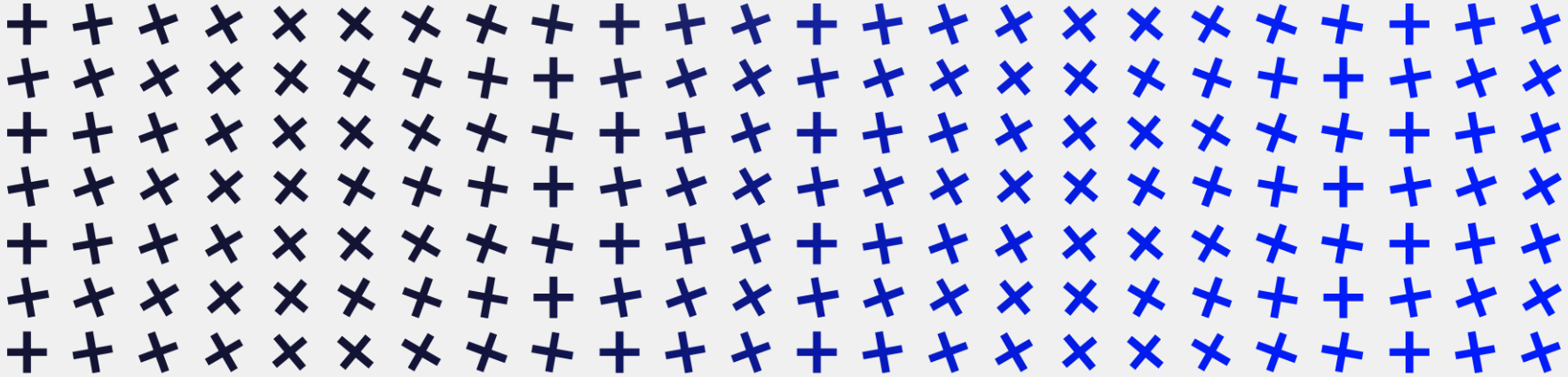


Designing DevOps to Support Data Science Teams

Dave Lemphers - CTO

easygo++

About Me

- CTO at Easygo - we make Kick, Stake and Carrot
- Co-founded and exited four AI/ML startups
- Former Principal Applied Data Scientist at Microsoft and AWS
- Author of four U.S. patents in machine learning and AI
- Advisor on technology strategy and digital transformation for public and private boards.

Introduction

- DevOps is critical for supporting machine learning workflows, particularly in production environments.
- This presentation focuses on the **model monitoring** aspect of DevOps and how it ensures the long-term success of ML models.
- We'll break down real-world implementations and practical steps to help data science teams keep models reliable in production.

The Importance of Model Monitoring in Production

+ Monitoring is the key differentiator between a successful and a failed ML deployment.
+ Once deployed, models face real-world variables: data drift, concept drift, and unexpected system performance changes.
+ Monitoring enables real-time visibility into how models are performing and gives teams a chance to react before issues become major failures.

Practical Design Step 1

Setting Up Monitoring Infrastructure

+ DevOps teams should first establish the right **infrastructure for monitoring.**
+ Use **Prometheus** for real-time data collection, paired with **Grafana** for visualization.
+ Ensure that monitoring captures both **model-specific metrics** (accuracy, precision, recall) and **infrastructure metrics** (latency, resource usage).

Practical Design Step 2

Defining Key Model Monitoring Metrics

+ Focus on **ML-specific metrics** that provide insight into model health: accuracy, precision, recall, and F1 score.
+ Monitor for **data drift** and **concept drift** by comparing real-time input data distributions with training data distributions.
+ Track **resource utilization** (CPU/GPU usage, memory, and inference latency) to understand how the model behaves under load.

Practical Design Step 3

Setting Thresholds and Automated Alerts

+ Establish **thresholds** for critical metrics (e.g., accuracy dropping below 85%, data drift exceeding 20%).
+ Use Prometheus to create **automated alerts** that notify teams when a threshold is crossed.
+ Alerts can trigger **automated responses**, such as model rollback or initiating retraining.

Practical Design Step 4

Monitoring for Data and Concept Drift

+ **Data drift:** Track how the incoming data distribution changes compared to training data.
+ **Concept drift:** Monitor if the relationship between input features and model predictions changes over time.
+ Set up **automated drift detection** that triggers alerts or retraining when drift is detected.

Practical Design Step 5

Feedback Looks for Continuous Improvement

+ Monitoring systems provide **feedback** to the data science team, allowing for model improvements.
+ Data drift and performance alerts can trigger **retraining pipelines** automatically.
+ Use this feedback to continuously improve models and prevent performance degradation.

Benefits to the Data Science Team

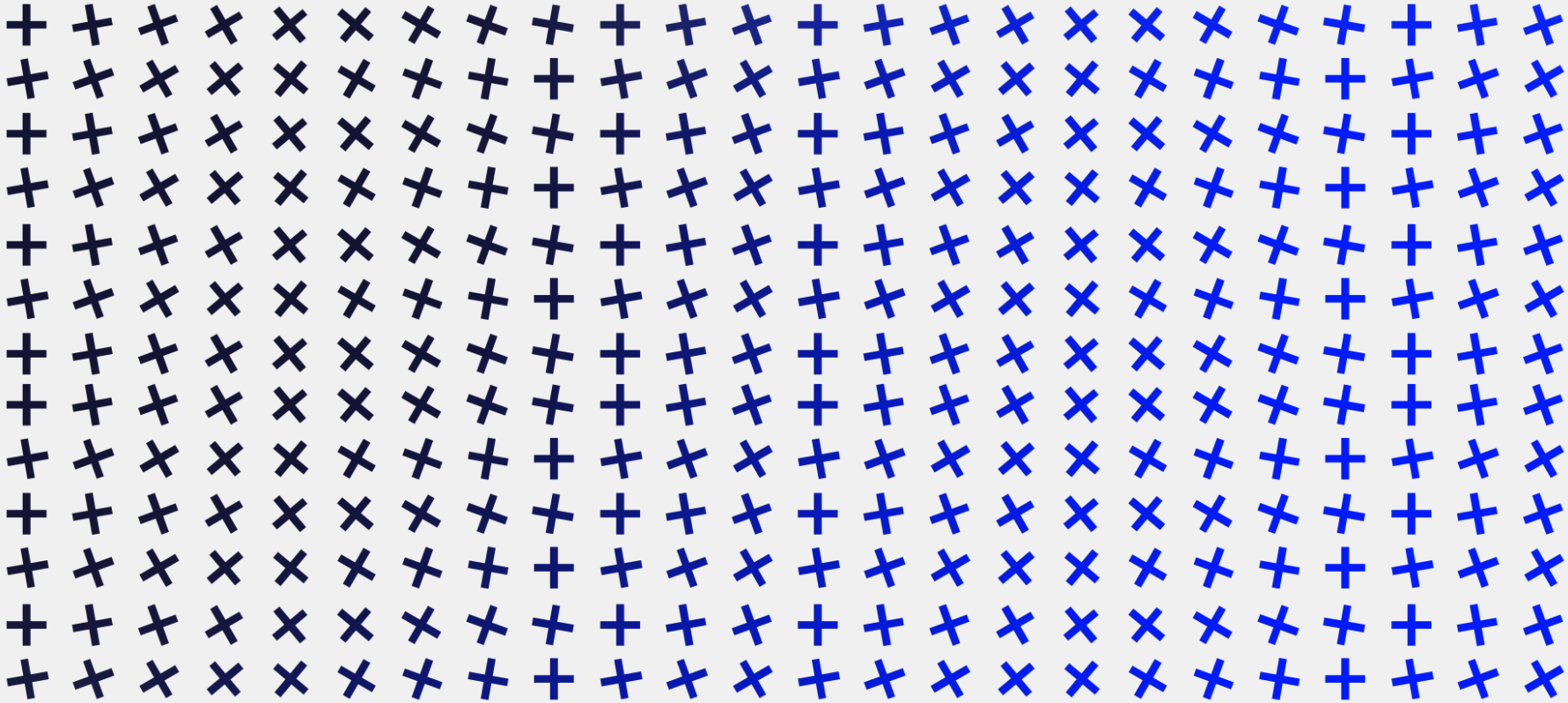
+ **Proactive issue detection:** Monitoring systems catch performance issues early, reducing downtime.
+ **Automated retraining:** Models stay up-to-date with minimal manual intervention, freeing data scientists to focus on innovation.
+ **Improved model reliability:** Continuous monitoring ensures models remain performant in dynamic environments.

Conclusion

+ Model monitoring is essential for ensuring the long-term success of ML models in production.
+ With the right infrastructure, metrics, and feedback loops, DevOps teams can enable data science teams to deliver reliable models at scale.
+ The future lies in further automation, enabling more seamless integrations between monitoring, feedback, and model retraining.

engineering++ the future

Join a select group of elite engineers driving innovation and excellence at Easygo Engineering. **Shape the future with us.**



easygo++