

# MLOps Zoomcamp Week 1

---

The course begins with a recap about Machine Learning as well as basics about MLOps maturity model.

Different levels of model maturity:

## 1. No Automation

- All code is present in a jupyter notebook written by data scientists working alone. Data scientists work in isolation, and the process is prone to disaster when transitioning from experimentation to production. The notebook is handed over to engineer for implementation. There is no automation, pipelining or experiment tracking involved.

## 2. DevOps, no MLOps

- This introduces some automation, with experienced developers assisting data scientists. Releases are automated. Model can be deployed similar to deploying a web service. Practices like automated releases, unit tests, and integration tests are used but they are not ML-specific. Operational metrics such as how many requests per second come handy in assessing a network situation. There is no experiment tracking and reproducibility of the models. This level is suitable for proof of concept (POC) projects going to production.

## 3. Automated Training

- There are automated machine learning training pipeline where jupyter notebooks are not needed to be loaded and executed to get the models. Model registry helps in experiment tracking (particularly the parameters) and knowing the models in production. The training is completely automated. Deployment is not necessarily automated but is low friction. Data scientists work closely with engineers at this stage.

## 4. Automated Deployment

- The stage suggests easy deployment of trained models and also involves A/B testing to compare different versions of a model. The models are available on an ML platform, and deployment can be part of the training pipeline.

## 5. Full MLOps automated operations

- The whole system is fully automated and easily monitored.

The importance of choosing the right level is based on the number of models in production and the use cases. It's suggested to invest in infrastructure for machine learning processes when there are multiple proven use cases.

Level	Description	Highlights	Technology
0	No MLOps	<ul style="list-style-type: none"><li>• Difficult to manage full machine learning model lifecycle</li><li>• The teams are disparate and releases are painful</li><li>• Most systems exist as "black boxes," little feedback during/post deployment</li></ul>	<ul style="list-style-type: none"><li>• Manual builds and deployments</li><li>• Manual testing of model and application</li><li>• No centralized tracking of model performance</li><li>• Training of model is manual</li></ul>
1	DevOps but no MLOps	<ul style="list-style-type: none"><li>• Releases are less painful than No MLOps, but rely on Data Team for every new model</li><li>• Still limited feedback on how well a model performs in production</li><li>• Difficult to trace/reproduce results</li></ul>	<ul style="list-style-type: none"><li>• Automated builds</li><li>• Automated tests for application code</li></ul>
2	Automated Training	<ul style="list-style-type: none"><li>• Training environment is fully managed and traceable</li><li>• Easy to reproduce model</li><li>• Releases are manual, but low friction</li></ul>	<ul style="list-style-type: none"><li>• Automated model training</li><li>• Centralized tracking of model training performance</li><li>• Model management</li></ul>
3	Automated Model Deployment	<ul style="list-style-type: none"><li>• Releases are low friction and automatic</li><li>• Full traceability from deployment back to original data</li><li>• Entire environment managed: train &gt; test &gt; production</li></ul>	<ul style="list-style-type: none"><li>• Integrated A/B testing of model performance for deployment</li><li>• Automated tests for all code</li><li>• Centralized tracking of model training performance</li></ul>
4	Full MLOps Automated Operations	<ul style="list-style-type: none"><li>• Full system automated and easily monitored</li><li>• Production systems are providing information on how to improve and, in some cases, automatically improve with new models</li><li>• Approaching a zero-downtime system</li></ul>	<ul style="list-style-type: none"><li>• Automated model training and testing</li><li>• Verbose, centralized metrics from deployed model</li></ul>