Техническое задание ML-классификатор «HS-код + контекст → санкционный риск» Документ предназначен для разработчиков, дата-сайентистов и команды, готовящей грантовые заявки (Минцифры «Развитие ИИ», ФСИ «Старт-ИИ-1»). Уровень детализации достаточен, чтобы:

начать работу без дополнительных встреч;

обосновать ИИ-компонент перед экспертами грантов и инвесторами;

измерять прогресс и приёмку результата.

1. Постановка задачи Параметр Значение Цель Автоматически определять, подпадает ли товар под санкционные ограничения, по данным из таможенной декларации. Вход hs_code (6-значный), country_origin, country_destination, consignee_name, description[1] Выход risk_score∈[0,1], risk_level∈{low, medium, high}, violations (список режимов/списков) Критические метрики Recall ≥ 0.95 по классу High (строгий запрет) Latency ≤ 200 мс на 95-м перцентиле при 20 req/s Бизнес-драйвер Ускорить комплаенс-проверку по параллельному импорту, снизить штрафы за ошибочные поставки. Сценарии использования 1) inline-проверка брокером, 2) пакетная прогонка деклараций, 3) REST-API для сторонних ERP. [1] description = текст из графы «Описание товара» (часто на ENG/RUS).

2. Данные 2.1 Сырьевые источники Категория Источник Формат Обновление Санкционные списки OFAC SDN/NS-CSV, EU FSF (XML), UN SC (XML) CSV/XML/JSON ежедн./недел. HS-коды + описания публичный датасет Harmonized-System CSV статич. Торговая статистика UN Comtrade API (пары код ⇔ страна) CSV раз в год Позитивные/негативные примеры Исторические декларации брокеров (по договору NDA) Parquet разово

2.2 Хранение pgsql Копировать Редактировать data/ raw/ sanctions_ofac.csv sanctions_eu.xml sanctions_un.xml hs_full.csv comtrade_2024.csv processed/ train.parquet # финальный датасет sanctions_flat.parquet 2.3 Обработка Нормализация санкционных списков Скрипт normalize_sanctions.py → поля: entity, country_iso, program, list_id, last_update.

Разметка HS-кодов Euristics v0 — ключевые слова + маппинг «dual-use», «explosives» и т.д. v1 — TF-IDF + логрег, v2 — fine-tuned MiniLM (Siamese).

Слияние датасетов → train.parquet (hs_code, features_json, label, source).

3. Модель Версия Архитектура Особенности Цель метрики v0 TF-IDF(eng+rus)+LogReg baseline ≤ 1 дня ROC-AUC ≥ 0.80 v1 MiniLM-S + табличные фичи (CatBoost) multi-input, mix embeddings + one-hot ROC-AUC ≥ 0.90 v2 Cross-encoder (E5-large) + LoRA-дообучение heavy, для офф-лайн F1-macro ≥ 0.92

Выбираемый вариант для продакшена — v1 (оптимальный баланс качество/латентность).

4. API-контракт (FastAPI) python Копировать Редактировать POST /v1/check Request: { “hs_code”: “860900”, “country_origin”: “CN”, “country_destination”: “RU”, “consignee_name”: “OOO Logistika”, “description”: “Electric locomotives control module” }

Response 200: { “risk_score”: 0.87, “risk_level”: “high”, “violations”: [“OFAC_SDN”, “EU_CFSP”], “model_version”: “2025.07.14_v1” } Дополнительно: энд-пойнт /status (здоровье модели) и /metrics (Prom-стиль).

5. Инфраструктура Компонент Технология Комментарий API-сервер FastAPI + Uvicorn докер-контейнер 200 MB Фоновое обновление санкций Docker cron job раз в день скачивает новые списки, триггерит перекласс. Хранение модели MLflow Artifacts (S3-совместимый minio) версионирование и откат. CI/CD GitHub Actions → Docker Hub → Fly.io линтер → pytest → docker build → deploy.

6. Метрики и мониторинг Группа Метрика Грана-тир Качество Recall@High ≥ 0.95 F1-macro ≥ 0.90 Производительность P95 latency ≤ 200 мс Throughput 20 req/s Дрифт Pop-recall-7d алерт при −5 pp Бизнес avg. ручных проверок/декларацию ↓

50 %

Мониторинг через Prometheus + Grafana; алерты Slack / Telegram.

7. План работ и вехи День Мильстоун Артефакт D + 0 Настоящее ТЗ утверждено doc v1 D + 3 train.parquet v0 80 k строк D + 5 Baseline v0 (notebook + pickle) ROC-AUC ≥ 0.80 D + 8 API /health + /v1/check (mock) Docker image D + 10 Модель v1 (MiniLM+CatBoost), unit-тесты 80 % tag v1 D + 12 Лоад-тест, опт. latency < 200 мс отчёт JMeter D + 14 Док-пакет грантов: описание модели, диаграмма, KPI PDF

(Дни считаются с момента старта работы над модулем — соответствуют «День 13 – 14» в общем плане.)

8. Критерии приёмки Тех: юнит-тесты pytest ≥ 80 % покрытие, CI зелёный.

ML: на hold-out сэмпле (2024Q4) — Recall@High ≥ 0.95.

Prod: деплой в Fly.io, 1000 запросов wrk — P95 ≤ 200 мс, ошибок 0.

Документы: README с инструкцией, ml_spec_v0.1.md, диаграмма Mermaid, результат JMeter.

9. Риски и меры Риск Воздействие Митигирование Списки меняются ежедневно модель быстро устаревает ежедневный cron-обновитель, retrain ↑ Разметка HS ↔□ санкций шумная падает Recall расширяем ручную валидацию top-false-negatives Latency > 200 мс в v1 неприемлемо для inline-проверок fallback: кеширование эмбеддингов + Faiss

10. Ответственные Роль Имя Зона ML-лид A. Bernikov датасет, модель Back-end N. Dev FastAPI, CI/CD DevOps P. Ops Docker, Fly.io, мониторинг PM / Гранты E. Manager дедлайны, подача заявок

Примечание для грантов Документ готов в формате, который можно приложить к пунктам «Описание ИИ-метода» (Минцифры) и «Программа НИОКР» (ФСИ). В финансовом разделе грантов достаточно сослаться на KPI и инфраструктуру, перечисленные выше.