Reaktor

MLOps training case assignments

2023 Reaktor / Saku Suuriniemi

V 1.1, Licence CC BY 4.0

IML4E

Intro

These exercises emphasize the MLOps adoption phase. You can assume the organization in question usually has fairly low MLOps maturity.

- In particular, no cases challenge with constraints from already adopted MLOps tools, so you are free to ideate and suggest additions and improvements.

To cover a range of themes, it makes sense to do at least 2-3 cases and regulate the amount of work with solution detail level and (in/ex)lusion of a practical implementation.

Intro

The cases call for analysis and aspects of more than one of the following

- People (communication, roles, motivations, stakeholder relations)
- Culture (teams' operating practices and principles)
- Technical solutions (automation, monitoring, artifact serving…)
- Ethics
- Internal performance objectives
- External constraints (certifications etc.)
- Resource /economical constraints

Intro

All solutions must contain a plan of how to address the problem.

Some cases specifically ask for an technical implementation.

- Partial implementations help gain skills, too.

There is a label such as (C2) to indicate the anticipated properties of each case:

- The letters indicate predominant theme: Technical, Conceptual, People
- The number 1-3 indicates challenge

Case 1: A disaster in a small company (TC1)

A four-person startup develops a machine vision quality assurance system for industry. One technical person works on the model and another one with the infra. The model maker gets into a serious accident. A computer vision expert - but from the media field - takes over. What arrangements would make

- the new modeller to know the objectives,
- the relevant data discoverable and understandable?
- How to get familiar with the development process of the model?
- (Bonus, TC2 with this: How to keep the production model alive and make decisions on update and developments needs?)

Case 2: The newcomer (PT2)

A young consultant - fresh from the university - joins a big ML team. The new person refuses to use anything else than (Jupyter) notebooks and keeps writing 1000-line-long “functions”. How to make this person a net-positive member to a team that’s otherwise using good SW development practices?

- Design a suitable notebook-to-Python workflow
- How to
  a. make the code quality tangible and
  b. motivate the newcomer to improve it?
- How to facilitate derivative work?
- Improve repeatability in the face of code dependency updates.

Case 3: The world changes (T2)

The customer maintains a web info board (Think of Gapminder) that has many many data sources that get updated with varying time intervals.

- How to detect that new data is available?
- How to update the data automatically?
- Preparation for things breaking up, e.g. outages, API changes?
- Indicators for update need of the analysis texts and predictive models?
- How to automate the updates?

Case 4: Let's always synthesize! (T1)

An essentially linear machine learning model didn't work well with the input features X,Y,Z, but the nonlinear functions f(X,Y) and g(Y,Z), h(Z) of the features give stellar results. Afterwards, there is a hunch why, but no rigorous argument.

- What tools and methods could be used to document and communicate the discovery process of the features to relevant parties?
- How to avoid the training-serving skew (i.e. make sure the transformation is always reliably done for both training data and production data for predictions)?

Case 5: Learning online (C2)

Recommender system for a starting subscription-based audiobook company with a rapidly growing catalogue recommends content based on likes and choices of customers plus genre labels. It is learning online (the nightmare scenario). However, the ground truth is quickly available, i.e. if the customer chooses to listen the recommended audiobook or not.

- Analyse the risks and challenges.
- Plan technical and procedural solutions to address them.

Case 6: Can’t know for sure if it’s still working (C2)

A new hobby recommender gets other hobbies, age, place of residence, interests etc. of the user as the input. It was trained once with a hefty sample of data gathered in a one-off campaign. The ground truth depends on the hobby updates of the users’ profiles (i.e. is highly uncertain).

- How to monitor if it still likely predicts OK?
- What signs to follow to decide on re-training?
- Suggest ways to improve the ground truth acquisition.

Reaktor

IML4E

Case 7: Block by block (TC3)

The system consists of several ML models that produce results to be consumed by the others.

- Intermediate results should be traceable
- How to enable re-training and deployment of new blocks?
- How to specify prior block performance requirements to keep or improve the composite performance?
- How to measure the quality of the composite results?
- How to leverage the analysis above in the training and CI/CD pipelines?

Case 8: Can you show it? (T1)

You have a data source that you query regularly. Devise a small monitoring dashboard to show if the source still works as intended. Use GET-queries to

https://catfact.ninja/fact

to have responses to work with. The JSON responses offer you a “fact” and the “length” of the fact.

Of interest would be the latency, failed requests, whether the length is correct, and whether the facts come properly randomized.

Case 9: Are we still in business?

A team of three data scientists develops different models to predict the supply of materials from a handful of unstable sources that exhibit sudden wide transitions between more stable states. How to keep track of

- what's been tried,
- on which data
- how did it work then, and
- how would it work now?

Case 10: Freeze! (TC2)

A medical application for clinical use depends on a certified model. Consequently, the certified input-output relation of the model must not change after certification. Robustness must be, however, improved on corner cases. Development of the code outside the model, and e.g. implementation of new APIs is allowed, as long as one can show that certified values are still provided.

You want to make at least internal releases often. However, each release requires you to provide some documents e.g. of the certified values still being provided. How to keep the release-costs low?

Case 11: Bad habits (PT2)

An experienced modeller produces Python code that you'd rather not work with. It's cluttered, same variables and functions get re-defined several times, comments defy specialist skills and naming conventions are unheard of. The modeller also has a habit of making big-bang single commits. It is thought that the behavior is not due to conscious resistance.

Then again, the modeller knows the statistics. The rest of the team needs to build reliable, working software that uses the modeller's magic to do the heavy lifting.

- Suggest strategies to improve the code quality.
- Think of both communication and tech/tool choices

Case 12: I’ve seen this before (T2)

A data ingestion pipeline has started producing duplicates. What’s worse, same duplicates appear in both training and test data. It is also unclear if the present production model was trained on compromised data or not.
The source of raw data is an incremental collection of parquet files on an object storage.

- Prioritize the actions in the situation.
- What do you need to investigate the present model?
  - Would you try to salvage it?
- If the duplicates were found out by chance, how would you make them automatically discoverable?
- Design necessary changes to preclude - to a fairly high certainty - that similar problem does not repeat.

Case 13: You just don’t understand (PC2)

An experienced Product Owner confronts a Data Scientist for failing to accomplish tasks in the backlog. The DS blames the data, ceases to make more “promises”, and attempts to make progress. The effort seems random and frivolous to the PO, and the reasons to not proceed sound obscure.

A test user group awaits. One member of the group has provided the training set when asked for “examples”. The machine learning component is supposed to help the main work tasks with suggestions.

- What misunderstandings may underlie the situation?
- Suggest ways to mitigate the problems.