Verta

Introduction to Model Monitoring

Key components of model monitoring

A robust model monitoring system will give you visibility in the following areas:

1. **Model Performance**
2. **Data Drift**
3. **Outliers**
4. **Model Service Health**

What is model monitoring?

A model monitoring system helps you ensure consistently high-quality results from your model. Model monitoring enables you to:

- ✓ Get real-time insights and alerts about model performance
- ✓ Monitor data characteristics
- ✓ Detect and debug anomalies
- ✓ Initiate proactive actions to improve ML applications

Why is it important?

Without real-time visibility to model performance, companies sometimes don't know about performance degradation until it's too late — they experience social media backlash or user complaints. Sadly, the very organizations that are working hard to implement new ML models into their products are sometimes the very last ones to know when things stop working correctly. Without a robust model monitoring system, organizations leave themselves exposed and open to risk.

Why do models fail?

AI/ML doesn't always work as expected. Models can fail or experience degraded performance for a variety of reasons:

- Data in production often differs greatly from data used to train a model.
- Complex data pipelines and frequently updated models increase the number of failure points.
- The underlying data-generating processes may change.
- Sometimes a model just becomes stale since the data used to build them is simply no longer relevant.
- The product may be used in new markets or with new users, which can lead to data and/or concept drift.
- There could simply be bugs or errors in models, ETL or serving code.

For most teams, **data** is the primary reason why models fail or degrade in performance.

Learn More

1

Model performance

Model quality metrics like accuracy, precision, recall, F1 score, and MSE are a few of the more common ways to measure model performance.

However, organizations often use different metrics for different model types and/or business needs. For example:

- IOU (Intersection Over Union) score for a computer vision model
- AUC (Area Under Curve) and confusion matrix for a classification model
- Perplexity and Word Error Rate for an NLP model

Identify the most relevant quality metrics and a monitoring system can help compute, track and detect performance deviations. When deviations are detected, a good monitoring system will give you the ability to drill down and perform root cause analysis.

Computing model performance using ground truth can pose challenges. Generating ground truth usually involves hand labeling, making ground truth difficult to access in a timely manner. Look for the ability to ingest delayed ground truth labels in a monitoring system for best results. In the absence of ground truth, we recommend using other proxies (e.g. click thru rates for a recommendation engine) for real-time quality feedback. Automate as much of this process as possible and don't rely on adhoc steps.

2

Data drift

Data drift is one of the key reasons why model performance degrades over time.

Data drift can be used as a leading indicator for model failures. Drift monitoring allows you to track distribution of input features, output predictions and intermediate results, and it should also allow you to detect changes.

There are two types of drift tracking

1. Drift over time helps see a sudden/gradual shift in production data (If you are tracking drift over time, make sure to take seasonality effects into account!)
2. Drift compared to training baseline helps show if feature distribution has changed significantly between training and production.

Different algorithms are used for drift detection (e.g. cosine distance, KL divergence, Population Stability Index (PSI) etc). Your data types will determine which drift detection algorithm you should choose, and your monitoring system should offer flexibility to choose.

Your monitoring solution should allow you to configure drift threshold both manually and automatically and give you the ability to get alerted when data drift is detected. For immediate remediation, consider building an automated retraining workflow that is triggered if certain features cross a drift threshold.

3

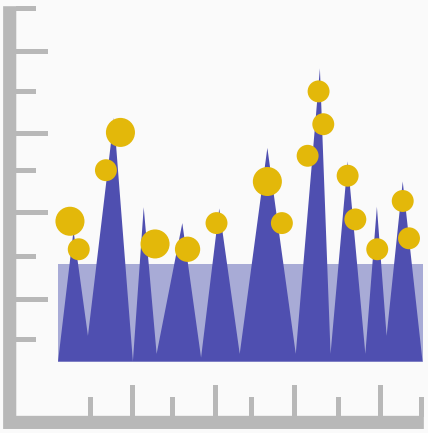
Outliers

Outlier detection is a great way to track anomalous input and output data.

Most often, outliers can help detect issues with data pipelines. For more sensitive models, use outliers to identify edge cases that require manual processing or further review.

A univariate outlier analysis is a good start where you are tracking outliers for a prediction or a single input feature over time. However, for most production use cases you may need to analyze impact across multiple variables; hence, multivariate outlier detection is useful.

Basic data quality monitoring (such as missing data, null values, standard deviation, mean, median, etc.) can be extremely helpful in production.



Outlier Detection Chart

4

Model service health

Along with model performance and data, organizations should monitor the overall service health of models through operational metrics like:

- Response time
- Latency
- Error rate
- Throughput

Inference service latency has a huge impact on user experience for realtime and near-real time systems and needs to have stringent SLAs. Typically, a model monitoring system built for ML may not be the platform of choice for service metrics. Most of the time, you'd be better served streaming those metrics to the APM system.

Learn more about best practices for model monitoring

Best practices for model monitoring

Model monitoring should occur in real time

Allow yourself to take immediate action when necessary. If you are dumping data into a database to post process or following ad hoc monitoring practices, you're introducing significant risk to your organization.

Connect model monitoring with with pre-production workflows and production models

Establishing seamless connections here can help with faster root cause analysis and issue resolution.

Model monitoring should be fully automated as soon as a model is deployed

Don't put yourself in a situation where you're relying on other data scientists or ML engineering to configure features, setup alerts, etc.

Model monitoring should fit your needs—not the other way around

Metrics and techniques used for model monitoring vary widely depending on type of data and problem. Monitoring systems should be customizable, extensible, so allow you to monitor a variety of data types. You should be able to compute and visualize data in different ways for different audiences.