# Full-stack Machine Learning

Alexander D’hoore

# Full-stack Machine Learning

- Introduction to MLOps
- Comparison to DevOps

- Experiment Tracking
- **Lab 1**

- Data Engineering
- **Lab 2**

- Model Deployment
  - Cloud, edge
  - Prototyping
- **Lab 3**

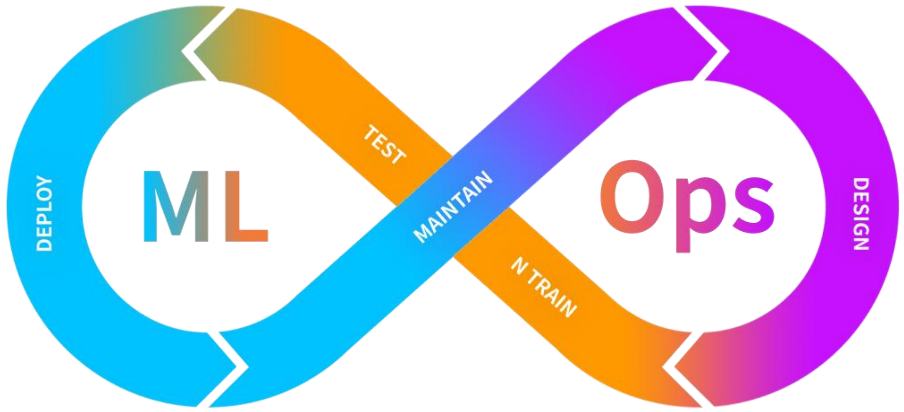
- Automation
  - Agents, pipelines
  - Hyperparameters
- **Lab 4**

# Introduction to MLOps

# What is MLOps?

- MLOps stands for **Machine Learning Operations**

- It is a set of practices that aims to
- **deploy and maintain** machine learning models
- in production **reliably and efficiently**.



# Why MLOps is Useful?

- **Facilitates collection** between data science and operations, by providing a systematic approach to managing ML workflows

- **Ensures reliability** and reproducibility of ML models by standardizing processes and enabling version control

- **ML lifecycle** involves several stages: data collection, preprocessing, model training, deployment, and monitoring

# Key Components of MLOps (1)

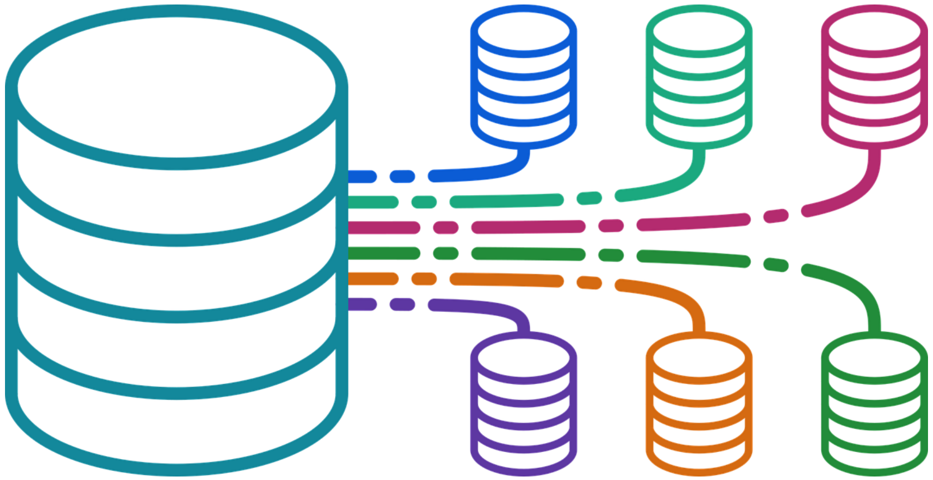
- **Data Engineering**: Involves data collection, preprocessing, and management
- **Model Development**: Iterative process of training, validating, and optimizing ML models
- **Model Deployment**: Reliably deploying the model to production
- **CI/CD Pipelines**: Automating training and deployment pipeline

# Key Components of MLOps (2)

- **Monitoring and Logging**: Track model performance, detect anomalies and data drift
- **Versioning and Reproducibility**: Record all changes to code, data and models
- **Closing the loop**: Automatic retraining of models, to ensure system stays up-to-date

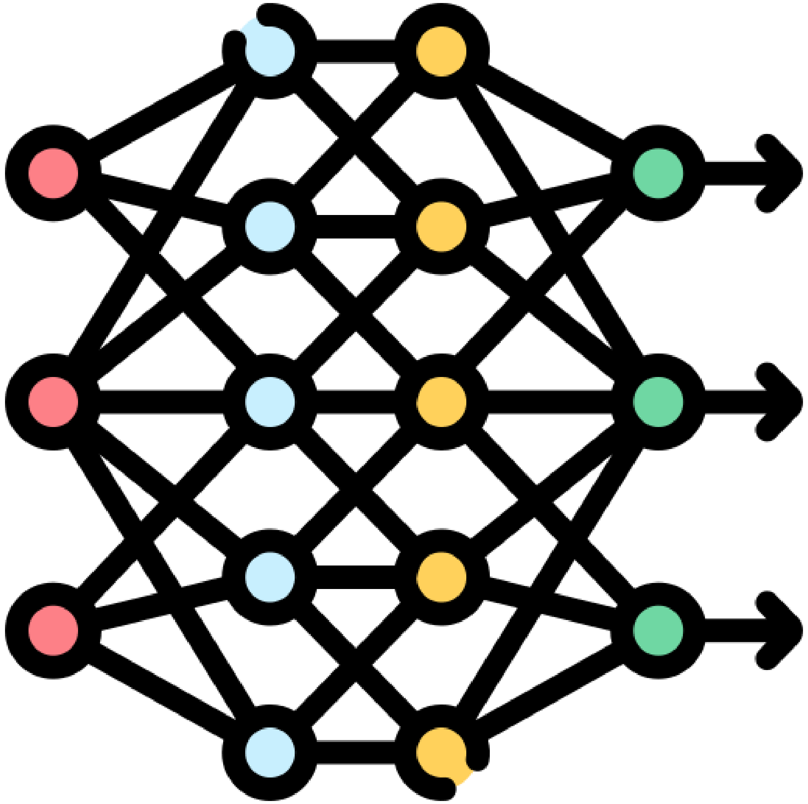
# Data Engineering in MLOps

- **Data Collection**: Gathering raw data from various sources
- **Data Labelling**: Annotate data with meaningful labels
- **Data Preprocessing**: Clean, transform, and normalize the data
- **Data Management**: Store and version data efficiently and securely



# Model Development in MLOps

- **Model Training**: Using data to train machine learning models
- **Model Validation**: Measure performance on unseen data (accuracy, precision, recall)
- **Hyperparameter Tuning**: Optimize model parameters for better performance



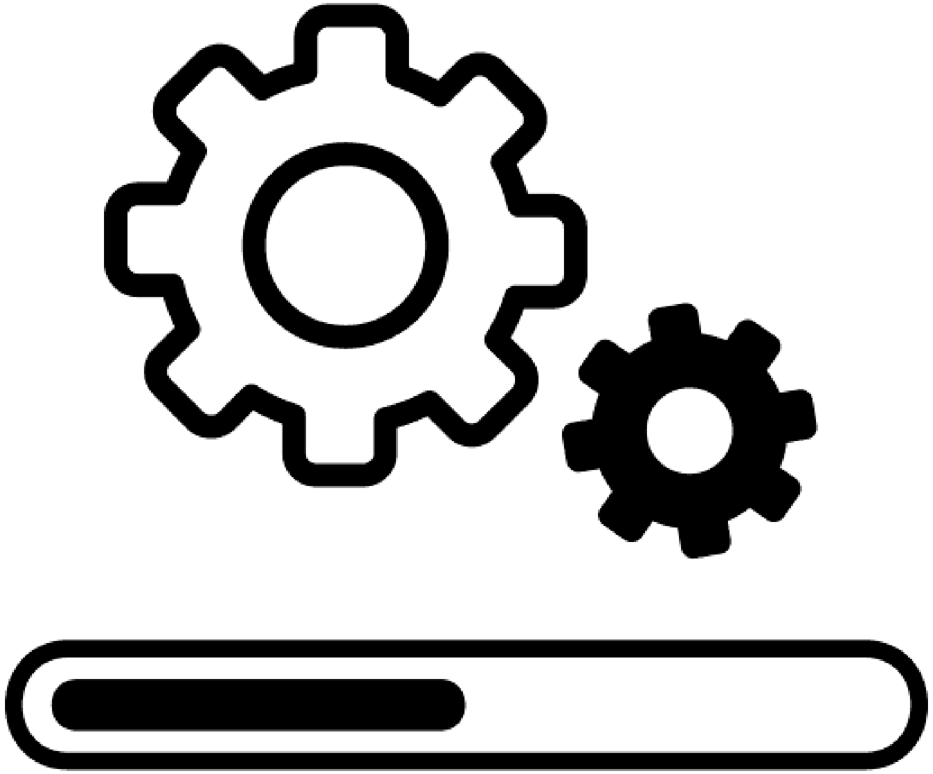
# Model Deployment in MLOps

- **Deployment**: Making the model available to end-users or applications

- **Edge vs cloud**: Where will the model run?

- **Scalability**: Ensuring the model can handle the required load

- **Latency**: Minimizing response time for predictions



# CI/CD Pipelines in MLOps

- **Continuous Integration**: Integrating code changes frequently, with automated testing
- **Continuous Deployment**: Automatically deploying the model to production after passing tests

- **Benefits**: Faster iteration, reduced manual errors, consistent deployments

# Monitoring and Logging in MLOps

- **Performance Monitoring**: Tracking model accuracy and other metrics
- **Data Drift Detection**: Identifying changes in input data distribution
- **Logging**: Keeping records of predictions and system behavior

# Model Governance in MLOps

- **Version Control**: Tracking changes and versions of models and data
- **Lineage Tracking**: Understanding data and model's journey from creation to deployment
- **Compliance**: Ensuring adherence to regulations and standards

# Summary of MLOps Introduction

- Crucial for effective deployment and maintenance of ML models
- MLOps leads to **reliable, scalable and efficient** ML workflows

- MLOps encompasses:
  - data engineering
  - model development
  - CI/CD pipelines
  - model deployment
  - monitoring
  - and governance

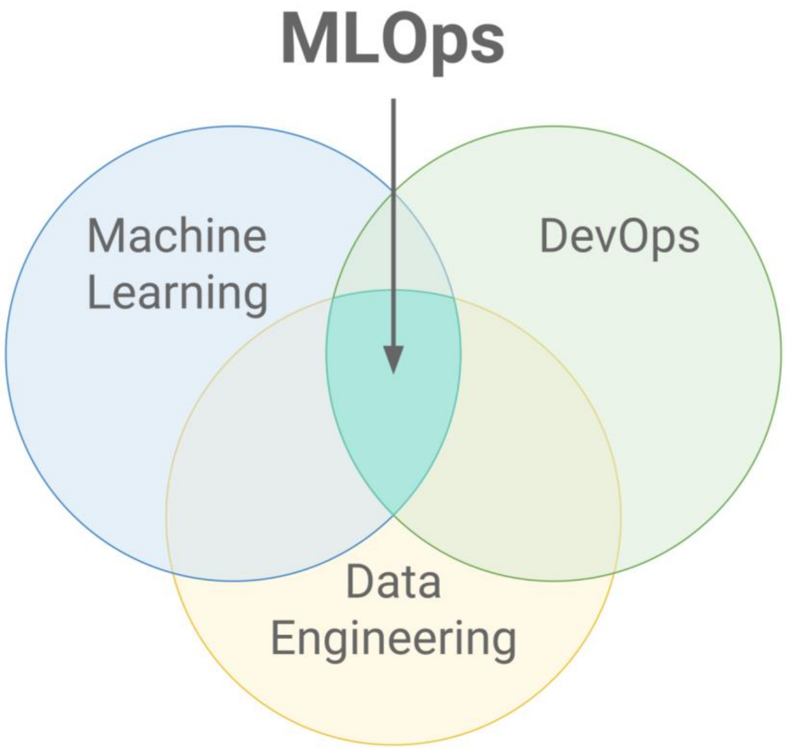
# Comparison to DevOps

# MLOps and DevOps Relationship

- **MLOps and DevOps**: Understanding the connection and differences

- Both practices aim to improve **collaboration, automation and efficiency** in the software lifecycle

- MLOps is basically **DevOps applied to Machine Learning**



# What is DevOps?

- **DevOps**: A set of practices that combines
  - software **development** (Dev)
  - and IT **operations** (Ops)

- **Goals**: Shorten the development lifecycle, deliver high-quality software continuously

# Key Components of DevOps

- **Continuous Integration (CI)**: Integrating code changes frequently
- **Continuous Delivery (CD)**: Automating the delivery of applications to production
- **Infrastructure as Code (IaC)**: Managing infrastructure through code

# MLOps vs DevOps: Similarities

- **Automation**: Both emphasize automation of workflows
- **Collaboration**: Foster collaboration between different teams
- **Continuous Improvement**: Focus on iterative improvements

# MLOps vs DevOps: Differences

- **Complexity**: MLOps deals with data, models, and code, making it more complex
- **Lifecycle**: MLOps includes unique steps like data preprocessing, model training, and model monitoring

# How MLOps Builds on DevOps

- Building on DevOps Practices

- **Extending CI/CD**: Incorporating model training and deployment into CI/CD pipelines
- **Infrastructure**: Applying IaC to manage data processing and model serving
- **Monitoring**: Adding model-specific monitoring (data drift, model accuracy)

# CI/CD in DevOps

- **Code Integration**: Frequent merging of code changes
- **Automated Testing**: Running tests automatically to ensure code quality
- **Deployment**: Automating the release process

# CI/CD in MLOps

- **Data Integration**: Regularly integrating new data
- **Model Testing**: Automating model validation and testing
- **Model Deployment**: Seamless deployment of new model versions

# DevOps Monitoring

- **Application Performance Monitoring (APM)**: Tracking application performance and uptime
- **Log Management**: Collecting and analyzing logs for troubleshooting

# MLOps Monitoring

- **Model Performance**: Tracking metrics like accuracy, precision and recall
- **Data Drift Detection**: Identifying shifts in input data over time
- **Operational Metrics**: Monitoring resource usage and latency

# DevOps Culture

- **Collaboration**: Bridging the gap between development and operations teams
- **Continuous Feedback**: Using feedback loops to improve processes

# MLOps Culture

- **Cross-functional Teams**: Bringing together data scientists, ML engineers, and operations
- **Shared Responsibility**: Ensuring everyone is accountable for the ML system’s performance

# Benefits of Integrating MLOps and DevOps

- Enhanced Efficiency
  - **Streamlined Processes**: Unified workflows reduce redundancy and inefficiencies
  - **Faster Time to Market**: Quicker deployment of ML models

- Improved Reliability
  - **Consistent Practices**: Standardized methods improve reliability and reproducibility
  - **Proactive Monitoring**: Early detection of issues in both code and models

# Summary of DevOps Comparison

- **Integration of Practices**: MLOps builds on DevOps principles, adding complexity to handle data and models
- **Holistic Approach**: Combining MLOps and DevOps leads to more robust and scalable solutions
- **Future Trends**: Increasing convergence of practices for better machine learning operations

# Experiment Tracking

# What is Experiment Tracking?

Experiment tracking involves **recording and managing information** related to machine learning experiments:

- hyper parameters,
- training metrics,
- artifacts (data, code, models)
- …

to facilitate **reproducibility and collaboration**.

# Why use Experiment Tracking?

- **Reproducibility**: Ensures that experiments can be recreated with the same inputs and conditions
- **Collaboration**: Facilitates sharing of results and insights among team members
- **Performance Monitoring**: Allows tracking of model performance over time

# Existing Tools: MLFlow

- Developed by Databricks, MLFlow provides tracking for experiments, packaging code, and managing models
- Offers a unified platform for managing the end-to-end machine learning lifecycle
- Features an intuitive UI for visualizing and comparing experiment results

# Existing Tools: Weights & Biases

- Weights & Biases offers experiment tracking, visualization, and collaboration tools for machine learning projects
- Allows tracking hyperparameters, metrics, and artifacts in a centralized dashboard
- Supports integration with popular machine learning frameworks and platforms

# Existing Tools: ClearML

- ClearML provides experiment management and automation for machine learning workflows

- Offers features like distributed training, hyperparameter optimization, and automatic experiment tracking

- Integrates with various ML frameworks and cloud providers for seamless deployment

# Model Registry

# What is a Model Registry?

A model registry is a **centralized repository**

for storing, versioning, and managing **machine learning models**

throughout their lifecycle.

# Why use a Model Registry?

- **Model Versioning**: Tracks different versions of models for reproducibility and comparison

- **Deployment Management**: Facilitates the deployment of trained models into production environments

- **Model Governance**: Ensures compliance and accountability in model usage and updates

# Existing Tools: MLFlow (again)

- In addition to experiment tracking, MLFlow provides model registry functionality for managing model versions and deployments
- Allows registering and serving models in a production environment with ease
- Supports model versioning and comparison for tracking model evolution

# Existing Tools: Weights & Biases (again)

- Weights & Biases includes a model registry feature for storing and versioning trained models

- Offers model deployment capabilities for deploying models to various deployment targets

- Provides visibility into model performance and usage metrics post-deployment

# Existing Tools: ClearML (again)

- ClearML extends its capabilities beyond experiment tracking to include model registry functionality
- Allows registering, versioning, and deploying models seamlessly within the ClearML platform
- Provides integration with popular ML frameworks and cloud platforms for streamlined model deployment

# Lab 1: Experiment Tracking

# Lab 1: Experiment Tracking

- We will learn to use **ClearML**

- Download exercises from Github
- Exercises will be **added 1 by 1**

https://github.com/AlexanderDhoore/240603-mlops-workshop