### Introduction

Best practices from software engineering are increasingly being brought to bear on machine learning. This includes lessons from DevOps (resulting in the term MLOps) and more recently, observability.

While the ML observability discussion has spawned a lot of opaque terms like “model drift”, “continual learning”, and “retraining triggers”, ML systems are similar to other consumers of data like dashboards in that they can greatly benefit from the basics of data observability. Data observability refers to the practice of setting up process, tooling, and infrastructure so that you have access to the state of your data and data pipelines at any given moment.

In this blog post, we cover 9 tips for incorporating data observability into your machine-learning operations:

### Understand what is Data Observability

Data observability is a concept derived from principles used in SRE (Site Reliability Engineering) and DevOps, which have been adapted for the world of data engineering and data science. It refers to the capacity to constantly monitor and have complete knowledge of the internal workings of all tables and pipelines in a data stack. Unlike traditional methods that involve data quality assessment or data pipeline testing based on pass/fail conditions, data observability continuously collects signals from datasets, enabling monitoring and anomaly detection to be conducted on those signals.

Data observability encompasses both “data pipeline monitoring” and “data quality monitoring”. Data pipeline monitoring is about overseeing underlying data infrastructure and ETL (Extract, Transform, Load) processes, ensuring a smooth flow of data across various stages of the pipeline. By doing so, it helps avoid data logjams and guarantees that data is current. Data quality monitoring, on the other hand, is focused on the actual contents of the data. It looks at things like the age of the values (freshness), the level of completeness (rate of nulls, blanks, etc.), duplication, and compliance with the prescribed format.

Data observability platforms like Bigeye typically sit on top of your data warehouse, running periodic queries to check certain statistics calculated on your data, and then automatically alerting you if those statistics go out of bounds. Platforms like Bigeye are also able to combine those alerts with additional data lineage information to help you discover the root cause of problems.

### Understand why you need data observability if you have ML models

Data observability is essential in managing machine learning (ML) models for several reasons:

1. **Data Quality:** ML models depend heavily on the quality of the input data. If there are issues in the data, such as missing values, outliers, or incorrect entries, these can significantly affect the model's performance. Data observability helps ensure the quality of the data by tracking and alerting about such anomalies.

2. **Data Drift**: Over time, the statistical properties of the input data to ML models may change, a phenomenon known as data drift. This can degrade the model's performance, even if the model itself has not changed. Data observability tools can monitor the data for signs of drift, allowing teams to address the issue before it significantly impacts the model's performance.

3. **Model Performance Monitoring**: Even if an ML model was accurate when it was deployed, its performance can degrade over time due to changes in the data or the environment in which it operates. Data observability allows for the continuous monitoring of model performance and can provide early warnings when the model's accuracy begins to drop.

5. **Operational Efficiency**: In many cases, data pipelines can break or fail silently, causing delays in data processing and ingestion. Data observability provides real-time visibility into these pipelines, ensuring that data is available when and where it is needed.

6. **Identifying issues as far upstream as possible:** Finally, the most important benefit of data observability when it comes to ML model training pipelines is being able to identify problems as far upstream as possible, before they impact models or customers. For example, suppose you find 10,000 unusable images in your data set after retraining your model, wasting significant resources. Data observability helps avoid these situations by monitoring data at every stage of the pipeline.

### Focus on Service Level Indicators

When machine learning teams get started with observability, it can be overwhelming. The bedrock of data observability (any kind of observability) - is monitoring metrics. But what metrics to monitor when you have a machine learning model in production? There are numerous statistics, like KL divergences for features, or even recall and/or precision, that are frequently talked about in the ML research context, but that don’t make sense to monitor in a data observability context.

Our suggestion is to start with a single “metric to rule them all” that is understandable to all stakeholders and directly tied to business value, like revenue. This is your service level indicator (SLI). Once you’ve picked an SLI, the second question is: when that service level indicator changes, why is it changing? This is where you can add additional metrics measuring data volume, freshness, correctness, model drift, feature drift, etc. to help you figure out why.

To summarize, changes in your service level indicator (SLI) should trigger alerts, and other metrics are then used to debug what happened. Don't get overwhelmed by monitoring hundreds of irrelevant metrics across your system.

### Monitor Pipelines, Not Just Models

Ideally, machine learning models are not static systems. Data is constantly flowing in for both training and inference, and flowing out in the form of predictions, recommendations, and determinations. These outputs are hopefully then graded, or labeled (by users or by an internal team), and used to improve the model.

While it can be tempting to just monitor model outputs like accuracy or recall, it actually makes more sense to monitor the whole data pipeline. Monitoring pipelines helps identify issues before they impact models, avoiding wasted time and effort. Models depend on the data provided by pipelines, so pipeline monitoring provides insight into model health.

Some concrete examples of what this might look like include:

* Monitoring your production database and “raw” tables as they land in your data warehouse, prior to any transformations, for volume and freshness, to ensure that the “loading” part of ELT has occurred correctly.
* Monitoring “transformed” tables in data warehouses to ensure that initial cleanup was performed correctly.
* Monitoring “feature” tables to ensure that feature calculations are correct.
* Logging all model outputs into the data warehouse and monitoring them to detect model errors and drift.

### Standardize metrics across models

If you have multiple models/pipelines in production, standardize the metrics you monitor for each one. Standardized metrics enable comparison and benchmarking between models.

### Tag and filter metrics for targeted insights

When you are monitoring a number of models, pipelines, data sources, etc., the quantity of data generated can be overwhelming. By tagging metrics based on attributes such as model type, data source, pipeline stage, team, etc., you can organize your data and filter it in a way that delivers targeted insights. For example, if you notice a dip in the performance of a particular model, you could filter all the metrics by that model's tag to quickly identify if the problem is specific to that model or part of a larger issue. Such a system also allows you to customize alerts based on specific tags, helping ensure that the right people are notified about relevant issues.

### Share dashboards of the metrics across teams.

Data observability is a team effort. While a single SLI may be aimed at high-level stakeholders, share more detailed metrics and dashboards across teams - data scientists, data engineers, product managers, etc. This enables all stakeholders to have a view of what's happening in real-time, fostering quicker decision-making and resolution of issues.

### Consider commercial tools

While open-source tools like Prometheus and Grafana are great for getting started with data observability, as your system grows in complexity, you may find that you need more sophisticated tools. Commercial vendors like Bigeye, Arize, and Weights & Biases offer powerful, purpose-built solutions for monitoring machine learning models, data, and systems. These tools come with features like automated anomaly detection, alerting, root cause analysis, and integration with popular data stack technologies, which can save you significant time and effort. They also provide advanced visualizations and dashboards that make it easier to understand and communicate about your data and models. Finally, they often provide support and resources to help you get the most out of your observability efforts. While there is a cost associated with these tools, the benefits in terms of saved time, improved performance, and reduced risk can make them a worthwhile investment.

### Foster a Culture of Continuous Improvement

A prerequisite to both MLOps and data observability is ultimately a culture of continuous improvement. Use the insights gained from data observability to identify areas of improvement, whether it's a data pipeline that's consistently slow, an ML model that's degrading in performance, or a specific type of data issue that keeps recurring, and incorporate these insights into your planning and prioritization process. Encourage teams to focus on resolving these issues and improving the system's reliability and performance.

Remember that data observability is not a one-time effort, but an ongoing practice. Regularly review your metrics and alerts, refine your SLIs, and adjust your monitoring as your data pipelines and ML models evolve.