The Next Frontier: Observability in Machine Learning Systems

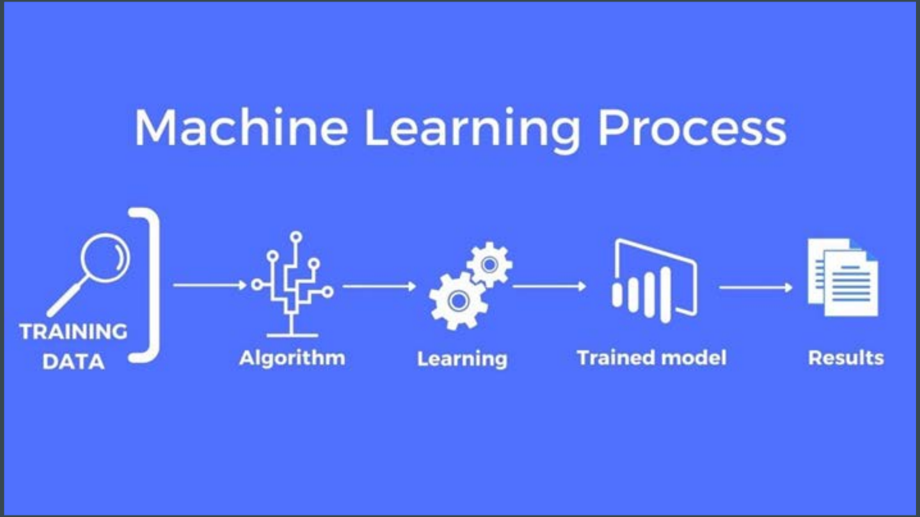
How to Ensure Reliability and Performance in Complex Models

Who Am I?

- Software Engineer With Over 5 years of experience delivering digital banking solutions to financial institutions in West Africa.
- Founder of two companies in Nigeria:
  - Waste Management Company
  - Music Company (Record Label)

Slide 3: What is Model Observability?

Model observability is the practice of validating and monitoring ML model performance and behavior. It involves measuring critical metrics, indicators, and processes to ensure models work as expected in production environments.



How is Model Observability Different From Model Monitoring?

Model Monitoring:

- Collects and analyzes metrics over time.
- Detects anomalies and trends.
- Ensures models operate within thresholds.
- Focuses on system health.

Model Observability:

- Provides real-time insights.
- Diagnoses issues within processes.
- Reveals underlying system dependencies.
- Understands why anomalies occur.

Monitoring
Observability

Why is Model Observability Important?

- **Transparency:** AI often functions as a "black box," lacking transparency in its processes.
- **Error Detection:** Users may not notice when large language models (LLMs) like GPT-4 make mistakes.
- **Credibility:** Unidentified errors can harm the credibility of the model.
- **Understanding:** Observability provides insights into why errors occur.
- **Trust:** Maintaining visibility and understanding of model behavior helps sustain user trust in AI systems.

Why is Model Observability Important: An Example

Google's chatbot Bard:

Right after its launch, Bard claimed in a promotional campaign that the James Webb Space Telescope took the first ever image of an exoplanet. *This was not true.*

<u>Consequences:</u>

- Customers raised a doubt of the model's efficiency
- Google reportedly lost $100 billion in market value because of the blunder

How ML Observability Helps

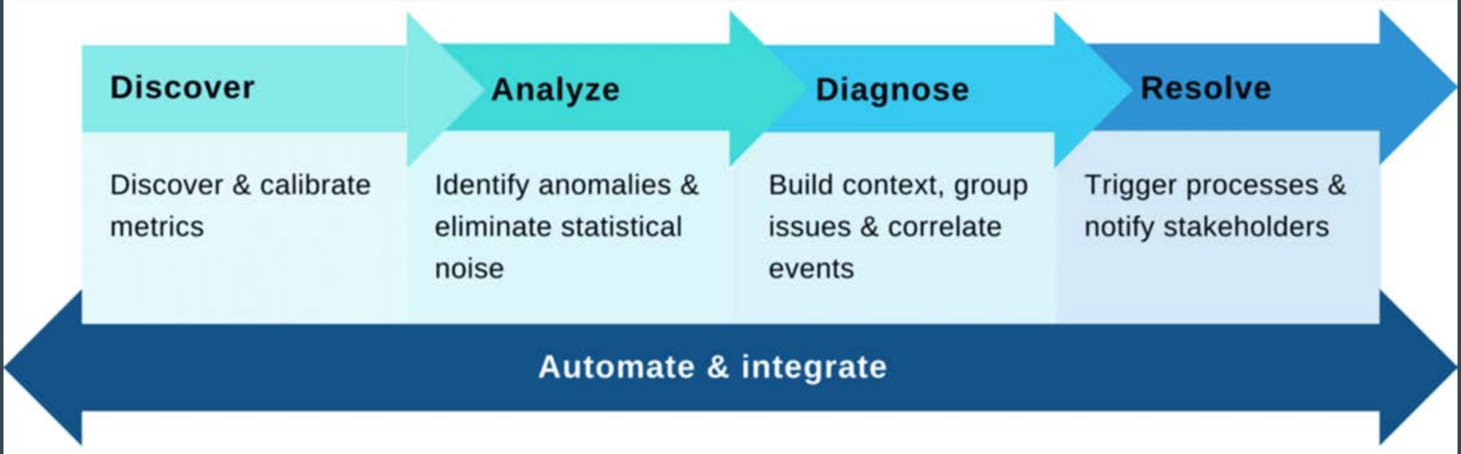
ML observability enables engineers to perform ***root-cause analysis***, identifying the reasons behind specific issues.

Benefits:

- Continuous performance improvement
- Ensures expected behavior in production
- Streamlined ML workflow
- Scalability
- Reduces time to resolution

Key Components of ML Observability

- **Event logging:** Detailed logs of model activities
- **Tracing:** Tracking data through stages
- **Model profiling:** Performance analysis
- **Bias detection:** Identifying and mitigating biases
- **Anomaly identification:** Detecting unusual patterns


Discover
Discover & calibrate metrics
Analyze
Identify anomalies & eliminate statistical noise
Diagnose
Build context, group issues & correlate events
Resolve
Trigger processes & notify stakeholders
Automate & integrate

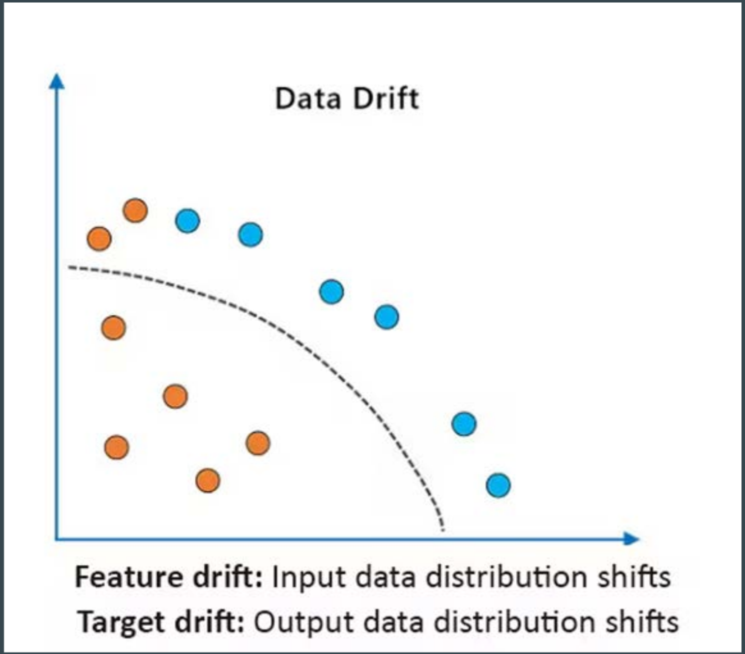
Key Challenges: Data Drift

Data Drift:

- Occurs when the statistical properties of the training data change over time.
- Can include covariate shift (changes in input feature distributions) and model drift (changes in the relationship between input and target variables).

Causes:

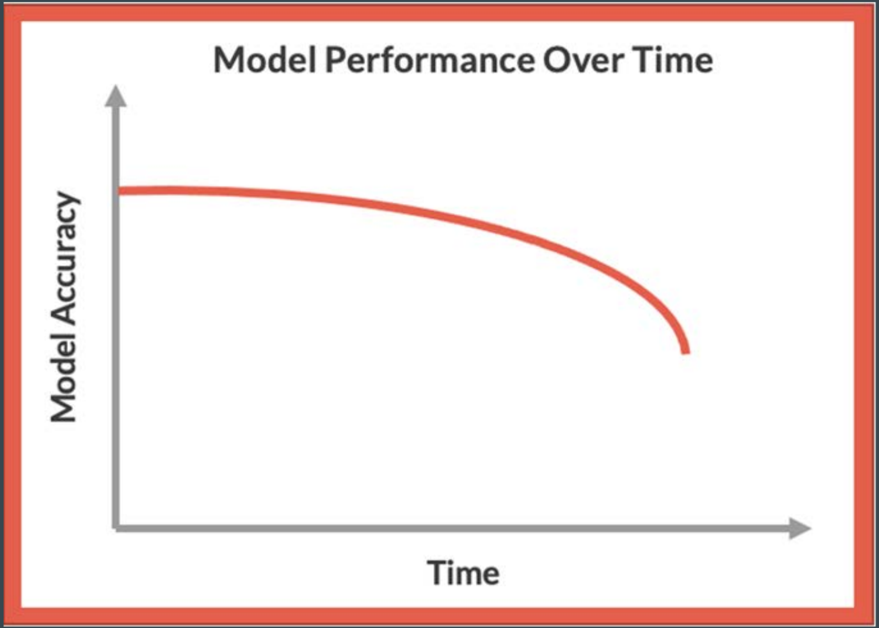
- Changes in customer behavior
- Shifts in the external environment
- Demographic changes
- Product updates and upgrades



Feature drift: Input data distribution shifts

Target drift: Output data distribution shifts

Key Challenges: Performance Degradation



As machine learning applications gain more users, ***model performance can decline*** over time due to:

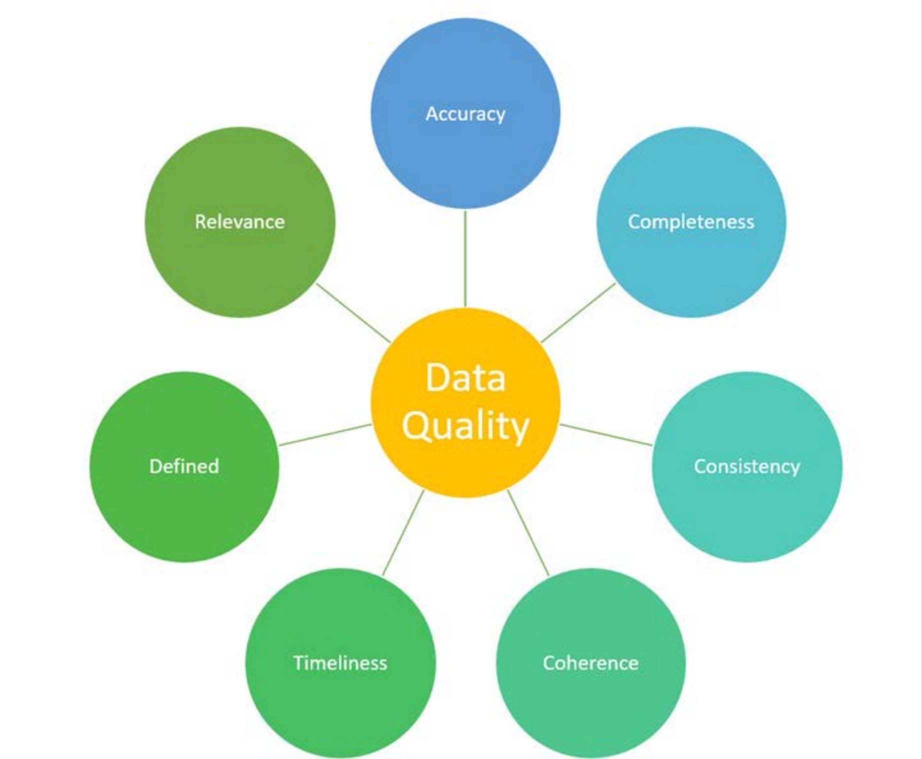
- Model overfitting
- Presence of outliers
- Adversarial attacks
- Changing data patterns

Key Challenges: Data Quality

Maintaining consistent data quality in production is challenging due to the reliance on various factors such as *data collection methods, pipelines, storage platforms, and preprocessing techniques*.

Possible issues include:

- Missing data
- Labeling errors
- Disparate data sources
- Privacy constraints
- Inconsistent formatting
- Lack of representativeness



Model Observability Challenges in LLMs

Large Language Models (LLMs) face unique issues:

- **Hallucinations:** Generating nonsensical or inaccurate responses.
- **No Single Ground Truth:** Multiple plausible answers make evaluation difficult.
- **Response Quality:** Responses may be correct but irrelevant or poorly toned.
- **Jailbreaks:** Prompts can bypass security, leading to harmful outputs.
- **Cost of Retraining:** Ensuring up-to-date responses requires expensive retraining.

Evaluation Techniques for LLMs

A tailored model observability strategy can help address challenges and improve evaluation.

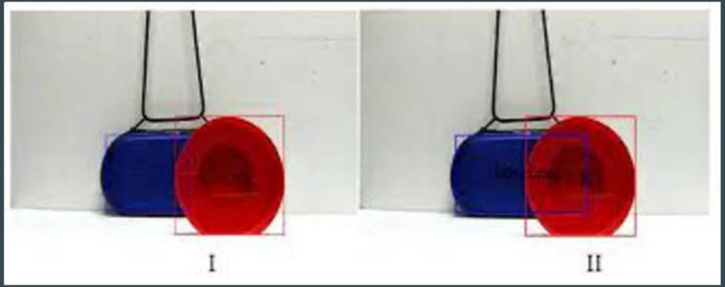
Common techniques include:

- **User Feedback:** Collect and assess reports on bias and misinformation.
- **Embedding Visualization:** Compare response and prompt embeddings for relevance.
- **Prompt Engineering:** Test various prompts to enhance performance and detect issues.
- **Retrieval Systems:** Ensure LLMs fetch correct information from relevant sources.
- **Fine-tuning:** Adjust the model with domain-specific data instead of full retraining.

Challenges in Computer Vision

- **Image Drift:** Changes in image properties over time, like lighting and background.
- **Occlusion:** Objects blocking the primary object, leading to misclassification.
- **Lack of Annotated Samples:** Difficulty in finding labeled images for training.
- **Sensitive Use Cases:** Critical applications like medical diagnosis and self-driving cars where errors can be disastrous.

An example of occlusion:



Components to Address Challenges in Computer Vision

- **Monitoring Metrics:** Measure image quality and model performance.
- **Specialized Workforce:** Involve domain experts in the labeling process.
- **Quality of Edge Devices:** Monitor remote devices like cameras and sensors in real-time.
- **Label Quality:** Ensure high-quality labeling with automation and regular reviews.
- **Domain Adaptation:** Indicate when to fine-tune models based on data divergence.

Monitoring Techniques in ML Observability

- **Standard ML Metrics:** Precision, recall, F1 score, AUC ROC, MAE.
- **LLM Metrics:** BLEU, ROUGE, METEOR, CiDER for automated scoring. Use human feedback, custom metrics, and RLHF for human-based assessments.
- **CV Metrics:** Mean average precision (mAP), intersection-over-union (IoU), panoptic quality (PQ) for tasks like object detection, classification, and segmentation.

Explainability Techniques in Standard ML Systems

Explainability is the capability of observability tools to provide clear, understandable insights into system behavior and performance, enabling stakeholders to easily interpret and act on the data.

Two techniques you can use to interpret a model's decision-making process:

- **SHAP:** Shapley Additive Explanations (SHAP) computes the Shapley value of each feature, indicating feature importance for global and local explainability.
- **LIME:** Local Interpretable Model-Agnostic Explanations (LIME) perturbs input data to generate fake predictions. It then trains a simpler model on the generated values to measure feature importance.

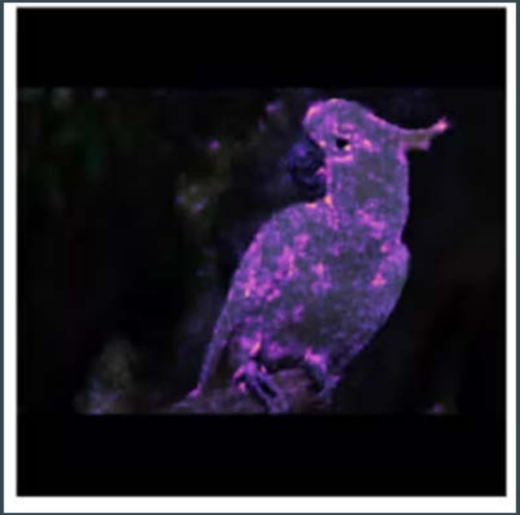
Explainability Techniques in LLMs

- **Attention-based Techniques:** Visualize which words the model considers most important in an input sequence. Useful in models like **ChatGPT**, **BERT**, and **T5** that use transformer architecture.
- **Saliency-based Techniques:** Compute gradients with respect to input features to measure their importance. Masking features and analyzing output variations can reveal crucial features.

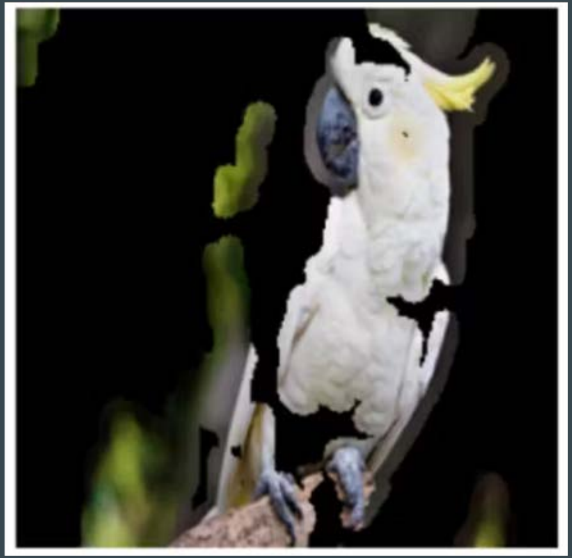
Explainability Techniques in CV

- **Integrated Gradients:** Builds a baseline image and adds features gradually, computing gradients to identify important features for object prediction.
- **XRAI:** Enhances Integrated Gradients by highlighting pixel regions instead of single pixels, segmenting similar image patches and computing saliency for each region.
- **Grad-CAM:** Generates a heatmap for CNN models, highlighting important regions by overlaying the heatmap on the original image.

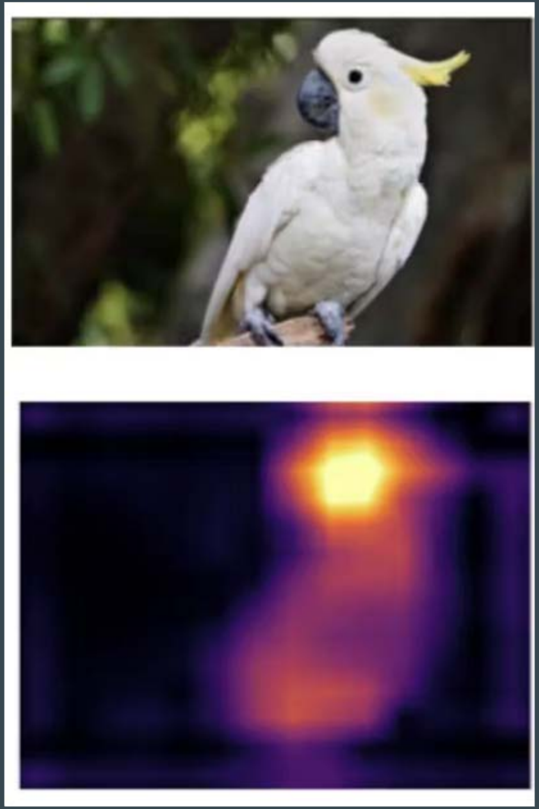
Explainability Techniques in CV



Integrated Gradients



XRAI



Grad-CAM

Future Trends in Model Observability

- **User-Friendly XAI:** Developing techniques to generate simple, understandable explanations.
- **AI Model Fairness:** Using XAI to visualize learned features and detect bias.
- **Human-Centric Explainability:** Combining insights from psychology and philosophy for better explainability methods.
- **Causal AI:** Highlighting why a model uses particular features for predictions, adding value to explanations and increasing robustness.

Thanks for Attention