Предсказание экстраверта/ интроверта

MLOps Pipeline

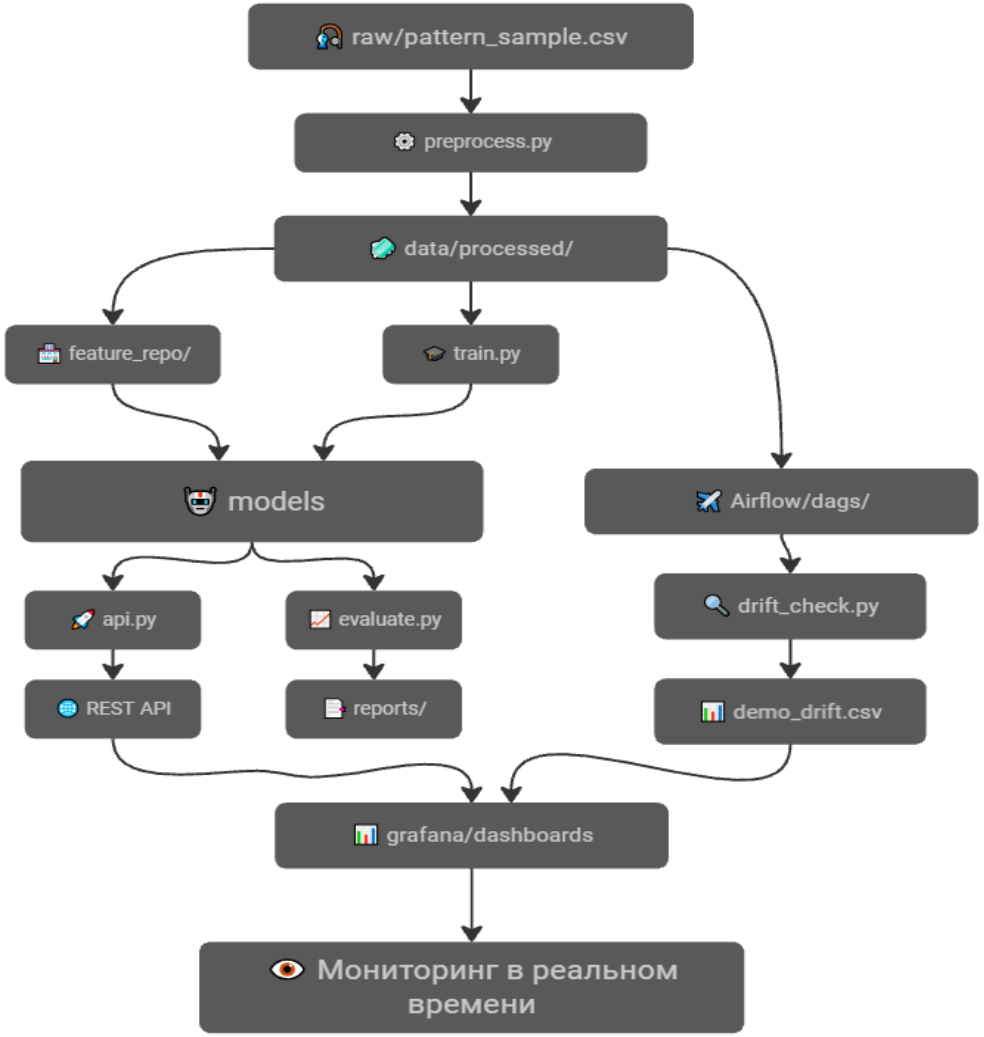
Выполнили:
Король А.Ю.
Раздоркина Д.С.
Банников Д.Ю.
Аврамчук И.А.

Структура и архитектура проекта

Архитектура

```
.dvc/                    #Управление версиями данных
Airflow/
├── dags/
│   ├── drift_detection.py      # Детекция дрейфа данных
│   └── personality_pipeline.py  # Основной пайплайн личности
├── Dockerfile
└── docker-compose.yaml
data/
├── raw/                  # Сырые данные
│   └── pattern_sample.csv
└── processed/            # Обработанные данные
    ├── processed.csv          # Базовая обработка
    ├── processed_feast.csv    # Для feature store
    ├── pers_features.parquet  # Признаки личности
    └── demo_drift.csv         # Демо-данные для дрейфа
feature_repo/
├── feature_store.yaml        # Конфигурация Feast
├── definitions.py            # Определения сущностей
├── features.py               # Feature
└── test_workflow.py          # Тесты Feast
grafana/
├── dashboards/
│   ├── dashboard.yml         # Конфигурация дашбордов
│   └── generate_load.py      # Генерация тестовой нагрузки
k8s/
models/
prometheus/
reports/
src/
├── api.py              # FastAPI/Fastify endpoint
├── preprocess.py       # Препроцессинг данных
├── train.py            # Обучение моделей
├── evaluate.py         # Оценка моделей
├── drift_check.py      # Проверка дрейфа
└── drift_utils.py      # Утилиты для дрейфа
```

Путь данных



Данные и обработка признаков

Пайплайн обработки

- **Очистка**: удаление дубликатов и выбросов.
- **Нормализация:** приведение численных данных к единой шкале
- **Кодирование:** One-hot и Label encoding для категорий.

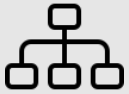
Целевая переменная и фичи

- **Target**: Extrovert/Introvert
- **Features:** Время проведения в одиночестве, страх сцены, посещение мероприятий, частота выхода из дома, чувство истощения после общения, круг друзей, частота постов в соц сетях.
- **Хранение:** Feast Feature Store обеспечивает доступность фичей для обучения и инференса.

Источник данных: kaggle.com

Объем данных: 3000 строк описаний личности
Train/Test: 80% / 20%

Инструменты разработки и качества



Pre-commit Hooks

Автоматический линтинг и форматирование кода:

- Black
- Flake8
- Check YAML

Feature store (Feast)

Управление признаками и версионированные данных

Orchestration

Airflow DAGs для регулярных задач:

- Обучение модели
- Ежедневный дрифт чек

Feature store: Feast

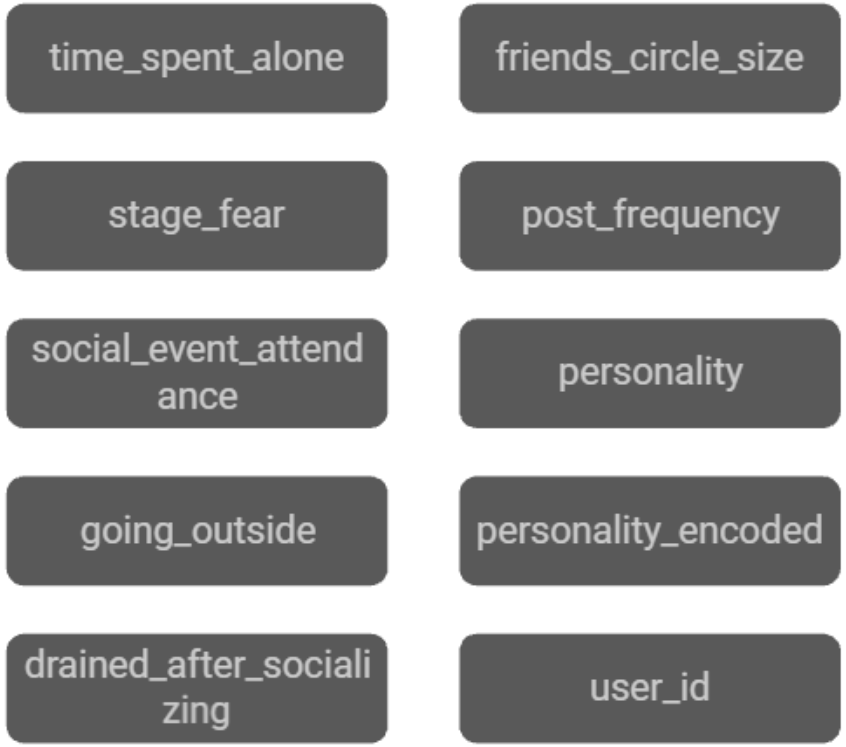
Консистентность признаков

Feast используется для унификации признаков между этапами обучения и инференса

Offline Store:

- Хранение набора данных для обучения модели RandomForest.
- Регистрация: хранение метаданных о признаках (версии, источники).

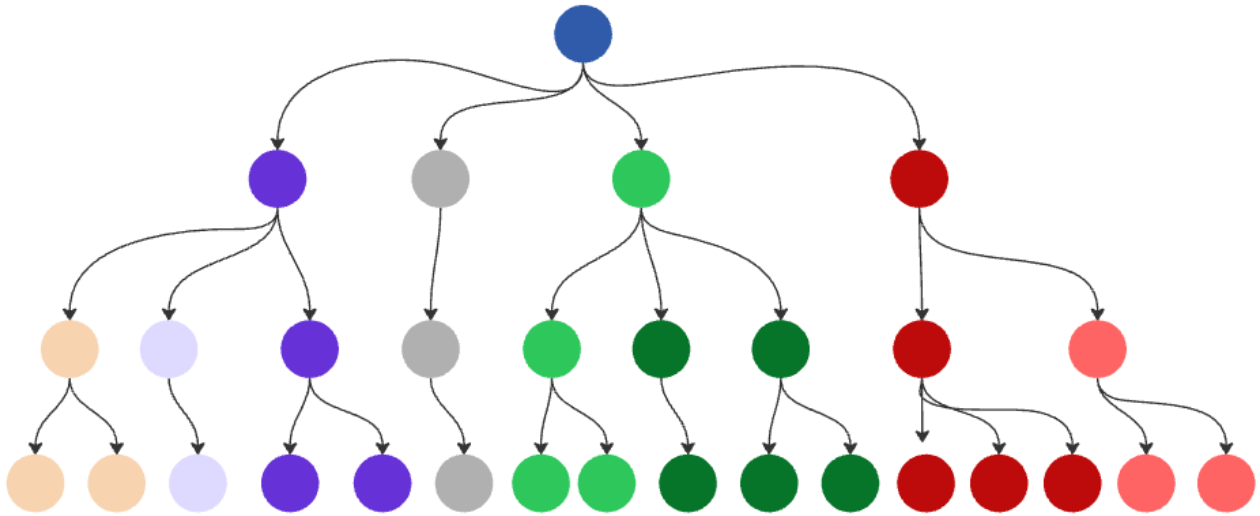
Признаки (всего 10):



Модель

RandomForest

Выбрана за интерпритируемость и удобства при работе с категориальными признаками

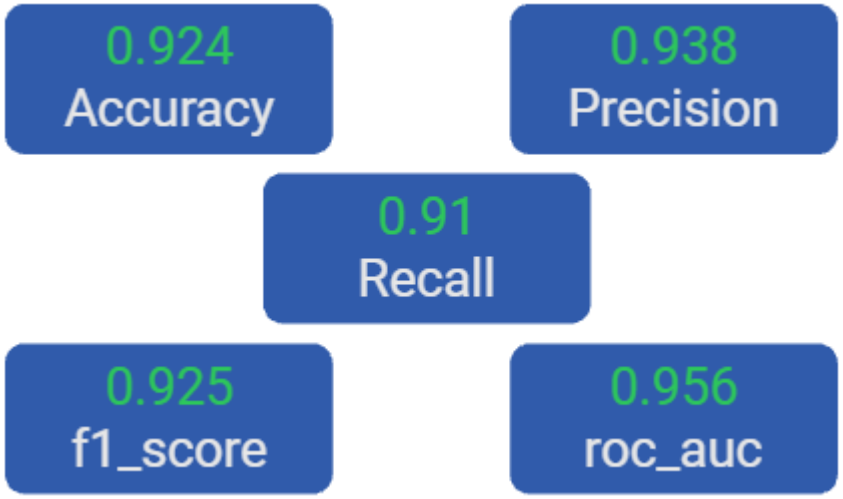


Параметры дерева

Кол-во деревьев	n_estimators=	100
Макс глубина дерева	max_depth=	10
Кол-во сэмплов для сплита	min_sam_split=	5
Кол-во сэмплов в листе	min_sam_leaf=	2
Стратегия выборки признаков	max_features=	sqrt
Критерий разделения	criterion=	gini
Бутстраппинг	bootstrap=	True
Веса классов	class_weight=	balanced
RANDOM_STATE	random_state=	42
Параллелизация	n_jobs=	-1

Результаты обучения

Метрики



Матрица ошибок

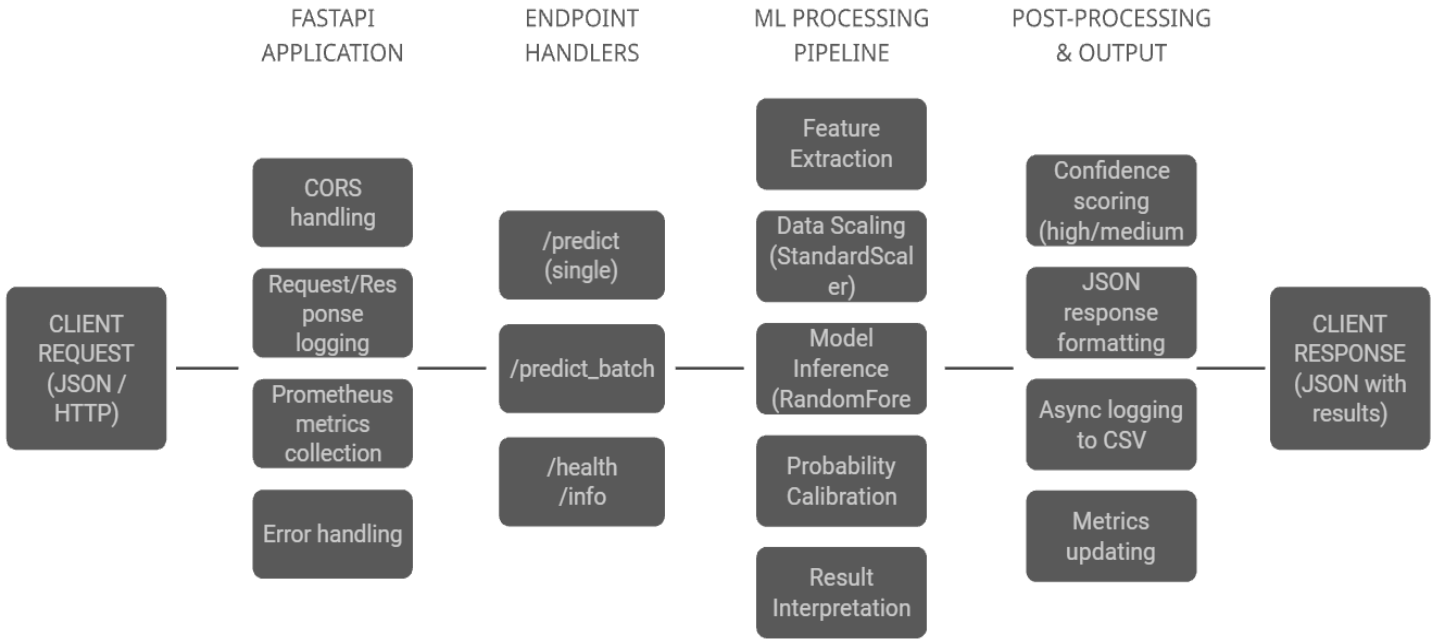
Матрица ошибок			
	Предсказано		
	Интроверт	Экстраверт	
Истина интров.	272	26	298
Истина экстрав.	18	264	282
	290	290	

Описание API (FastAPI)

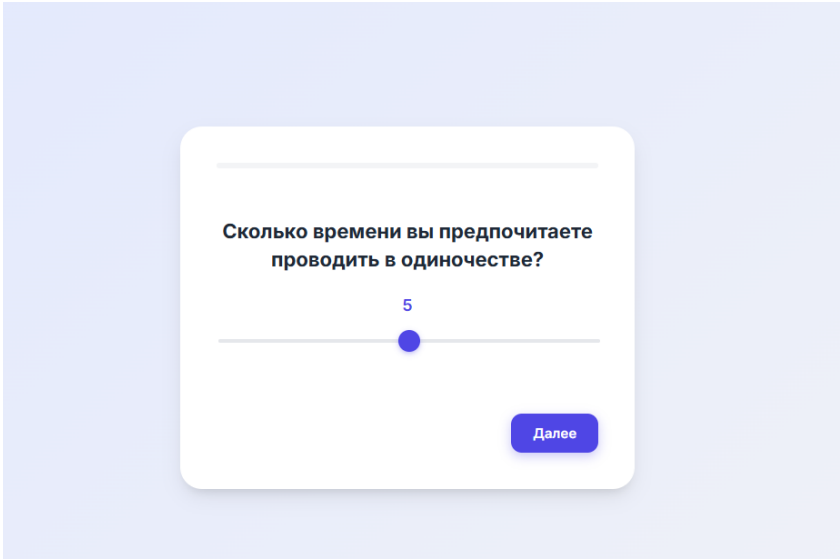
Основные Эндпоинты

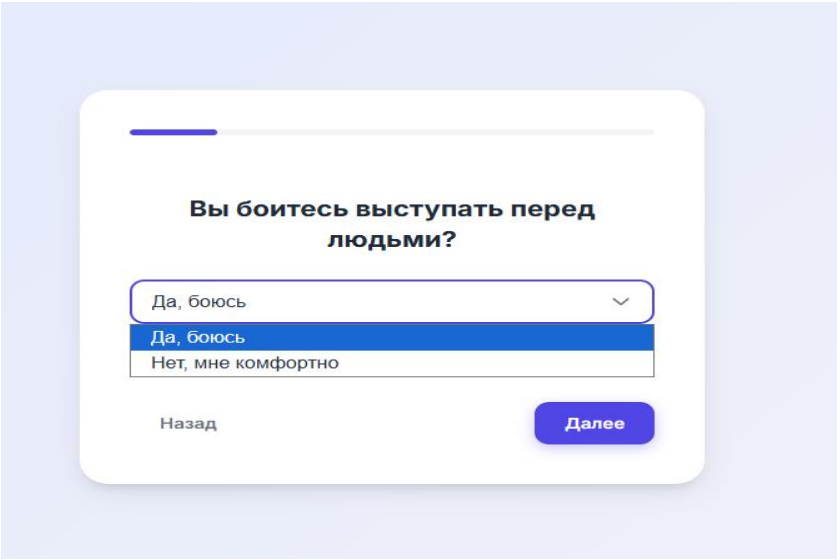
Документация доступна через Swagger UI.

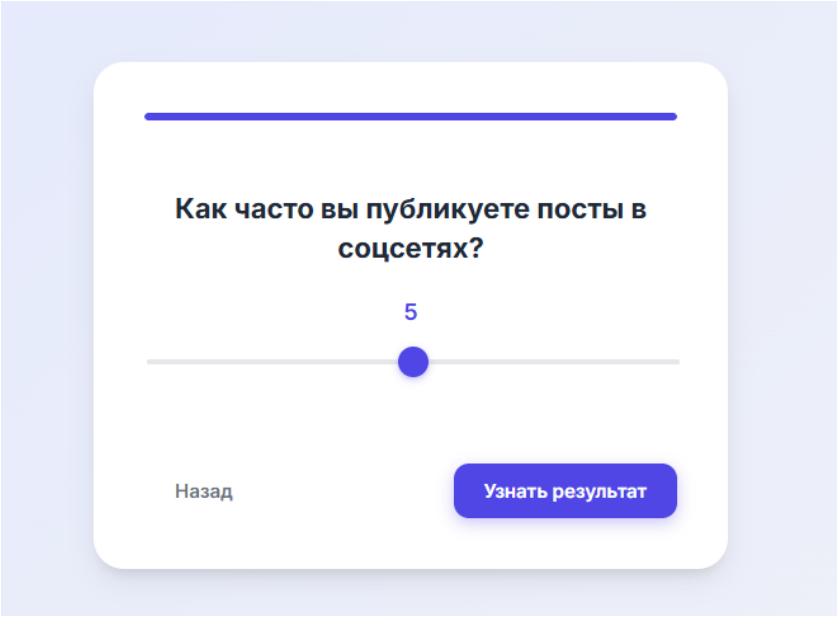
Архитектура API

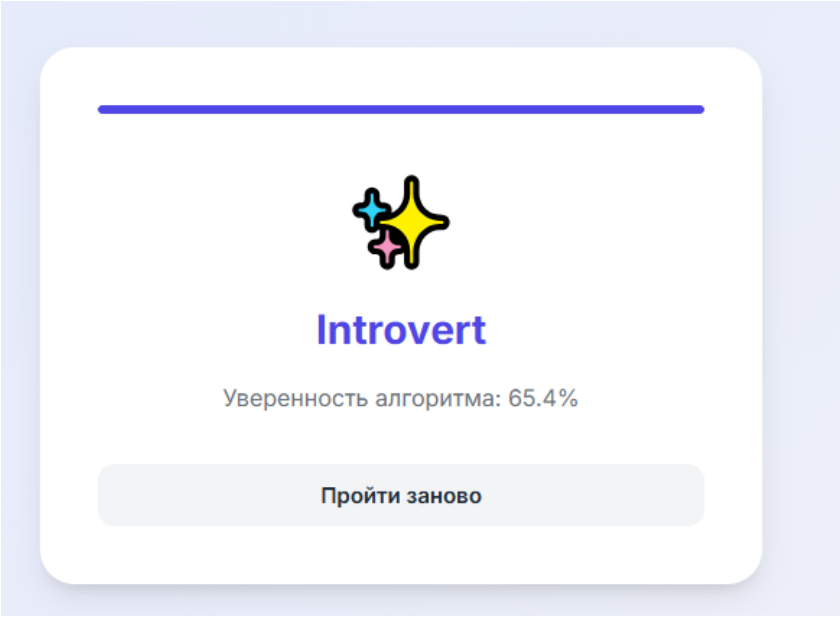


API + UI


Сколько времени вы предпочитаете проводить в одиночестве?
5
Далее


Вы боитесь выступать перед людьми?
Да, боюсь
Да, боюсь
Нет, мне комфортно
Назад
Далее

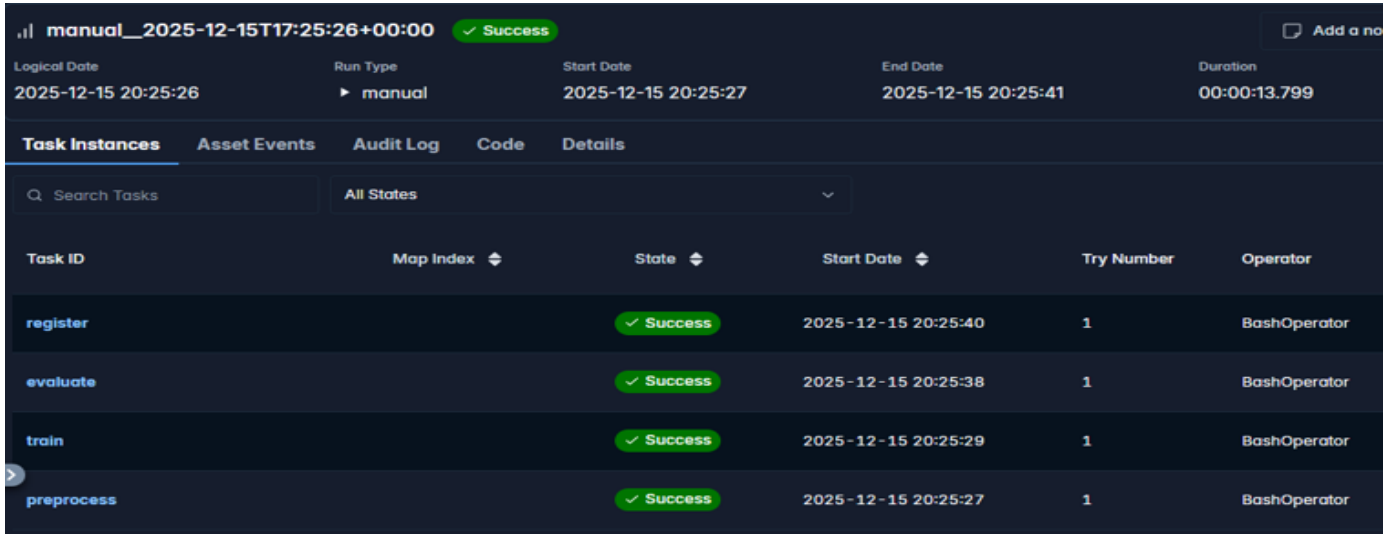

Как часто вы публикуете посты в соцсетях?
5
Назад
Узнать результат

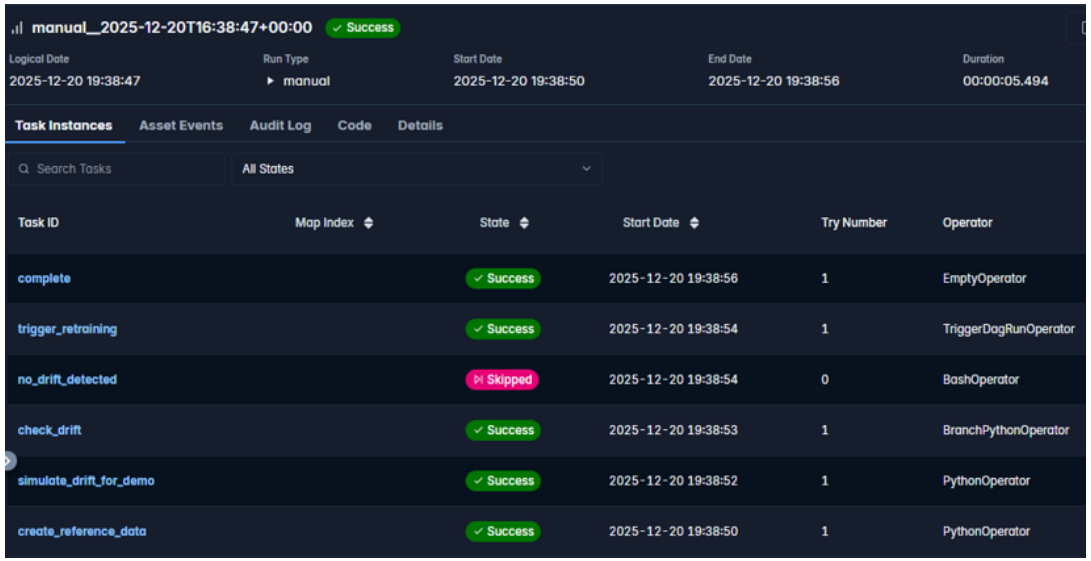

Introvert
Уверенность алгоритма: 65.4%
Пройти заново

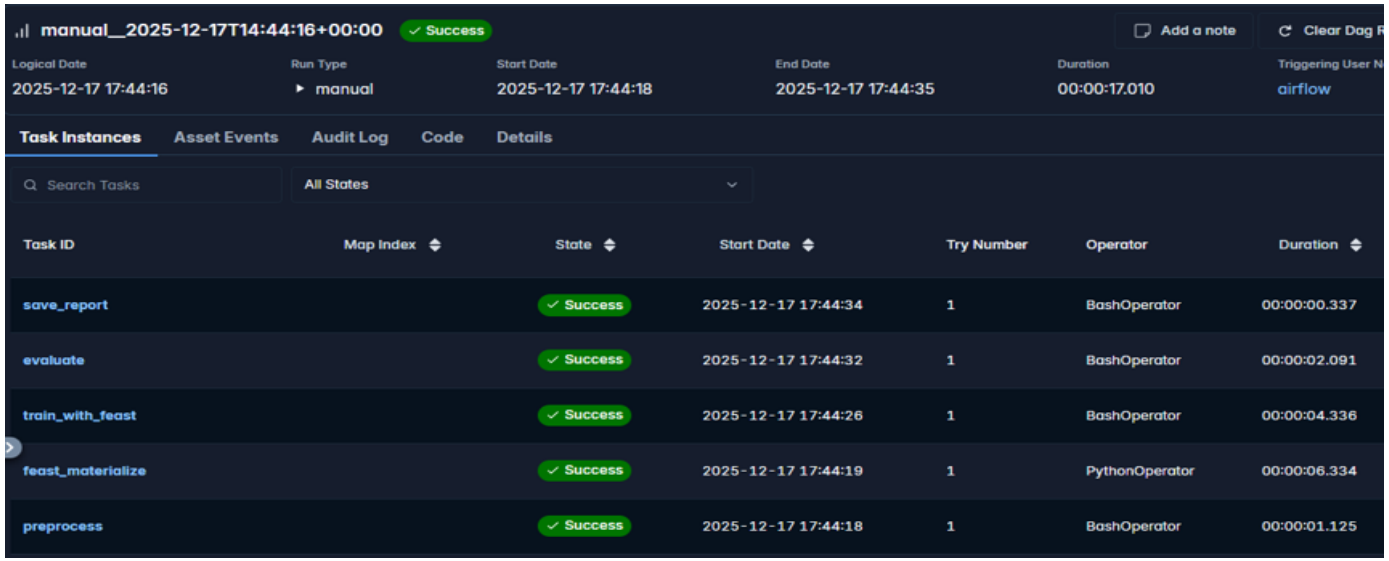
Автоматизация и оркестрация Apache Airflow

Инструменты

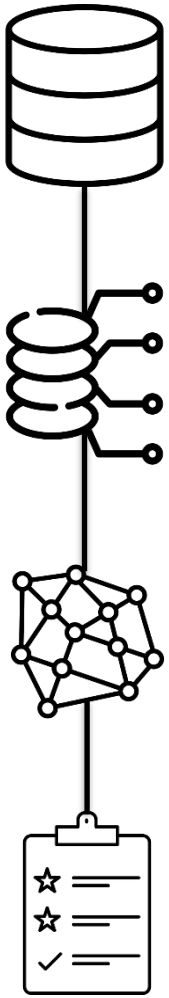
- Настроен DAG для ежедневного обнаружения дрифта данных (Data Drift)
- Мониторинг: сравнение распределения признаков между тренировочным и текущим production - набором.
- Триггер: если дрифт превышает установленный порог, Airflow может автоматически запустить DAG переобучения.







ML-пайплайн (DVC):



preprocess

Очистка данных, кодирование, нормализация. Выход: `data/processed/processed.csv`

split

Разделение датасета на тренировочный и тестовый (сохраняется в DVC).

train

Обучение модели RandomForest. Выход: `models/personality_model.joblib`

evaluate

Оценка модели, логирование метрик и артефактов в MLflow. Выход: `models/classification_metrics.json`

CI/CD Pipeline

Code

Commit & Linting

Проверка качества кода:
- Flake 8
- Black
- Pylint

ML

- Валидация данных
- Автоматическое обучение Feast+Mlflow
- Регистрация модели

Unit Tests

Запуск unit – тестов через pytest:
- test_preprocess
- conftest

Build

Сборка Docker образа программного обеспечения и загрузка в Docker Hub Registry

Deploy

- Развертывание в кластер Kubernetes
- Тестирование масштабирования и автоскейлинга

Мониторинг и observability

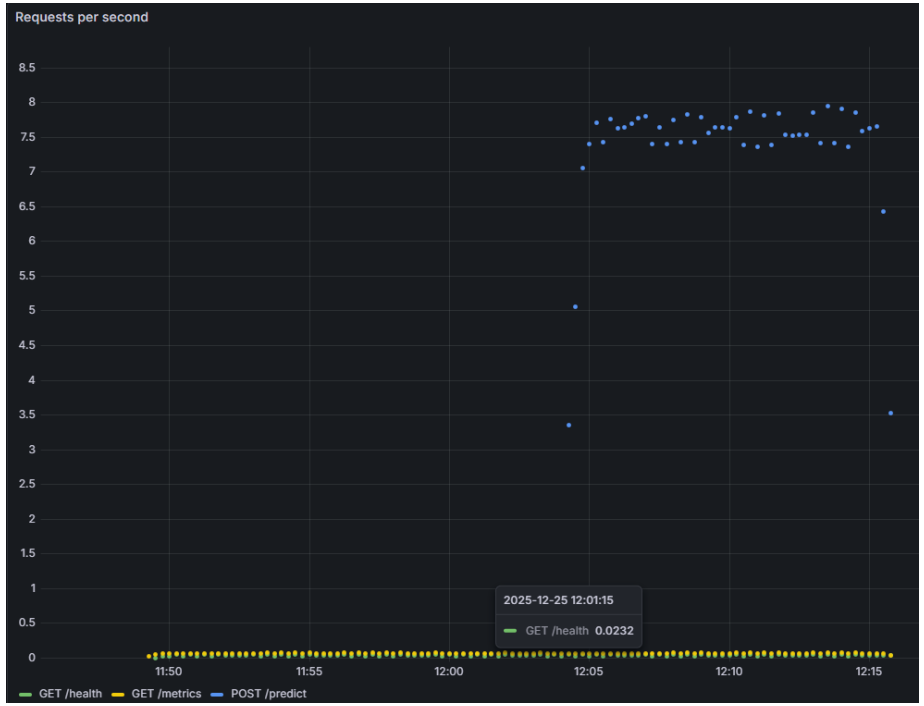
Инструменты

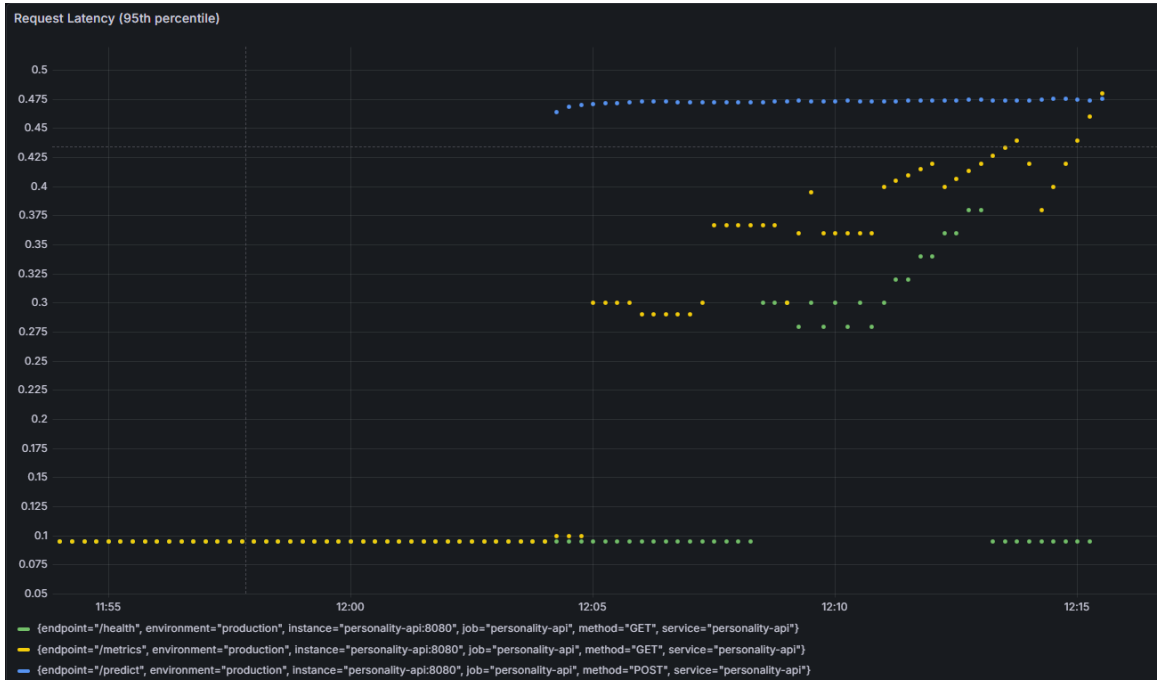
- Prometheus: сбор метрик с микросервисов
- Grafana: визуализация дашбордов

Отслеживаемые показатели

- Сколько запросов API обрабатывает в каждую секунду
- Среднюю нагрузку на систему

```
Worker 0: 300 requests, latency: 0.108s, status: 200
Worker 2: 300 requests, latency: 0.133s, status: 200
Worker 4: 300 requests, latency: 0.115s, status: 200
Worker 3: 300 requests, latency: 0.101s, status: 200
Worker 1: 320 requests, latency: 0.117s, status: 200
Worker 0: 320 requests, latency: 0.407s, status: 200
Worker 4: 320 requests, latency: 0.115s, status: 200
Worker 2: 320 requests, latency: 0.154s, status: 200
Worker 3: 320 requests, latency: 0.169s, status: 200
Worker 0: 340 requests, latency: 0.102s, status: 200
Worker 1: 340 requests, latency: 0.186s, status: 200
Worker 4: 340 requests, latency: 0.161s, status: 200
Worker 2: 340 requests, latency: 0.117s, status: 200
Worker 3: 340 requests, latency: 0.118s, status: 200
Worker 0: 360 requests, latency: 0.106s, status: 200
Worker 1: 360 requests, latency: 0.109s, status: 200
Worker 4: 360 requests, latency: 0.123s, status: 200
Worker 3: 360 requests, latency: 0.143s, status: 200
Worker 2: 360 requests, latency: 0.118s, status: 200
```





Команда

Король Алексей

Настроил Git-репозиторий с полной структурой проекта (src/, data/, models/, tests/)

Инициализировал DVC с локальным remote для версионирования данных

Реализовал DVC пайплайн с stage для предобработки данных

Добавил CI-реплику в виде базового GitHub Actions workflow

Разработал REST API на FastAPI с эндпоинтом /predict для онлайн-предсказаний

Раздоркина Дарья

Разработала unit-тесты для функций предобработки (tests/test_preprocess.py)

Создала тесты API (tests/test_api.py) Настроила pytest конфигурацию и фикстуры в conftest.py

Реализовала Airflow DAG для оркестрации полного ML пайплайна

Настроила задачи Airflow - preprocess, train, evaluate, register

Добавила мониторинг выполнения - логи, ретраи для DAG задач

Банников Дмитрий

Разработал ML пайплайн обучения (src/train.py) с RandomForestClassifier

Интегрировал MLflow для логирования экспериментов, параметров и метрик

Реализовал оценку модели с вычислением ROC AUC, precision/recall, confusion matrix

Настроил MLflow Model Registry

Создал фронтенд-интерфейс (Swagger UI/ReDoc) для тестирования API

Аврамчук Игорь

Интегрировал Docker-сборку в GitHub Actions workflow

Реализовал деплой в Minikube - создал k8s манифесты

Добавил мониторинг Prometheus - инструментировал API метриками (/metrics эндпоинт)

Развернул Grafana dashboard для визуализации метрик

Реализовал детекцию дрейфа

Настроил автоматический retraining при обнаружении дрейфа