publicis
sapient

# ML Model Monitoring

AI & DATA COE



July 19, 2022

# About the Speaker

Ravi Ranjan is working as Manager Data Science at Publicis Sapient (India) with expertise in budling scalable ML solutions. He is a certified Google Cloud Architect. He is contributor and member at Kubeflow (ML Platform by Google).

publicis
sapient

# Session Logistics

1. Recording of the session will be accessible on MS Teams.
2. Presentation and source code will be available on PS Code. [ https://bit.ly/model-monitoring-artefacts ]
3. We will address the Q&A at the end of the session.
4. Pulse Check - https://bit.ly/ML-Monitoring

SCAN ME

SCAN ME

publicis
sapient

## Learning Outcome

1. Understand the importance of ML Monitoring in ML Lifecycle.
2. Hidden technical debt in ML systems.
3. What is ML Model Monitoring?
4. How to build real-time Model Monitoring pipeline?

# Agenda

1. The ML System Lifecycle
2. Hidden technical debt in ML systems.
3. Need of Model Monitoring
4. What is Model Monitoring?
5. Working demo
6. Best Practices
7. Tools and Tech Stack
8. Q&A

Section 01

# Machine Learning Lifecycle

# ML Life-cycle - Introduction

Why ?

# Problem

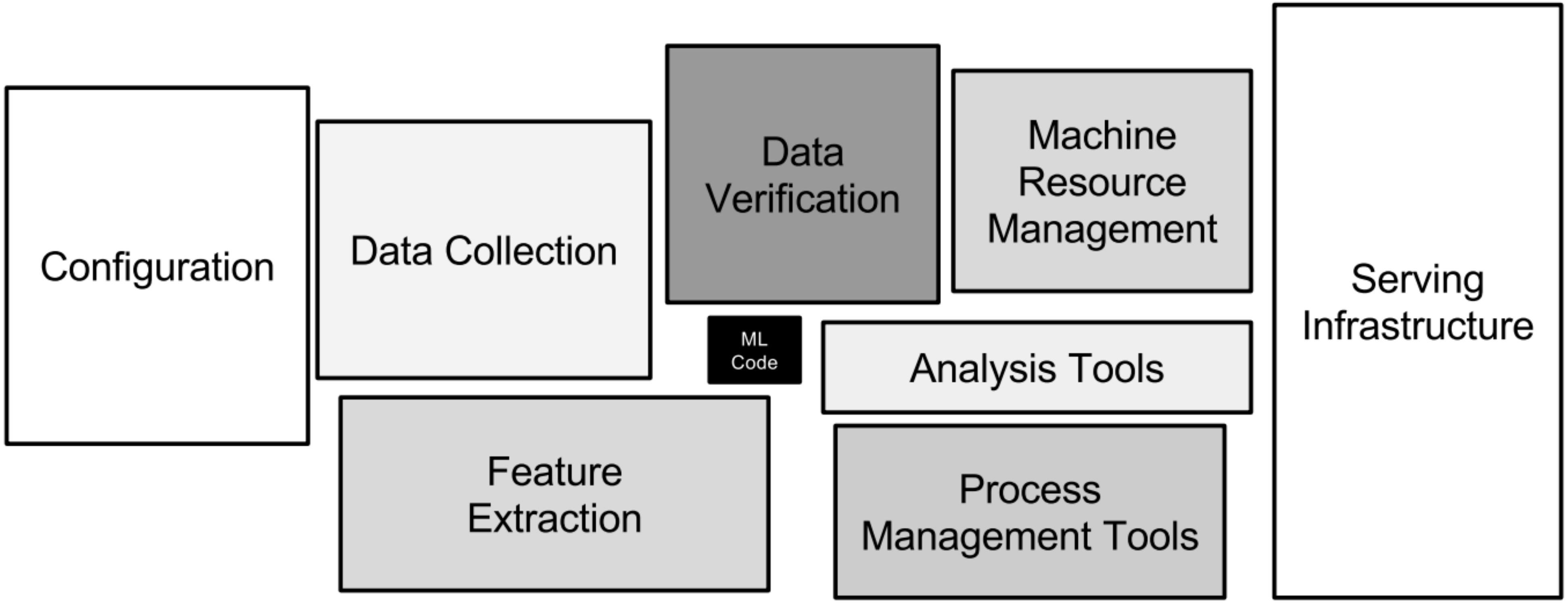
We don’t have a ML workflow. What we have done is take our **old traditional SW development** workflows and stuck some ML applications in there and it lacks things that ML needs.

ML experimentation is like the wild west. Ad-hoc tools and processes because of lack of standardized tooling. Forget **reproducibility** , it is difficult to track experiments and results.

While **model freshness** is important to the business , we only update our models once every six months, because it is **a manual process**

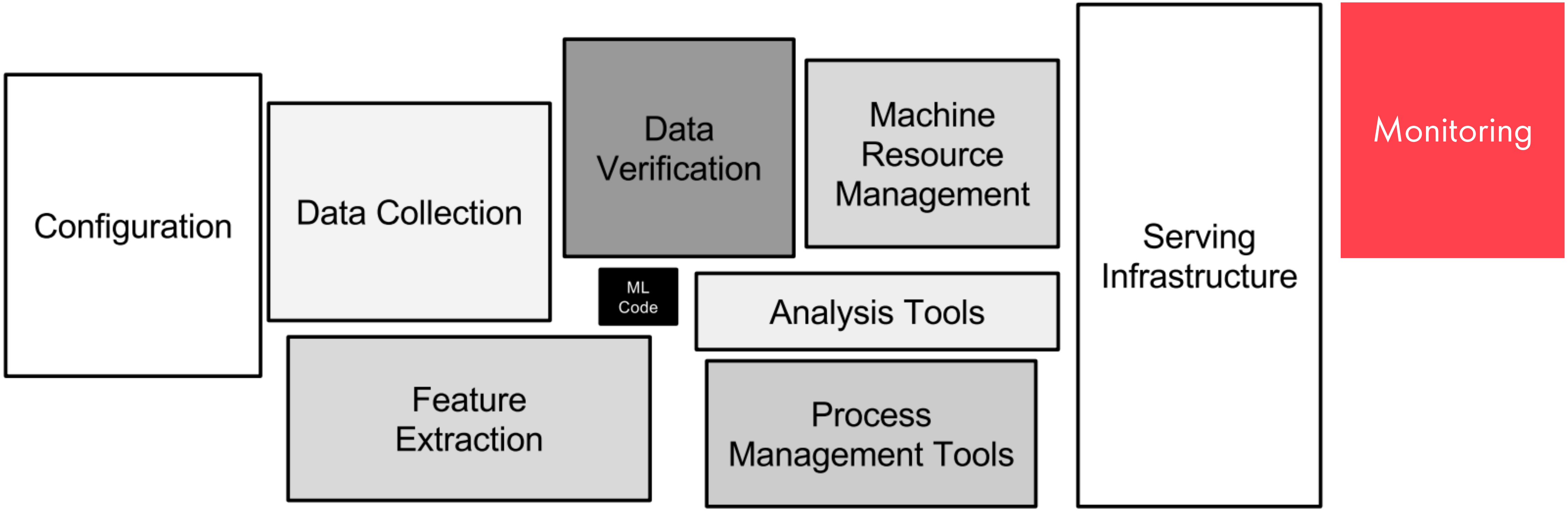
Section 02

# Hidden technical debt in ML systems



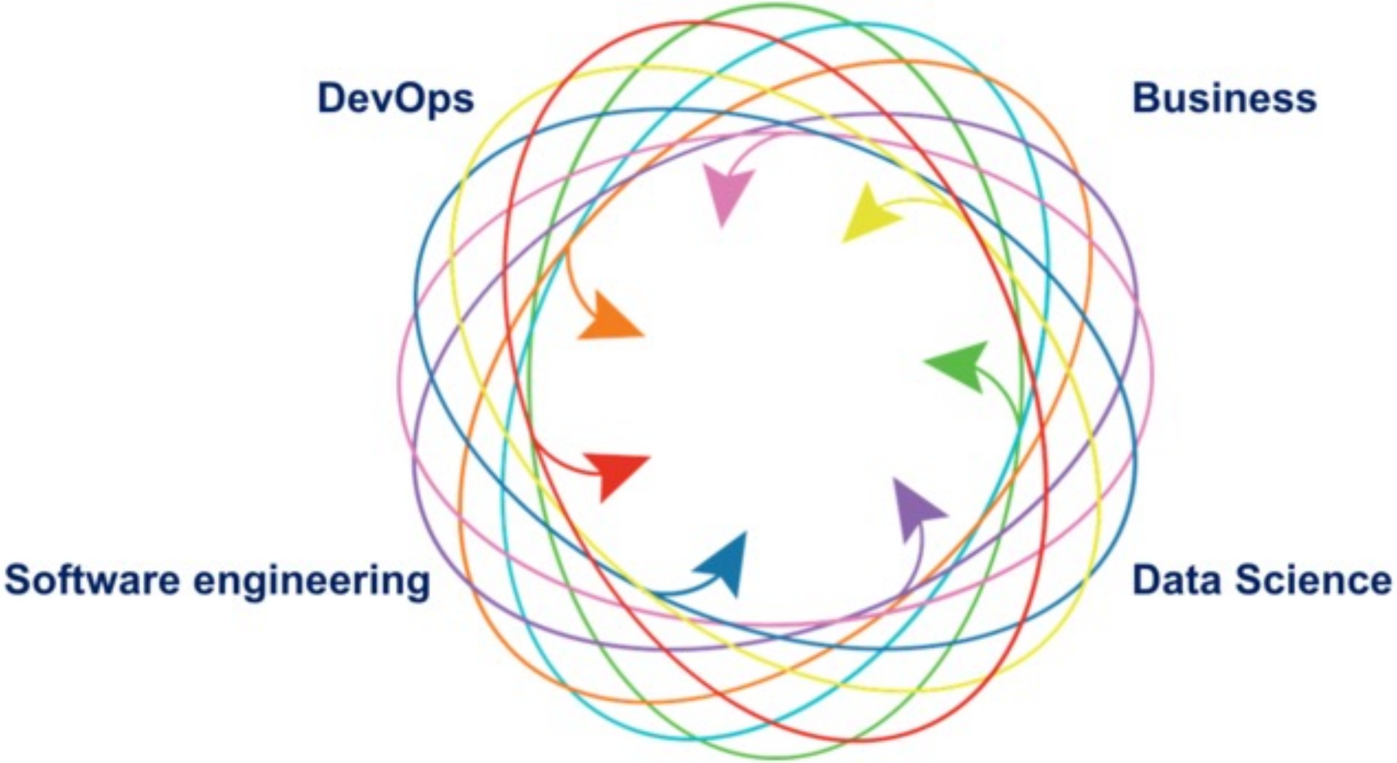

As we talk about machine learning models in production, we are talking about **ML Systems** in place **of ML Applications**


PLEASE
SILENCE
HUMANS




As we talk about machine learning models in production, we are talking about **ML Systems** in place **of ML Applications**

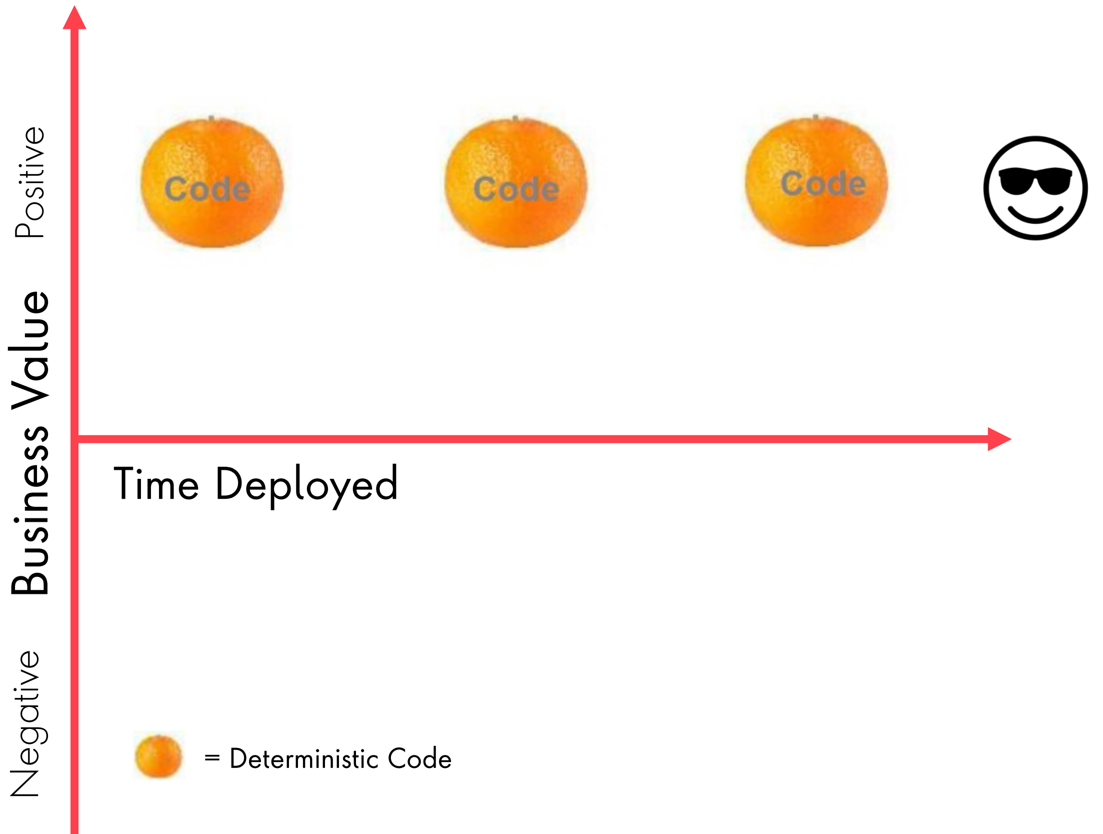
# The Responsibility Shift



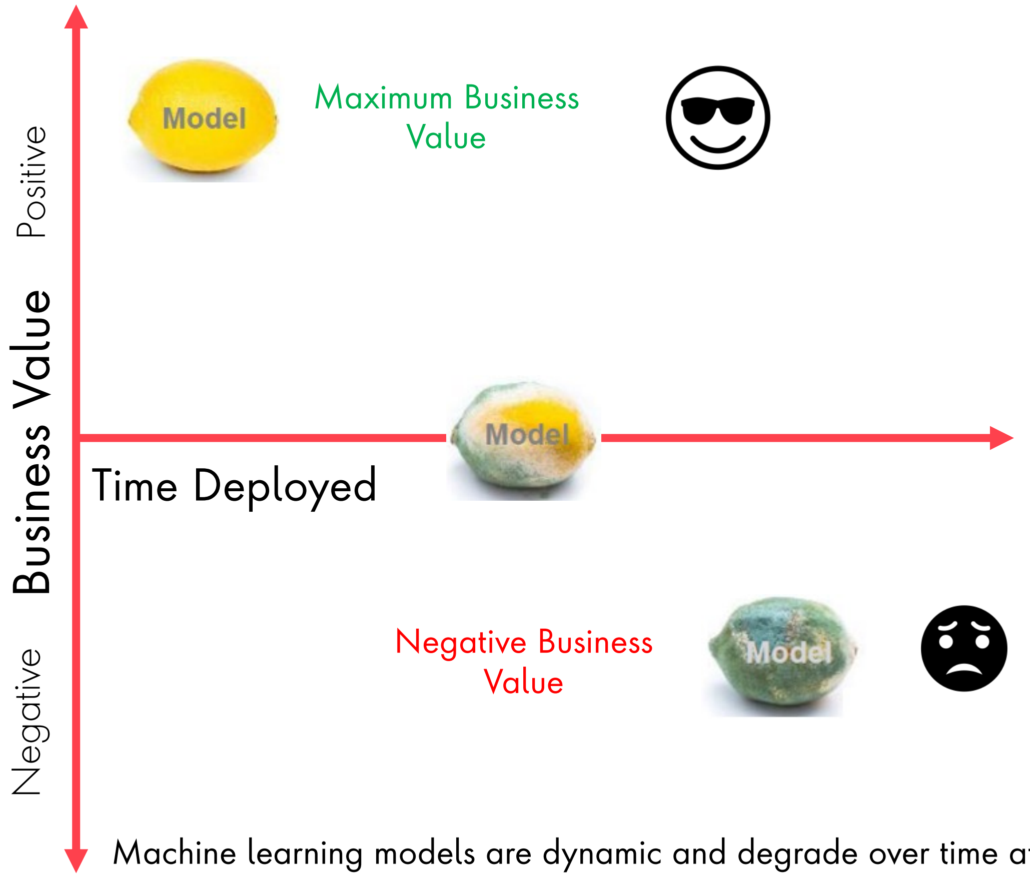
Section 03

# Need of Model Monitoring

# Traditional Software vs ML Systems

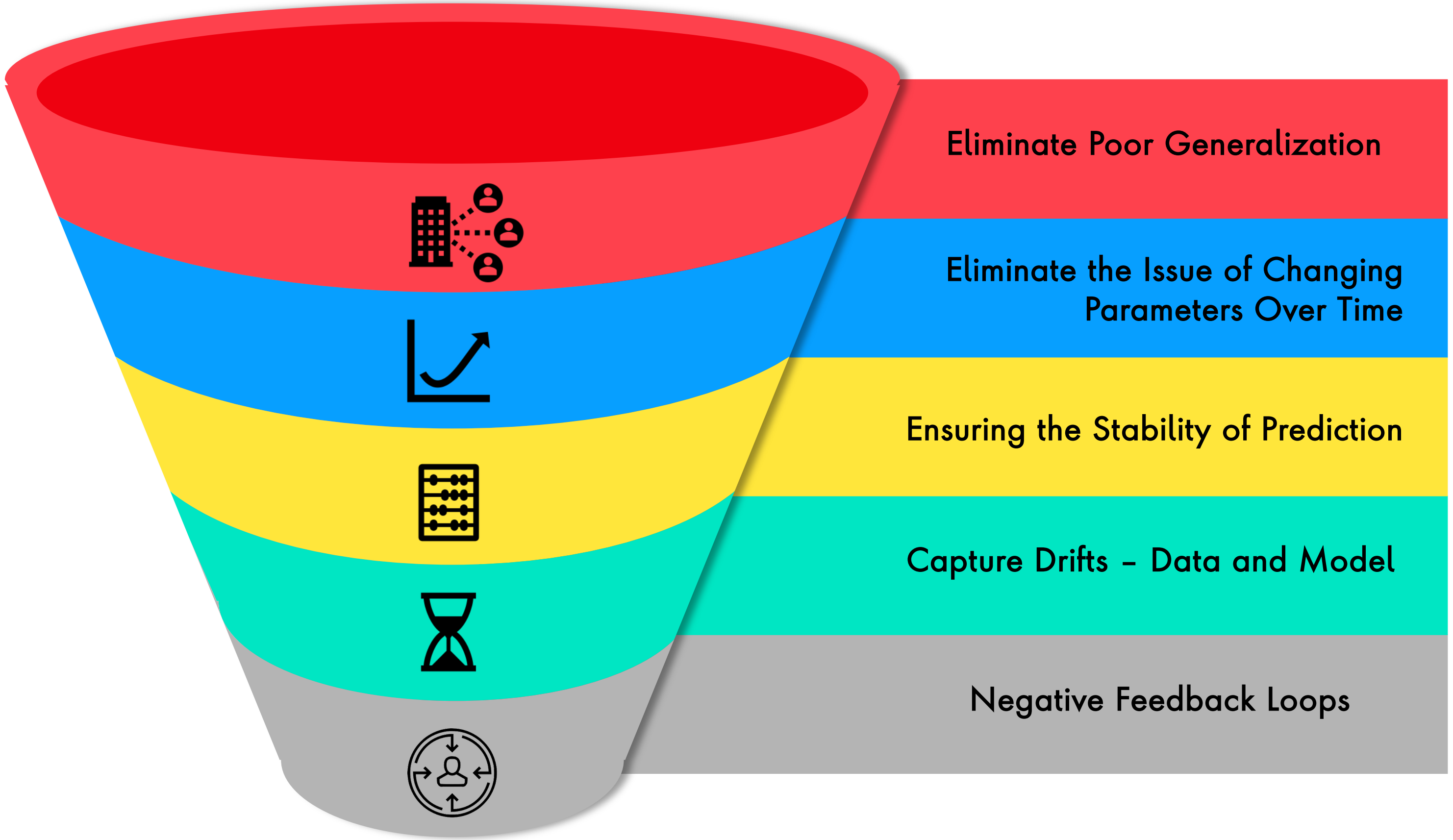



Code is deterministic and always runs as written

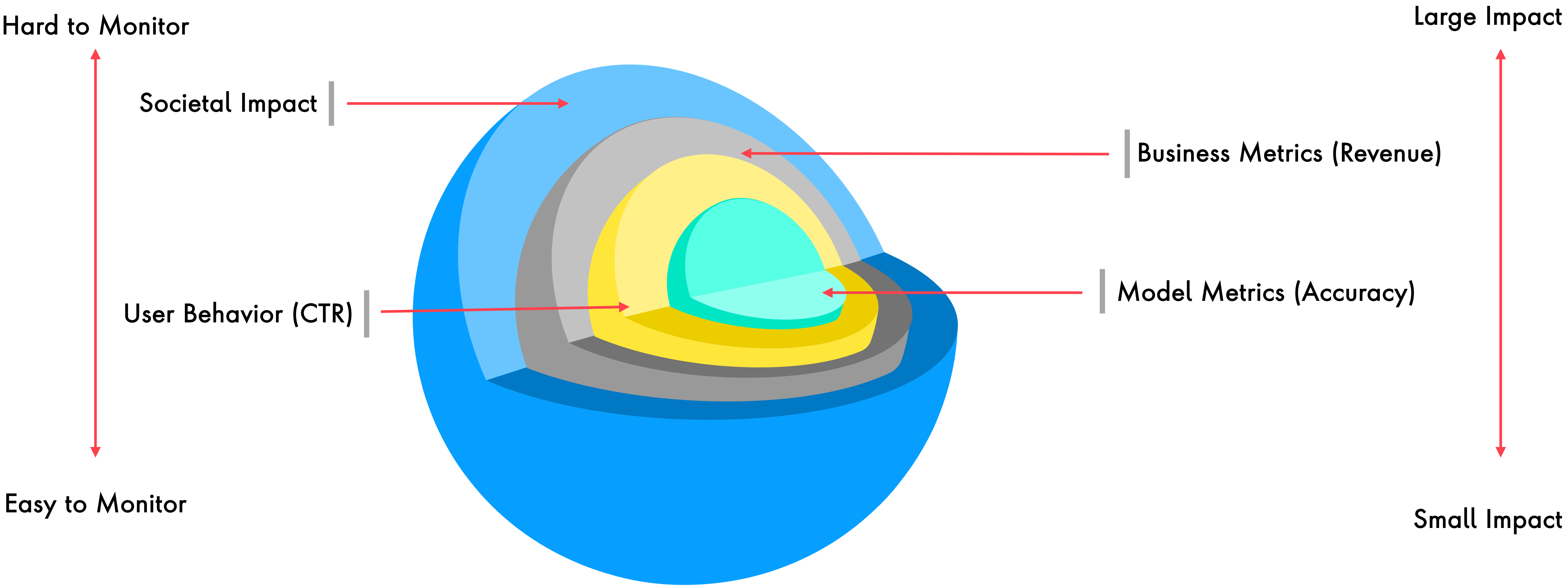



Machine learning models are dynamic and degrade over time after being deployed to production

# Why Is Model Monitoring Needed After Deploying the Model Into Production?

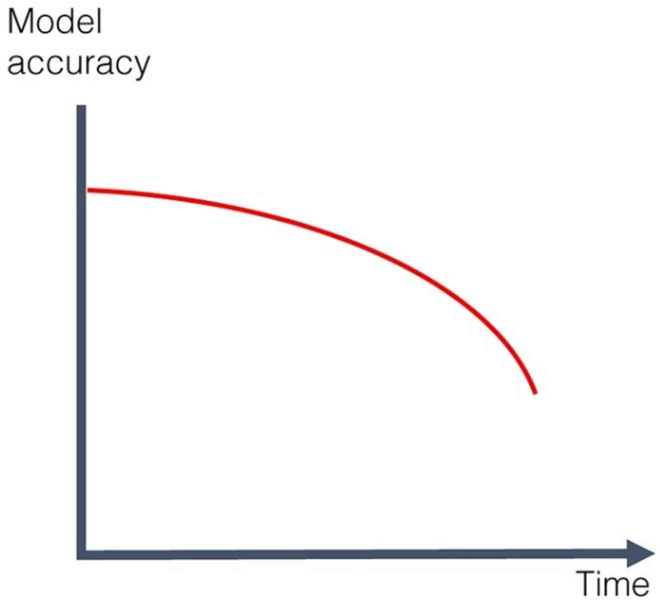


# N$^{th}$ Order Effect

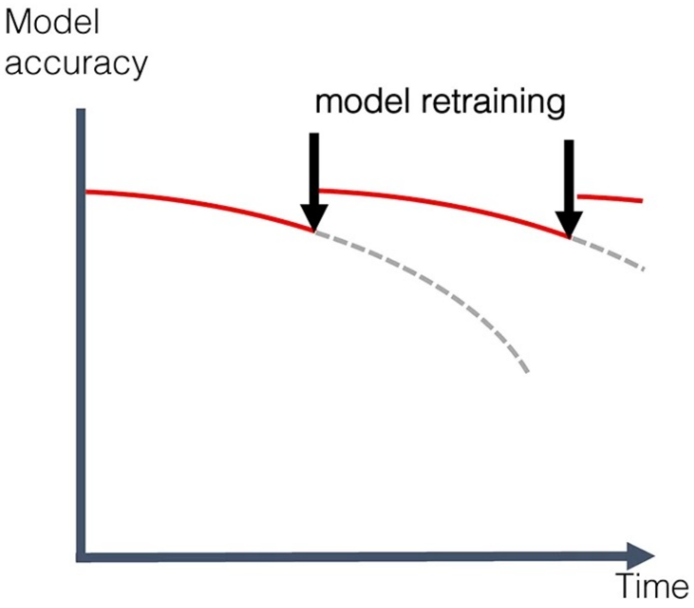



Monitoring should be designed to provide early warnings to the myriad of things that can go wrong with a production ML model.

# No model lives forever, but the speed of decay varies.




Model decay over time




Regularly updated model

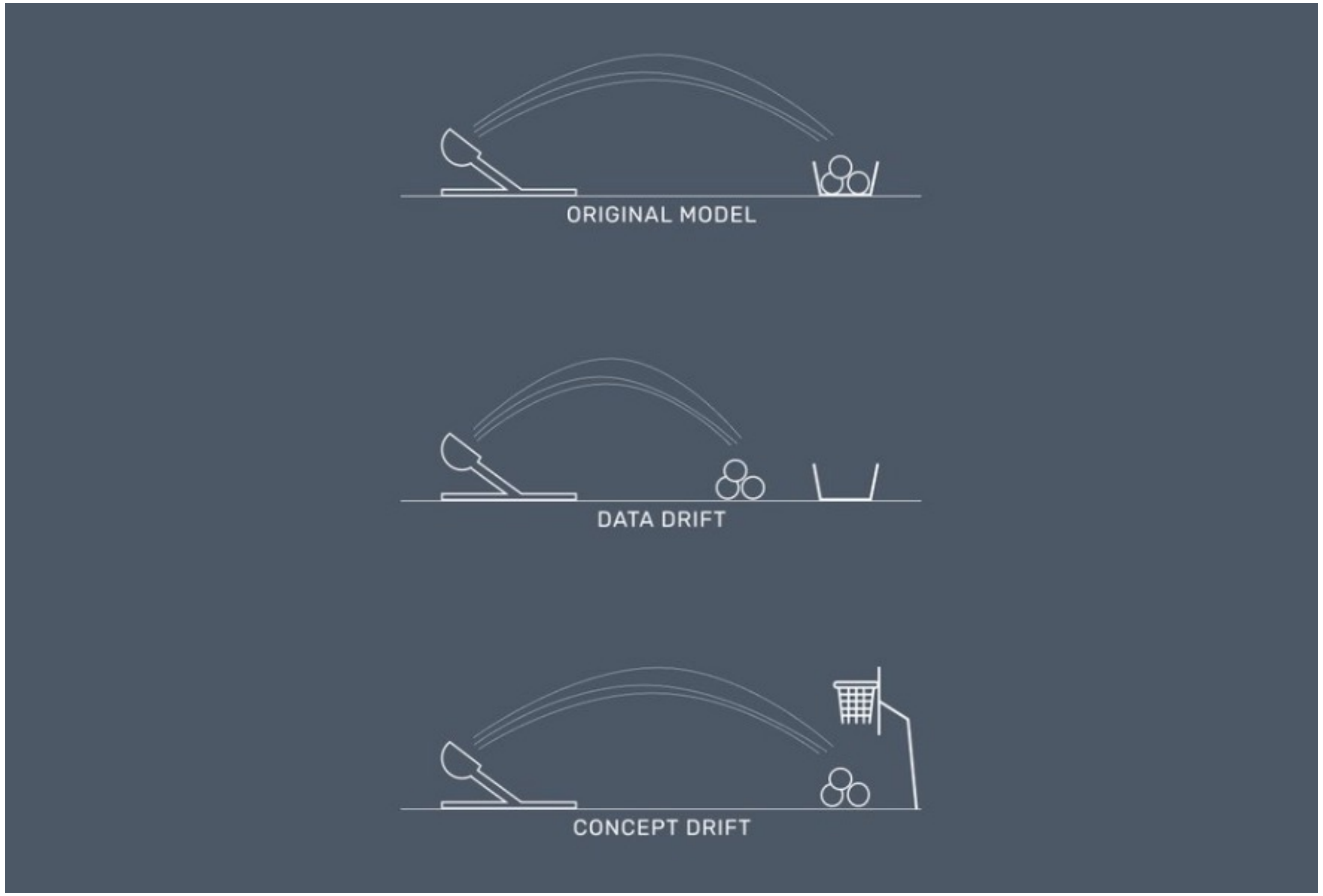
# Drifts

## Data Drift

- Input data has changed
- Meaningful change in distribution between the training data and production data
- The model performs worse on unknown data regions.

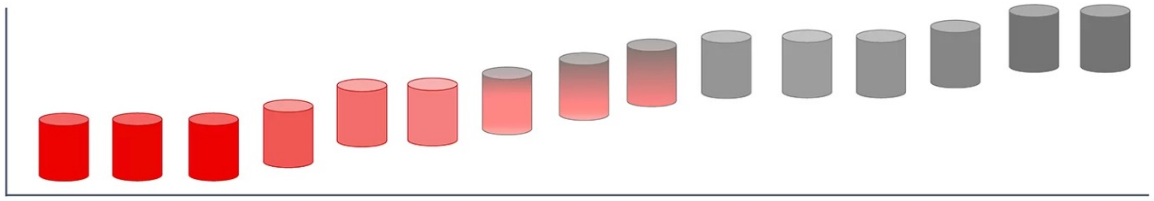
## Concept Drift

- Distributions might remain the same
- Relationships between the model inputs and outputs change
- The world has changed, and the model needs an update.

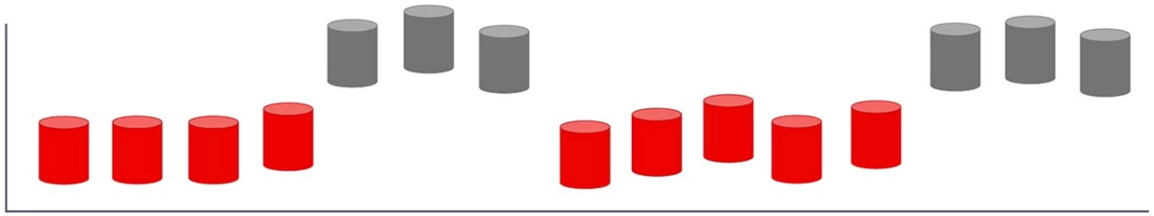


# Concept Drift Patterns

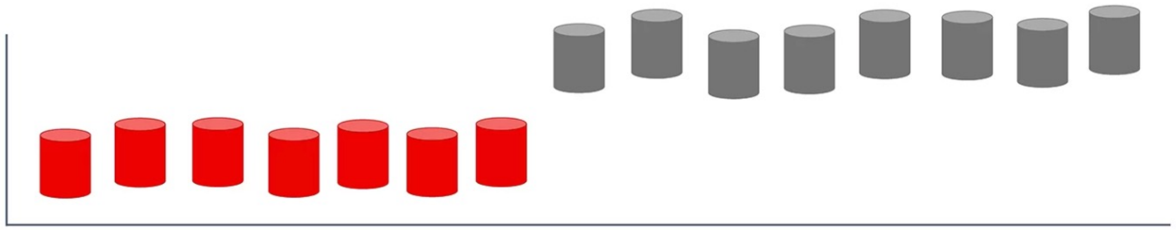
Gradual concept drift



Recurring concept drift



Sudden concept drift



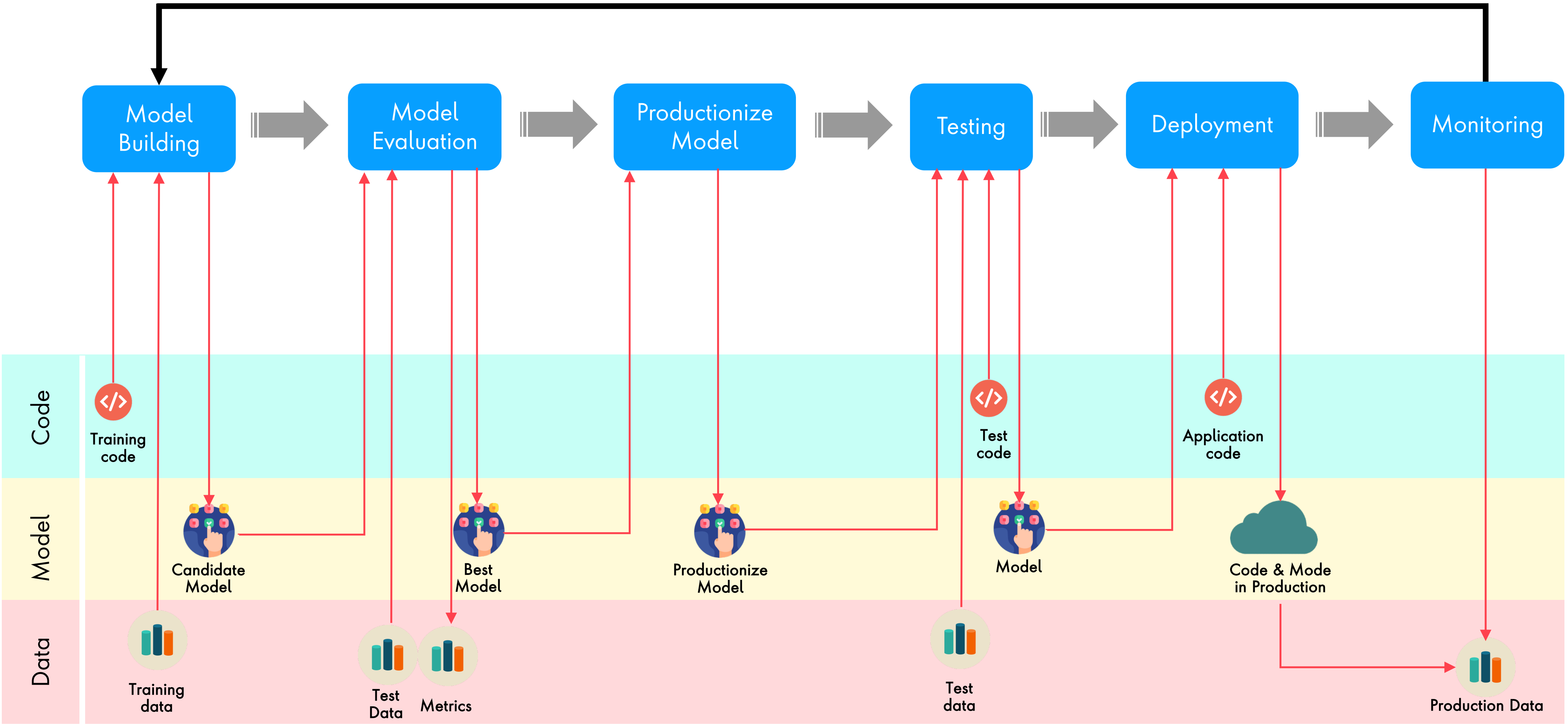
# Drift detector – statistical tests

| Detector | Tabular | Image | Time Series | Text | Categorical Features | Online | Feature Level |
|---|---|---|---|---|---|---|---|
| Kolmogorov-Smirnov | ✓ | ✓ | | ✓ | ✓ | | ✓ |
| Cramér-von Mises | ✓ | ✓ | | | | ✓ | ✓ |
| Fisher's Exact Test | ✓ | | | | ✓ | ✓ | ✓ |
| Maximum Mean Discrepancy (MMD) | ✓ | ✓ | | ✓ | ✓ | ✓ | |
| Learned Kernel MMD | ✓ | ✓ | | ✓ | ✓ | | |
| Context-aware MMD | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Least-Squares Density Difference | ✓ | ✓ | | ✓ | ✓ | ✓ | |
| Chi-Squared | ✓ | | | | ✓ | | ✓ |
| Mixed-type tabular data | ✓ | | | | ✓ | | ✓ |
| Classifier | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Spot-the-diff | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Classifier Uncertainty | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Regressor Uncertainty | ✓ | ✓ | ✓ | ✓ | ✓ | | |

Section 04

# Model Monitoring

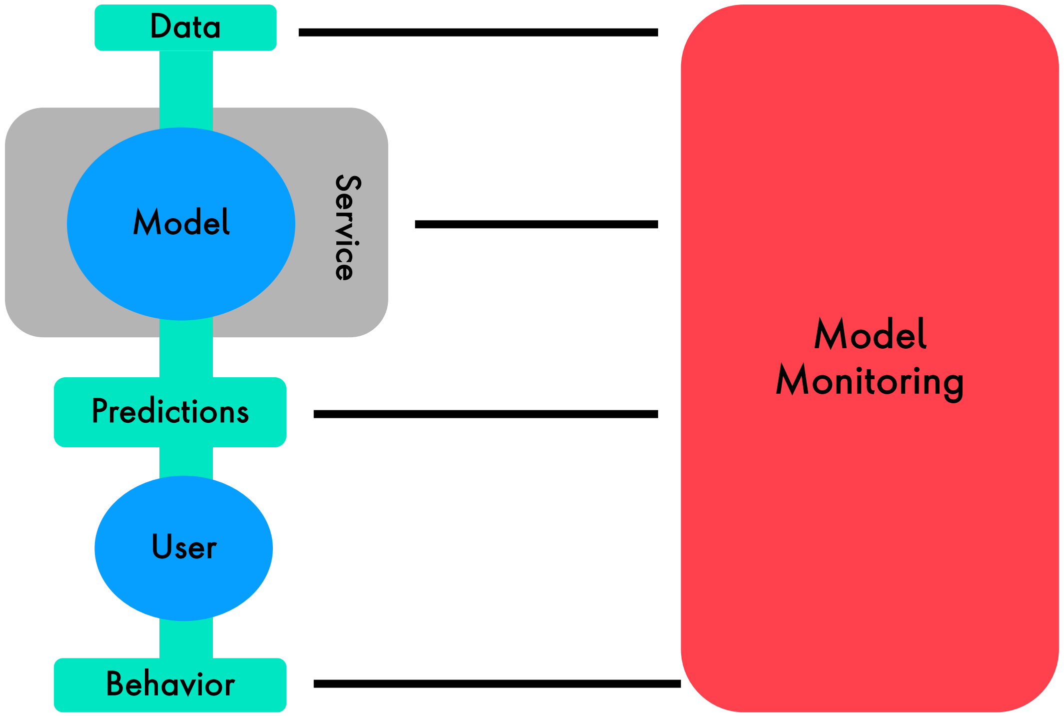
# ML Pipeline



Model Building
Model Evaluation
Productionize Model
Testing
Deployment
Monitoring
Code
Model
Data
Training code
Candidate Model
Training data
Test Data
Metrics
Best Model
Productionize Model
Test code
Model
Test data
Application code
Code & Mode in Production
Production Data

# What Is Model Monitoring?

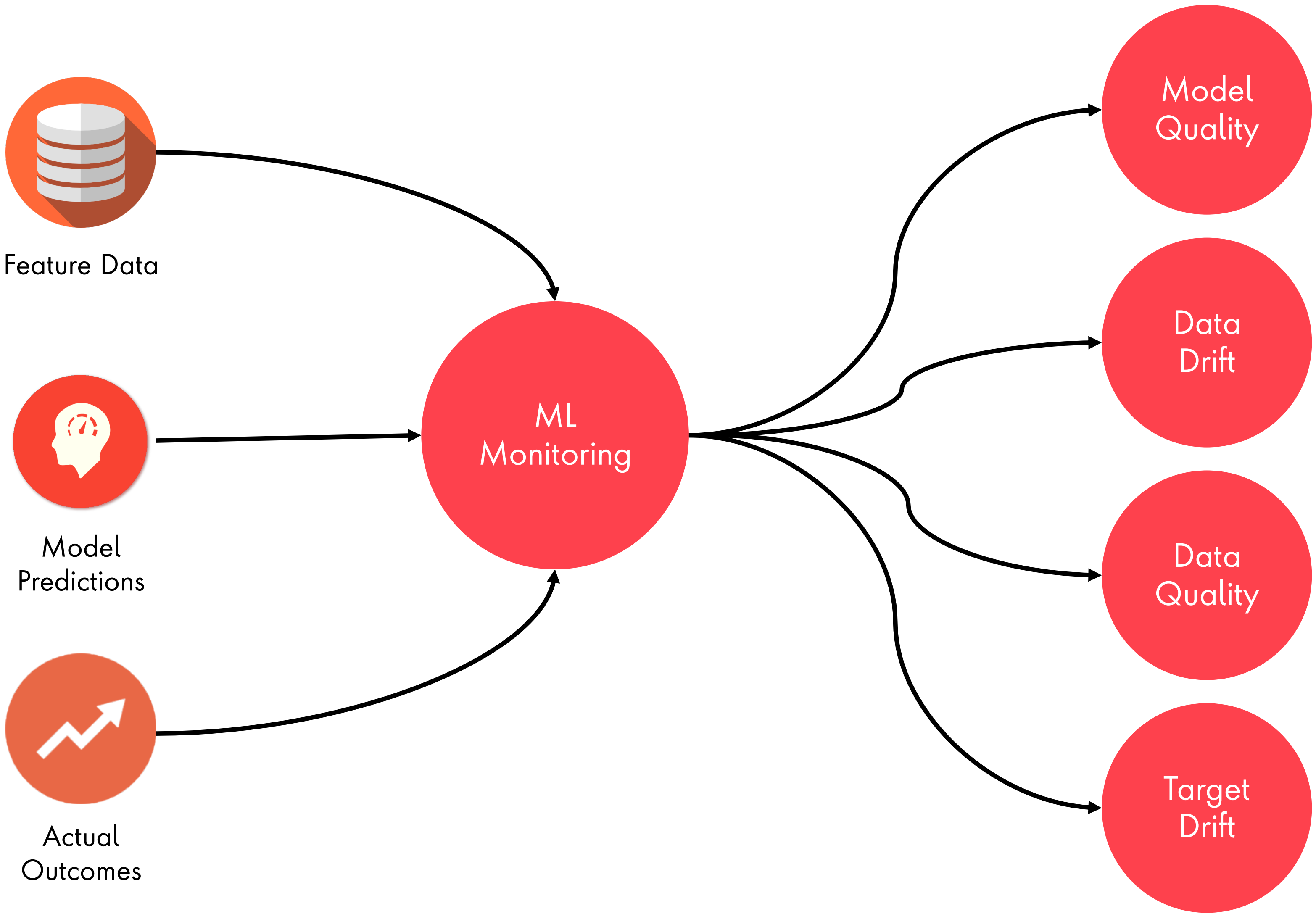
Model monitoring refers to the process of closely tracking the performance of machine learning models in production. It enables the AI team to identify and eliminate a variety of issues, including bad quality predictions and poor technical performance. As a result, the machine learning models deliver the best performance.



# What Should One Monitor?

| System Monitoring | ML Monitoring | | |
|---|---|---|---|
| | Data | Model | Predictions |
| Latency | Data Quality Issues | Model Drift | Model Evaluation Metrics |
| IO/Memory/Disk Utilization | Data / Feature Drift | Model Configuration | Prediction Drift |
| Uptime | Outliers | Model Version | |
| Auditability | | | |

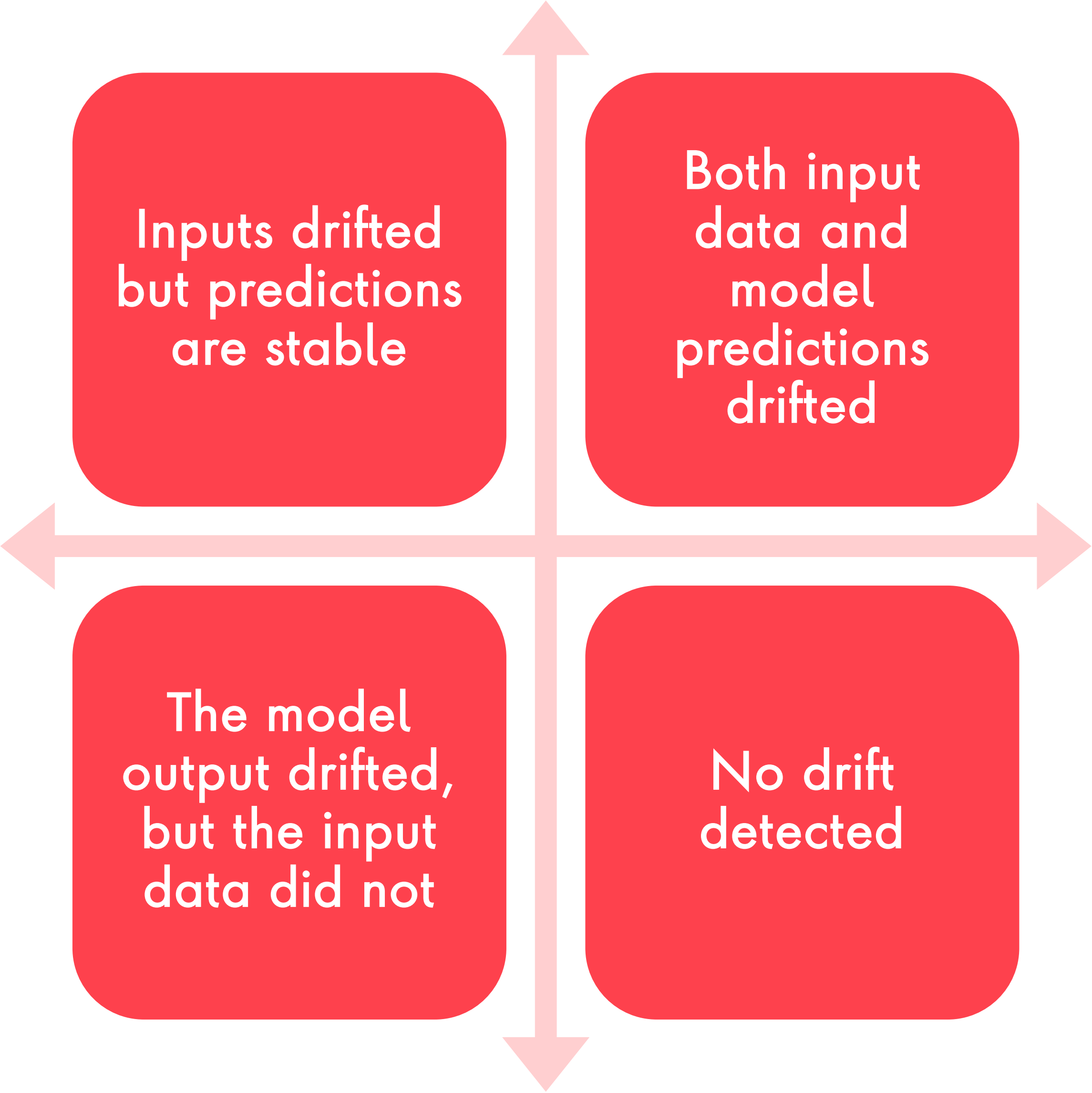
# How to Perform Model Monitoring?



Section 05

# Demo

# Drift scenarios and actions



# Inputs drifted but predictions are stable

## Positive Interpretation : the model is all set, and the drift does not matter

- Important features are stable
- The model is robust and can adapt to drift

### Actions

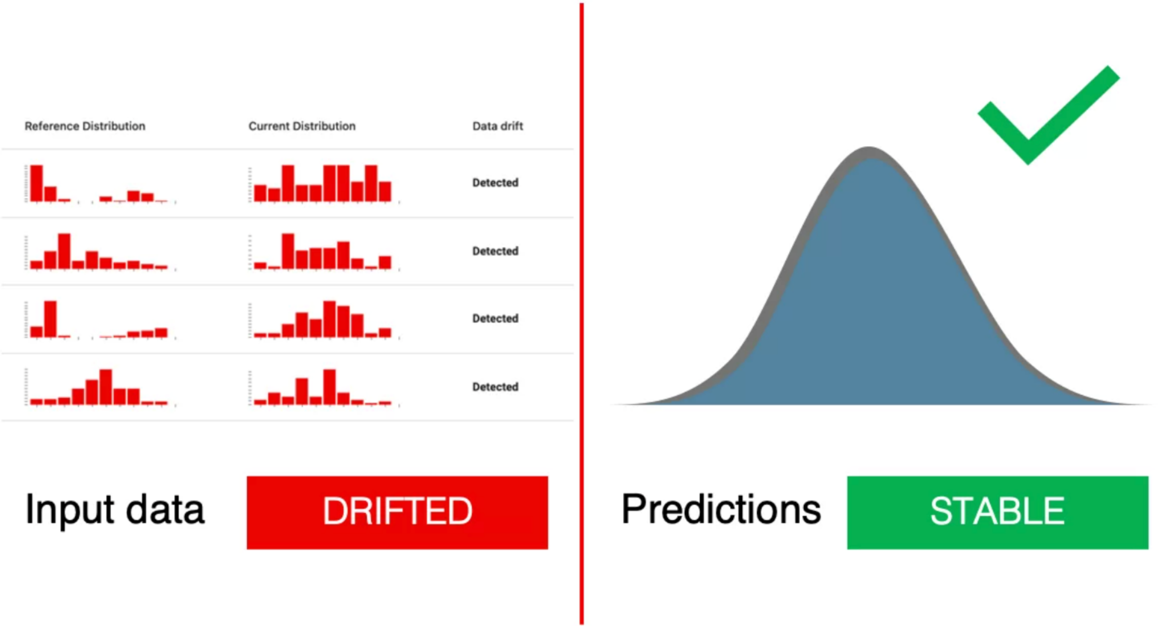
- No need to adjust/retrain model
- Adjust data drift detection approach
    - Limit drift detection to important features
    - Change size of comparison window
    - Pick less sensitive statistical test

## Negative Interpretation : the model is unreasonable!

- Important features changed
- The model should have reacted, but it did not

### Actions

- Retrain model/Rebuild model



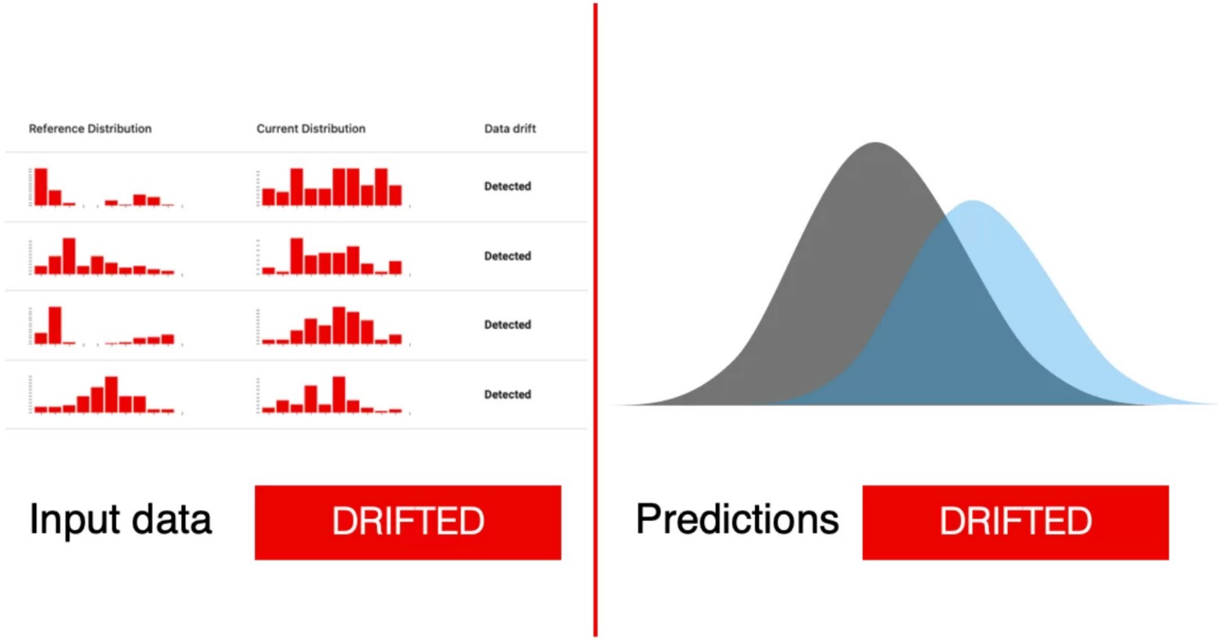
# Both input data and model predictions drifted

## Positive Interpretation : model handles drift well!

- Important features changed
- The model is robust and reacts well

### Actions

- No need to intervene
- If changes continue accumulating, might need to calibrate or rebuild the model




## Negative Interpretation : the model is unreasonable!

- Important features changed
- The model behaviour is unreasonable

### Actions

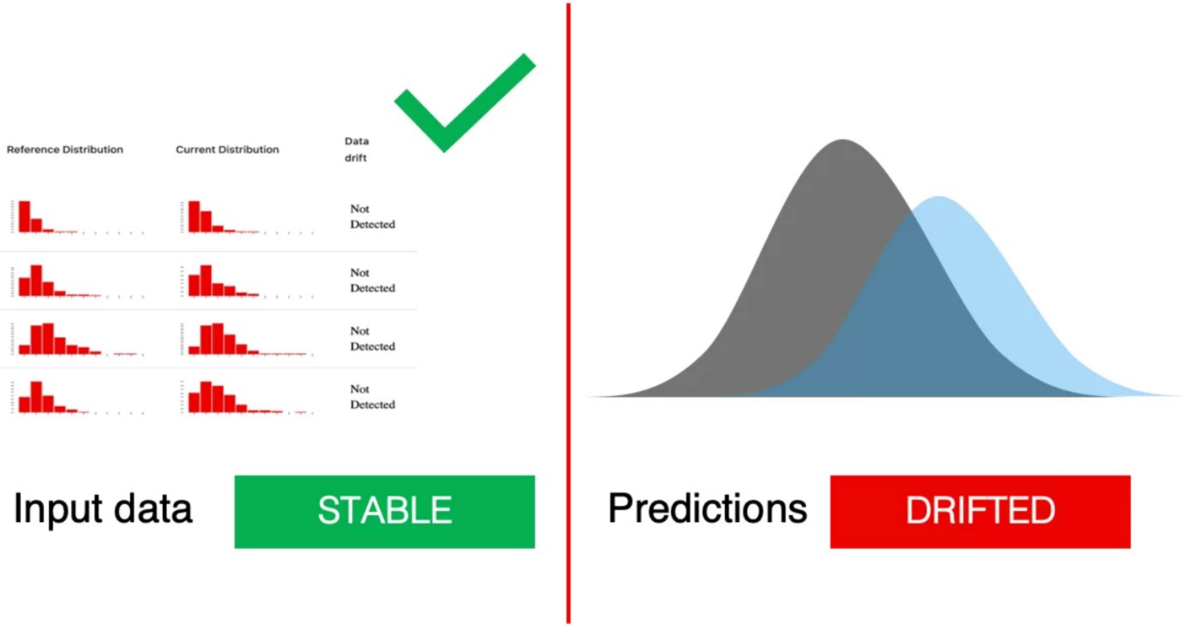
- Start from the investigation of causes
- Solve the data quality issues
- Retrain or rebuild the model

# The model output drifted, but the input data did not

**Output drift is always a solid signal to dig deeper**

- **Signal of an error**
  - Data quality issues
  - Bug in the code that performs drift test

- **Signal to review sensitivity of your drift test.**
  Examples,
  - Input drift detector is set to react only for major shifts to avoid alert fatigue. It might not trigger with lot of small changes in each feature. But it affects the performance of the model
  - Output drift detector is overly sensitive and react even to minor variation.



# No drift detected

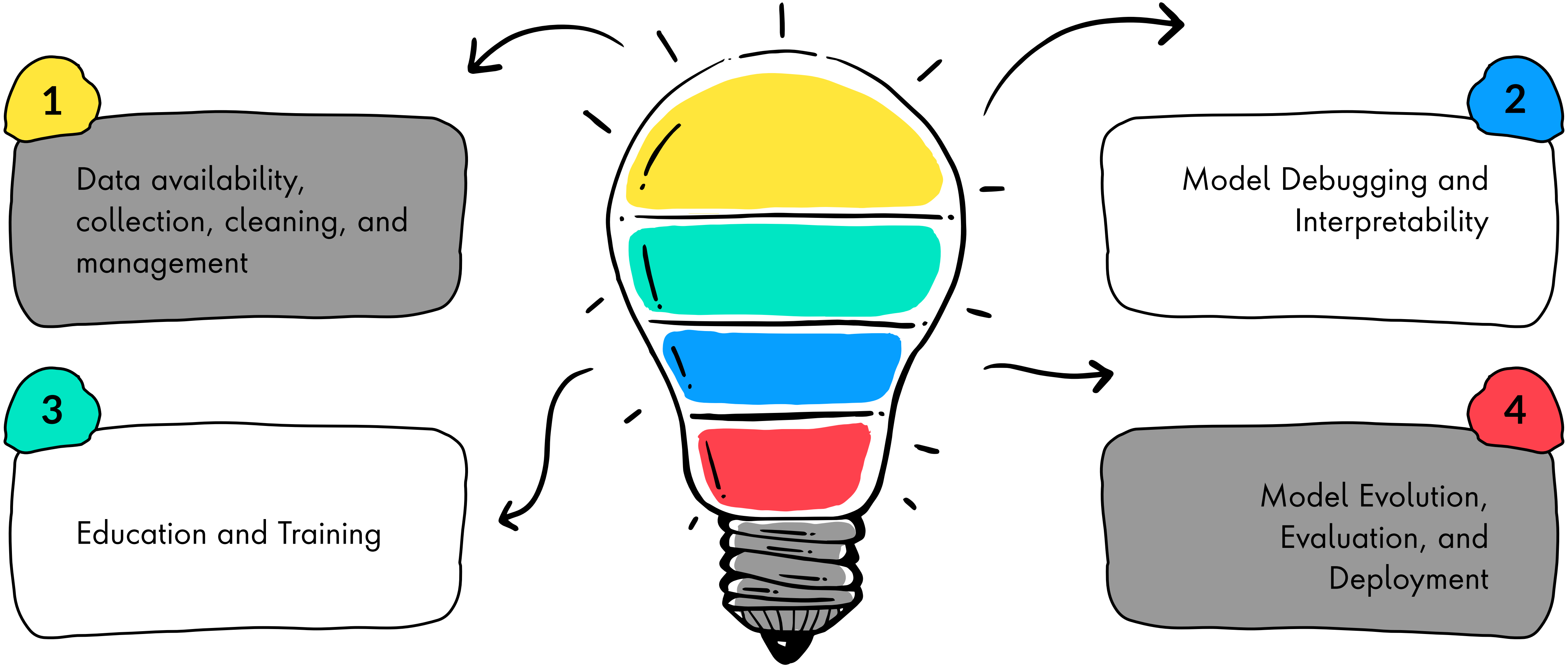
**Positive Interpretation** : Awesome! Let's grab a coffee.

**Negative Interpretation** : Did the drift detection job even work?

Section 06

# Best practices

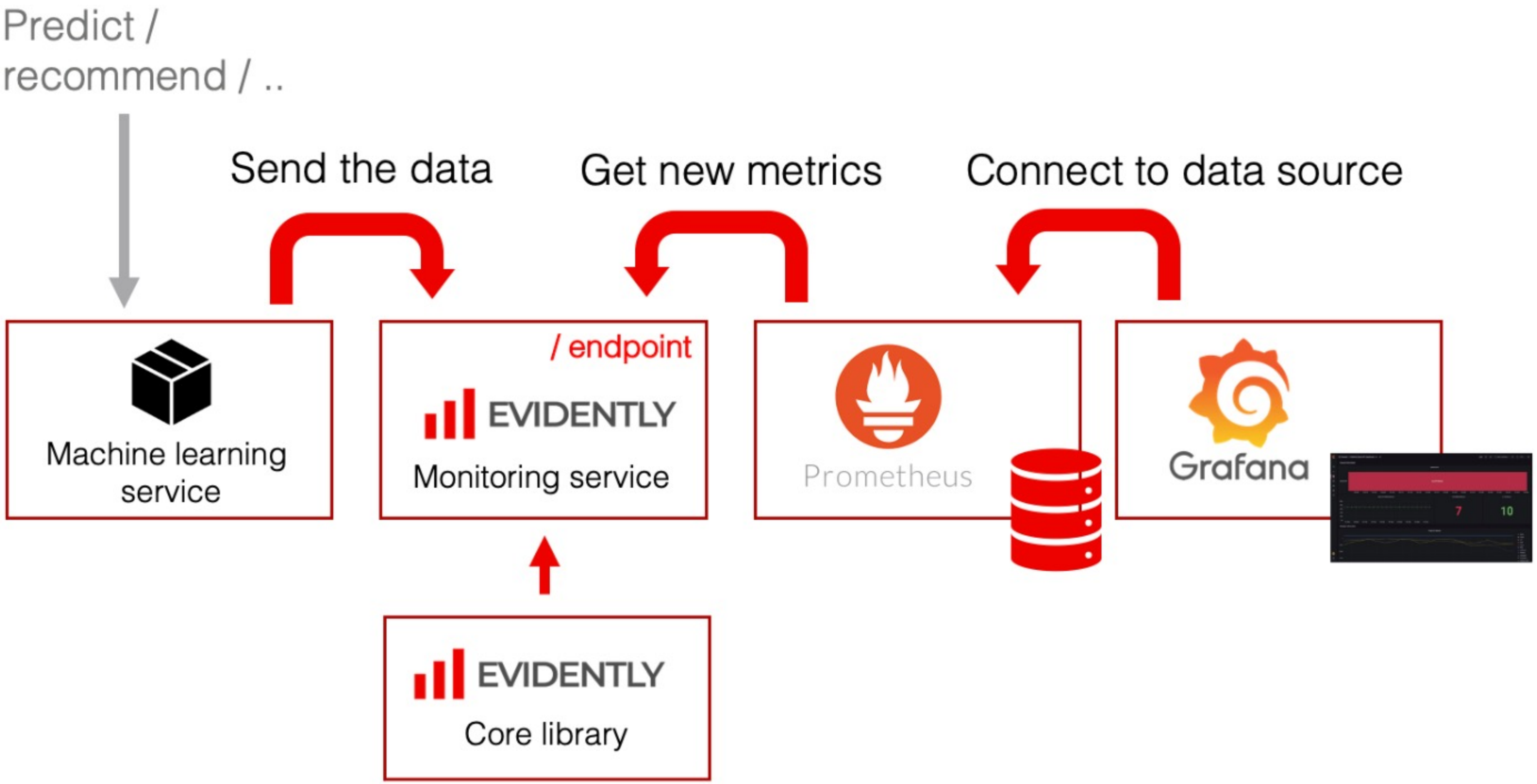
# Best Practices with Machine Learning



Section 07

# Other Tools & Tech for ML Monitoring

# Real-time ML Monitoring



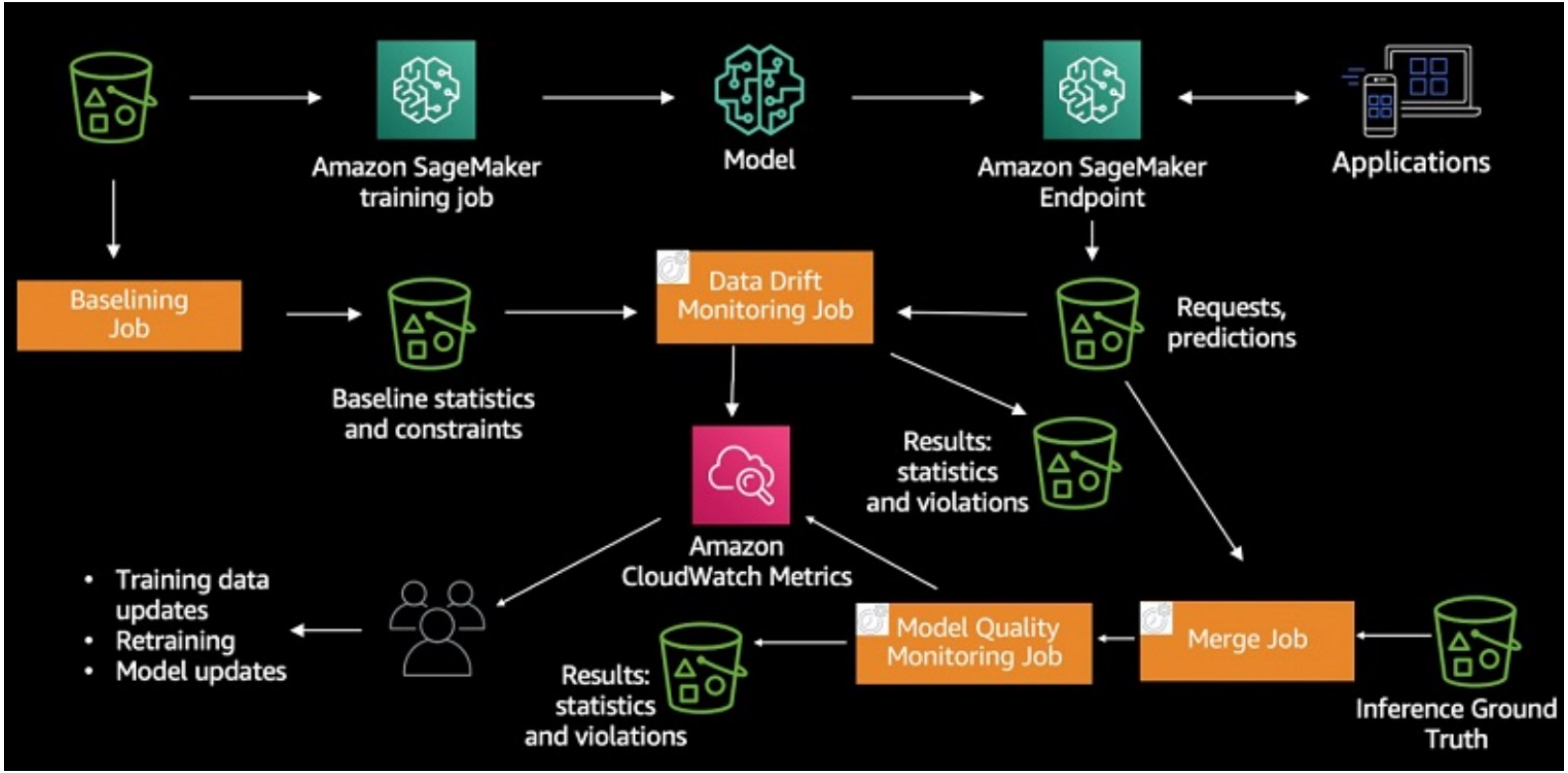
# Batch ML Monitoring

EVIDENTLY AI

+

Apache Airflow

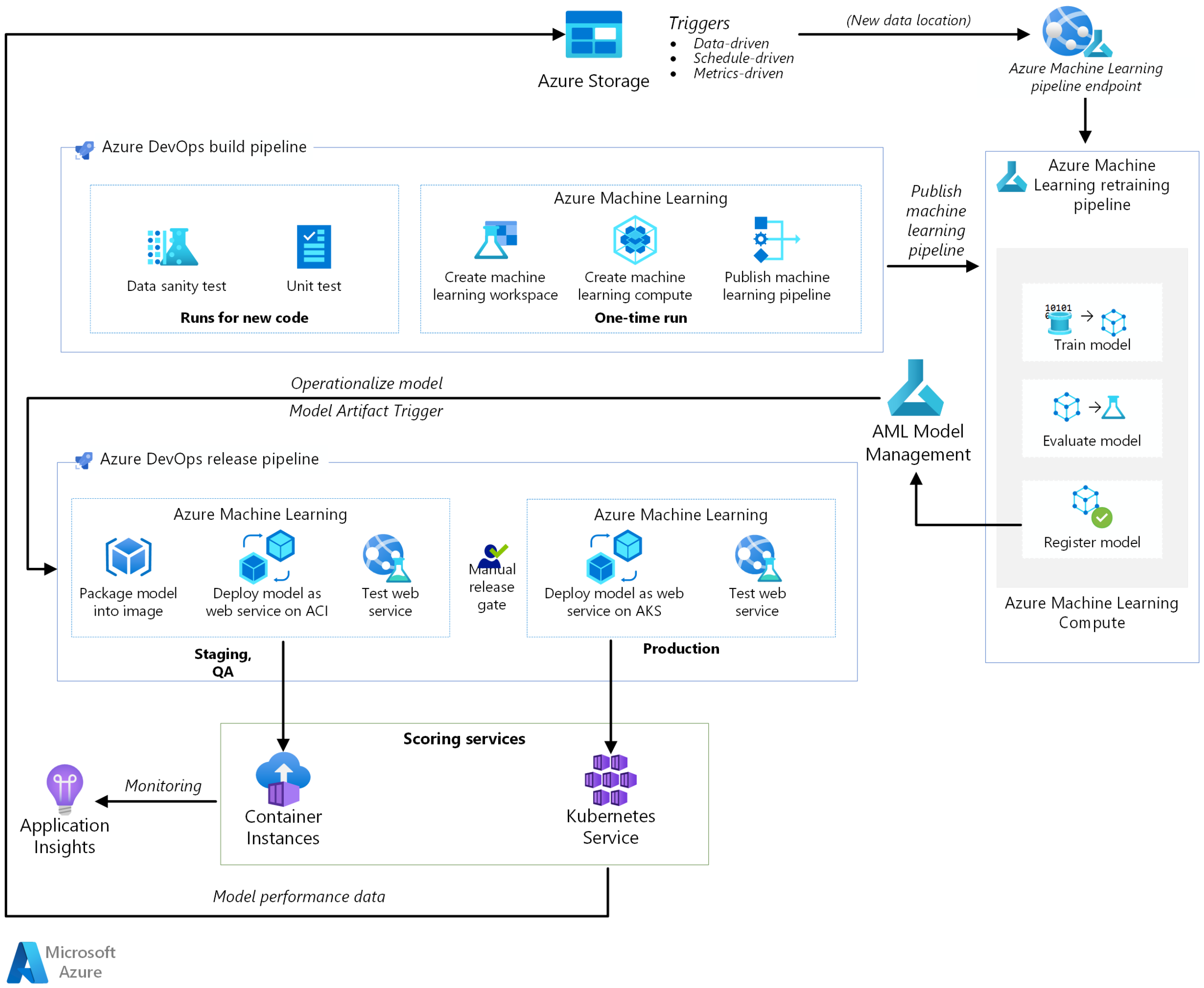
# ML Monitoring - AWS






Source: https://aws.amazon.com/blogs/machine-learning/monitoring-in-production-ml-models-at-large-scale-using-amazon-sagemaker-model-monitor/

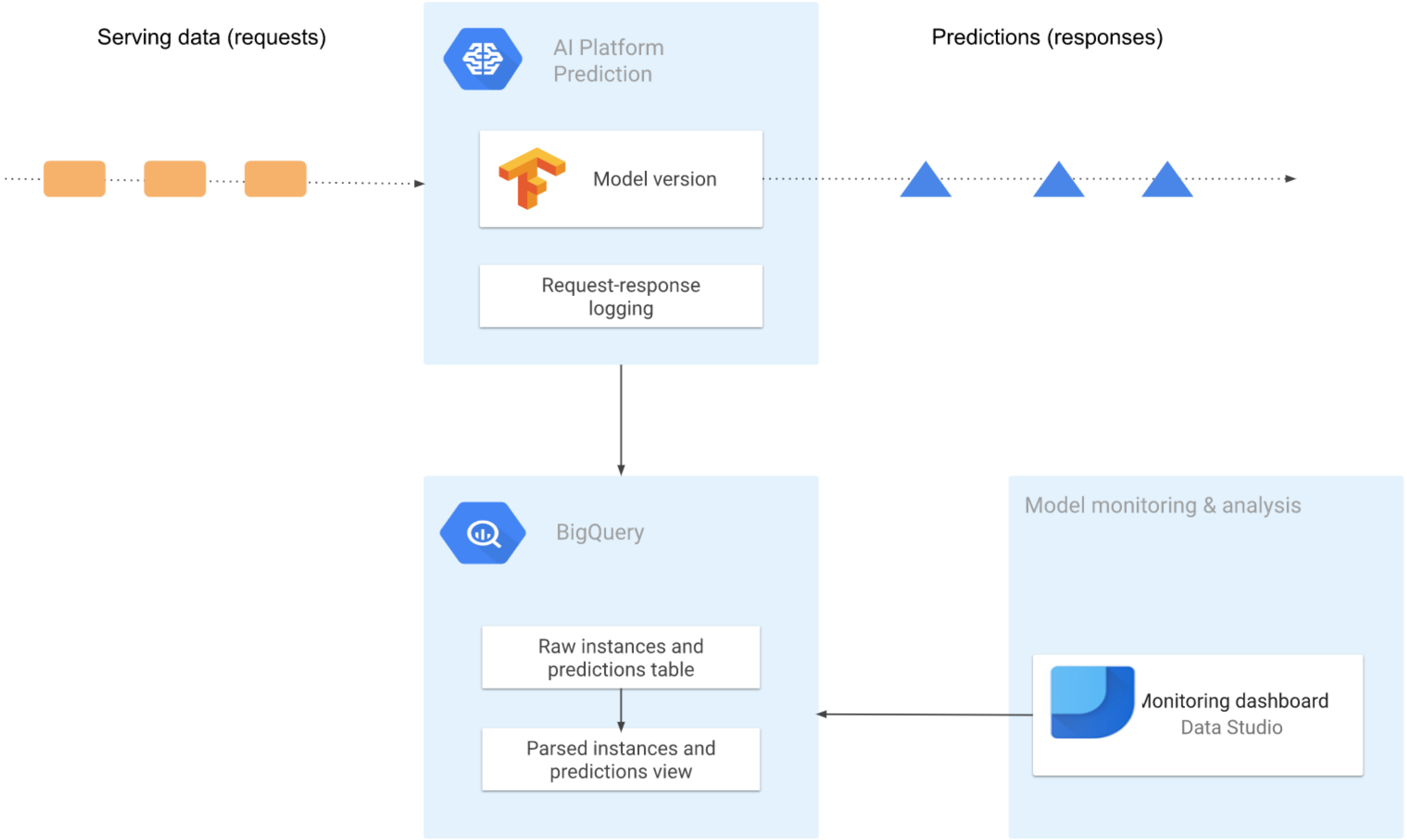
# ML Monitoring – MS Azure






Source: https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/ai/mlops-python

# ML Monitoring – GCP






Source: https://cloud.google.com/architecture/ml-modeling-monitoring-logging-serving-requests-using-ai-platform-prediction?hl=en

## Paid ML Monitoring Tools

- neptune.ai
- fiddler
- arize
- censius
- WHYLABS
- Qualdo™

## Open-source ML Monitoring Tools

- Grafana
- Prometheus
- EVIDENTLY AI
- anodot/MLWatcher
- KServe
- CORE
- whylogs

thank you

# References


1. https://neptune.ai/blog/ml-model-monitoring-best-tools
2. https://github.com/evidentlyai/evidently
3. https://www.youtube.com/watch?v=eQoKK5KNGLY
4. https://deepchecks.com/how-to-monitor-ml-models-in-production/
5. https://towardsdatascience.com/technical-debt-in-machine-learning-8b0fae938657
6. https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide
7. https://arize.com/model-drift/
8. https://analyticsindiamag.com/concept-drift-vs-data-drift-in-machine-learning/
9. https://valohai.com/model-monitoring/
10. https://christophergs.com/machine%20learning/2020/03/14/how-to-monitor-machine-learning-models/#lifecycle
11. https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide
12. https://www.fiddler.ai/blog/drift-in-machine-learning-how-to-identify-issues-before-you-have-a-problem
13. https://storage.googleapis.com/pub-tools-public-publication-data/pdf/aad9f93b86b7addfea4c419b9100c6cdd26cacea.pdf
14. https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse-2019_Software_Engineering_for_Machine_Learning.pdf
15. https://www.aporia.com/machine-learning-model-monitoring-101/
16. https://evidentlyai.com/blog/machine-learning-monitoring-data-and-concept-drift
17. https://evidentlyai.com/blog/data-and-prediction-drift