

Построение обратной петли (feedback loop)

Частичная перепроверка

1. Все скоры модели сохраняются в БД
2. Часть новых данных периодически отправляется на ручную разметку
 - а. для оценки метрик выбираются случайные объекты за интервал
 - б. для обогащения обучающей выборки выбираются объекты с пограничными вердиктами (active learning)
3. В интерфейсе модели предусмотрен ответ “Я не знаю”
 - а. при таком вердикте объект отправляется на ручную обработку
 - б. для некоторой доли уверенных ответов “Я не знаю” проставляется случайным образом
4. на основе ручной разметки для новых данных периодически пересчитываются метрики качества за выбранных период

Объем ручной перепроверки

Плюсы большого объема:

- маленький доверительный интервал для важных метрик
- большая скорость роста обучающей выборки
- быстрое изменение метрик со временем

Минусы большого объема:

- только небольшая часть работы оказывается автоматизирована
- большой объем ручной работы по перепроверке

Рекомендации по выбору объема

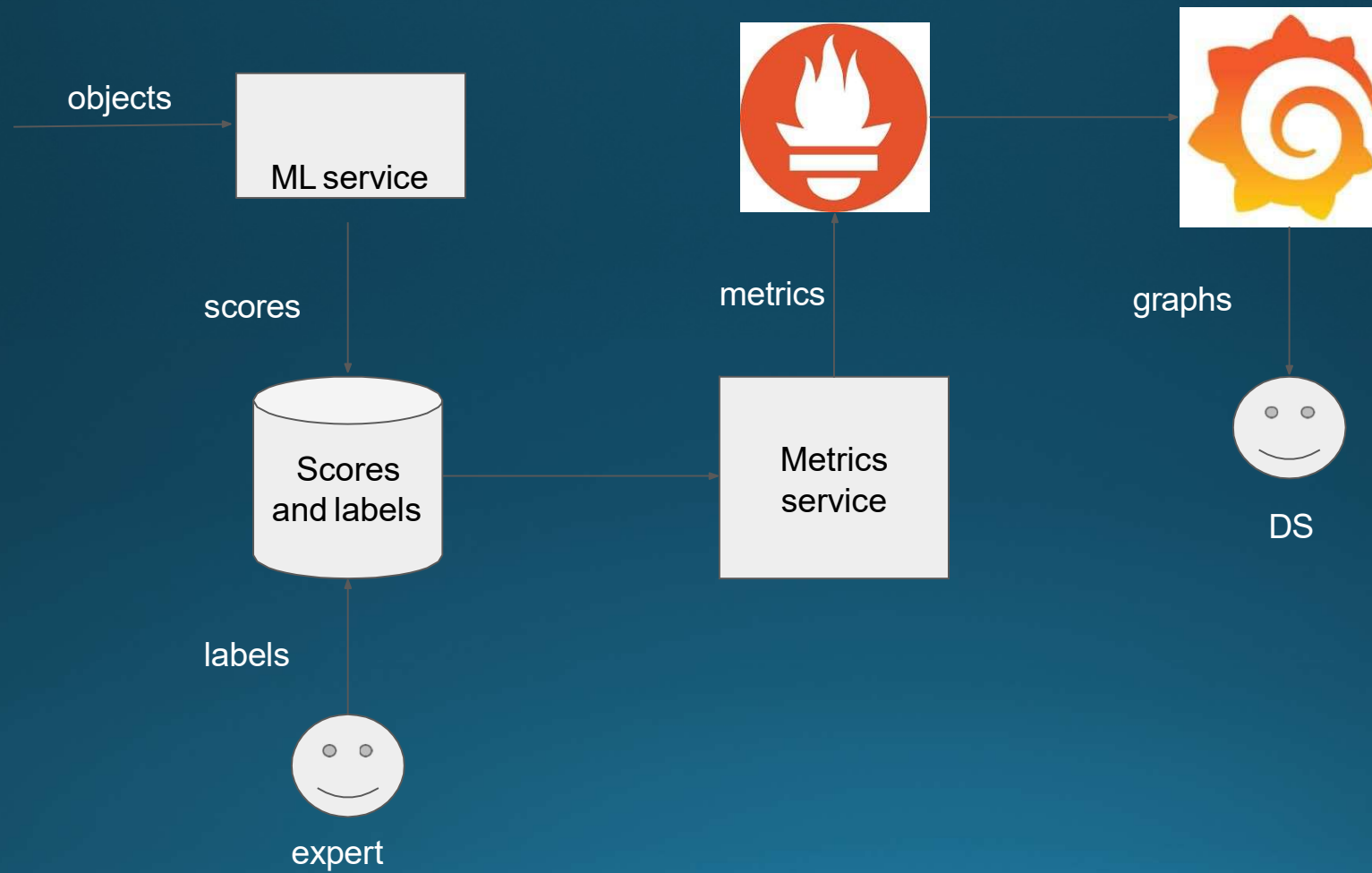
- требования к величине доверительного интервала для метрик
- ограничения на объем ручной перепроверки
- желаемое время обнаружения деградации модели

Пересчет метрик качества

Сервис расчета метрик

- периодически ходит в БД и забирает все объекты с ручной разметкой за выбранное время и вердикты модели
- вычисляет необходимые метрики
- сохраняет их в БД или отдает системе мониторинга
- отдельный дашборд настроен для визуализации метрик во времени

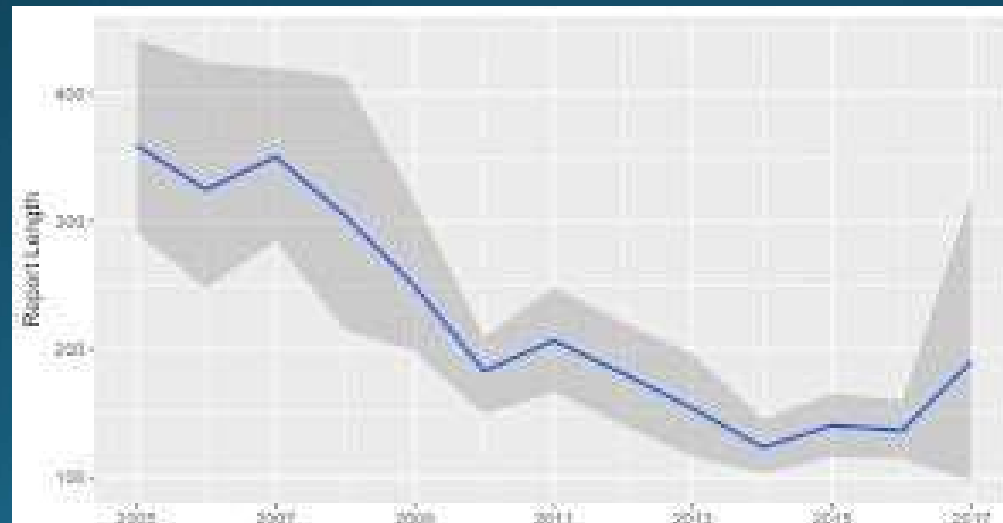
Схема взаимодействия



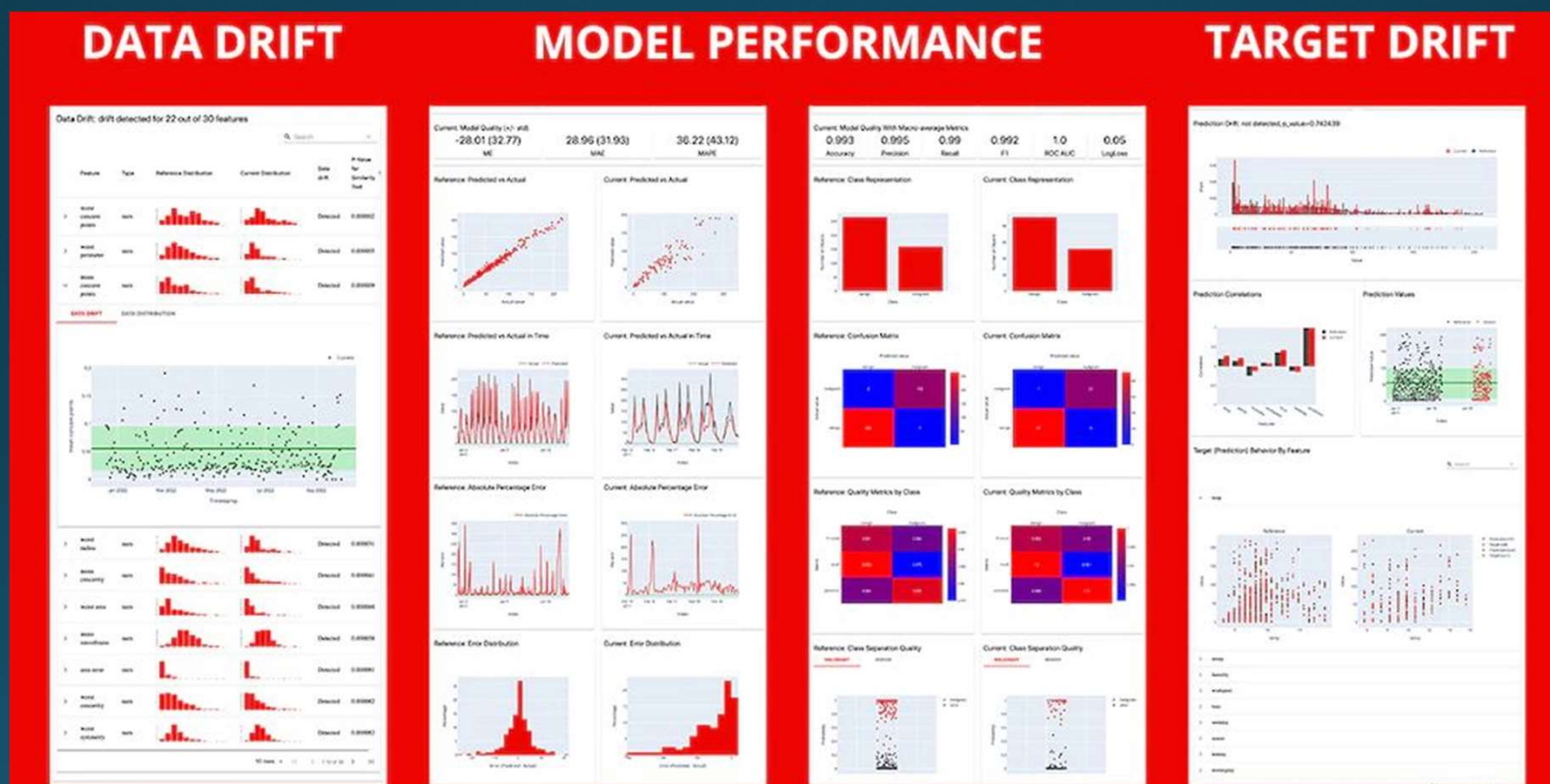
Мониторинг метрик качества

Визуализация

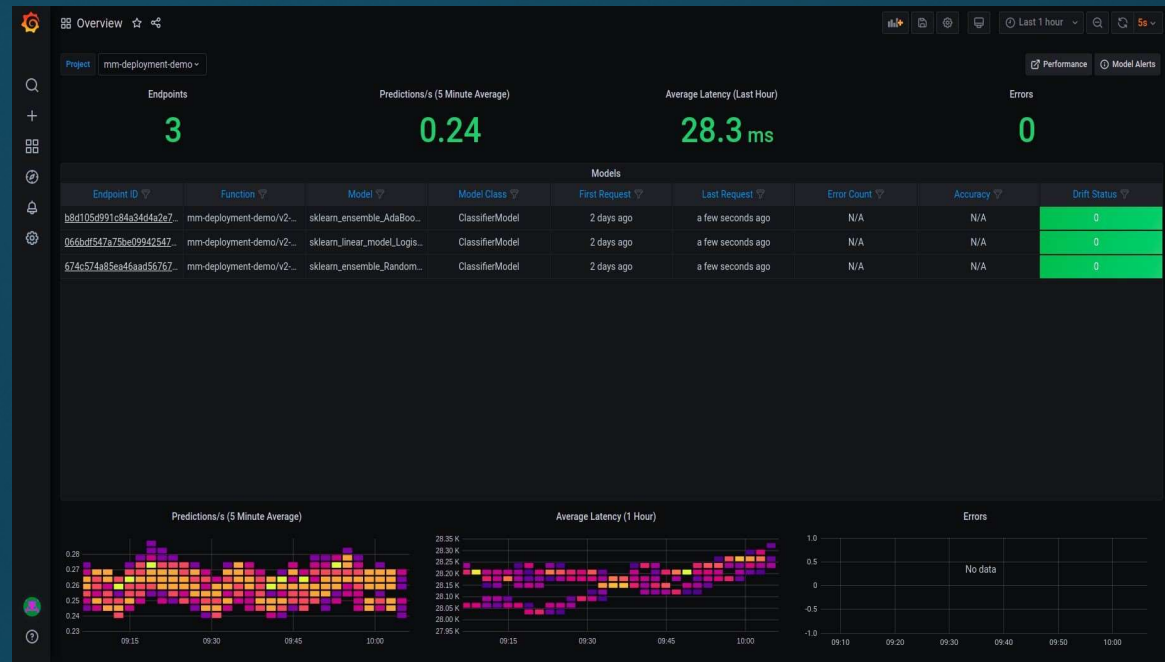
1. За разные интервалы времени может быть разное число примеров
2. Удобно визуализировать доверительный интервал
3. Строить метрики качества в необходимых разрезах
4. Подписывать версии моделей



Существующие решения



MLRun model monitoring



Monitor, Explain & Improve Your ML Models

```
MODEL_SERVING.PY

@app.post("/my-model/predict")
def predict(request):
    # Preprocess request
    X = preprocess(request)

    # Perform inference
    y = model.predict(X)

    # Log prediction to aporia
    aporia.log_prediction(X, y)

    return {"result": y}
```

Data Drift

New Value

Model Activity

Prediction Drift

Performance Degradation

Model Staleness

Aporia