Riitta Piela

2 February 2022

Quarterly Report Q1

Workstream 4

Organization

Workstream 4 started the work by forming an active working group consisting of five persons: Shirin and Abel from Canada, Claus from the UK, and Rok & Riitta from Statistics Finland. First, we discussed the necessity to refine the scope of the Workstream 4. But instead of formally defining the scope we decided to approach the scope the agile way; we prioritize the upcoming tasks and after solving the prioritized tasks, we look at the scope again and if necessary, the scope will be refined. In practice this means that we work in three-week periods (sprints) and hold a sprint review at the end of each sprint to which all members of the workstream are invited.

We decided to use Quire to organize the work of the workstream. Quire is a free (at the time of writing), cloud-based SaaS project management tool. Quire offers the possibility to organize tasks of the workstream and view them in three main mods in Task List view, Kanban view and Timeline.

Literature review

In the first sprint (ends at 29th of march) we prioritized identifying all types of drift (circumstances) that require some sort of mitigation measure to be undertaken to keep model performance acceptable. In addition to retraining, retuning of weights was identified as another possible mitigation measure. Weight retuning entails weighting the observations as they come, with a higher weight given to the more recent observations. If drift is detected, the weights may be “retuned” to return model performance to an acceptable level. Because model retraining is not the only mitigation measure, we are likely to focus on drift detection metrics (mitigation measures are out of scope).

First sprint. For each identified drift, we propose to identify the items to be monitored and metrics to be used. We have also started writing up a vocabulary, which will first only include types of drift, but will later be expanded to other concepts related to our domain.

Data in WS4

We have discussed the data and models to be used to test the proposed drift detection metrics. Because Statistics Finland cannot share its data with outside partners, we decided to use the ECOICOP-dataset and models Statistics Poland has published on its GitHub page during its involvement in UNECE Machine Learning Group 2019-2020. We may also include other open data if suitable datasets are found. A GitHub repository has been created on GitHub account of Statistics Finland to facilitate code sharing for this activity. https://github.com/StatisticsFinland/ml-data-metrics.

Riitta Piela

2 February 2022

Quarterly Report Q2

Workstream 4

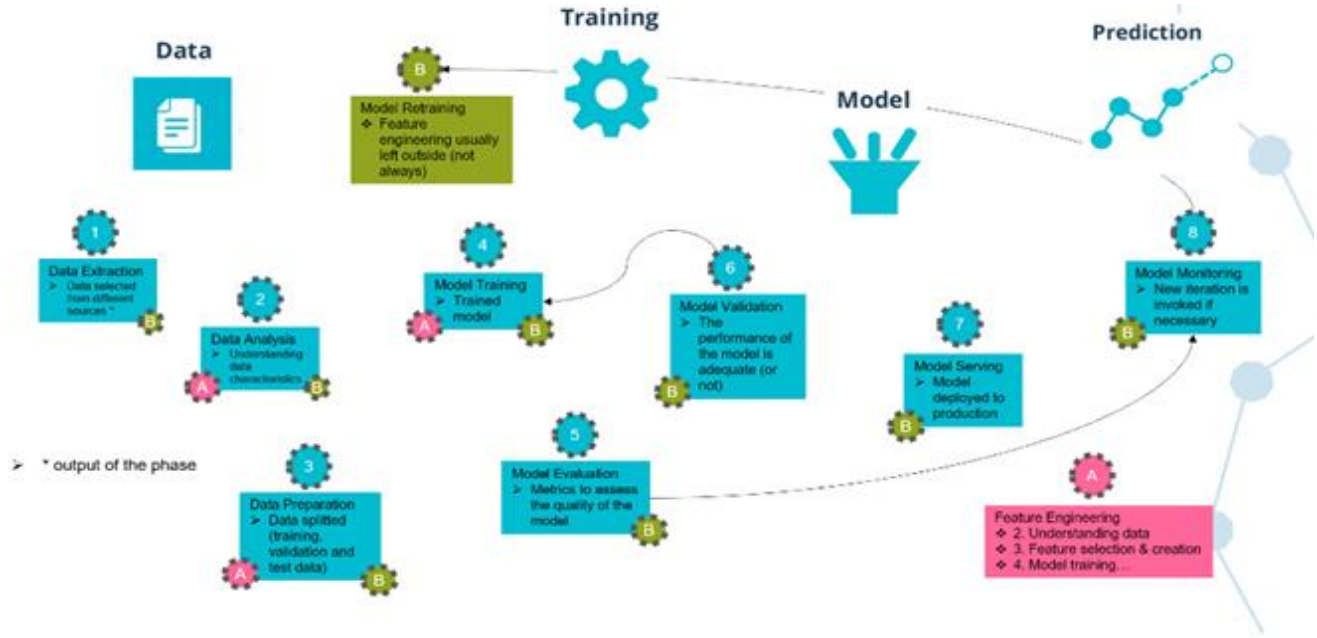
Organization

The original idea in organizing the work was to run the work in three weeks periods (sprints) and at the end of every sprint share with the whole workstream what had had been found out. A number of those who actively took part in the work of workstream (“workgroup”) was at the beginning six persons of which we already in quite early phase partly lost two persons. Although, there were only four people left the group managed to go through at least part of the work that was planned. However, some of the working stream's meetings had to be canceled during May and June because the active workgroup had not been able to produce anything new. Also, a little bit more active participation was expected of all those who were interested in the theme of the workstream 4. However, the leader of the workstream 4 would still encourage all those involved to take a more active part in the group's activities. In addition, the leader would also like to stress that this is primarily a studying group where previous expertise in machine learning is not required and yet, everyone’s skills and knowledge can be beneficial.

MLOps process

The WS4-workgroup first looked at machine learning from the perspective of the Google Cloud’s MLOps machine learning process. Through that process, the group tried to outline the parts related to machine learning and at the same time understand the formation of various unwanted states in that process (especially those related to ML model’s training/retraining). Group classified machine learning into “data and model issues”. Model issues come after all data issues are solved. Data issues mean that the data is extracted, analyzed, and prepared for the models training process. At the result of those data issues, the quality of data, should be at such a level that the process can move to these model issues. Quality in, quality out. Conversely, this means that if the data is not of sufficient quality, the model will not produce a quality outcome either. Machine learning does not achieve its goal.

Riitta Piela 2 February 2022



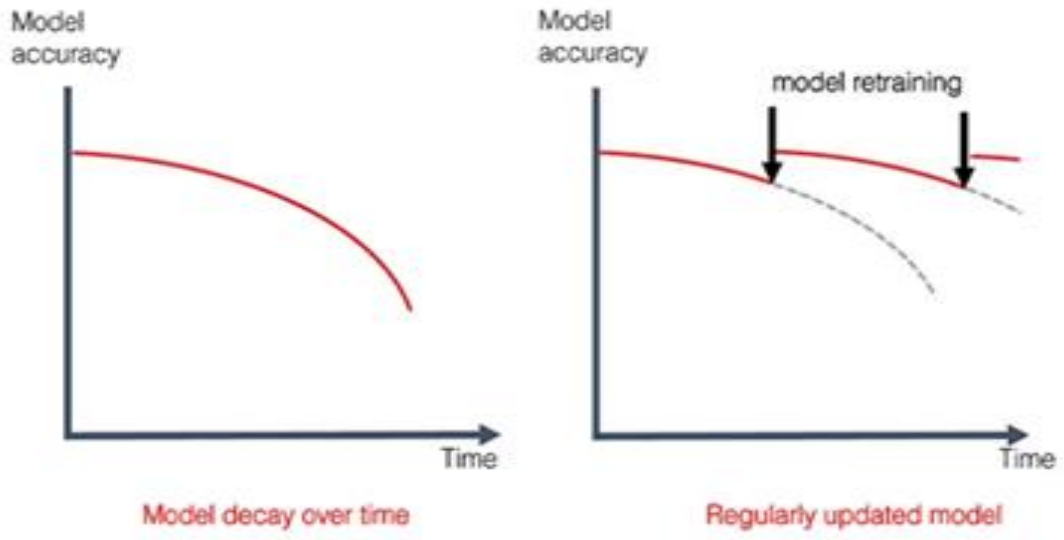
The MLOps process of Google Cloud

Literature review, drifts

But why getting the machine learning models into a production is so difficult? That is one of the questions Workstream 4 wanted to find answers. It has been said that ML model's accuracy will be at its best until you start using it! In production model generates predictions from real data. The world changes around the model and so does the data the model takes in. It means that the predictive performance of the model decreases over time, degradation happens. If the ML model is not monitored the probable degradation cannot be detected. Monitoring the performance of the model is one the key issues. What to monitor was the next question the group wanted to find answers. Drifts are kind of unwanted states that can be monitored to identify possible degradations of the ML model. Definition for drift: a change in an entity with respect to a baseline.

In the literature review workstream identified several drifts, but finally it concluded that there are only two (or maybe three) drifts with real meaning. These two main drifts are data drift and concept drift. Other drifts have been identified in the literature, but they appeared to be refinements or combinations of these two main drifts.

In the machine learning society, it is often talked about model drift (aka model decay or model staleness) although it is caused by either data drift or concept drift. Model drift means that the model just got worse, and it got worse because of changes in data. In other words, model received data that it has not seen during the model's training phase.



Model drift

When there is a data drift behind the model drift it means, to put it even more simply, that input data has changed, and the distribution of the variables is meaningfully different. As a result of that, the trained model is not any more relevant for this new data. The model would still perform perfectly on the data that is like the old data. So, the model is still fine, as much it can be in an isolated world. A second reason that can hide behind the model decay is a concept drift. It occurs when the patterns the model has learned no longer hold.

In contrast to the data drift, in concept drift the distributions can remain the same. What happens in concept drift is that relationships between the model inputs and outputs change. Change in P(Y|X). In essence, the meaning of what we are trying to predict is evolving. Depending on the scale, this makes the model less accurate or even outdated. A real-life example of concept drift: competitor of an online store launches new products. Consumers have more choices, and their behavior changes. As should sales forecasting models. Concept drift occurs.

Methods and metrics for detection of drifts

In the literature there seemed to be a wide variety of methods and metrics available for detection of drifts. For example, quantifying concept drift one could use distance-based metrics, measure how different two different distribution are (Wasserstein metric). For detecting data drifts possible statistical metrics like Kullback-Leiber or Jensen-Shannon could be useful.

WS4 group noticed that it should not dive in too deep in the world of detection of drifts and decays, because that world seemed to be partly undiscovered. There were no ready-made answers available, and the group did not have the opportunity to begin in-depth research at that area. Because different articles had different methods and metrics for calculating drifts, our workstream decided to concentrate on studying the possibilities of ready-made libraries and services. Very promising methods were found in Scikit-Multiflow - library’s ADWIN.

In machine learning, there are two ways in performing model training, batch learning (aka offline learning or static learning) and stream learning (aka online learning or dynamic learning or real-time streaming analytics or incremental learning). The only problem with ADWIN is that it is more suitable for stream learning, where the model is regularly re-trained as new data arrives as data streams. Stream learning is usually the case for systems that use time-series data. Stream learning could still accept data as mini batches, where the system continuously updates itself. However, the test case of Statistics Finland was more batch-learning, but despite that the group believed that ADWIN's possibilities should be simulated in practice. ADWIN and some other drift detection methods were also studied in more depth in the literature review by Abel Dasylva.

The working group also wanted to explore what kind of ready-made tools are available for monitoring the performance of ML models. There seemed to be surprisingly many services available and a couple of them were selected for closer look. The possibilities of Fiddler, Anodot and Evidently (open source) were shortly analyzed, but this area really needs to be further explored. There is no sense in reinventing the wheel again.

Identifying Data Drift Cause with Fiddler

Simulation In addition to the literature review, the purpose of the workgroup was to simulate various findings made in literature review. The group obtained test data through the NSO of Poland and created a test environment for simulation. However, because of insufficient human resources, the simulation could not be carried forward as planned. This is the area, where WS4 will concentrate during the next period.

Riitta Piela

2 February 2022

Quarterly Report Q4 and some final conclusions

Workstream 4

Simulating drifts

Originally the main target of workstream 4 was to concentrate on simulating drifts and finding the right metrics to identify drifts. Due to underestimation of the complexity of research area, too little time was left for the drift simulation. However, in the end WS4 managed to start the simulation phase. In Jupyter Notebook there is a simulation of the process where data drift is analyzed in different phases of ml model lifecycle and Drift detection link in GitHub. Libraries and tools that were used in simulation: Alibi Detect and Evidently (open source tool to analyze data drift).

Conclusions of simulation phase

For drift simulations a dataset of Polish products was used. The dataset contains nearly 16700 product names labelled to sixty-one different categories.

To detect drift, we first need to generate some drifted data. This is done by dividing the product data to two different subsets with different characteristics. The first subset represents the data as it should be, the reference data. The second subset represents drifted data. Splitting the data to different subsets is done with the help of Non-Negative Matrix Factorization (NMF). It should be noted that this method is needlessly complicated, as it is hard to grasp the logic behind it. It would be better to replace this step with a simpler method, hence we will not attempt to explain NMF here.

After we have our data set up, we can move to drift detection. That is, we verify that the drift we have generated is indeed detected. Different methods to detect feature drift (referred as data drift in the notebook) were tested. Concept drift was not simulated due to time constraints. Concept drift would also require additional steps in setting up the data.

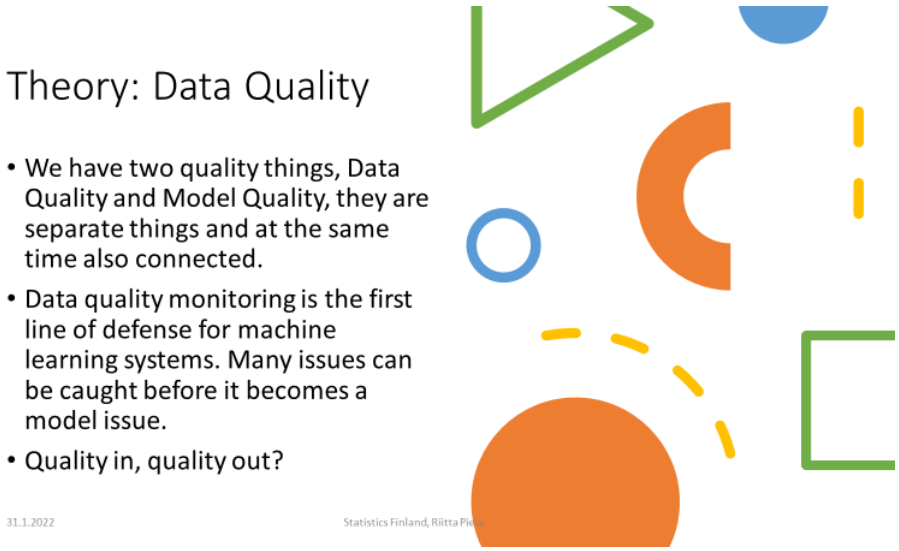
Feature drift detection is first demonstrated by comparing the features (or independent variables) between the reference data and drifted data. Drift detection models work in similar way to machine learning models. The model is first tuned (or trained) with our reference data to learn what kind of data to expect. After tuning, the model can be used to determine if any new data set has drifted.

From the Alibi Detect library Kolmogorov-Smirnov and Maximum Mean Discrepancy algorithms are used. Both algorithms detect the drift we have generated. However, the drift detection in Evidently library fails to detect our drift. This seems to be due to a different threshold used. It also indicates that drift detection is reliant on subjective estimations of what constitutes drift, and there may not be a single answer for every kind of data.

We also demonstrate drift detection with Evidently by comparing the distribution of predicted labels from a ML model to distribution of labels in our reference data. With this method Evidently succeeds in detecting drift. It is probably easier to determine drift from a single variable than from the huge feature set we had in the earlier method.

Data Quality

All though, the workstream 4 was named as Quality of Training Data, the processing of data quality issues in machine learning stayed quite limited in our workstream. Instead of Data Quality, we concentrated on model quality.



As well as finding right metrics for detecting drifts, the so-called data issues in machine learning are an area for which no ready-made solutions are made. Although, IBM research center in India had made some progress at that area. In their presentation Data Quality for Machine Learning Tasks, they have identified special data quality metrics for different ml types, and they name special metrics to classification and regression cases. For example, detecting label noise is especially important in classification cases. IBM researchers also address that among metrics, ordering/sequence of data quality metrics can improve the performance of the model and can serve as a powerful framework to optimize the data quality assessment process. They also stress that measuring data quality in a systematic and objective manner, through standardized metrics, can improve the reliability of the model's performance. Albeit data quality is very important area in producing high quality predictions, much research is still needed in this area.

Model retraining

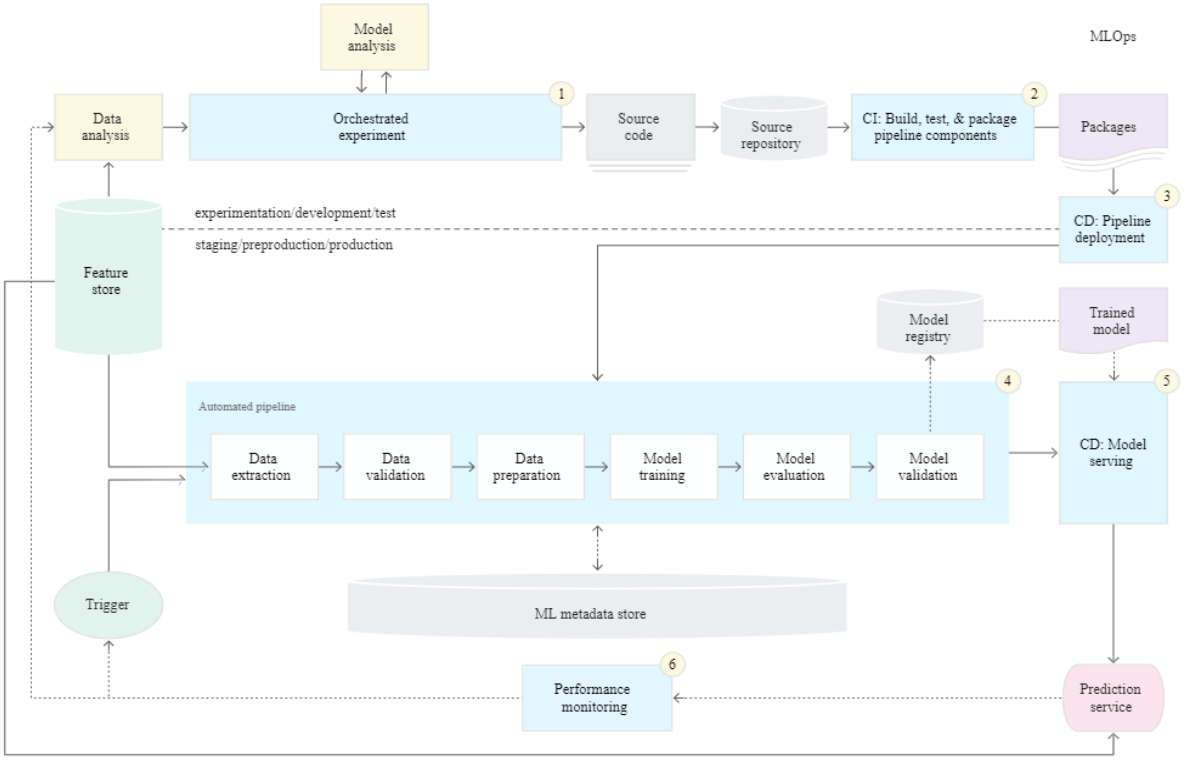
However, why is it so important to identify when model performance is no longer sufficient? As the degradation of the model starts, the model should usually be retrained. How to decide when to retrain the model? It can be based on many factors. First, retraining can be based on interval. It can also be

based on changes in data, or it can be retrained just on demand. Last but not least is retraining on performance-base of the model. Monitoring the performance of a model using the right metrics is a kind of target state. There, the retraining is triggered automatically on performance-base. MLOps provides a platform and process for the model where the performance of the model is monitored and, if necessary, model is also automatically retrained.

Another question is, how to retrain? Also, the retraining approaches vary. All available data can be used in retraining, with assigning higher weights to the new data so that the model would give priority to the recent patterns. All data can be used also without assigning higher weights to new data. If enough data exists, old data can be just dropped. However, the most important thing is that retraining strategy exists.

MLOps

MLOps aka Machine Learning Operationalization Management is an ML engineering practice that aims at unifying ML system development (Dev) and ML system operation (Ops). Practicing MLOps means defending automation and monitoring at all stages of ML system construction. MLOps architecture includes components and processes that are necessary for a high-quality machine learning system. Without MLOps and all the automation and monitoring it offers, implementing machine learning models just as is, may lead to a situation, where replacing labor-intensive process with machine learning solutions, just shifts the need for labor elsewhere (model maintenance for example).



Google Cloud MLOps Level 2.