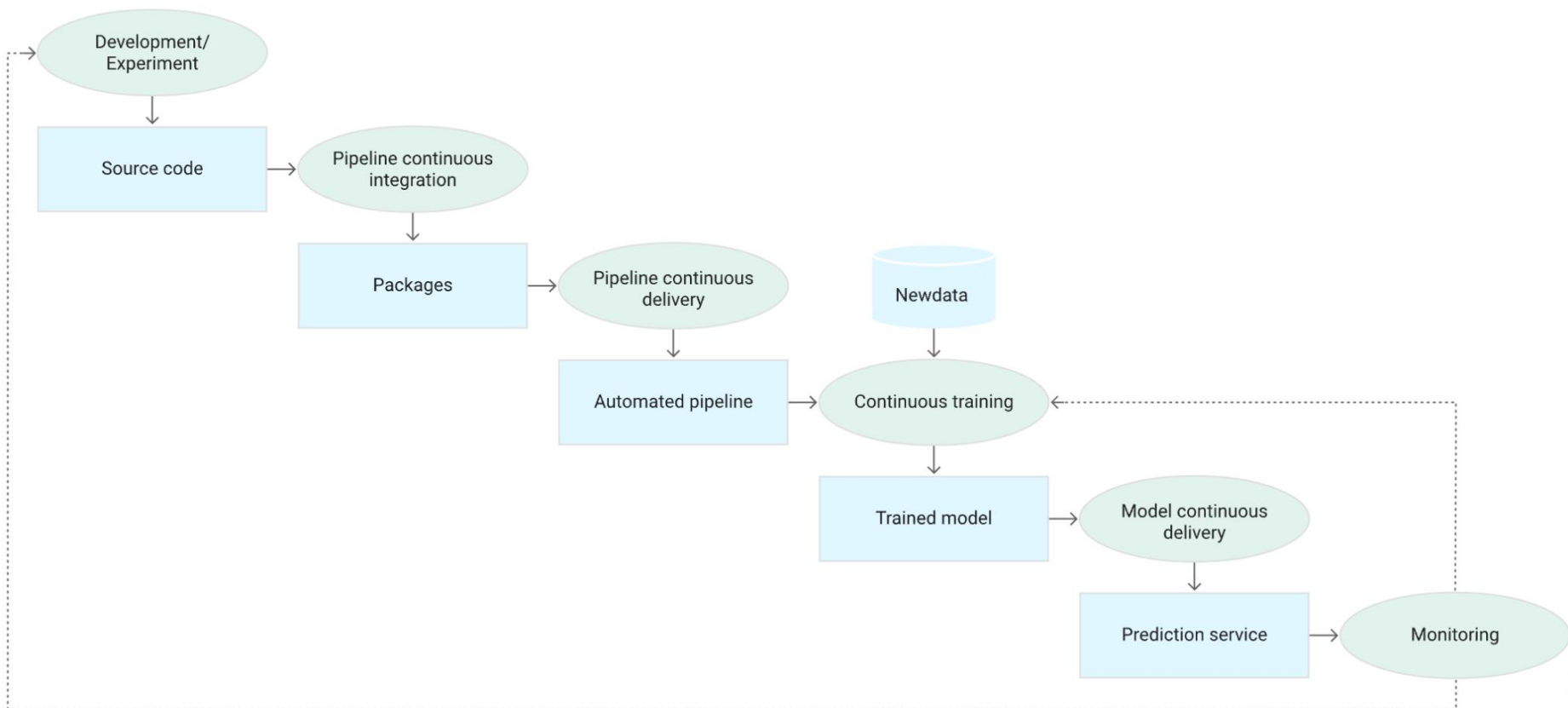


*<https://towardsdatascience.com/a-gentle-introduction-to-mlops-7d64a3e890ff>

MLOps: ML Pipeline Automation

- At this stage, teams recognize the need to manage models within a CI/CD pipeline, with continuous training and validation on new data. The ML pipeline evolves to look like this:
 - Experiments are conducted more quickly thanks to automation throughout the ML process. Data scientists can formulate a hypothesis and deploy it to production with minimal delay.
 - The model undergoes continuous testing and retraining with updated data, driven by feedback from real-time model performance.
 - The experimental and production environments are aligned to prevent discrepancies between model training and serving (training-serving skew).
 - All components used for building and training the model are designed to be reusable and shareable across multiple pipelines.

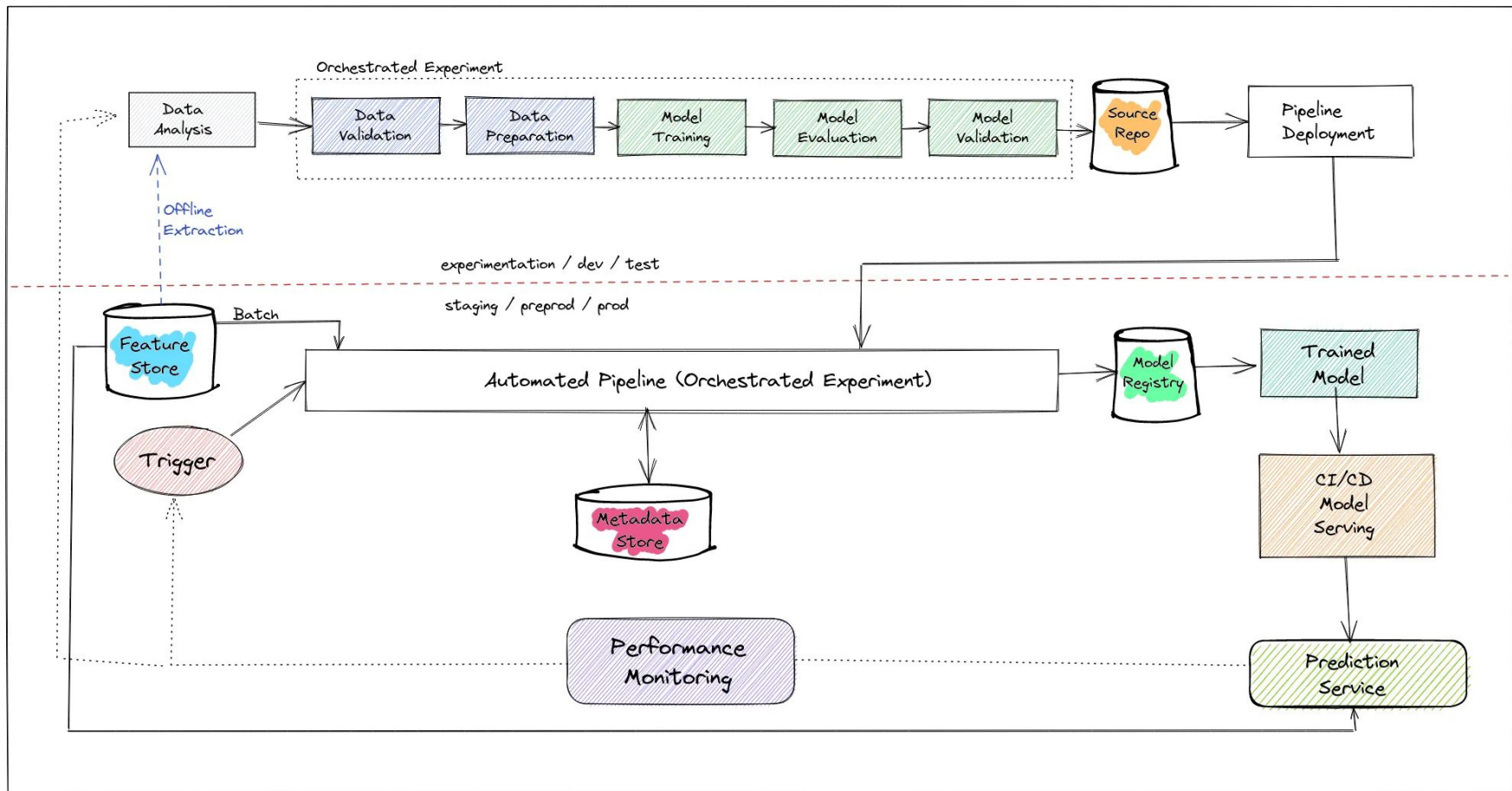
MLOps Level 2: Full CI/CD Pipeline Automation



*[https://cloud.google.com/architecture/images/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning-5-st](https://cloud.google.com/architecture/images/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning-5-stages.svg)

ages.svg

MLOps Level 2: Full CI/CD Pipeline Automation



MLOps: Essential Best Practices

- **Automate Model Deployment**

- Consistency
- Faster time-to-market
- Seamless updates

- **Keep the First Model Simple and Build the Right Infrastructure**

- Faster iteration
- Easier debugging
- Scalability
- Integration
- *To build a robust infrastructure, consider the following components:*
 - Data ingestion
 - Model training
 - Model deployment
 - Monitoring and logging
 - Security and compliance

MLOps: Essential Best Practices

- **Enable Shadow Deployment**
 - Validation
 - Risk mitigation
 - Performance comparison
 - *General steps of how this process can work:*
 - Infrastructure
 - Data routing
 - Model outputs
 - Monitoring and evaluation
- **Ensure Data Labeling is Strictly Controlled**
 - Develop clear labeling guidelines
 - Train and assess annotators
 - Use multiple annotators
 - Monitor and audit the labeling process

MLOps: Essential Best Practices

- **Use Sanity Checks for External Data Sources**
 - Data validation
 - Detect anomalies
 - Monitor data drift
- **Write Reusable Scripts for Data Cleaning and Merging**
 - Modularize code
 - Standardize data operations
 - Automate data preparation
 - Version control for scripts
- **Enable Parallel Training Experiments**
 - Accelerated model development
 - Efficient resource utilization
 - Improved model performance
 - Experiment management
- **Evaluate Training Using Simple, Understandable Metrics**
 - Alignment with business objectives
 - Interpretability
 - Trade-offs

MLOps: Essential Best Practices

- **Automate Hyper-Parameter Optimization**
 - Grid search
 - Random search
 - Bayesian optimization
 - Genetic algorithms
 - Gradient-based optimization
 - *Incorporating HPO into an MLOps pipeline can have significant benefits, including:*
 - Improved model performance
 - Increased efficiency
 - Consistency and reproducibility
 - Continuous improvement
- **Continuously Monitor the Behaviour of Deployed Models**
 - Detecting model drift
 - Identifying issues
 - Maintaining trust
 - Compliance and auditing
 - *Continuous monitoring in MLOps typically involves:*
 - Performance metrics
 - Data quality
 - Resource usage
 - Alerts and notifications

MLOps: Essential Best Practices

- **Enforce Fairness and Privacy**
 - Assess fairness
 - Use privacy-preserving techniques
 - Regularly review policies
- **Improve Communication and Alignment Between Teams**
 - Establish clear objectives
 - Maintain documentation
 - Hold regular meetings
 - Use version control

MLOps tools and platforms



Amazon SageMaker



databricks



TensorFlow Extended



lakeFS

Conclusion

- MLOps is critical for maintaining and scaling ML models.
- Automation in the ML pipeline unlocks faster innovation.
- Companies that embrace MLOps will gain a competitive advantage.

Questions?