Lecture 2 - MLOps

Advanced Machine Learning

Miroslav Čepek, Zdeněk Buk, Rodrigo Da Silva Alves, Vojtěch Rybář, Petr Šimánek

FIT CTU

2. 3. 2023

Why repeatable models

- Training process review.
- Models need to be investigated when failed.
- Recreate the model.
- Model selection and parameter tuning.
- (Automated) refresh of the model on top of new data.

Reproducibility of ML Model

- The problem with ML project is that the result depends on too many and too diverse factors, including:
  - Starting Dataset (and selection of training data)
  - Data preprocessing
  - Model type, architecture, optimisation procedure and hyper-parameters
  - Random seed
  - ...
- In order to get the some model twice, you need to get all above right.

Reproducibility Challenges (1)

Phase	Collecting Data
Challenges	Generation of the training data can't be reproduced (e.g due to constant database changes or data loading is random)
Ensure Reproducibility	1. Always backup your data. 2. Saving a snapshot of the data set (e.g. on the cloud storage). 3. Data sources should be designed with timestamps so that a view of the data at any point can be retrieved. 4. Data versioning.

Taken from ML Ops Principles from ml-ops.org

Reproducibility Challenges (2)

Phase	Feature Engineering
Challenges	Scenarios: 1. Statistic based feature engineering Missing values imputation, scaling in normalisations. Categorical $\rightarrow$ Numerical and vice versa conversions. 2. Non-deterministic feature extraction methods.
Ensure Reproducibility	1. Feature generation code should be taken under version control. 2. Require reproducibility of the previous step "Collecting Data"

Taken from ML Ops Principles from ml-ops.org

Reproducibility Challenges (3)

Phase	Model Training / Model Build
Challenges	Non-determinism
Ensure Reproducibility	1. Ensure the order of features is always the same. 2. Document and automate feature transformation, such as normalization. 3. Document and automate hyperparameter selection. 4. For ensemble learning: document and automate the combination of ML models.

Taken from ML Ops Principles from ml-ops.org

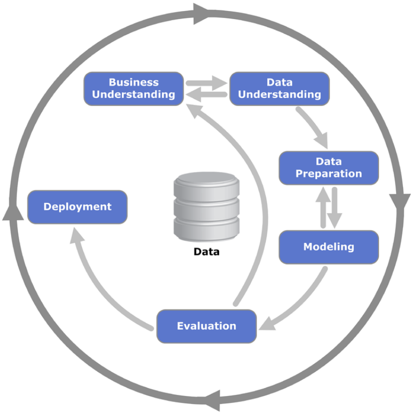
Reproducibility Challenges (4)

Phase	Model Deployment
Challenges	1. Training the ML model has been performed with a software version that is different to the production environment. 2. The input data, which is required by the ML model is missing in the production environment.
Ensure Reproducibility	1. Software versions and dependencies should match the production environment. 2. Use a container (Docker) and document its specification, such as image version. 3. Ideally, the same programming language is used for training and deployment.

Taken from ML Ops Principles from ml-ops.org

CRISP-DM

- Attempt to define standardised steps of machine learning process.
  1. Business Understanding
  2. Data Understanding
  3. Data Preparation
  4. Modeling
  5. Evaluation
  6. Deployment
- Set of actions and checkboxes to be marked in each stage before you move further (or back).
- Some commercial tools supports the flow.



Wikipedia Image
CRISP-DM The New Blueprint for Data Mining

Machine Learning Operations

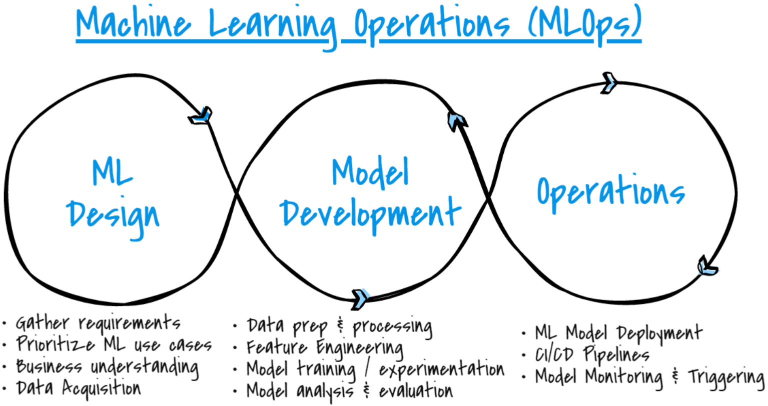
- Evolution of the reproducible ML idea.
- Envelop the whole process into series of steps and support it with software tools.

Machine Learning Operations

- MLOps is a paradigm - including best practices - as well as a development culture when it comes to the end-to-end conceptualisation, implementation, monitoring, deployment, and scalability of ML products.
- It is a SW engineering practice that leverages three contributing disciplines: ML, software engineering and data engineering.
- MLOps aims to facilitate the creation of machine learning products by leveraging these principles: CI/CD automation, workflow orchestration, reproducibility; versioning of data, model, and code; collaboration; continuous ML training and evaluation; ML metadata tracking and logging; continuous monitoring; and feedback loops.

Machine Learning Operations (MLOps): Overview, Definition, and Architecture

Machine Learning Operations



Pratik Sherma - 10 Best MLOps Tools in 2022.

MLOps vs DevOps

- DevOps – describes processes and interactions between developers and operations people. Ways to hand over code, automate tests, validations and deployment into production. Monitoring service availability and quality.
- MLOps – describes processes between ML/DS/AI practitioners and rest of the organisation. Improve record keeping, simplify creation, deployment and monitoring of the machine learning models.
- In both cases it focuses on automation and reproducibility.

MLOps Areas

- Experiment Tracking
- Model Registry
- Data Versioning
- Feature Store
- ML Model Deployment and Monitoring
- Pipeline (Project) Management

Experiment Tracking

- is a process and tool(s) to save all important experiment related information. So you can later return, review and compare different experiments.
- Values to store includes for example:
  - Code of the experiment,
  - System (environment) configuration, ie. libraries and their versions,
  - Versions of the data, training/evaluation split,
  - Hype-parameters
  - Evaluation metrics
  - Performance visualisations (confusion matrix, ROC curve, learning curves, example predictions, etc.)

Inspired by Neptune AI Blogpost

Model Registry

- A place to store all the trained models (their binary serialisation).
- Models are typically identified by ID.
- MR works closely with Experiment Tracking.
- Records model life-cycle stage (experimental, staging/testing, in production, retired).

Feature Store/Data Versioning

- Feature store is a data management system that manages and serves features to machine learning models.
- Provide single and unified way to calculate individual features.
- Goto place for training and deplyment data.
- Part/Layer above data lakes/databases.
- Data versioning – different snapshots of the data (ie different times).

Feature Store 101

ML Model Deployment and Monitoring

- System/Process to put ML model into production (make it available to customers).
- Some MLOps systems (ie MLFlow) contains a deployment service for this purpose.
- ML Model Deployment Strategies

- Monitoring is needed due to data drift (natural changes in world as captured in the data).
- Automatically evaluate performance metrics and alerting.

MLFlow

- https://www.mlflow.org
- Experiment Tracking — is an API and UI for logging parameters, code versions, metrics, and artifacts when running your machine learning code and for later visualizing the results.
- Projects – to unify structure of different experiments into single shape (described by YAML file), which can be understood and executed by MLFlow.
- MLflow Models offer a convention for packaging machine learning models in multiple flavours (TF, Torch, GLUON, SK-L, ...), and a variety of tools to help you deploy them.
- MLflow Registry offers a centralized model store to manage the full lifecycle of an MLflow Model. It provides model lineage, model versioning, stage transitions, and annotations.

MLFlow

```
mlflow run https://github.com/mlflow/mlflow-example.git -v 0651d1c962aa35e4dd02608c51a7b0efc2412407 -b local -P
```

› Description Edit

⌄ Parameters (2)

Name	Value
alpha	0.4
l1_ratio	0.1

⌄ Metrics (3)

Name	Value
mae	0.617
r2	0.192
rmse	0.791

› Tags

⌄ Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt

Full Path:mlflow-artifacts:/0/4041f98a9b15463a8a4...

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

Source Code
Cloud FIT MLFLow

Weights and Biases

- Cloud based service available at `wandb.ai`.
- Track experiments, results (artefacts), automates selected tasks
  - Hyperparameter tuning
  - Reporting
  - Alerting
- Can be deployed locally.
- Free tier for personal use (with much more generous free offer for university students).

- Notebook with Example Experiment
- Publicly accessible WandB Tracking UI

Neptune.ai

- Cloud based service available at `http://neptune.ai`.
- Experiment tracker and model registry.
- Allows you to record code, metrics and media (images, videos) and artefacts (serialised models).
- Free tier for personal use (with much more generous free offer for university students).

- Notebook with Example Experiment
- Publicly Accessible Neptune Tracking UI

ML Orchestrator - Metaflow

- Allows you to build a repeatable pipeline.
- Pipeline is a series of steps, that can be automatically executed by scheduler in serial/parallel manner (and can be distributed in cloud/k8s environment)

Feature Stores / Data Versioning

- Pachyderm – pachyderm.com
  - *Pachyderm is cost-effective at scale, enabling data engineering teams to automate complex pipelines with sophisticated data transformations.*
- Data Version Control – dvc.org
  - *DVC is a tool for data science that takes advantage of existing software engineering toolset. It helps machine learning teams manage large datasets, make projects reproducible, and collaborate better.*
- Feast – feast.dev
  - *Feast is a standalone, open-source feature store that organisations use to store and serve features consistently for offline training and online inference.*