Model evaluation in model monitoring

You wouldn't push a new software release to a production environment without running the necessary testing. Similarly, you wouldn't cease the monitoring of a program simply because it's been moved to production. Consistent, ongoing monitoring is an imperative part of the development process. The same is true for the creation and implementation of machine learning (ML) models. While models are being built, trained, and deployed, it's imperative that model monitoring and model evaluation are intentionally embedded into the MLOps lifecycle. By doing so, teams are better able to identify risks, malfunctions, biases, and other issues that can hinder model performance and user experience.

Evaluation in model selection and monitoring is a complex, ongoing process — and one that is essential for building responsible AI. In this article, we'll explore the answers to commonly asked questions about model evaluation:

- When does model evaluation occur?
- Why is evaluating a model important?
- What are the model evaluation methods that should be used?

What is model evaluation used for?

At a high level, model evaluation is used to assess whether or not a model is performing effectively. Of course, no machine learning model can ever be completely accurate one hundred percent of the time due to the inherent limitations of statistical estimations and datasets. But model evaluation metrics and techniques can help assess the accuracy of models, data drift, outliers, bias, and more to ensure that a model is operating as expected during training, testing, and post-deployment. While model evaluation is a crucial component of the entire MLOps lifecycle, there are two fundamental steps where it is most important.

Model training and offline evaluation

In the earlier stages of the MLOps lifecycle, teams identify a problem, collect data, and explore metrics and features. Next, machine learning models must be built, trained, and evaluated. This is done to choose the model and approach that offer the best performance. Most of the data collected during the early stages of the MLOps lifecycle (roughly 70-80%) will go to the training of a model. Yet, that same data cannot be used to assess how a model will perform when introduced to new data. Model evaluation in machine learning, then, is achieved by reserving a portion of the data collected (typically 20-30%) to act as an independent dataset. By introducing this new dataset, developers can mimic a production environment to determine how successful a model will be once it's introduced to real-world data.

Model release and monitoring

Once a model has been deployed into the production environment, it's necessary to continue to monitor it. At this step, model evaluation is used to monitor the ongoing behavior of a model. Elements like data drift, model bias, and performance degradation can all occur—especially as new data is introduced over time. Continuing to monitor and evaluate models can also provide insight into how a model can be retrained with new data in order to improve factors such as overall accuracy and performance.

Why is model evaluation necessary?

There are a number of reasons why model evaluation is an imperative step in the MLOps lifecycle. Perhaps most fundamental, it empowers teams to understand how a model is performing and whether or not it is accurate. But this certainly isn't the entire reason why model evaluation is so critical, nor does it provide the full picture. Here are just a few reasons why monitoring ML models in production and training stages is so necessary.

- **Ensuring the proper training of models:** Launching a machine learning model into a production environment is exciting, but without the proper training and evaluation in place it can quickly spell disaster. When a model hasn't come into contact with real-world datasets, model drift and bias, as well as a host of other issues may occur. To prevent these sorts of problems, developers may split their training data into parts—essentially replicating the process of introducing a model to new, real-world data. This empowers teams to better evaluate models for accuracy and performance, resulting in a better model being pushed to production.

- **Avoiding model bias:** A machine learning model can only be as good as the data it's been trained with. Using incomplete data will likely result in a model that's biased and inaccurate. Regular model evaluation—both during training and post deployment—can ensure that these biases, should they exist, are caught and resolved in a timely fashion.

- **Improving model performance and accuracy:** As mentioned in the previous point, data sets are inherently limited due to factors like scope. Once a model is introduced to new data during testing or post-deployment, the potential for degradation and inaccuracy persist. From selecting a model that performs most effectively to monitoring deployed models for performance and accuracy, model evaluation helps to ensure that a model continues to perform as expected.

- **Maintaining compliance:** Consistent model monitoring and evaluation empower teams to avoid potential regulatory risks that could negatively impact a brand. As model, data, and regulatory requirements change, evaluating a model to ensure it continually meets said standards is business-critical.

How do you evaluate model performance?

Whether you're assessing the effectiveness of a test model or a live one, there are a number of model evaluation methods that can be used. In this next section, we'll take a deeper dive into two of the most important categories: classification and regression metrics. We'll also briefly discuss some other categories of model performance measurement that may be helpful, depending upon the type of model you are evaluating.

Understanding these model monitoring techniques—and any others that apply to your specific models—helps ensure that model training and deployment go as smoothly as possible.

Classification metrics

Model evaluation techniques for classification are used to determine how well a model classifies and segments data into discrete values. When analyzing classification metrics, a confusion matrix is helpful to determine the number of cases that a model correctly and incorrectly classifies. Confusion matrices consist of four primary categories:

- **TN (true negative):** The number of negative cases correctly classified
- **TP (true positive):** The number of positive cases correctly classified
- **FN (false negative):** The number of positive cases incorrectly classified as negative cases
- **FP (false positive):** The number of negative cases incorrectly classified as positive cases

With this information, we can then look at some of the most common classification metrics and their mathematical formulas.

- **Accuracy:** Accuracy measures the percentage of total predictions that were classified correctly. It is calculated with the following formula:

$$ACC = \frac{tp + tn}{tp + fp + tn + fn}$$

- **False positive rate:** False positive rate measures how often a model incorrectly classifies a positive. It is calculated with the following formula:

$$FPR = \frac{fp}{fp + tn}$$

- **Precision:** Precision measures the percentage of positive cases that were correctly classified. It is calculated with the following formula:

$$FPR = \frac{fp}{fp + tn}$$

- **Recall:** Recall measures the percentage of actual positive cases that were correctly classified. It is calculated with the following formula:

$$TPR = \frac{tp}{tp + fn}$$

- **F1 score:** The F1 score balances both recall and precision by calculating their harmonic mean. It is calculated with the following formula:

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

- **Logarithmic loss:** Logarithmic loss is simply a measure of how many errors a model has made. The closer to zero, the more accurate the model is at classification.
- **Area under curve:** This key metric measures how a model is performing by visualizing the true positive rate against the false positive rate.

Regression metrics

Model evaluation techniques for regression provide for a method of measuring continuous output, rather than the discrete values produced by classification metrics. Common regression metrics include:

- **Coefficient of determination:** Otherwise known as R-squared, this metric measures variance, or how well a model fits the actual data. It is calculated with the following formula:

$$R\text{-}Squared = 1 - (UnexplainedVariation/TotalVariation) = 1 - (Variance(model)/Variance(Average))$$

- **Mean squared error:** Mean squared error (MSE) measures the average amount of deviation from the observed data within a model. It is calculated with the following formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

- **Mean absolute error:** Mean absolute error (MAE) measures the average amount a model deviates from the observed data, essentially measuring the horizontal or vertical distance between a data point and the linear regression line.
- **Mean absolute percentage error:** Mean absolute percentage error (MAPE) essentially takes MAE metric and displays it as a percentage.
- **Weighted mean absolute percentage error:** The WMAPE measures the percentage errors by their actual values, rather than by absolute values.

Other common metrics

While classification and regression metrics are perhaps the most common, there are other forms of metrics to evaluate a model that may be useful, depending upon the specific use case. These include:

- **Ranking metrics:** For ranking models and recommendation systems, it's necessary to understand performance and precision. Common metrics include mean average precision (MAP), and normalized discounted cumulative gain (NDCG).
- **Statistical metrics:** The evaluation of statistical models for accuracy and precision is critical. Machine learning model monitoring metrics for statistical models include correlation, computer vision metrics, intersection over union, and more.
- **Natural language processing (NLP) metrics:** For models that deal with language, NLP metrics are used to assess how well a model deals with different types of language tasks. Common model evaluation criteria in this category include the bilingual evaluation understudy store and the perplexity score.
- **Deep learning metrics:** Deep learning metrics specifically assess a model's neural network and how effective it is. Common metrics include the inception score and the Frechet inception distance.

How to monitor a machine learning model more effectively

The ongoing training, monitoring, and evaluating of a model is critical to its success. Yet, without the right tools, observability into the way a model is functioning is limited at best. Model performance management (MPM) platforms can empower MLOps and data science teams to better manage and monitor their models. When seeking the right solution, look for features and capabilities including:

- Performance monitoring

- Data drift detection
- Prediction drift
- NLP and CV monitoring
- Class imbalance detection
- Feature qualify detection
- Ground truth updates
- Real-time alerts

Ready to explore how a model performance management tool can empower your team's ML lifecycle? Learn more about Fiddler's capabilities today.