

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет имени  
академика С.П. Королева»  
(Самарский университет)

Институт информатики, математики и электроники  
Факультет информатики  
Кафедра технической кибернетики

### ЛАБОРАТОРНАЯ РАБОТА №3

по курсу  
Инженерия данных

Группа 6232

Студент \_\_\_\_\_ М.К. Макарова  
*(подпись)*

Преподаватель \_\_\_\_\_ Р.А. Парингер  
*(подпись)*

Самара 2025

Целью лабораторной работы было освоение полного цикла ML-проекта с использованием платформы ClearML, начиная с подготовки датасета, организации экспериментов и подбора гиперпараметров, заканчивая регистрацией модели и развертыванием инференс-сервиса. Нужно было реализовать FastAPI сервиса для прогноза температуры на 7 дней вперед и настройку n8n-пайплайна, который автоматически запрашивает прогноз и отправляет его в Telegram.

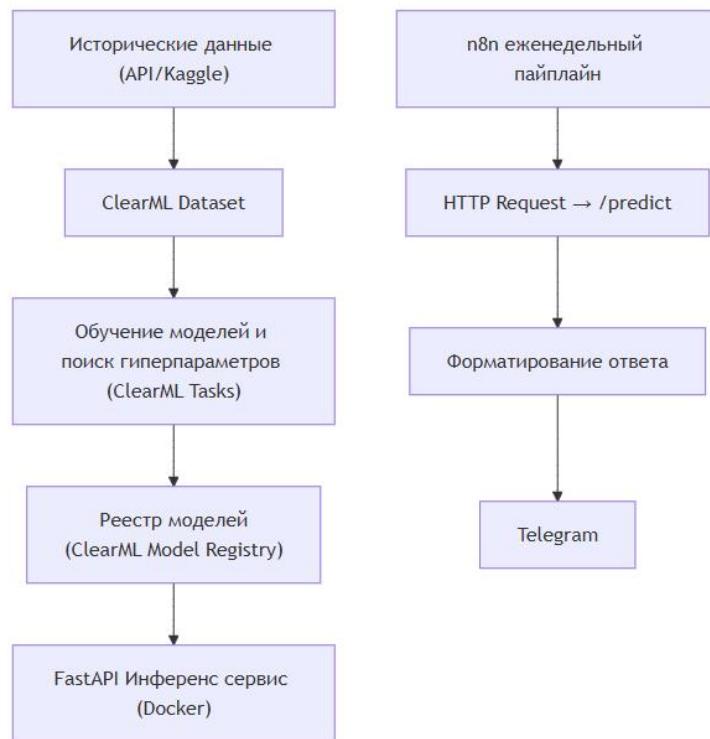


Рисунок 1 - Схема пайплайна

## Источники

Исторические данные о погоде собирались с использованием Open-Meteo API. Данные охватывают период с 01.01.2020 по 01.01.2024. В качестве целевой переменной использовалась среднесуточная температура (temperature\_2m\_mean). Данные были сохранены в CSV и зарегистрированы как ClearML Dataset.

## Архитектура и детали реализации

Подготовка данных: prepare\_dataset.py загружает исторические данные и сохраняет в CSV, который затем регистрируется в ClearML Dataset.

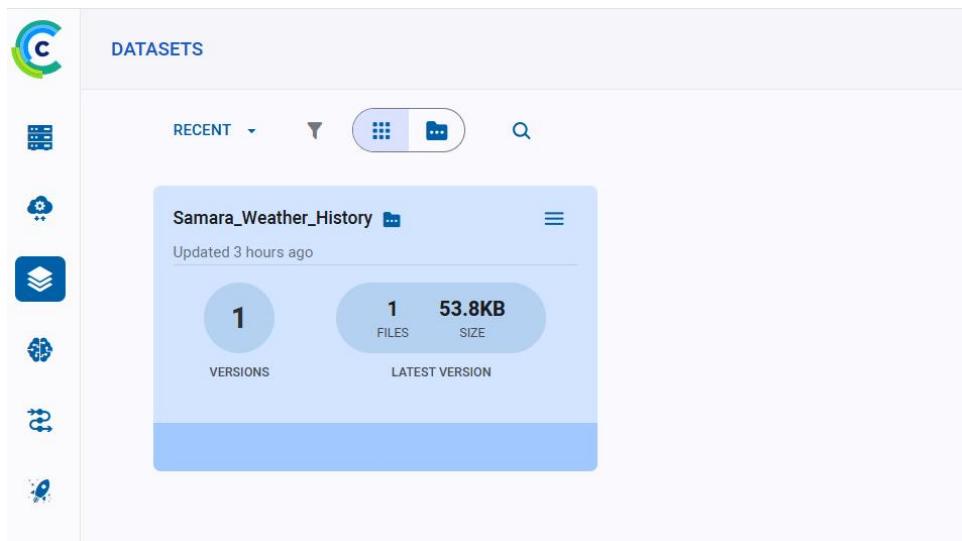


Рисунок 2 - Пример датасета в ClearML

Обучение модели: train.py извлекает данные из ClearML Dataset, выполняет feature engineering. Модель CatBoost обучается и валидируется, метрики логируются в ClearML.

HPO: hpo.py использует Optuna через ClearML HyperParameterOptimizer для поиска оптимальных гиперпараметров модели. Лучшие конфигурации публикуются в Model Registry.

Инференс-сервис: main.py реализует FastAPI сервис с эндпоинтом /predict, который принимает город и массив дат, формирует признаки и возвращает прогноз с доверительными интервалами. Также есть /health для проверки состояния сервиса.

Контейнеризация: Dockerfile создает образ с моделью и API, который развертывается в Docker, а docker-compose.yml (был взят [отсюда](#)) используется для поднятия ClearML Server и его сервисов на локальной машине.

Для автоматизации отправки прогноза погоды в Telegram был создан workflow в n8n, представленный на рисунке 3, который выполняется еженедельно. Основные шаги пайплайна следующие:

- Schedule Trigger срабатывает раз в неделю в заданное время.
- Code генерирует массив дат на следующие 7 дней и указывает город.
- HTTP Request отправляет POST-запрос к FastAPI сервису на эндпоинт /predict, передавая город и список дат. Сервис

возвращает JSON с прогнозом температуры и доверительными интервалами на каждый день.

- Code обрабатывает ответ API, формирует текстовое сообщение для Telegram.
- Telegram node отправляет сообщение в указанный чат.

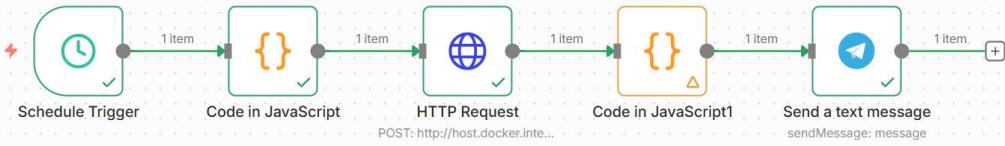


Рисунок 3 - Workflow из n8n

## Результаты

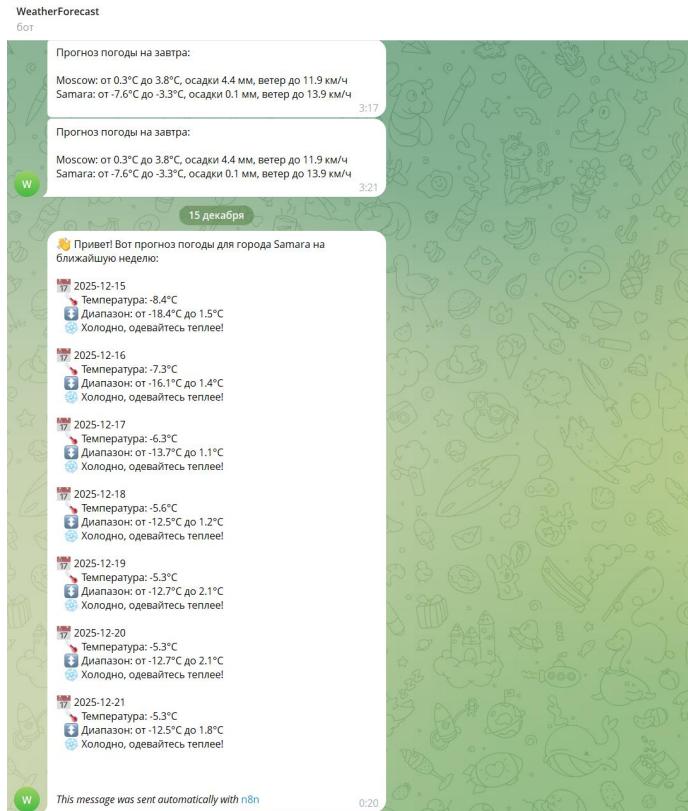


Рисунок 4 - Пример сообщения в Telegram

## Выводы

В ходе лабораторной работы был реализован полный цикл оркестрации ML-проекта с использованием ClearML, включающий автоматизированный прогноз погоды на 7 дней через n8n с последующей отправкой уведомлений в Telegram.

Возникшие трудности включали настройку признаков для модели и согласование формата входных данных между историей погоды и моделью CatBoost.

Для дальнейшего улучшения можно расширить прогноз на несколько городов, добавить новые признаки: осадки, скорость ветра и др.