Improving Automated Retraining of Machine-Learning Models

By Rachel Brower-Sinning

Machine-learning (ML) models are increasingly used to support mission and business goals, ranging from determining reorder points for supplies, to event triaging, to suggesting courses of action. However, ML models degrade in performance after being put into production, and must be retrained, either automatically or manually, to account for changes in operational data with respect to training data. Manual retraining is effective, but costly, time consuming, and dependent on the availability of trained data scientists. Current industry practice offers MLOps as a potential solution to achieve automatic retraining. These industry MLOps pipelines do achieve faster retraining time, but pose a greater range of future prediction errors because they simply offer a refitting of the old model to new data instead of analyzing the changes in the data. In this blog post, I describe an SEI project that seeks to improve MLOps pipelines by adding automated exploratory data-analysis tasks.

Improved MLOps pipelines can

- reduce manual model-retraining time and cost by automating initial steps of the data-retraining process
- provide immediate, repeatable input to later steps of the data-retraining process so that data scientists can spend more time on tasks that are more critical to improving model performance

ML systems that incorporate an MLOps pipeline with improved automated data analysis will therefore be able to more quickly adapt models to operational data changes and reduce instances of poor model performance in mission-critical settings. As the SEI leads a national initiative to advance the emergent discipline of AI engineering, the scalability of AI, and specifically machine learning, is crucial to realizing operational AI capabilities.

Proposed Improvements to Current Practice

Current practices for refitting of an old model to new data have several limitations: They assume that new training data should be treated the same as the initial training data, and that model parameters are constant and should be the same as those identified with the initial training data. Refitting is also not based on any information about why the model was performing poorly; it has no informed procedure for how to combine the operational dataset and the original training dataset into a new training dataset.

An MLOps process that relies on automatic retraining based on these assumptions and informational shortcomings cannot guarantee that its assumptions will hold and that the new retrained model will perform well. The consequence for systems relying on models retrained with such limitations is potentially poor model performance, which leads to reduced trust in the model or system.

The automated data-analysis tasks that our team of researchers at the SEI is developing to add to the MLOps pipeline are analogous to manual tests and analyses done by data scientists during model retraining, shown in Figure 1. Specifically, the goal is to automate Steps 1 to 3—analyze, audit, select—which is where data scientists spend much of their time. In particular, we are building an extension of the MLOps pipeline—a *model operational analysis* step—that will execute after the *monitor model* module of the MLOps pipeline signals a need for retraining.

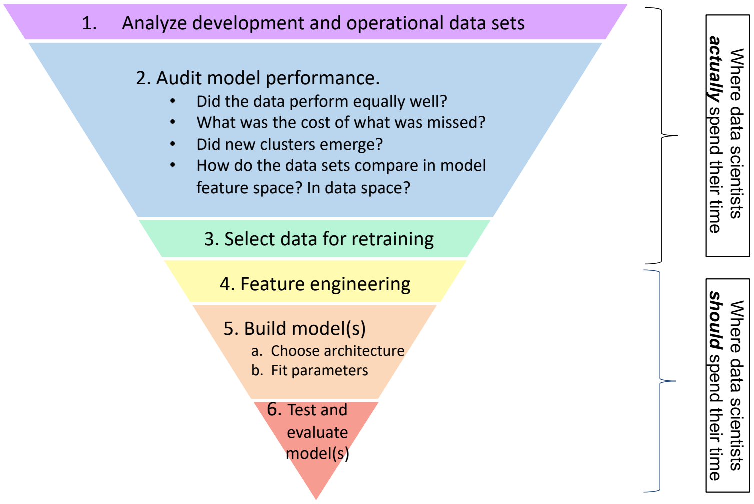
1. Analyze development and operational data sets
2. Audit model performance.
Did the data perform equally well?
What was the cost of what was missed?
Did new clusters emerge?
How do the data sets compare in model feature space? In data space?
3. Select data for retraining
4. Feature engineering
5. Build model(s)
a. Choose architecture
b. Fit parameters
6. Test and evaluate model(s)
Where data scientists actually spend their time
Where data scientists should spend their time

Figure 1: Manual Model Retraining Process Performed by a Data Scientist (Modeled After David Bianco's Pyramid of Pain)

This module will incorporate parts of the analysis, audit, and select steps (Steps 1 to 3) detailed above, and feed the results either to the data-science team to execute Steps 4 to 6, or to the MLOps retraining pipelines to improve automated retraining, as shown in Figure 2.

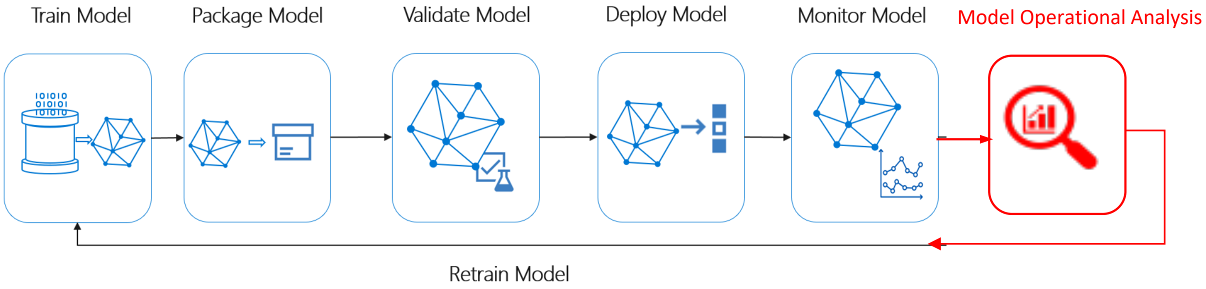

Figure 2: Extended MLOps Pipeline (Adapted From MS Azure MLOps Pipeline)

Approach for Retraining MLOps Pipelines

The goal of our project is to develop a model operational analysis module to automate and inform retraining in MLOps pipelines. To build this module, we will need to answer the following research questions:

- What data must be extracted from the production system (i.e., operational environment) to automate "analyze, audit, and select"?
- What is the best way to store this data?
- What statistical tests, analyses, and adaptations on this data best serve as input for automated or semi-automated retraining?
- In what order must tests be run to minimize the number of tests to execute?

We are following an iterative and experimental process to answer these research questions:

Model and dataset generation—We are developing datasets and models for inducing common retraining triggers, such as general data drift and emergence of new data classes. The datasets used for this task are (1) a simple color dataset (continuous data) with models such as decision trees and k-means, and (2) the public fashion Modified National Institute of Standards and Technology (MNIST) dataset (image data) with deep neural-network models. The output of this task is the models, and the corresponding training and evaluation.

Identification of statistical tests and analyses—Using the performance of evaluation datasets on the models generated in the previous task, we are determining the statistical tests and analyses required to collect the information for automated retraining, the data from the operational environment, and how this data should be stored. This is an iterative process to determine what statistical tests and analyses must be executed to maximize the information gained, yet minimize the number of tests performed. An additional artifact created in the execution of this task is a testing pipeline to determine (1) differences between the development and operational datasets, (2) where the deployed ML model was lacking in performance, and (3) what data should be used for retraining.

Implementation of model operational analysis module—We implement the model operational analysis module by developing and automating (1) data collection and storage, (2) identified tests and analyses, and (3) generation of results and recommendations to inform the next retraining steps.

Integration of model operational analysis model into MLOps pipeline—Here we integrate the module into an MLOps pipeline to observe and validate the end-to-end process from the retraining trigger to the generation of recommendations for retraining to the deployment of the retrained model.

Validating the Model Operational Analysis Module

To validate the efficacy of the model operational analysis module, we begin by comparing the results and recommendations from the developed module with those done by a data scientist. We first establish a baseline by having a data scientist perform the manual retraining of the model. Next, we execute the model operational analysis step for the same model and follow the recommendations to retrain the model. We then compare the recommendations generated by the data scientist in the first three steps of the manual process with those generated by the module. In addition, we compare the performance of the automated retrained model with that of the retrained baseline model.

Expected Outputs of This Project

Our goal is to demonstrate that we can integrate the data analyses, testing, and retraining recommendations that would be done manually by a data scientist into the MLOps pipeline, both to improve automated retraining and to speed up and focus manual retraining efforts. We will produce the following artifacts:

- statistical tests and analyses that inform the automated retraining process with respect to operational data changes
- prototype implementation of tests and analyses in a model operational analysis module
- extension of MLOps pipeline with model operational analysis

Additional Resources

Read the SEI blog post, Software Engineering for Machine Learning: Characterizing and Detecting Mismatch in Machine-Learning Systems.

[Distribution Statement A] Approved for public release and unlimited distribution.

Read other SEI blog posts about artificial intelligence and machine learning.

Read the National AI Engineering Initiative report *Scalable AI*.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM21-0932