AI Sweden - Deep Dive Sessions for Data Scientists

Fredrik Strålberg, Cloud Solution Architect AI
2022-05-25
Handling the complexity of model deployment challenges
Turn captions on

Go to your meeting controls and select More options ...
More options button >
Turn on live captions.
Agenda

Part 1 - Most common challenges
What is the most common challenges after a model is trained and ready for deployment.

Part 2 - How to handle the challenges
Key considerations during model deployment and before releasing model for production.

Part 3 – Tooling and Processes to Overcome Challenges
First steps to some resources available in Azure and importance of MLOps.
Fredrik Strålberg

Cloud Solution Architect within AI - Microsoft

Data Scientist – Microsoft 5½ years
Delivering end-to-end Data Science and MLOps solution for customer across different industries worldwide. Focus on computer vision, forecasting and machine learning solutions.

Master in Mathematical Statistic
Umeå university. Industrial engineering and management, computer science.

Personal Note
High-level model deployment challenges
Model Deployment Challenges – Company

**Business Challenges**
We do not have the people or processes in place to deploy and start consuming predictions from models develop.

**Technical Challenges**
We do not have right tools or infrastructure to deploy a model, or we see MLOps as a one-time task and not a continuous lifecycle.
Model Deployment Challenges – Company

**Technical Challenges**
We do not have right tools or infrastructure to deploy a model, or we see MLOps as a one-time task and not a continuous lifecycle.

**Business Challenges**
We do not have the people or processes in place to deploy and start consuming predictions from models develop.
Model Deployment Challenges – Roles

Data scientist often wear many hats and works together with many different roles at companies. It often occurs grey areas of responsibilities between different roles.

**Data Scientist**
- Data wrangling
- Develop model
- Discover business value
- Solving/stating problem statement

**AI/Ops Engineers**
- Model operationalization
- MLOps
- Monitoring

**AI Architect**
- Architecting solution
- Infrastructure

©Microsoft Corporation
Azure
Most Common Challenges
Most Common Challenges

- **Difference in training and production data**
  The training domain differs from the production domain.

- **No monitoring of model in production**
  Metrics, KPIs and telemetry data from model in production.

- **No feedback to algorithm**
  Capturing and labelling data for modelling retraining.

- **Model performance is not sufficient**
  Model deployed do not achieve required business outcomes.

- **Model drift**
  Detecting that the underlying dataset is changing in production and alarming for re-training.

- **Lacking Supporting infrastructure**
  We do not have the architecture, tools or process to deploy models.
Difference in training and production data

- Different data distributions between training and production
- Data collection – different lightning conditions, environment, weather, or seasons not considered
- Devices – different sensors and cameras being tested/evaluated during model development.

Side note:
- Data collection – Are there any defects in production? And what production site should be used as test site?
Model Drift

• No clear approach/strategy of how to capture model drift defined

• Sensors being replaced - upstream process changes

• Broken sensors - data quality issues

• Natural drift - mean temperature suddenly changing

• Covariate shift – sudden change in relationship between features
Monitoring Model Service in Production

- Model interpretability not defined or used correctly
- No monitoring (streamed or stored data) in place
- Metrics, alerts and events are not clearly defined and linked to predictions
- Lacking quality of the service
- Unclear responsibility for who should implement monitoring features
Supporting infrastructure

- MLOps not used
- Supporting processes not implemented or considered in solution
- Latency expectations of services
- Full end-to-end flow/architecture not clearly defined
- Cost management not considered
- Model approach selection not fitted to purpose
What kind of model deployment challenges are you/your organization currently facing?
How to Handle the Challenges
How to Handle the Challenges

Recommendations from a business and technical aspect

Avoiding Domain Shift
• Best practices from the field

Supported Processes to Consider
• Key processes identified

Infrastructure and Tooling
• Going from a local DS team to an enterprise DS team
Recommendations

**Business Aspect**
Think about the solutions as a full product lifecycle and how this will look in an actual production environment before starting the development.

**Technical Aspect**
Do not deprioritize features for supported processes that might need to be implemented, be transparent of time consuming and costly workloads and take time to define MLOps strategy.
Avoiding Domain Shift

• Clearly define annotation process and label definition before start. Document!

• Identify and involve actual production SME and get them involved from the beginning.

• True production data available upfront

• Mechanism and features implemented (or planned early) for capturing changes in production

• Decide and prioritize how you will monitor your model in production from the start (data drift)
Supported Processes to Consider

- **Release process** – KPI release or manual approval.
- **Environment management** – condition between dev to test and test to prod.
- **Rollback mechanism** – how do we rollback if everything goes wrong.
- **Audit process** – traceability end-to-end from data to deployed model.
- **Quality control** – human in the loop (HITL) implementation with traffic light release to production. Human handling low confidence predictions feeding those back to algorithm (active learning).
- **Security** – GDPR, zero trust and ethical AI.
Infrastructure and Tooling – From local DS Team to Enterprise DS Team

• Driving the conversations around defining the CI/CD process and strategy on your organization
  CI Process – code version and quality control gates
  CD Process – deployment mechanism is used
  Monitoring Process – reporting to understand model status

• Scalable infrastructure – traceability, expected throughput, on-prem or cloud
Breakout Session

How did you find a way forward connected to the challenges?

What is your organization current CI/CD process and/or what is the pain points?
Tooling and Processes to Overcome Challenges
Tooling and Processes to Overcome Challenges

Asset management and orchestration services  MLOps
Asset management and orchestration services to assist in the lifecycle of model training and deployment workflows.
Azure Machine Learning

Operationalise and manage models with ease

Bring models to life quickly
- Build and deploy models in minutes
- Iterate quickly on serverless infrastructure
- Easily change environments

Proactively manage model performance
- Identify and promote your best models
- Capture model telemetry
- Retrain models with APIs

Deploy models closer to your data
- Deploy models anywhere
- Scale out to containers
- Infuse intelligence into the IoT edge
Azure Machine Learning

Components

Azure Storage Account
Azure Container Registry
Azure Key Vault
Azure Application Insights

Dependencies

Linked services
Managed resources

Compute instances
Compute clusters

Assets

Environments
Experiments
Pipelines
Datasets
Models
Endpoints
What is DevOps?

“DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users.”

*Donovan Brown*
DevOps Practices

- Version Control
- Infrastructure as Code
- Continuous Integration
- Microservices
- Continuous Delivery
- Monitoring and logging
MLOps Practices

The ability to continuously integrate, automatically test, build and deploy Machine Learning artifacts such as Data & Training pipelines and models.
**CI/CD Pipelines**

**Continuous integration (CI)**
Automate the build and testing of code every time a code changes requested.

Drives the ongoing merging and testing of code, which leads to finding defects early.

**Continuous delivery (CD)**
Build, test, configure and deploy from a dev to a production environment.

Ensures that code and infrastructure are always in a repeatable and production-deployable state.
Continuous Integration in MLOps

Training Pipeline

Tests → Launch training → Evaluate models → Register model → Publish artifact

Experiment → Packaged model

Azure DevOps
Continuous Deployment in MLOps

CD Pipeline

Deploy model in DEV → Tests → ... → Manual approval → Deploy model in PROD

Webservice (development environment)

Webservice (production environment)
MLOps v2
Breakout Session

Open discussion
Thank you.
Invent with purpose.