MLOps: Managing Data and Workflows for Efficient Model Development and Deployment

Konstantinos Balafas – Head of AI Data
Carlo Dal Mutto – Director of Engineering
Acubed by Airbus
ML Ops and Data Ops
System Productization (MLOPS)

MLOPS

Discipline that comprises a set of tools and principles to support progress through the lifecycle of a ML project
ML Project Lifecycle

Scoping
- Define project
- Define data and establish baseline

Data
- Label and organize data
- Select and train model

Modeling
- Perform data analysis

Deployment
- Deploy in production
- Monitor and maintain system
Project Scoping

Understand if there is **business value**

- Fundamental questions:
  - Is the project providing any business value?
  - What system performances would make the business viable?
    - System requirements

Understand if the problem is **technically achievable**

- Establish a baseline and confirm that system requirements can be achieved
  - **Human level performance (HLP)**
  - Literature search
  - Quick and dirty implementation

If HLP is not satisfactory, improve the considered sensors or the instruction to the humans
Data are first-class citizens in data-centric AI

New trend: big data → good data

Which data

- Raw vs processed data
- Metadata
  - Provenance: where the data came from
  - Lineage: which are the processing steps applied to the data

Data pipeline

- Raw data → Cleaning Engineering → Labelling → Partitioning → Dev. dataset → Test dataset
Diversity is more important than quantity

Need to be deliberate about new data

Leverage already collected data in an iterative fashion for data driven decisions

Core ETL Tools and Processes

Identify data that cannot or should not be used downstream

Validate quality of Data Acquisition (time syncing, outliers, etc.)

Potentially engineer ML features

In-house manual labeling typically yields highest quality but is not scalable

Algorithmic solutions exist

ML models

Data labeling vendors

QA is paramount

Partitioning strategy should be requirements-driven

Prevent target leakage

Ensure that models are evaluated on a representative dataset

Dataset versioning is critical

Several code version control-inspired solutions

Raw dataset or metadata-driven versioning

Many high-quality vendors exist
Data pipeline - Key challenges

- **Raw data**
  - Costly

- **Cleaning Engineering**
  - Significant upfront investment and ongoing maintenance

- **Labelling**
  - Difficult balance between scalability and quality

- **Partitioning**
  - Dataset lineage becomes very complex

- **Dev. dataset**
  - Test dataset

While automation steps are possible, it is very difficult to remove manual steps entirely from this loop.
Model-centric AI development → data-centric AI development

Model development preliminary steps

1. Define development set, then split it into training and validation set
2. Define business-centric and model-centric metrics
   ○ Account for data distribution artifacts (skewness, unbalancing, fairness...)

Model development main steps

- Get an initial dataset
- Feature engineering (normalization, dimensionality reduction, etc.)
- Data augmentation
- Architecture exploration
- Finetune model

Start simple! Efficient train + deploy! Squeeze more out your data! Be creative! Back to business!

© 2022 Airbus
Types of deployment

- New product/problem capability
- Automate/assist human
- Replace a previous model/pipeline

Gradual deployment → progressively increase amount of automation

Monitoring

- Define which metrics should be tracked during deployment
- Implement monitoring dashboards (manage under- vs over-monitoring)
- Pipeline monitoring. For each component monitor: input metrics, output metrics, software metrics
Post-Deployment

Model deployment → Model monitoring → Decision point → Model re-training → Model evaluation

- Metrics on input data (concept shift)
- Metrics on model performance (require ground truth)
- Decision point implies continuous monitoring and event-triggered outer loop
- Avoid catastrophic forgetting
- Training data should represent operational conditions
- Dataset versioning is important to ensure reliable model evaluation and avoid target leakage

New data collection → Data integration

- Necessary in the absence of ground truth
- Diversity is key

New data need to "look and feel" like old data
Wayfinder
Problem Statement

• Develop a vision-based perception system for autonomous flight

• Development streams
  • Sensing development (cameras, radars, INSs)
  • SW development
  • Perception development (including ML)
Autonomous Landing

- Goal: vision-based equivalent of the Instrument Landing System (ILS)
- INPUT: images, database of airport locations
- OUTPUT: position of the aircraft wrt the runway
ML Ops and Data Ops at Wayfinder
ML Ops and Data Ops

Data Ops
- Image quality analysis
- Image annotation
- Dataset/label versioning
- Data consumption and “insights”

ML Ops
- Train/validate/test split
- Model versioning
  - efficient experiment tracking
  - Verification & Validation
- Model monitoring
Image Quality Analysis

- Need to determine whether an image is suitable for training (or testing!) before an image is included in an ML-bound dataset

- Leveraging "classical" CV techniques (e.g. brightness, contrast)

- Exploit model predictions/confidence as an additional dimension

© 2022 Airbus
Image Annotation

- **Requirements**
  - Need to calculate ground truth for model training
  - Very high accuracy requirements
  - Need to scale to millions of images

- **Solution**
  - Leverage an ensemble of:
    - Classical CV techniques
    - Machine learning models
    - Human annotation
  - Establish both a per-image and per-algorithm “quality” metric
  - Develop a dense 3D database of airport features
Image Annotation Pipeline

- Image collection
- Selection of images to manually label
- Outsource manual labeling
- Label audit
- Labeling using CV techniques
- Label fusion
- Labeling using ML techniques
- ML training
- Label quality metrics
Dataset/Label Versioning

- Data versioning is important for Verification and Validation of our autonomous flight system
  - Images used for training different ML versions and modules
  - Labels denoting ground truth
- Leveraging logging of runs of the aforementioned pipeline to ensure traceability of our labels
- Development of a similar “delivery” pipeline for versioning datasets
  - Consider datasets “immutable”
Data Consumption and Insights

- Collected data has value beyond ML

- Enable data-driven decisions across Wayfinder
  - Data labeling
  - Future data collection
  - Systems-level requirements

- Developing purpose-built dashboards
ML Ops: Pre-Deployment

- Train/validate/test split
  - Account for sequential nature of data
  - Ensure similar distributions of key features across sets
  - Ensure sufficient coverage across operational envelope
- Model versioning
  - Efficient experiment tracking
  - V&V efforts
ML Ops: Post-Deployment

- **Current state**
  - Model is evaluated on a per-project pre-defined test set that has been curated to meet system requirements

- **Monitoring Constraints**
  - Ground truth is unavailable “online”
  - Expected large volumes of diverse data arriving in multiple batches

- **Moving forward**
  - Establish an annotation and evaluation strategy
  - Objective is to maximize diversity of a continuously growing evaluation set based on
    - Image content
    - Key metadata “dimensions” (e.g. airports, weather conditions, aircraft position, etc.)
    - Current model performance (and/or model confidence on different predictions)
There are several operational and conceptual challenges associated with ML and Data Ops

- Labeling
- Data curation
- Monitoring

We try to address them by:

- Breaking down the data life-cycle in semantic stages
- Organizing our code in modular pipelines
- Leveraging both classical and ML-based Computer Vision algorithms
- Automating the different pipeline steps to the extent possible
Thank you!
We are hiring!
acubed.airbus.com/careers