

Описание проекта

Целью проекта является разработка и реализация сервиса для предсказания погоды в нескольких городах при помощи ML и TS на основе исторических данных о погоде, полученных через API. На данный момент реализовано только предсказание температуры для Москвы.

Архитектура системы

Система состоит из двух основных компонентов: серверной части, реализованной с помощью FastAPI, и клиентского приложения, реализованного с помощью Streamlit.

Серверная часть (FastAPI)

Серверная часть отвечает за обработку запросов, загрузку предобученных моделей, обучение новых моделей и предсказание погоды. Она включает в себя следующие основные компоненты:

- Модели Pydantic для представления данных запросов и ответов
- Обработка запросов с использованием соответствующих HTTP-методов
- Логирование с механизмом ротации
- Загрузка предобученной модели при запуске сервера
- Обучение модели в отдельном процессе
- Предсказание погоды с использованием активной модели
- Создание новой модели с возможностью загрузки данных из CSV-файла

Клиентская часть (Streamlit)

Клиентская часть представляет собой веб-интерфейс, который позволяет пользователю взаимодействовать с сервером. Она включает в себя следующие основные компоненты:

- Загрузка данных из CSV-файла
- Анализ данных (EDA)
- Создание новой модели с возможностью выбора гиперпараметров
- Просмотр информации о загруженных моделях
- Инференс с использованием обученной модели
- Логирование с механизмом ротации

Реализация

Серверная часть (FastAPI)

Модели Pydantic

В проекте используются следующие Pydantic-модели для представления данных запросов и ответов:

- ModelDesc: Описание модели
- SetActiveModelRequest: Запрос на установку активной модели
- SetActiveModelResponse: Ответ на установку активной модели
- FitModelResponse: Ответ после запуска обучения модели
- PredictRequest: Запрос на предсказание
- PredictResponse: Ответ с предсказаниями
- LoadNewModelResponse: Ответ после загрузки новой модели
- CSVContent: Представление содержимого CSV-файла
- LoadNewModelRequest: Запрос на загрузку новой модели

Обработка запросов

Серверная часть обрабатывает следующие запросы:

- GET /models: Получение списка загруженных моделей
- POST /set_model: Установка активной модели
- POST /fit: Запуск обучения активной модели
- POST /predict: Получение предсказаний от активной модели
- POST /load_new_model: Загрузка новой модели
- POST /upload_csv: Загрузка CSV-файла на сервер

Логирование

В проекте реализовано полноценное логирование с механизмом ротации. Логи сохраняются в папку logs/.

Загрузка предобученной модели

При запуске сервера происходит загрузка предобученной модели из файла.

Обучение модели

Обучение модели происходит в отдельном процессе. Пользователь может отправить запрос на обучение модели, передав необходимые гиперпараметры. Если обучение занимает более 10 секунд, процесс будет прерван, и пользователю будет возвращен соответствующий ответ.

Предсказание

Пользователь может отправить запрос на предсказание погоды, передав начальное время для прогноза. Сервер вернет предсказания, сделанные активной моделью.

Загрузка новой модели

Пользователь может загрузить новую модель, передав необходимые параметры, такие как путь к CSV-файлу, имя модели и количество эпох для обучения. Если указан путь к CSV-файлу, он будет загружен в базу данных, и на его основе будет создана новая модель.

Streamlit приложение (далее — приложение)

Приложение имеет 3 страницы:

1. Стартовая страница (вкладка Start)
2. Страница с EDA-частью (вкладка EDA)
3. Страница с ML-частью (вкладка ML)

Для переключения между страницами пользователь должен кликнуть на соответствующую вкладку слева.

Описание содержания и функционала страниц

1. Вкладка Start

1. Краткая навигация по приложению.

2. Вкладка EDA

1. Содержит графики временных рядов для следующих погодных данных: температура, влажность, давление, скорость ветра за последние несколько дней (на основе доступных данных), а также их прогноз на следующие сутки и некоторую описательную статистику.

3. Вкладка ML

1. Раздел «Загрузка данных»

1. Пользователь загружает последние данные о погоде в формате csv-файла для обучения моделей.
 1. Если пользователь не загрузил данные, выводится информационная панель «Пожалуйста, загрузите не пустой CSV-файл.»
2. После загрузки файла пользователю открывается окно данных со скроллингом для их предпросмотра.
3. Далее пользователь нажимает кнопку «Сохранить загруженные данные».
 1. Файл с последними данными достаточно загрузить 1 раз.
 2. Если пользователь не нажал кнопку, приложение будет обучать новые модели на дефолтных данных (до

20.09.2024).

2. Раздел «Создание нового класса модели»

1. Пользователь вручную вводит id и гиперпараметры новой модели, которую он далее хочет обучить, и нажимает кнопку «Создать новый класс для модели».
2. Если пользователь не ввел id или не нажал кнопку, приложение будет использовать модель по умолчанию в следующих пунктах.
3. Если гиперпараметры не введены, то используются гиперпараметры по умолчанию.
 1. На данном этапе работы в бэкенде фактически реализован только учет параметра epochs.

3. Раздел «Обучение модели»

1. В разделе описаны параметры и кривая обучения предобученной LSTM-модели.

4. Раздел «Загруженные модели»

1. При нажатии на кнопку «Показать все загруженные модели» приложение выводит список всех загруженных и обученных ML-моделей.

5. Раздел «Установка активной модели»

1. Пользователь вручную вводит id модели, которая далее будет обучаться и по которой будет строиться прогноз, и нажимает кнопку «Установить активную модель».
2. Если пользователь не ввел id или не нажал кнопку, приложение будет использовать модель по умолчанию в следующих пунктах.

6. Раздел «Обучение активной модели»

1. Пользователь нажимает кнопку «Обучить активную модель».
2. При успешном обучении модели приложение выводит сообщение «Модель {id} обучена»

7. Раздел «Прогноз активной модели»

1. Пользователь вручную вводит дату и время, на которое строится прогноз, и горизонт прогнозирования.
 1. Если дата и время не введены, приложение выводит ошибку.
2. Если горизонт не выбран, то используются горизонт по умолчанию.
3. Пользователь нажимает кнопку «Показать прогноз температуры».
4. При успешном получении прогноза, приложение выводит на экран таблицу с почасовым предсказанием и график температуры за последние несколько дней и прогноз.