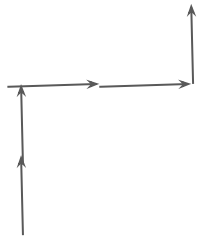
Мониторинг скоринговых моделей в продакшене

DevFest 2025
Alexey Turov, Data Science Expert,
Beeline Kyrgyzstan

Проверим кто тут data scientist

pred	target
0.55	0
0.8	1
0.3	1
0.95	1
0.7	0

pred	target
0.95	1
0.8	1
0.7	0
0.55	0
0.3	1



Кто может сказать какой AUC-ROC для данного результата

AUC за 30 секунд

- Отсортируйте по score ↓ → метки: **[1, 1, 0, 0, 1]**.
- **m=3** (единицы), **n=2** (нули) → решетка **3×2**, старт в (0,0).
- За **1** шаг **вверх** (TPR ↑), за **0** шаг **вправо** (FPR ↑).
- Площадь под маршрутом = **4 клетки из 6** = **AUC = 4/6 = 0.67**.
- Интуиция: вправо — **ложноположительные** по нулям (FPR), вверх — **верно пойманные** позитивы (TPR).

Source: xxxxx

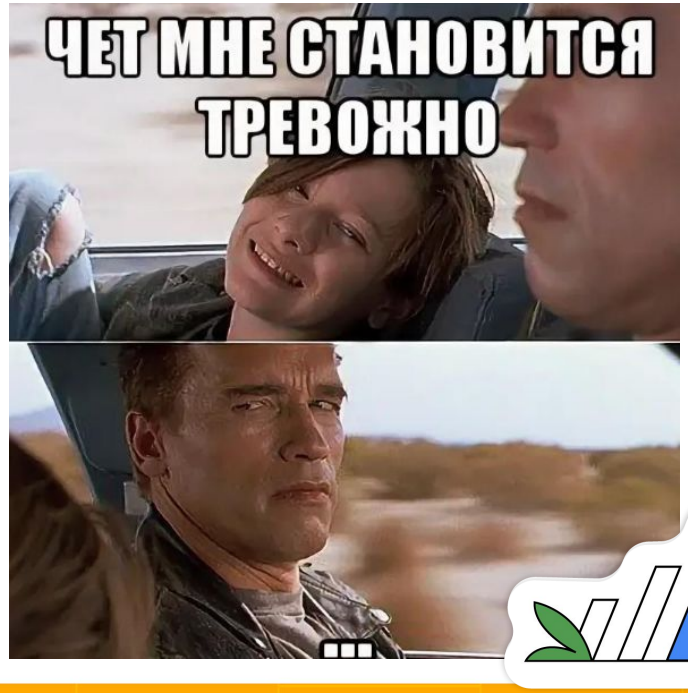

```
code_slides.txt

// A couple more notes on code slides:
//  *When possible, use the “move in” (bottom to top) transition
//  *Use the Google Sans Mono font
//  *Set line spacing to 1.2
//  *Don’t use font sizes below 25
```

Путь модели в бизнесе

Как обычно «видят» путь модели.

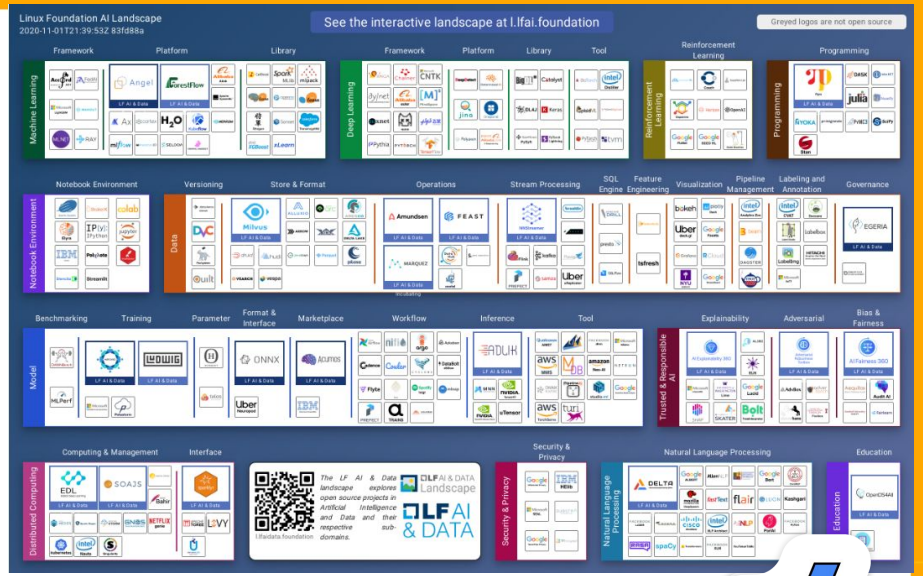
Бизнес: «Нужен ML!»
DS: «Сделали крутую модель»
Prod: «Катим!»
А что дальше?



Source: xxxxx

Технологии и проблемы

- Сегодня существует огромный стек DS/ML-технологий
- Ошибка выбора → финансовые и временные потери
- Интернет перегружен информацией → легко запутаться
- В этом докладе: на чём мы остановились и как мониторим модели в проде

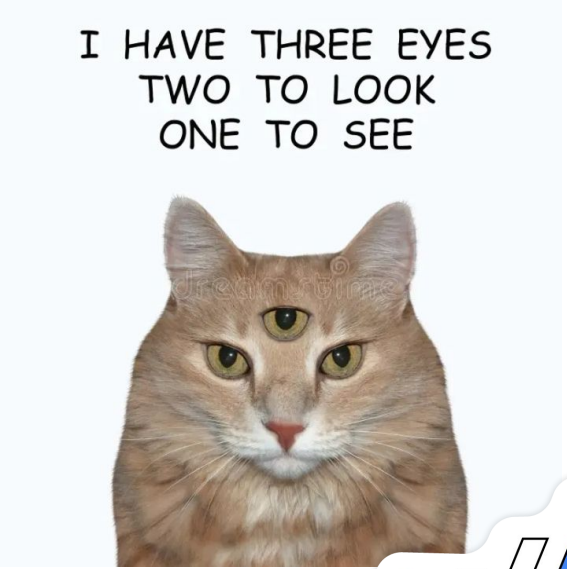


https://landscape.lfai.foundation/

Зачем мониторинг (3 аудитории)

1. **Бизнес** — видит FPD/SPD/TPD, выдачи, заявки, динамики, понимает, «живёт» ли модель и где риски.
2. **Разработка** — следит за сервисом: ошибки, SLA, задержки, входящий поток.
3. **Data Science** — замечает дрейф данных/скор баллов и деградацию качества

«Важно: это не «один дашборд для всех». Технические метрики ≠ бизнес-метрики. Мы разделяем зоны ответственности и каналы алёртов»..



Source: xxxxx

Бизнес мониторинг

Стек технологий:

1. Tableau, Power BI, Grafana
2. Python, Sql

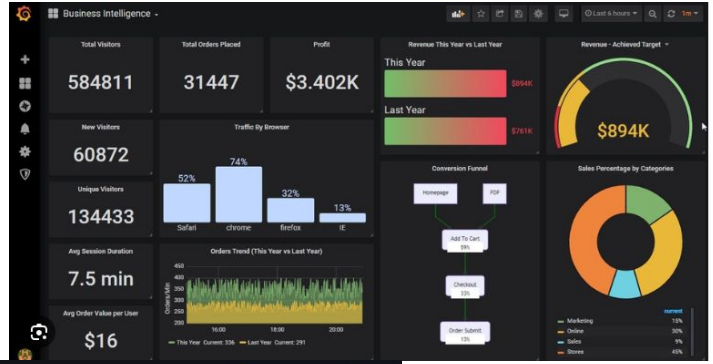
Что отслеживаем:

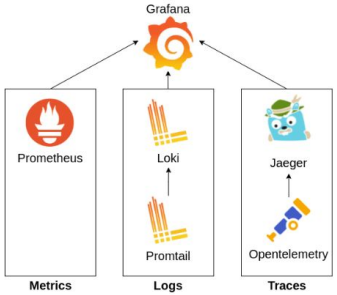
- Бизнес-прокси (FPD, TPD, Доходы, Заявки, выдачи)
- Порог/Threshold(влияние порога на Approval rate)
- Кто подает заявки(пол, возраст и т.д.)
- Алерты(выход за граничные показатели)

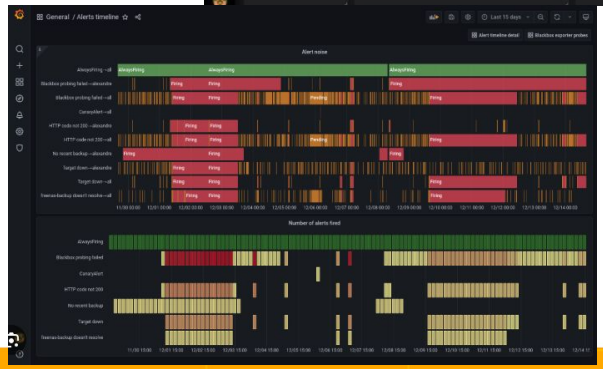
Source: xxxxx

Тех Мониторинг

«Мы выбрали классический стек. Приложение инструментируем OTel: метрики/логи/трейсы летят в Collector. Collector отдаёт метрики Prometheus'у, логи — в Loki, трейсы — в Jaeger. Grafana всё это объединяет в один экран. Alertmanager шлёт алерты в Telegram







https://rafed.github.io/devra/posts/cloud/django-mlt-observability-with-opentelemetry/

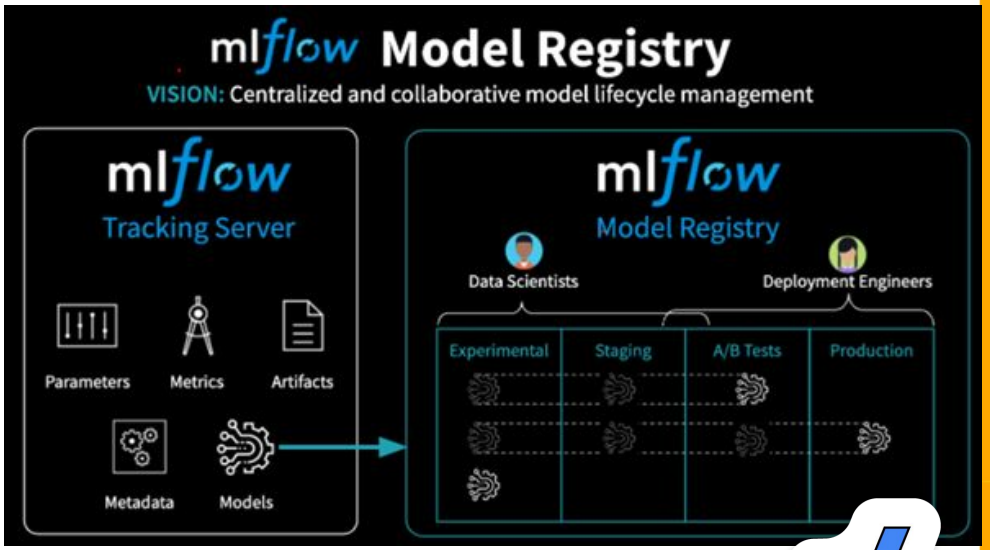
Data science monitoring

Стек техлогий и практик:

1. **MLflow / ClearML / Kubeflow** и другие — трекинг экспериментов и Model Registry
   параметры, метрики, артефакты; стадии: Staging → Production
2. **Airflow** — оркестрация и автозапуски
   ETL/фичи/скоring/ретрейн; SLA/ретраи/backfill
3. **Feature Store (offline/online)** — point-in-time корректность
   ключи сущностей, версионирование фич, backfill на истории, online-выдача на inference
4. **CI/CD + Docker** — поставка моделей
5. **Data & Score Drift** — стабильность входа/выхода
   числовые: KS/PSI, категориальные: χ^2; score-дистрибуции, approve-rate
6. **Статпорога** (быстрые ориентиры)
   PSI: <0.1 ok, 0.1–0.25 предупреждение, >0.25 — дрейф; p-value χ^2/KS < 0.01 — тревога
7. **CBPE** — качество без таргета сейчас
   онлайн-оценка AUC/Gini по уверенности модели и историческим калибровкам; коридоры и алёрты

MLflow / ClearML / Kubeflow

- Что это: трекинг экспериментов и Model Registry
- мы работаем не с кодом, а с экспериментами
- Зачем: хранить параметры, метрики, артефакты
- Ключевая идея: модель ≠ файл .pkl, а полный эксперимент (train code, dataset, метрики, артефакты)
- Стадии: Staging → Production



Airflow

- Что это: оркестрация процессов
- Зачем: автоматизировать ETL, retrain, скоring
- Примеры: SLA мониторинг, backfill историй, ретраи при ошибках

«Airflow — это «дирижёр» процессов. Можно настроить пайплайн: подготовка данных → обучение → валидация → деплой. И всё это с графиком и алертами.»

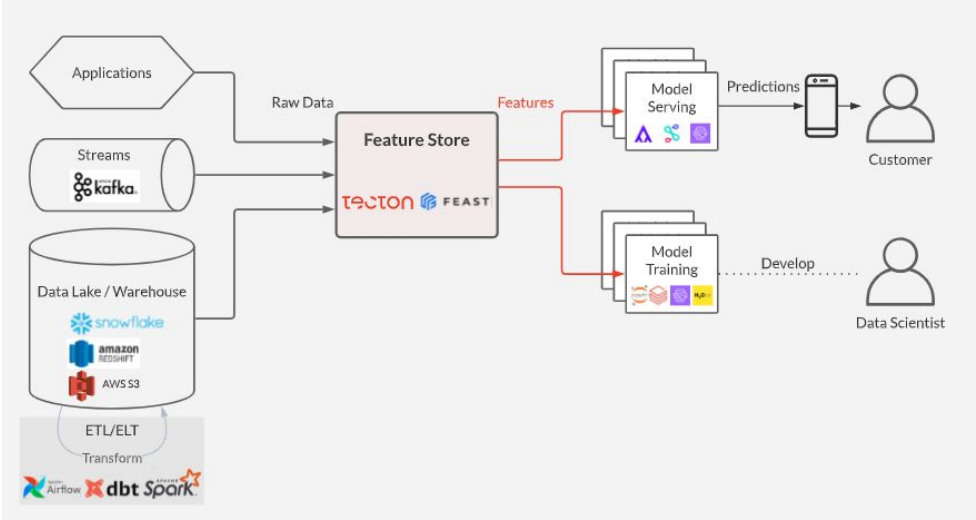
https://airflow.apache.org/docs

Feature Store

Что это: единое хранилище фич (offline/online)
Зачем: консистентность между train и inference
Ключевые моменты:

- point-in-time корректность (без data leakage)
- версионирование фич
- offline для обучения, online — для продакшена

«**Feature Store** — это библиотека признаков. Берём одинаковые фичи и в train, и в проде. Это сильно снижает риск data leakage.»

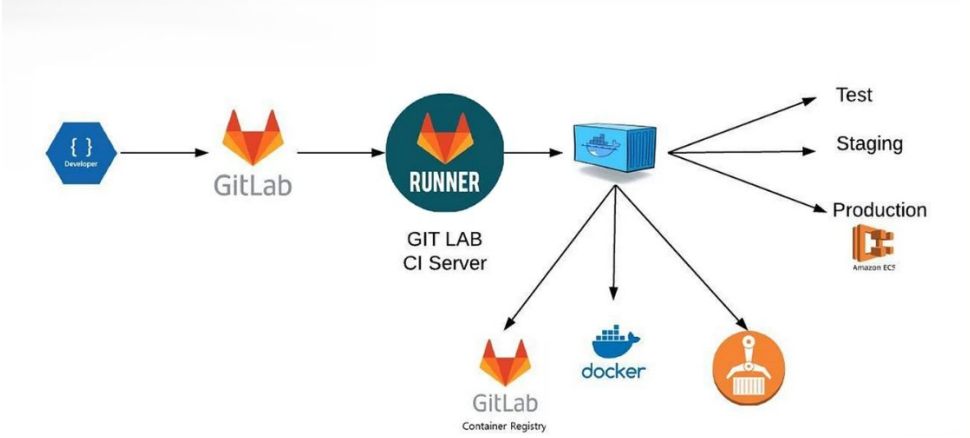


https://airflow.apache.org/docs

CI/CD + Docker

- **Что это:** поставка моделей
- **Зачем:** деплой быстрый и безопасный
- **Примеры:** unit/data/contract-тесты, версии образов, blue-green/canary деплой

«CI/CD мы используем для моделей так же, как для кода. Docker с тегами model_version позволяет понять, какая именно модель крутится в проде.»



Data & Score Drift

Data drift: входные признаки меняются (доход, возраст, канал)
Score drift: распределение скорбаллов уходит от исторического
Методы:

- числовые: KS/PSI
- категориальные: χ^2
- drift detected, drift concept

Сигналы: approve-rate, распределения около порога

В скоринге drift — это почти норма: пришли новые сегменты, сезонность. Важно не пропустить, а правильно интерпретировать.

code_slides.txt

```
protected void onTryUpdate(int reason) throws RetryException {
    // Do some awesome stuff
    int foo = 15;
    publishArtwork(new Artwork.Builder()
            .title(photo.name)
            .imageUri(Uri.parse(photo.image_url))
            .viewIntent(new Intent(Intent.ACTION_VIEW,
                    Uri.parse("http://500px.com/photo/" + photo.id)))
            .build());
    scheduleUpdate(System.currentTimeMillis() + ROTATE_TIME_MILLIS);
}
```

Copy / paste your code using this tool for formatting using the 'Dark Alternate' theme

Use this template for code snippets longer than 6 lines

Статпороги (ориентиры)

PSI:

- <0.1 — ок
- 0.1–0.25 — предупреждение
- 0.25 — критический дрейф

χ^2/KS: p-value <0.01 ⟶ тревога

https://airflow.apache.org/docs

CBPE (Confidence-Based Performance Estimation)

- **Что это:** оценка качества без свежего таргета
- **Зачем**: факты дефолтов приходят через 3–6 месяцев
- **Как работает:** сравниваем распределение вероятностей текущей модели с историей
- **Что даёт:** ранние алерты по AUC/Gini, доверительные интервалы

CBPE не заменяет таргет, но позволяет понять: если модель ушла в деградацию — сигнал сработает сегодня, а не через полгода

https://airflow.apache.org/docs

Выводы по секции

- Работайте с экспериментами, а не с кодом (MLflow, Kubeflow).
- Автоматизируйте (Airflow + CI/CD).
- Следите за drift и калибровкой.
- CBPE спасает, когда нет таргета. Также можно посмотреть корреляцию, например 5 месяца и 12

https://airflow.apache.org/docs

code_slides.txt

```
protected void onTryUpdate(int reason) throws RetryException {
    // Do some awesome stuff
    int foo = 15;
    publishArtwork(new Artwork.Builder()
            .title(photo.name)
            .imageUri(Uri.parse(photo.image_url))
            .viewIntent(new Intent(Intent.ACTION_VIEW,
                    Uri.parse("http://500px.com/photo/" + photo.id)))
            .build());
    scheduleUpdate(System.currentTimeMillis() + ROTATE_TIME_MILLIS);
}
```