

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 и кибернетики  
Кафедра технической кибернетики

ЛАБОРАТОРНАЯ РАБОТА № 3  
MLOps с использованием ClearML | недельный прогноз погоды

по курсу  
Инженерия данных

Группа 6232  
Студент

*(подпись)*

К.Р. Донец

Преподаватель

*(подпись)*

Р.А. Парингер

Самара 2025

## АРХИТЕКТУРА

На рисунке 1 представлен пайплайн из заглавного задания.



Рисунок 1 – Схема пайплайн

Кратко по архитектуре:

разбита на 2 части ClearML + FastApi server(с n8n)

**MLOps платформа для оркестрации ML экспериментов**

Зачем сделано:

1. Автоматизация обучения моделей на исторических данных
2. НРО - автоматический поиск лучших параметров модели
3. Версионирование моделей, данных и экспериментов
4. Визуализация метрик обучения в реальном времени
5. Повторяемость экспериментов

Роль в системе: Обучение и отладка моделей.

**Python веб-сервер с ML моделью**

Зачем сделано:

1. ML инференс в реальном времени (прогноз температуры)

2. REST API для интеграции с другими системами
3. Загрузка НРО-оптимизированных моделей из ClearML
4. Поддержка разных городов с разными температурными профилями

Роль в системе: делает прогнозы по запросу

## N8N

Low-code платформа для автоматизации потоков (редактируемые бизнес-процессы)

Зачем сделано:

1. Автоматизация еженедельных прогнозов (каждый понедельник в 7:00)
2. Интеграция с FastAPI для получения прогнозов
3. Форматирование сообщений для Telegram по паттерну
4. Обработка ошибок и повторные попытки
5. Визуальный мониторинг выполнения workflow'ов

Роль в системе: "Руки" системы - автоматически выполняет рутинные задачи

## ХОД РАБОТЫ

На рисунке 2 представлен скриншот запущенных контейнеров для работы ClearML, собрать его было непростой задачей, так как официальный билд у меня крашился, пришлось его подкорректировать для корректной работы.

| ● | mlops-clearml-pipeline |              |                                    | 0.87% 3 hours ago                                     |
|---|------------------------|--------------|------------------------------------|---|
| ● | clearml-webserver      | e6861f4ac5e4 | allegroai/clearml:latest           | 8080:80 ↗ 0% 3 hours ago                              |
| ● | clearml-filserver      | 8c771ceb7c78 | allegroai/clearml:latest           | 8081:8081 ↗ 0.01% 3 hours ago                         |
| ● | clearml-apiserver      | 23c1e364d5a1 | allegroai/clearml:latest           | 8008:8008 ↗ 0.07% 3 hours ago                         |
| ● | clearml-mongo          | 00fa14f17311 | mongo:6.0                          | 0.44% 3 hours ago                                     |
| ● | clearml-redis          | 5201c56ee896 | redis:8.4.0                        | 0.12% 3 hours ago                                     |
| ● | clearml-elasticsearch  | 5c17b7d73737 | elasticsearch/elasticsearch:8.13.4 | 9200:9200 ↗ 0.23% 3 hours ago                         |
| ● | Inference              |              |                                    | 0.1% 23 minutes ago                                   |
| ● | weather-forecast-n8n   | c2209755d67f | inference-weather:n8n              | 5678:5678 ↗ 0.1% 23 minutes ago<br>Show all ports (3) |

Рисунок 2 – Скриншот контейнеров для ClearML

На рисунке 3 представлен UI ClearML раздела projects, перед этим надо было инициализировать среду выполнения через clearml-init.

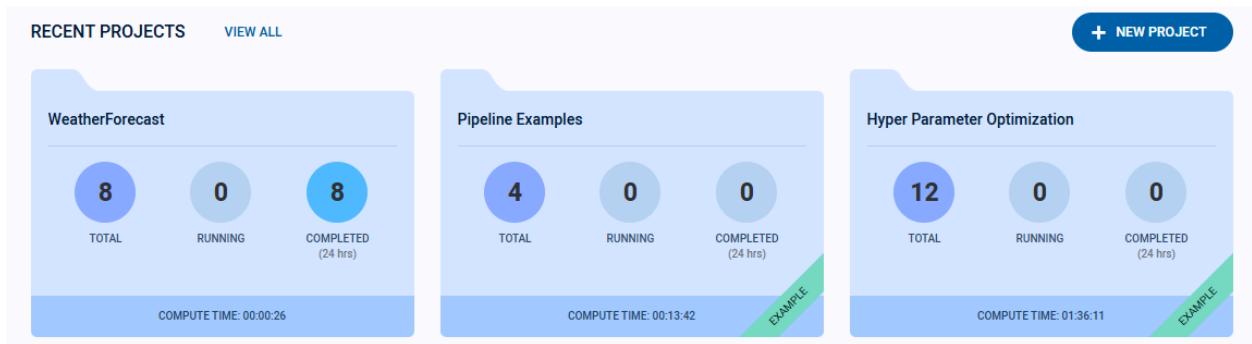


Рисунок 3 – Project ClearML

С помощью скрипта 01\_create\_dataset.py загружается датасет в ClearML. Датасет генерировался через апишку OPen Meteo, используя данные с 21 по 23 год для городов Austin, London, Tokyo и sydney. На рисунке 4 представлен загруженный датасет.

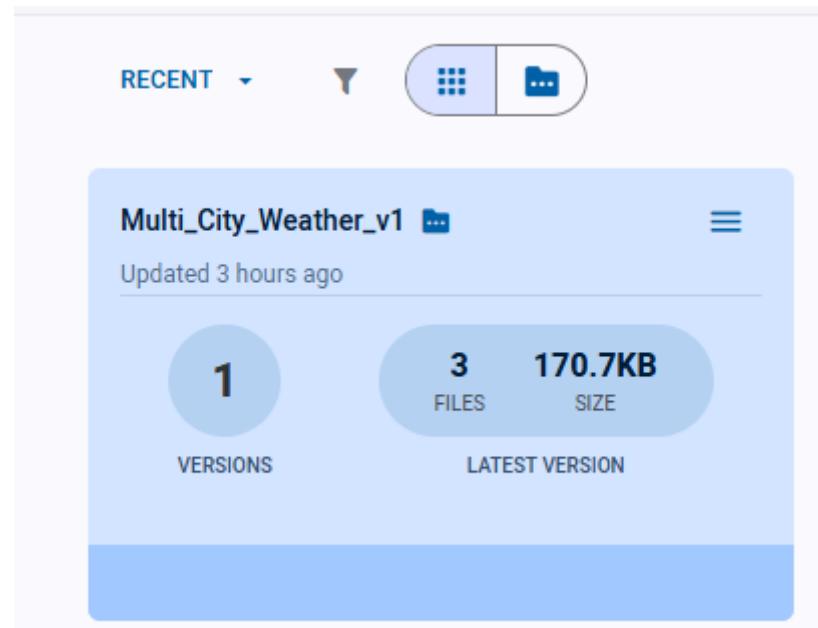


Рисунок 4 – Датасет в ClearML.

Далее был запущен скрипт 02\_train\_model.py. А после был этап НРО, чтобы затюнить модель.



Рисунок 5 – UI ClearML после hpo

После этого лучшую модель регистрируем по ID в ClearML.

Модель зарегистрирована в Model Registry: 65e30b5beafb4967b58839838817e97d

Рисунок 6 - Логи регистрации

### FAST API + n8n часть

После обучения модели был поднят сервер для выполнения предикта, поэтому /predict должен получать город и список дат, и возвращать прогнозы. Эндпоинт /predict реализован как гибридная система, где модель машинного обучения дополняется актуальными метеоданными через Open-Meteo API. Ключевая особенность - обогащение прогноза реальными историческими температурами. При получении запроса с городом и списком дат, сервис сначала обращается к Open-Meteo API за последними 30 днями исторических температур для указанного города. Эти реальные данные используются для заполнения лаговых признаков (Temperature\_C\_lag\_1d, Temperature\_C\_lag\_2d и т.д.) вместо синтетических значений.

В n8n все сильно проще чем во второй лабораторной работе. Здесь был поставлен schedule trigger который каждую неделю запускает скрипт по прогнозу погоды. Сначала генерируя даты для прогноза - а потом делая запрос к развернутому серверу. После получения ответа - редактирует сообщение по

шаблону и отправляет в указанный чат.

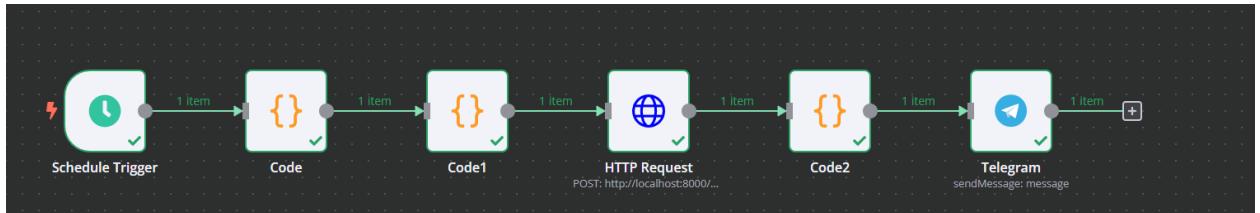


Рисунок 7 - Логи регистрации

Отправляется сообщение представленное на рисунке 8.

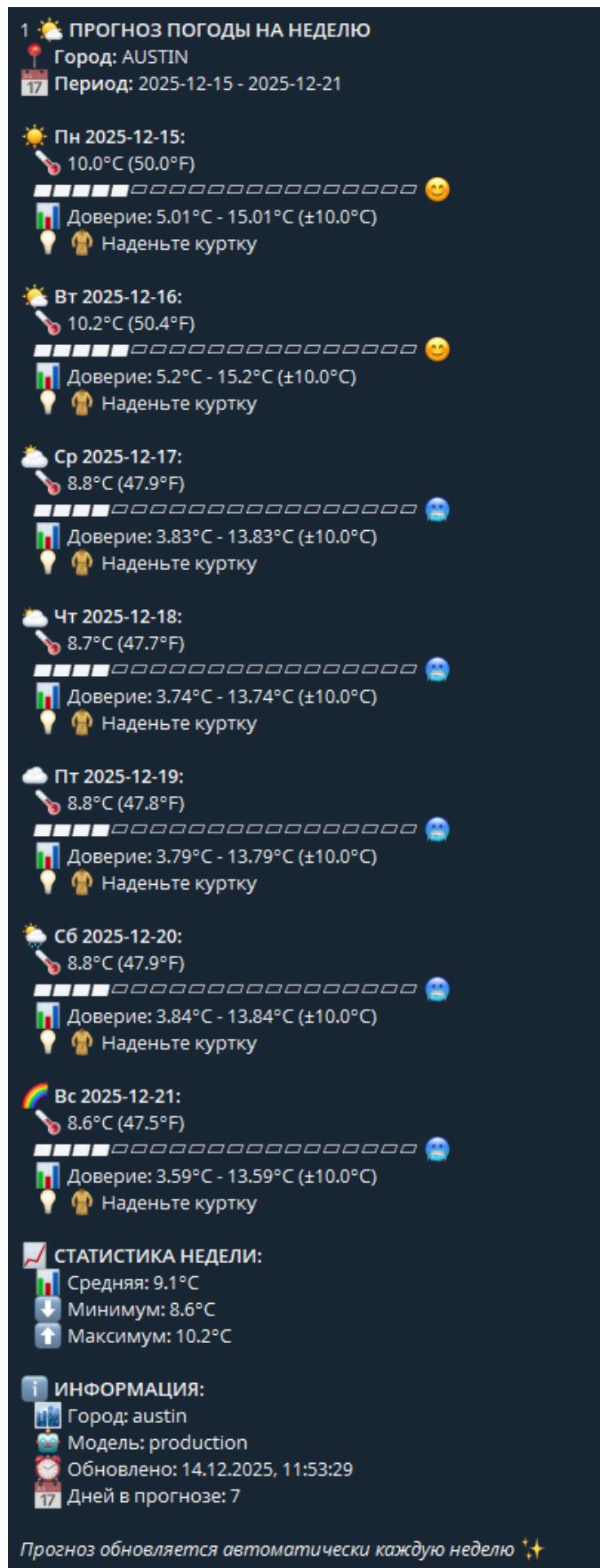


Рисунок 8 - Сообщение от бота

## **ВЫВОДЫ**

Самый тяжелый этап лабораторной собрать докеры и запустить без ошибок..... С этим возникли сложности, так как в один момент я уже начал “жонглировать” версиями mongo и redis, чтобы у меня не крашился api и web server. В конечном итоге все запустилось, но с трудом. Подключение к ClearML и все что с ним связанное стало уже не такой сложной задачей.

В ходе лабораторной работы был реализован полный ML-пайплайн в ClearML: от создания датасета и запуска экспериментов с подбором гиперпараметров до регистрации лучшей модели. Была обучена модель для предсказания погоды и добавлена в инференс-сервис на FastAPI в Docker. Также был настроен n8n-пайплайн, который раз в неделю автоматически запрашивает прогноз и присыпает его в Telegram.