# Мониторинг ML-систем

**Ирина Степановна Трубчик**

https://t.me/+PsC-JDrwrvsxNmVi

Лекция 9

# Цели занятия

1. Зачем мониторить ML
2. Где нужен мониторинг в ЖЦ  ML-модели
3. Что именно нужно мониторить
4. Примеры инструментов мониторинга

*приведите реальные примеры “тихого падения” ML-моделей.*

# Жизненный цикл ML-модели: где нужен мониторинг


Мониторинг – единственная стадия, которая работает 24/7 после релиза

*Каждая ML-система требует постоянной обратной связи по качеству модели, состоянию данных, инфраструктуре и метрикам бизнеса.*

# Что мониторить?
# Основные типы метрик

- Инфраструктурные метрики (доступность, latency, CPU/mem)
- Метрики данных (drift, полный объем, пропуски)
- ML-метрики (accuracy, recall, F1, AUC)
- Бизнес-метрики (конверсия, прогнозирование дохода (revenue))

# Инфраструктурный мониторинг (SRE)

- **Инструменты: Prometheus, Grafana, Alertmanager, ELK/EFK Stack**
- **Мониторинг CPU, памяти, числа запросов, ошибок, времени ответа**
- **Алерты и автоматизация действий при сбоях**

SRE (Site Reliability Engineering) — это практика автоматизации задач по обеспечению надежности и доступности ИТ-инфраструктуры и приложений.

SRE-инженеры используют программные инструменты для мониторинга, управления, диагностики и отладки систем, фокусируясь на «четырех золотых сигналах»: **задержке, трафике, ошибках и насыщении**, чтобы оптимизировать работу сервисов и минимизировать сбои

# Мониторинг данных и фичей

- **Почему нужны, примеры data drift (drift категорий, средних, seasonality)**
- **Метрики: распределения, пропуски, уникальные значения, корреляции**
- **Пример: внезапно в поле “region” появляются новые значения, которых не было при обучении**

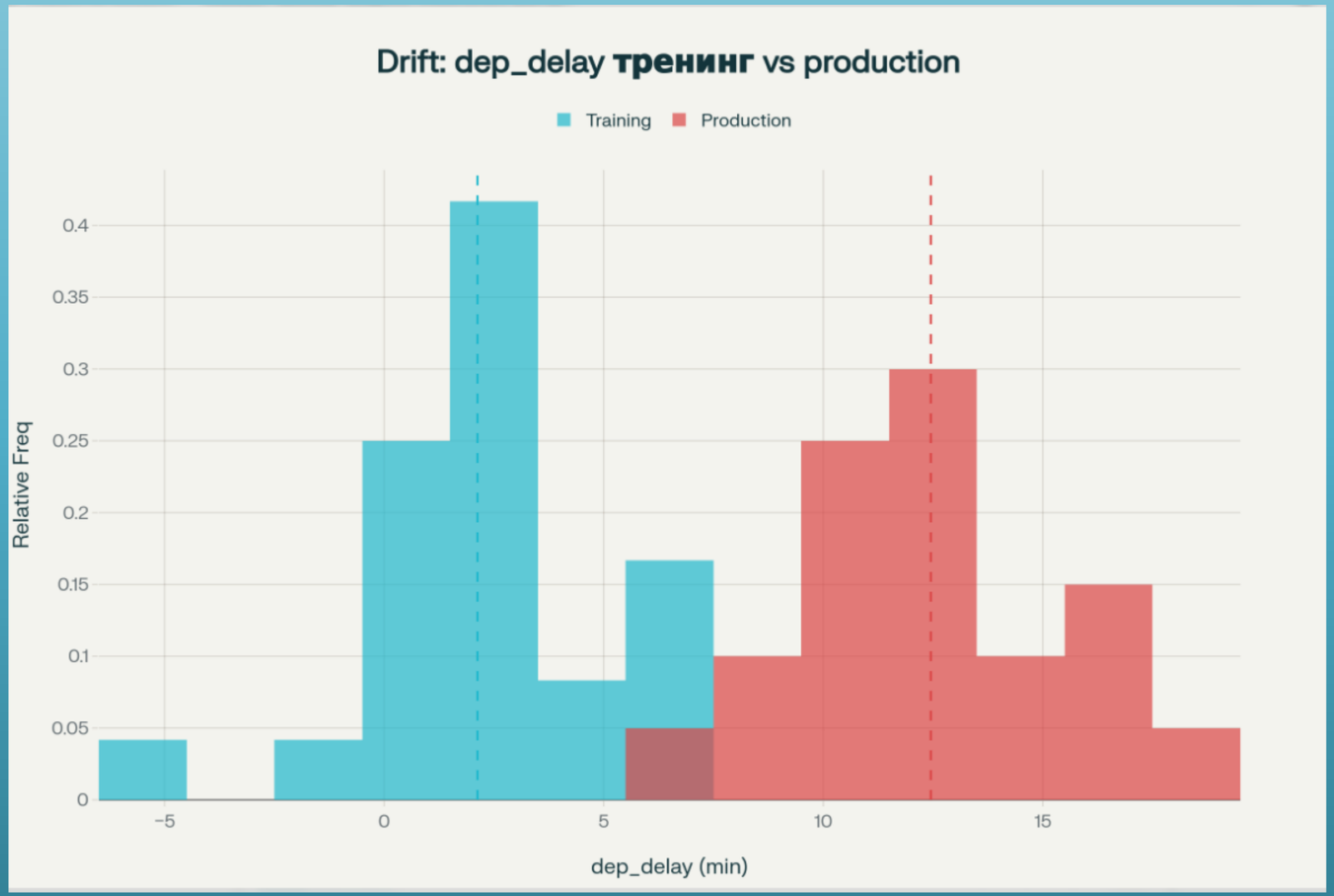
*Контроль качества поступающих данных — ключевой фактор при работе production ML. Повышенное число пропусков или неожиданные категории должны вызвать алерт.*

# Детектирование drift и деградации

- **Визуализация drifts: histograms до/после, KL divergence, PSI, Wasserstein, t-test**
- **Готовые библиотеки: Evidently, WhyLabs, Alibi-detect, River**
- **Автоматизация сверки: отчет nightly, telegram/email-алерты**
- **Пример: график изменения распределения ключевых признаков**

*Automated tools помогают выявлять и сигналить о значимых изменениях статистики данных (drift) или качества модели.*

# Сдвиг распределения признака dep_delay: тренировочные vs production данные



# Мониторинг ML-метрик (качества)

- **Как часто можно оценивать метрики: если есть delayed ground truth (например, click в рекламе)**
- **Метрики классификации, регрессии, вероятностные**
- **Система алертов при падении accuracy ниже порога**

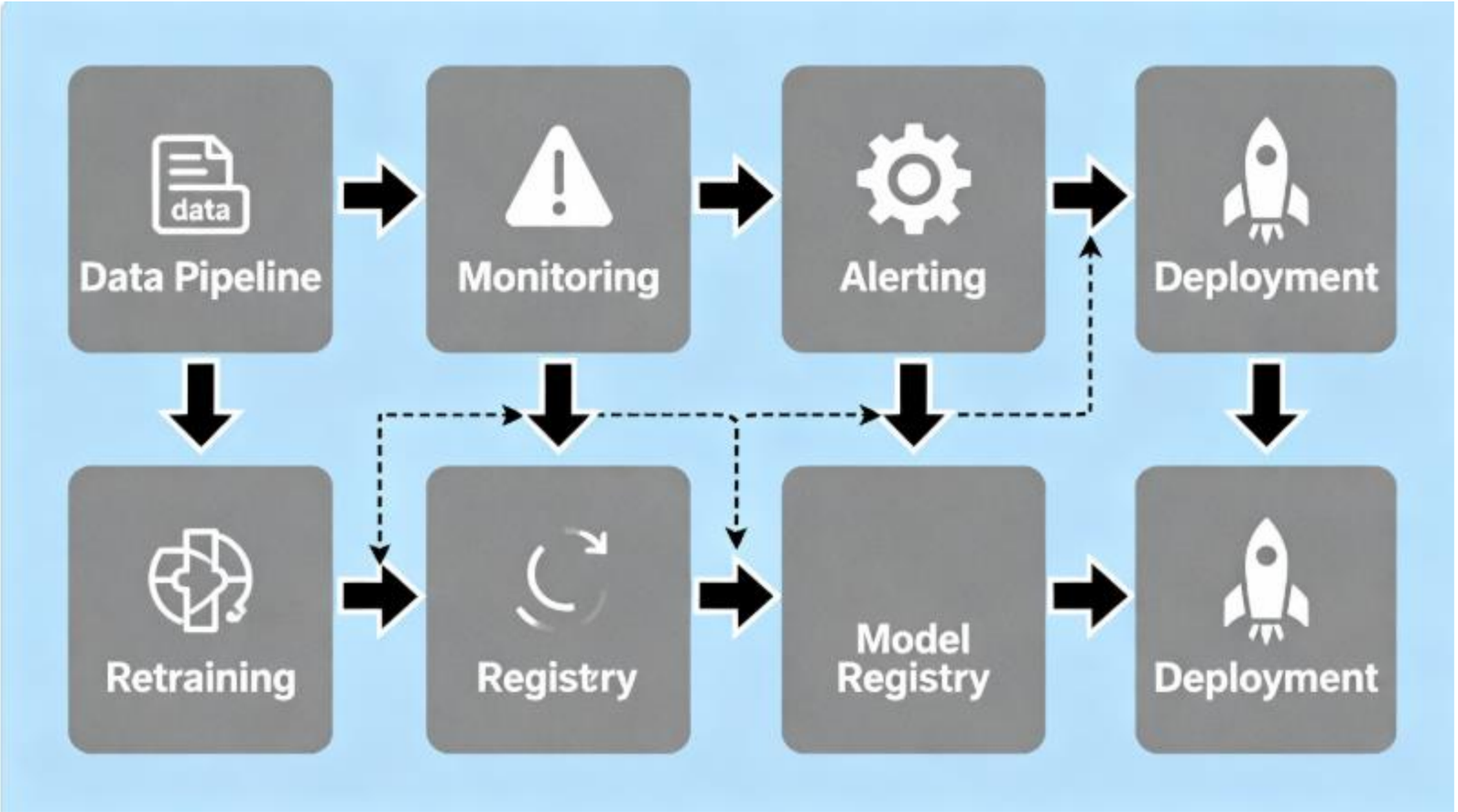
*Даже с задержкой истинных ответов (например, отклик пользователя через неделю) нужно ежедневно считать метрики и отслеживать деградацию.*

# Бизнес-метрики и A/B тесты в мониторинге

- **конверсия, доход, показатель отказов, NPS (индекс лояльности клиентов)**
- **В чем разница между “качество модели” и “бизнес-эффект”**
- **Автоматические сравнения бизнес-метрик между версией A (prod) и B (обновления)**

*Успешная модель должна улучшать бизнес-показатели.*
*Мониторинг включает сравнение ключевых метрик между версиями модели.*

# Архитектура мониторинга ML-систем




- Data pipeline → Monitoring → Alerting → Retraining → Registry
- Взаимосвязь мониторинга с CI/CD и auto-retrain

# Примеры инструментов мониторинга

- **Prometheus/Grafana (SRE)**
- **ELK/EFK Stack (инфраструктурные и пользовательские логи)**
- **Evidently (drift, distribution, ML performance)**
- **MLflow (model registry + трекинг метрик)**
- **WhyLabs, Alibi-detect (open-source и коммерческие)**

*Стек мониторинга выбирается под задачу: простого SRE достаточно для infrastructure, для данных и ML-метрик нужны другие инструменты.*

# Практика: мониторинг модели через Evidently

- ✓ *Демонстрация простого пайплайна: запись prediction в лог, генерация отчетов по drifts*
- ✓ *Код примера: используем Evidently для анализа data drift каждую ночь*
- ✓ *Визуализация: автоматический email/slack отчет*

# Практика: Prometheus для latency и запросов

- ✓ *Написание метрик (FastAPI/Flask + prometheus_client)*
- ✓ *Графики в Grafana: latency, error rate, throughput*
- ✓ *Настройка алерта по threshold*

# Чек-лист best practices мониторинга

- Записаны все внешние и внутренние алерты
- Есть метрики по доступности, latency, дрейфу данных, ключевым ML-метрикам и бизнес-эффекту
- Корректно настроены SLA/SLO
- Есть план реагирования на алерты
- Мониторинг регулярно тестируется (“test alert”)

*Мониторинг — живой процесс: алерты и метрики нужно регулярно дорабатывать и совершенствовать!*

# Ключевые вопросы для самопроверки:

1. Зачем мониторить ML-системы, если accuracy высокой была на валидации?
2. Какие бывают типы drift, и как их выявлять?
3. Почему важно отделять инфраструктурные алерты от алертов на ML-метрики?
4. Как организовать автоматическое переобучение при деградации?
5. Какие инструменты подходят для мониторинга данных vs. инфраструктурных логов?
6. Как переводить результат мониторинга в бизнес-решения?

# Материалы и ссылки

Evidently: https://evidentlyai.com/

Prometheus: https://prometheus.io/docs/introduction/overview/

Grafana: https://grafana.com/docs/grafana/latest/

WhyLabs: https://whylabs.ai/

ELK Stack: https://www.elastic.co/what-is/elk-stack

“Machine Learning Operations” (O’Reilly, Mark Treveil, Alok Shukla)

Coursera: “Data Science in Production: Monitoring, Testing, and Logging”

YouTube: “Monitoring for ML in Production” (DataTalksClub, 2023)

# Вопросы

Телеграм https://t.me/+PsC-JDrwrvsxNmVi

**СКИФ (https://do.skif.donstu.ru/course/view.php?id=7508)**