

# SMS Spam Detector

# Проблематика проекта Фильтрация SMS спама и фишинга



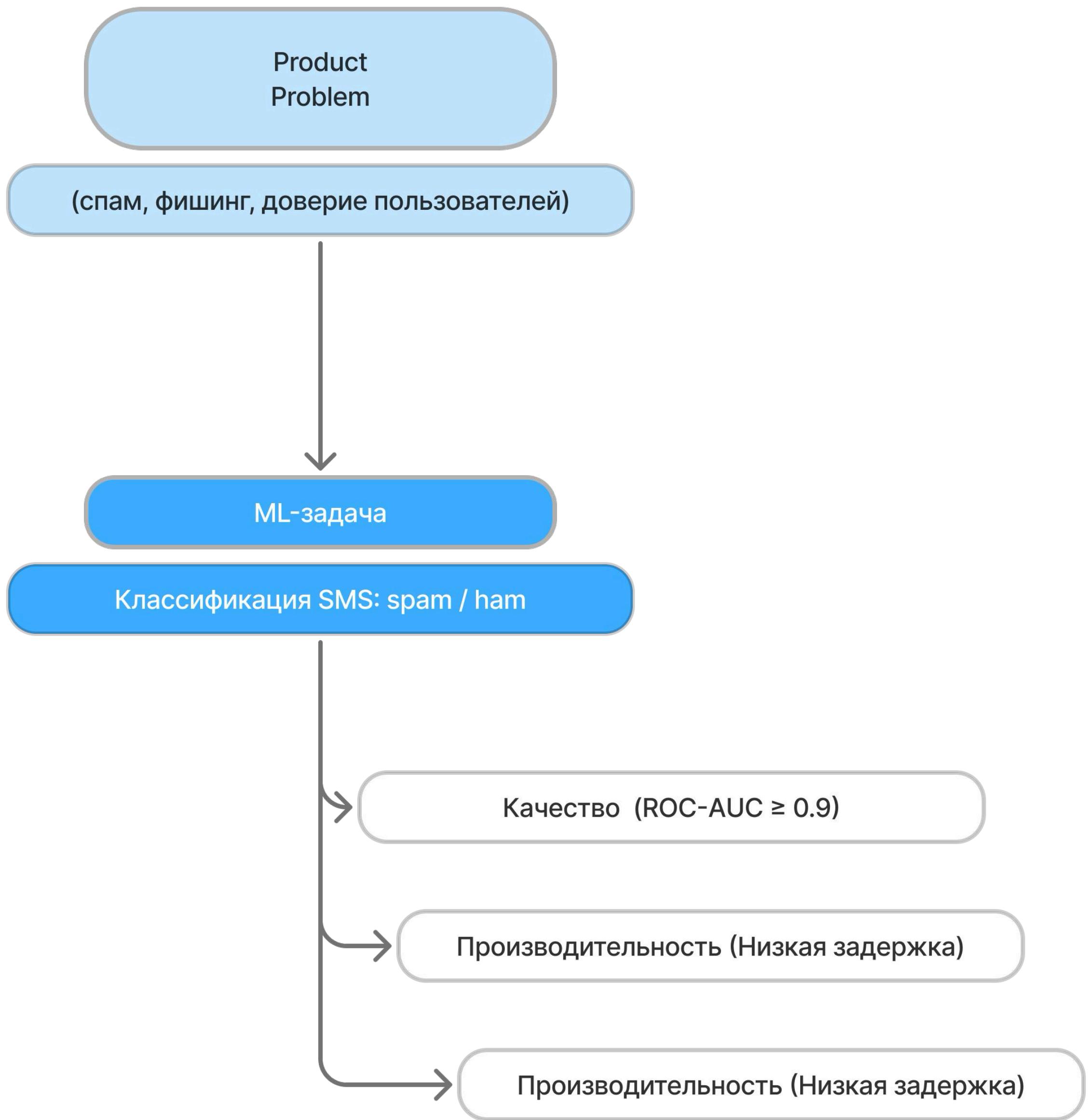
**47 %**

Процент сообщений,  
которые относятся к  
фишингу

47.27 % в Рунете

48.57 % в мире

# Постановка Задачи



# Модель Проекта



## Данные и предобработка

.01

Взяли

Открытый датасет  
SMS (spam / ham)

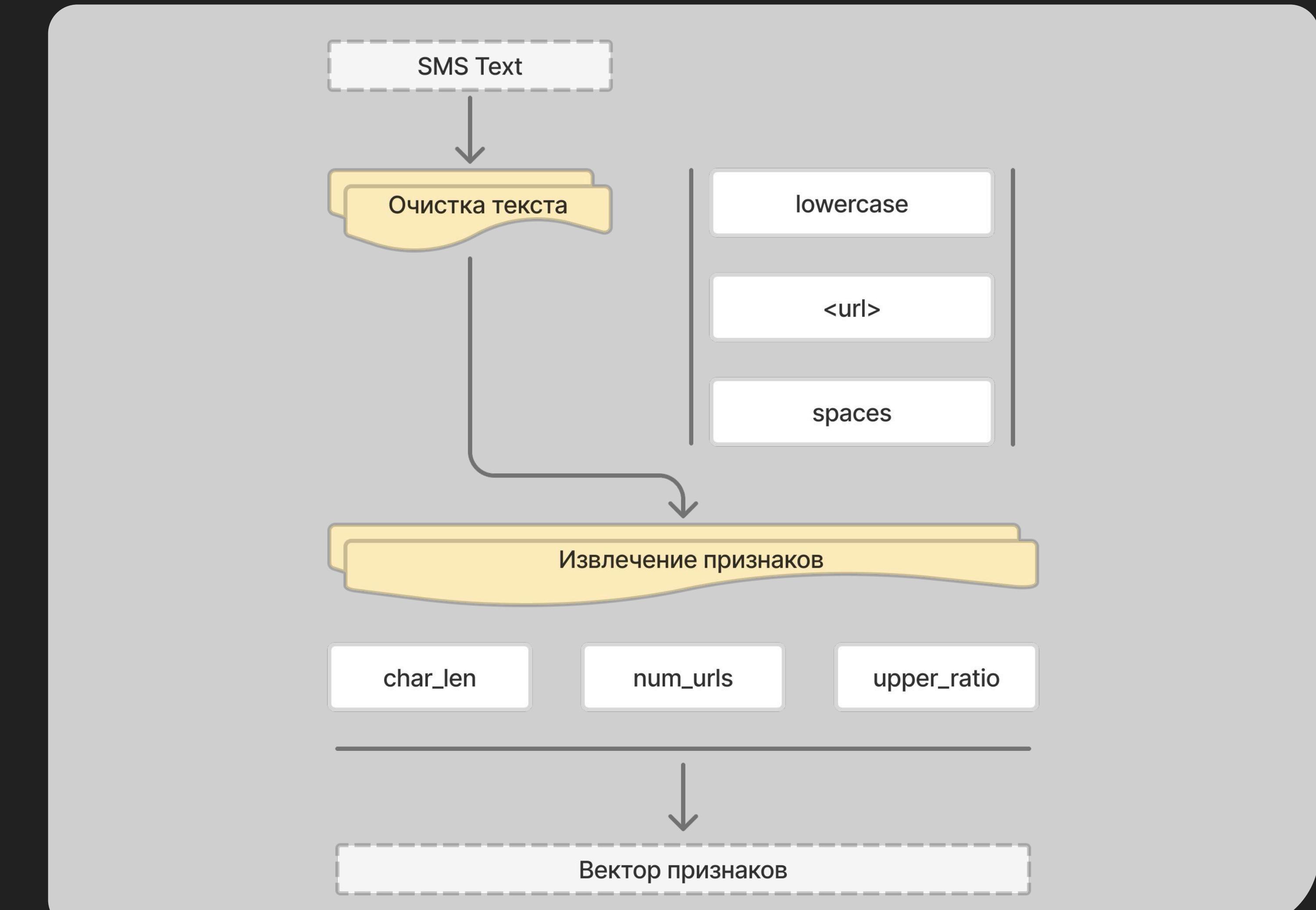
.02

Детерминирован  
ная очистка текста

.03

Простые  
интерпретируемые  
признаки

# Модель Проекта Структура модели



как предобращаются данные

# Модель Проекта

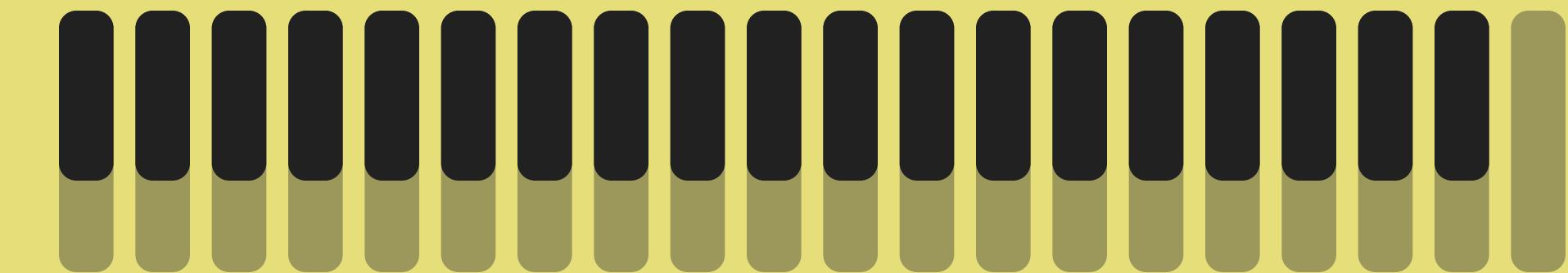


## Контроль качества

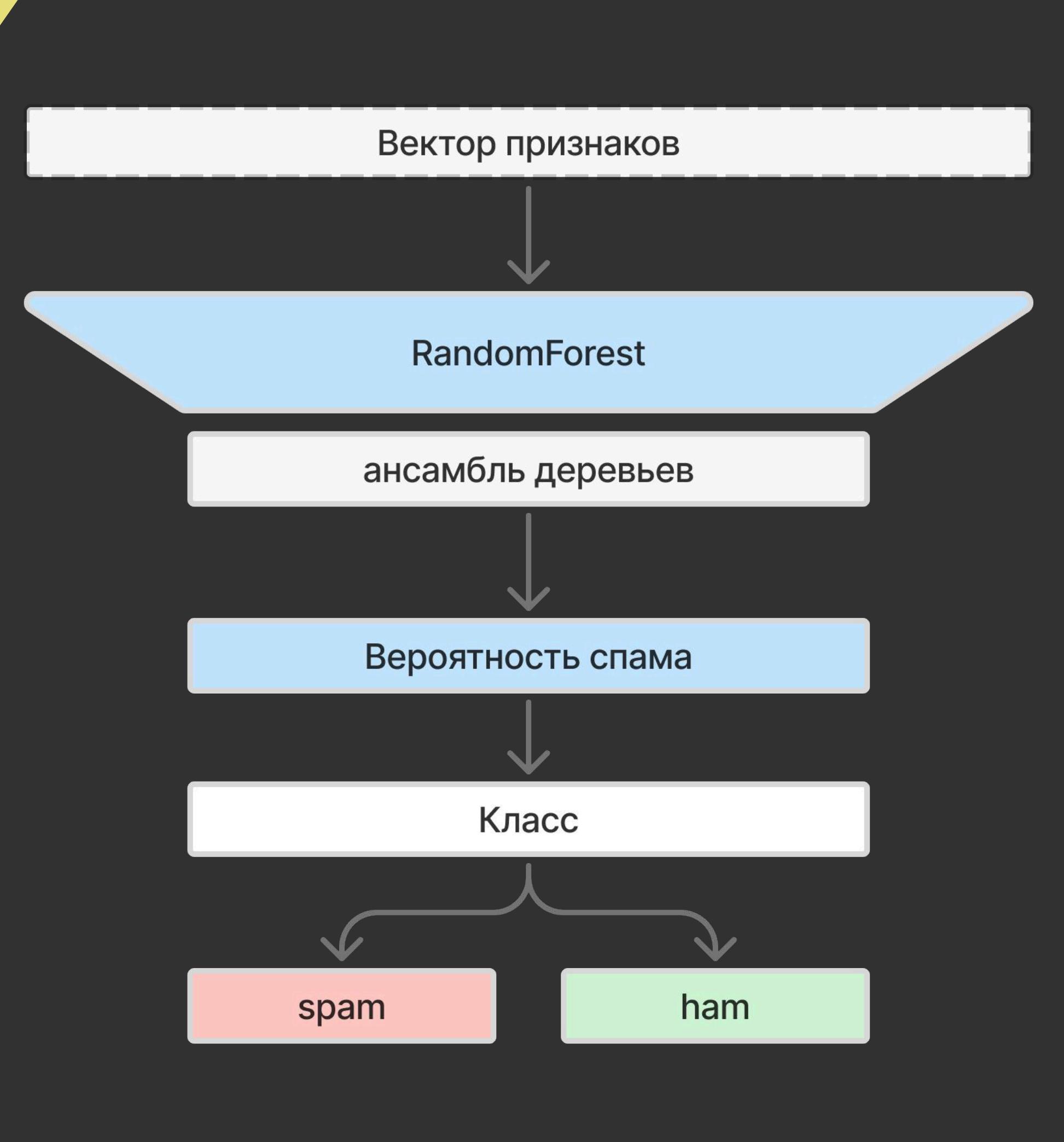
QUALITY  
GATE

>0.95

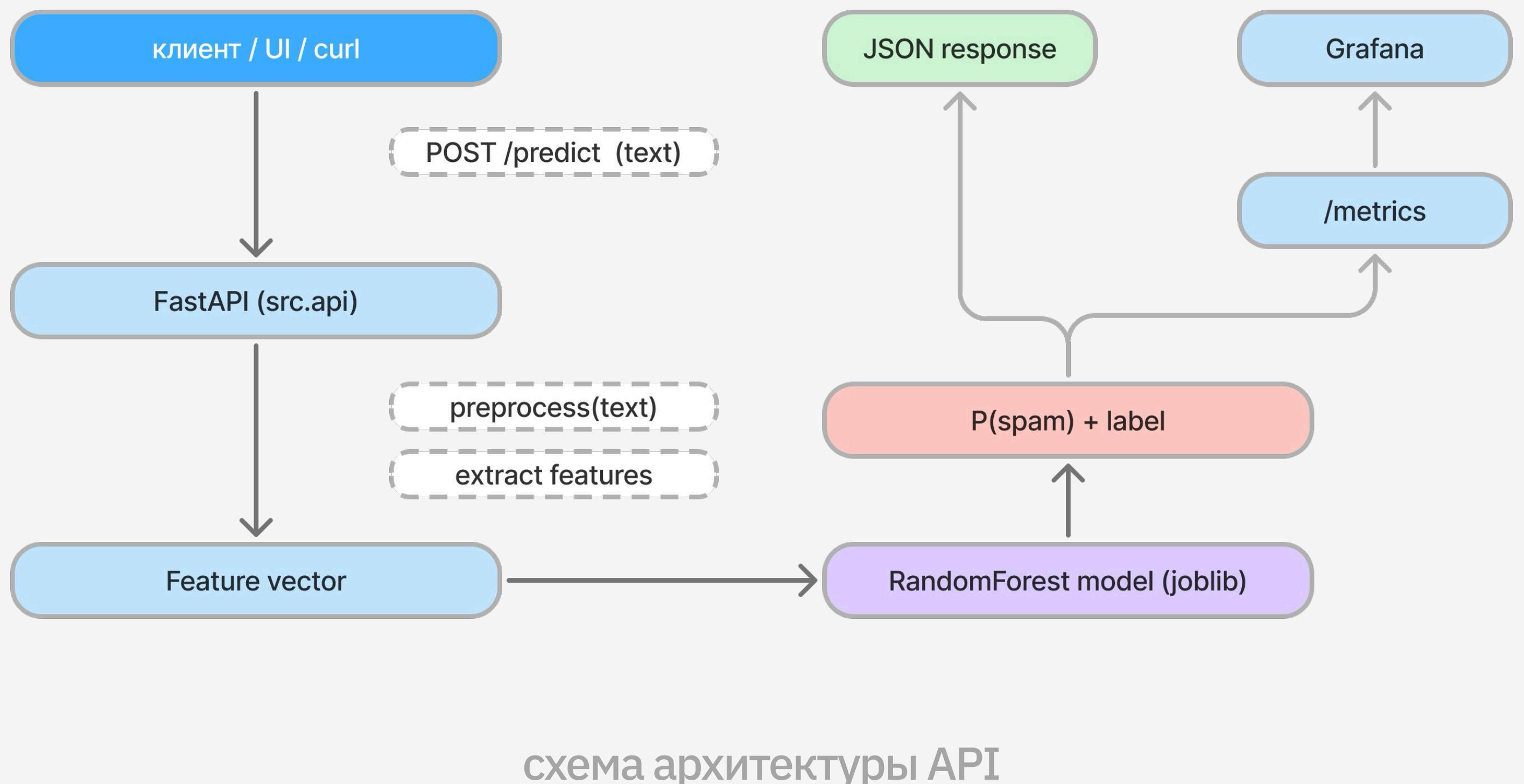
В проекте реализован механизм quality gate:  
модель регистрируется и допускается к деплою  
только в том случае, если значение ROC-AUG  
превышает заданный порог



АВТОМАТИЧЕСКАЯ  
ОЦЕНКА  
ПОСЛЕ ОБУЧЕНИЯ  
МОДЕЛИ

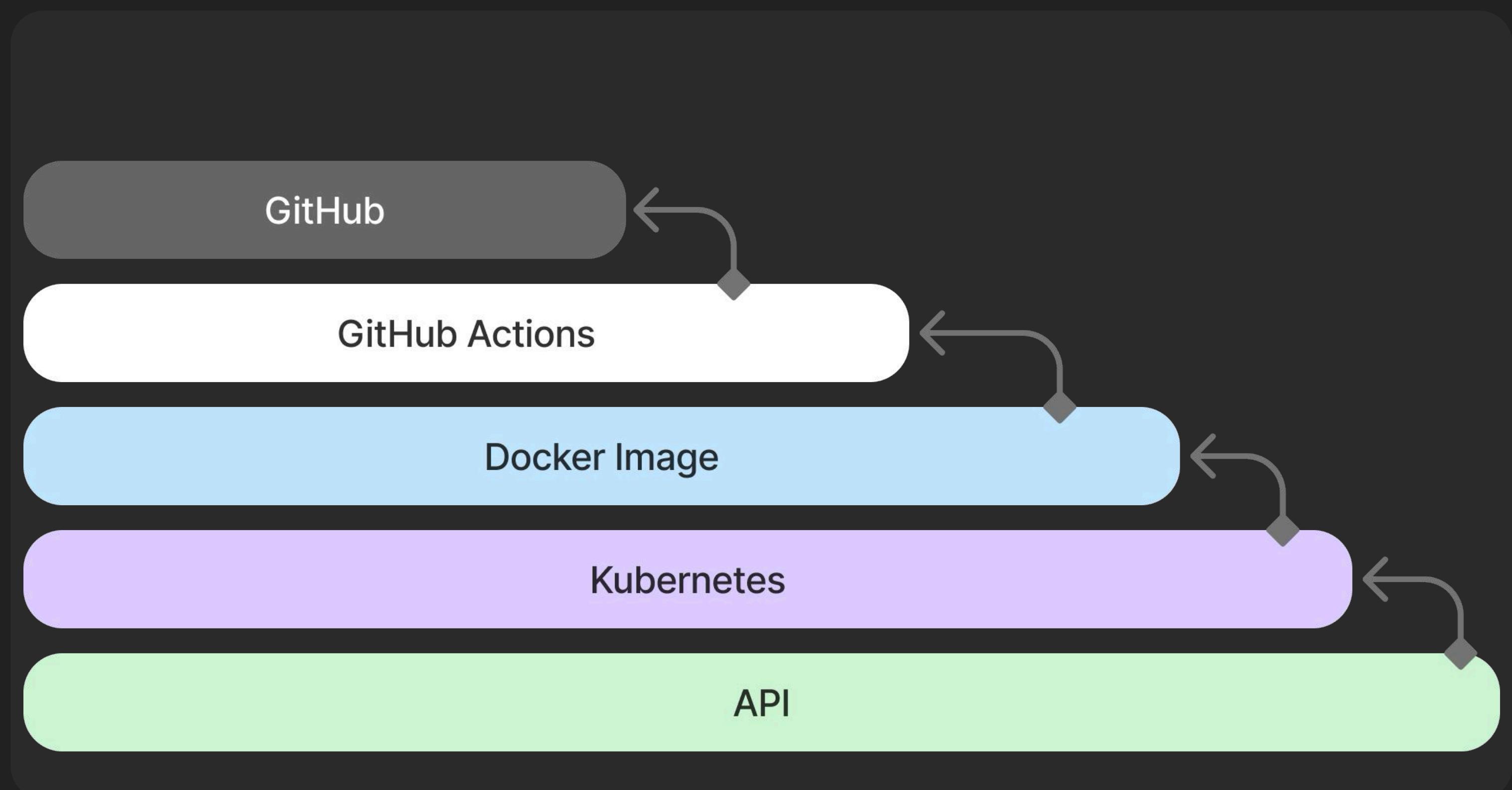


# API и сервис инфере́нса



**FAST-API СЕРВИС  
ДЛЯ ОНЛАЙН-ИНФЕРЕНСА  
МЕТРИКИ И  
HEALTHCHECK ИЗ КОРОБКИ**

# Деплоймент и CI/CD



**CI: тесты → Docker → registry**

**CD: автоматический rollout в  
Kubernetes**

**Наблюдаемость:  
Prometheus + Grafana**

# Быстрая проверка SMS на спам

Модель RandomForest на ручных признаках. Введите текст и получите вероятность спама.

[Метрики Prometheus](#) [Репозиторий](#)

## Введите текст сообщения

Модель RandomForest на ручных признаках. Введите текст и получите вероятность спама.

Ham пример

Spam пример

Проверить

## Результат

HAM

P(spam)

7.6%

Модель: model\_store/random\_forest.joblib

## История запросов

последние 3

Модель RandomForest на ручных признаках. Введите текст и получите вероятность спама.

HAM

P(spam) = 7.6%

**URGENT! You have won \$1000. Click here to claim.**

HAM

P(spam) = 22.0%

**Win a free iPhone if you reply NOW!**

HAM

P(spam) = 10.5%

# Команда

Горшколепов Ярослав  
топовый разработчик



Рясной Владимир  
главный разработчик



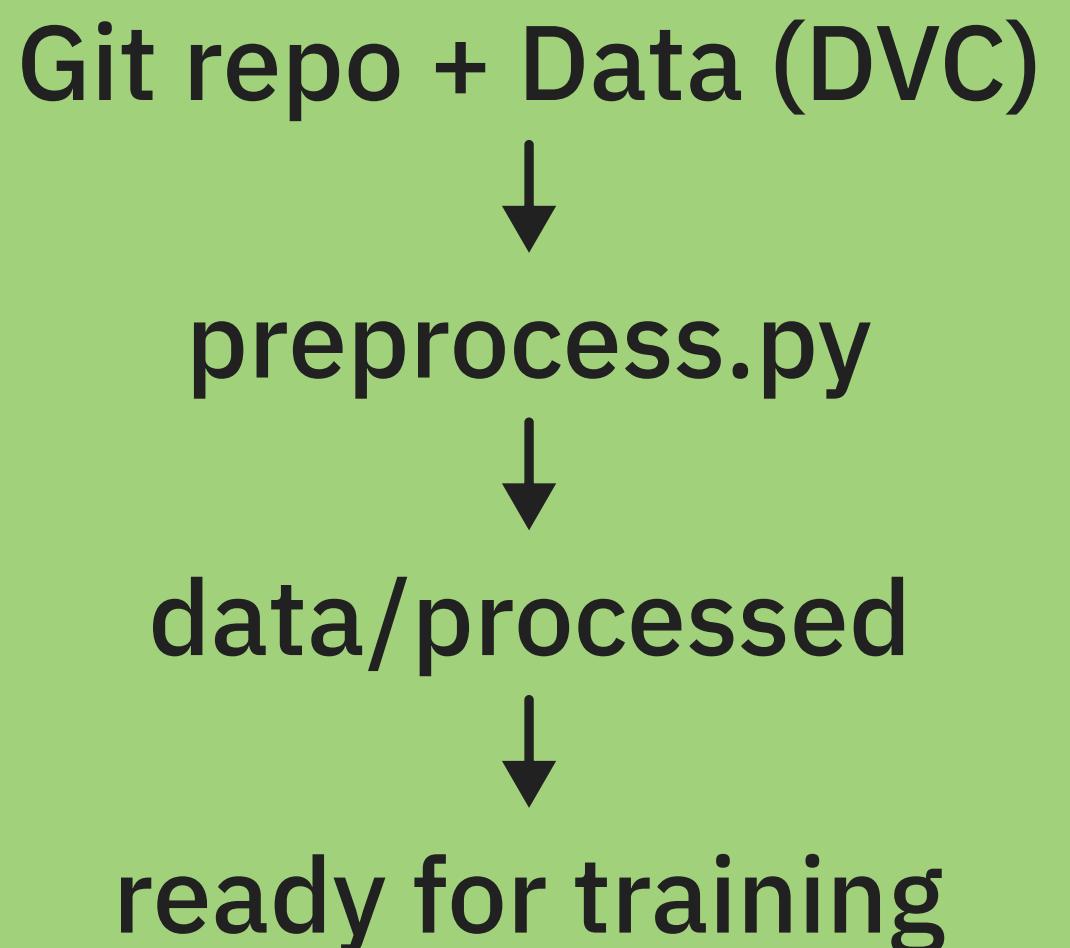
Жаров Игорь  
тимлид, разработчик



# Владимир

Data + ML часть проекта

- Структура репозитория (src/data/  
models/tests)
- DVC: данные в версии + remote
- raw → processed + генерация признаков



Data



Preprocess



Features

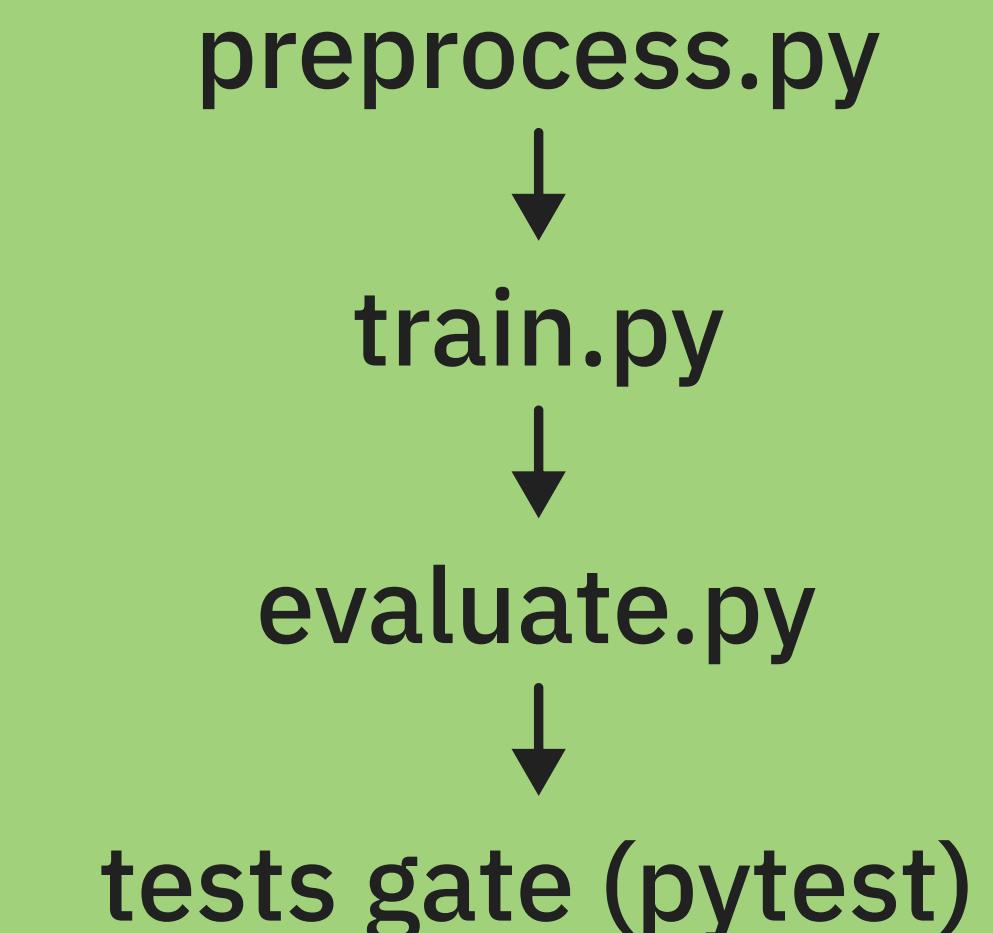


Train

# Владимир

Data + ML часть проекта

- `train.py`: split → модель → метрики → save
- `evaluate.py`: отчёт + ROC-AUC / Precision / Recall
- `tests/test_preprocess.py`: юнит-тесты + фикстуры



Preprocess



Train



Evaluate

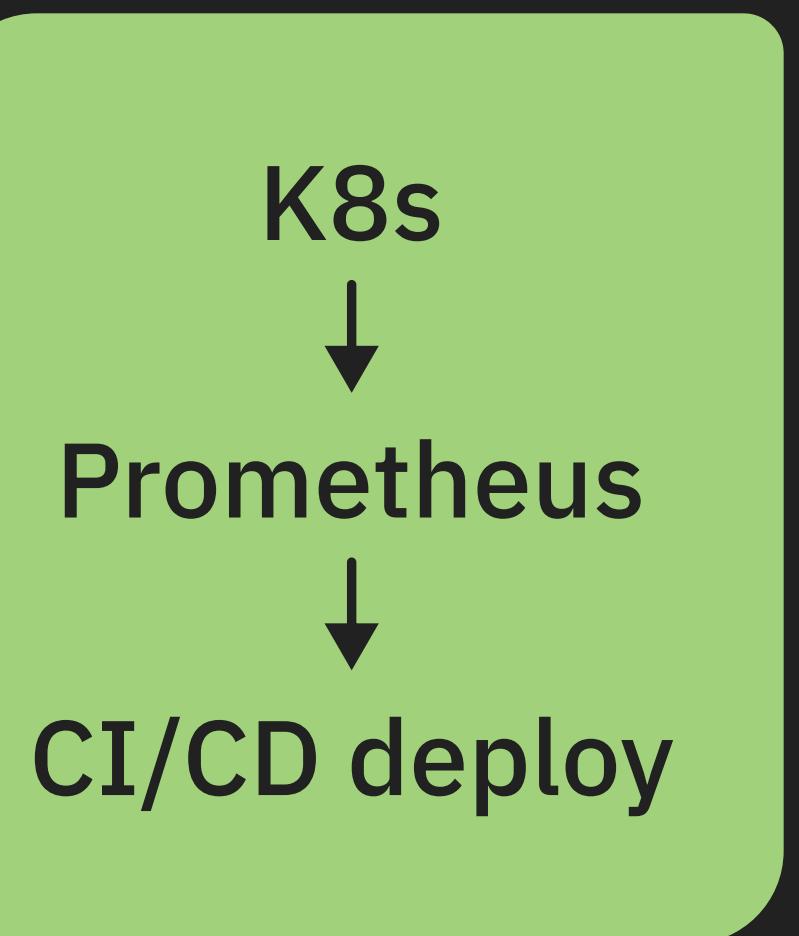
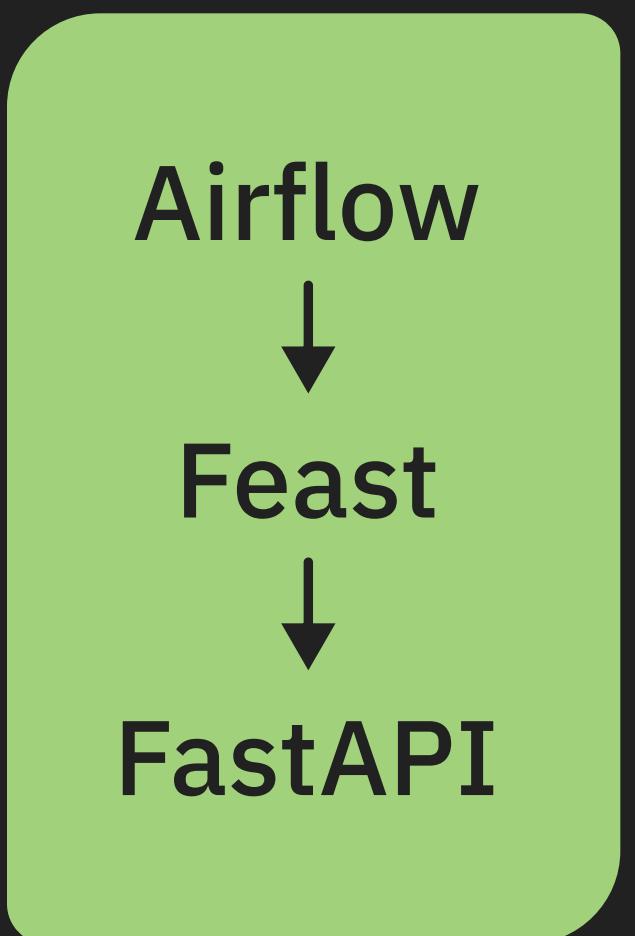


Tests

# Владимир

Production loop / Infra

- Airflow: окружение (compose + init), DAG-запуски
- K8s: манифесты + probes + HPA (autoscaling)
- Наблюдаемость и деплой: Prometheus / metrics + GitHub Actions CI/CD



Airflow



API



K8s (HPA)

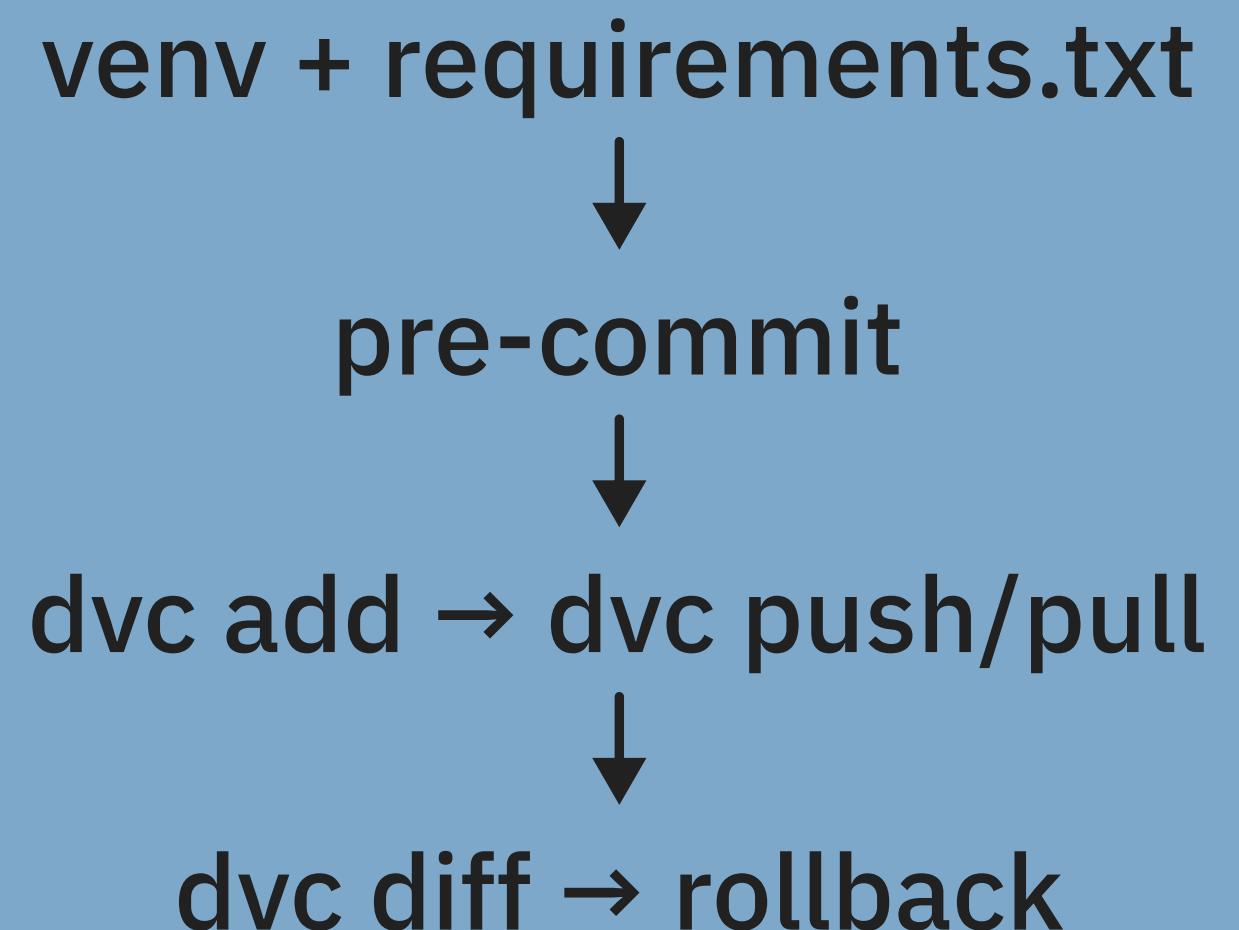


Monitoring

# Игорь

Dev experience + DVC

- Окружение проекта: `venv + requirements.txt`
- `pre-commit: black + isort + flake8`  
(проверки перед PR)
- DVC-версионирование данных: `add → diff → rollback`



Env



Code style



DVC

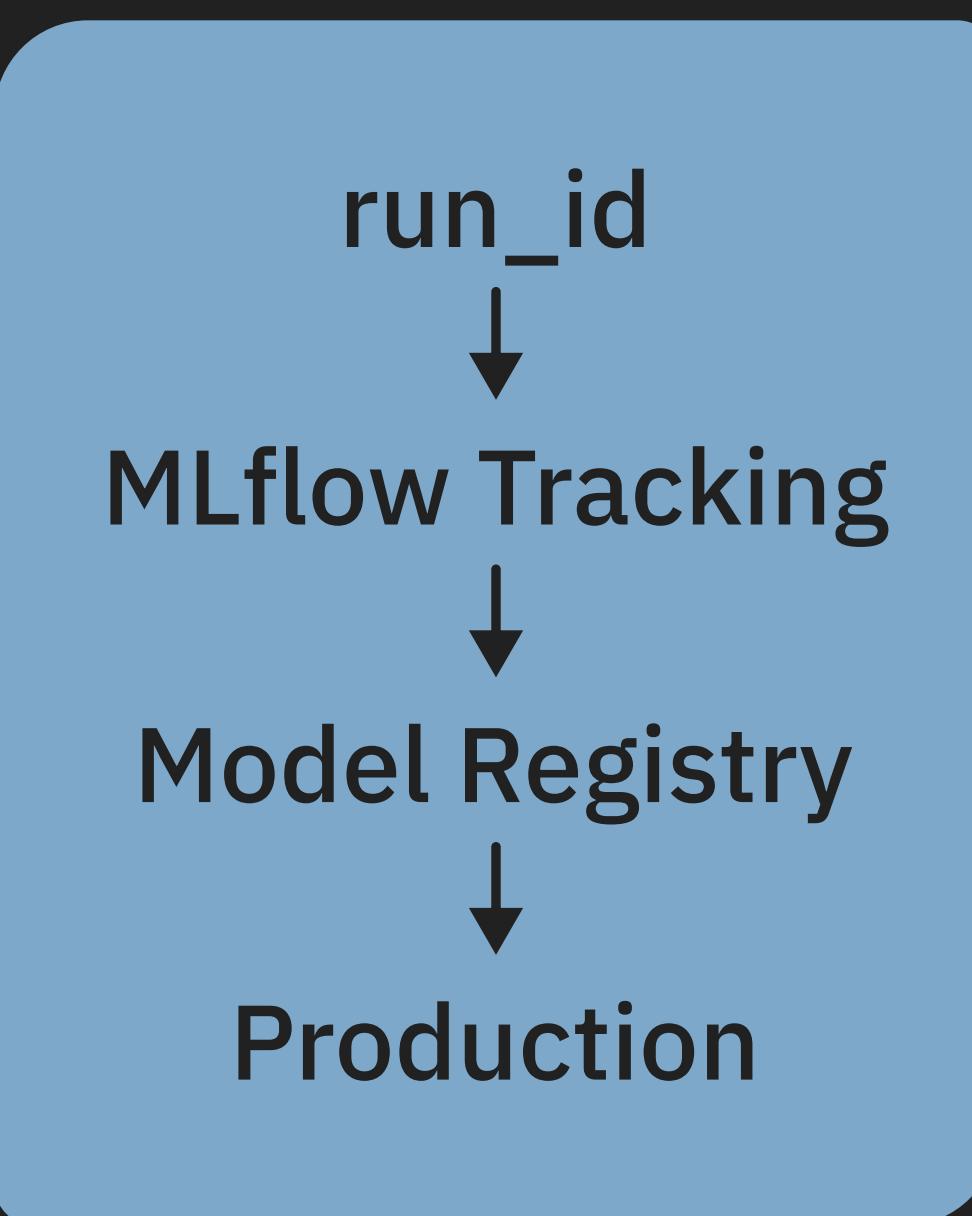


Rollback

# Игорь

MLflow / Registry

- Логирование экспериментов: параметры, метрики, артефакты
- MLflow UI: сравнение прогонов (runs)
- Model Registry: register.py – quality gate и продвижение модели



Tracking



UI



Registry

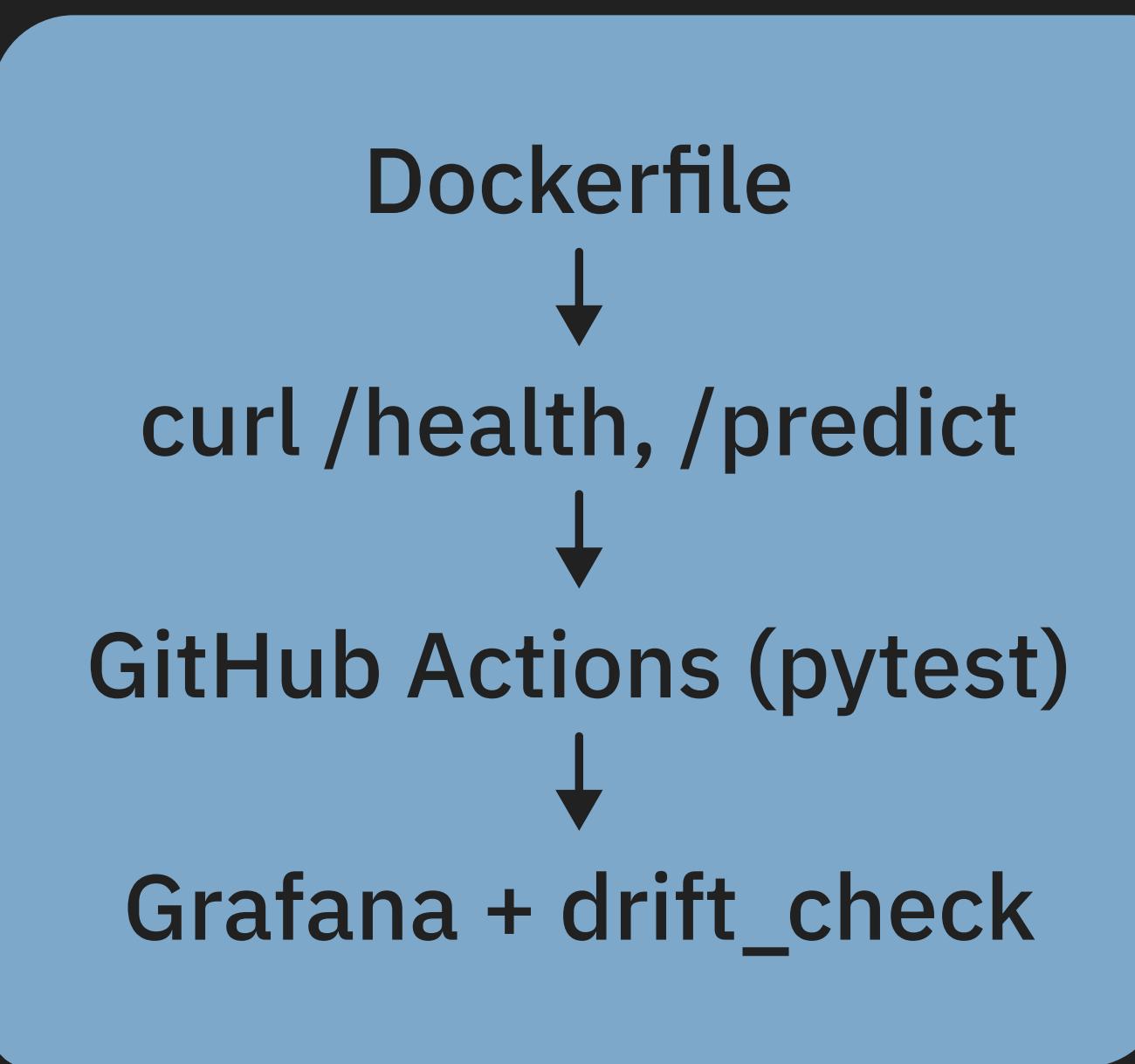


Promotion

# Игорь

Docker + CI + Monitoring

- Docker: образ API + проверка /health и /predict
- CI (GitHub Actions): build → pytest  
→ push образа
- Monitoring: Grafana dashboard + drift\_check отчёт



# Ярослав

Team workflow / Integration

- Репозиторий: `.gitignore` + `dvcignore` + `environment.yml`
- Командная работа: PR → review → merge в main
- Воспроизводимость: `dvc pull` + `dvc repro` (чистый запуск)

feature branch  
↓  
PR + review  
↓  
CI checks  
↓  
merge + dvc repro



# Ярослав

Airflow + API tests

- API-тесты: `tests/test_api.py` + `conftest.py` (TestClient)
- Контракт `/predict`: ключи + типы + обработка ошибок
- Airflow DAG: `flight_pipeline` + Variables + логи

download/preprocess  
↓  
train  
↓  
evaluate  
↓  
register



# Ярослав

Feast + K8s + Alerts + Drift

- Feast-интеграция: offline store → get\_historical\_features (учим на признаках)
- K8s деплой + масштабирование: Minikube, / health + /predict, HPA/replicas
- Мониторинг + дрейф: Prometheus rules/alerts + drift validation + kubeconfig для CI

