

## Практическая работа

**Тема: Поиск аномалий вручную и с помощью ИИ через n8n и Ollama**

### Цель работы

Научиться:

- 1 находить аномалии в данных вручную
  - 2 анализировать таблицы и выявлять подозрительные записи
  - 3 создавать workflow в n8n
  - 4 подключать локальную ИИ-модель Ollama через HTTP Request
  - 5 передавать данные из сайта в n8n и получать ответ обратно
  - 6 сравнивать результат студента, результат ИИ и правильный ответ
- 

### Суть работы

В этой работе студент выполняет анализ данных в два этапа.

#### Этап 1

Студент самостоятельно ищет аномалии на сайте

#### Этап 2

Студент собирает workflow в n8n, который:

- 1 принимает данные с сайта
- 2 отправляет их в локальную Ollama
- 3 получает ответ от ИИ
- 4 возвращает результат обратно на сайт

После этого сайт показывает:

- 1 что нашёл студент
  - 2 что нашёл ИИ
  - 3 какой ответ является правильным
- 

### Что такое аномалия

Аномалия — это запись, которая:

- 1 сильно выбивается из общей картины
- 2 выглядит нелогично
- 3 содержит ошибку ввода
- 4 может указывать на сбой
- 5 может быть подозрительной или необычной

Примеры:

- 1 высокий балл при почти нулевой посещаемости
  - 2 очень дорогой заказ при одном товаре
  - 3 резкий скачок температуры оборудования
  - 4 отрицательный доход
  - 5 возраст 200 и более лет
-

## Что понадобится для работы

### Программное обеспечение

- 1 локальная папка с сайтом
- 2 браузер
- 3 n8n
- 4 Ollama
- 5 загруженная локальная модель в Ollama

### Что должно быть готово заранее

- 1 студенту выдана папка с сайтом
  - 2 у студента установлен n8n
  - 3 у студента установлена Ollama
  - 4 в Ollama загружена модель, например llama3.1
- 

## Общая структура работы

### Часть 1 Работа на сайте

- 1 открыть сайт
- 2 изучить таблицу
- 3 вручную отметить аномалии
- 4 написать пояснения

### Часть 2 Работа в n8n

- 1 создать workflow
- 2 добавить Webhook
- 3 добавить HTTP Request
- 4 подключить локальную Ollama
- 5 разобрать ответ через Code
- 6 вернуть результат через Respond to Webhook

### Часть 3 Проверка

- 1 вставить webhook в сайт
  - 2 отправить данные в ИИ
  - 3 получить результат
  - 4 показать сравнение
  - 5 сделать вывод
- 

## Часть 1 Работа с сайтом

### Шаг 1 Открытие сайта

- 1 Откройте папку с сайтом
- 2 Найдите файл index.html
- 3 Откройте его в браузере
- 4 Убедитесь, что страница загрузилась

### Шаг 2 Заполнение данных

Если на сайте есть поля:

- 1 введите ФИО
- 2 введите группу
- 3 пока не заполняйте webhook, если workflow ещё не создан

### **Шаг 3 Выбор сценария**

На сайте может быть один из сценариев:

- 1 оценки и посещаемость
- 2 подозрительные заказы
- 3 датчики оборудования
- 4 контроль качества данных

### **Шаг 4 Изучение таблицы**

Внимательно изучите все строки:

- 1 просмотрите значения по всем столбцам
- 2 найдите необычные числа
- 3 обратите внимание на резкие выбросы
- 4 сравните строки между собой
- 5 ищите нелогичные сочетания

### **Шаг 5 Ручной поиск аномалий**

- 1 Отметьте строки, которые считаете аномальными
- 2 Для каждой строки напишите пояснение
- 3 Не отмечайте слишком много строк без причины

### **Как правильно писать пояснения**

Плохо:

- 1 ошибка
- 2 странно
- 3 аномалия

Хорошо:

- 1 слишком высокий балл при крайне низкой посещаемости
- 2 сумма заказа слишком большая для одного товара
- 3 температура резко выше остальных значений
- 4 возраст нереалистично высокий
- 5 доход не может быть отрицательным

### **Что важно**

На этом этапе:

- 1 правильный ответ скрыт
- 2 сравнение скрыто
- 3 студент работает самостоятельно
- 4 ИИ ещё не запустился

---

## **Часть 2 Создание workflow в n8n**

### **Логика workflow**

Workflow должен работать так:

- 1 сайт отправляет JSON в Webhook
  - 2 Webhook принимает данные
  - 3 HTTP Request отправляет запрос в локальную Ollama
  - 4 Ollama анализирует таблицу
  - 5 Code преобразует ответ к нужному JSON
  - 6 Respond to Webhook возвращает результат на сайт
- 

### Итоговая структура workflow

Webhook -> HTTP Request -> Code -> Respond to Webhook

---

### Шаг 1 Создание нового workflow

- 1 Откройте n8n
  - 2 Создайте новый workflow
  - 3 Назовите его, например:  
Поиск аномалий через Ollama
- 

### Шаг 2 Добавление узла Webhook

- 1 Нажмите Add node
- 2 Найдите узел Webhook
- 3 Добавьте его на рабочую область

### Настройка Webhook

Укажите:

- 1 Method — POST
- 2 Path — anomaly-check

В результате вы получите адрес webhook.

Пример:

`http://127.0.0.1:5678/webhook/anomaly-check`

Именно этот адрес потом нужно будет вставить в сайт.

---

### Шаг 3 Понимание входных данных

Сайт будет отправлять JSON примерно такого вида:

```
{
  "scenario": "grades",
  "studentName": "Иванов И.И.",
  "groupName": "ИСП-21",
  "variantSeed": 1001,
  "rows": [
    {
      "id": 1,
```

```
"student": "Иванов И.",
"avgScore": 4.8,
"attendance": 96,
"lateCount": 1
},
{
  "id": 2,
  "student": "Петров А.",
  "avgScore": 5.0,
  "attendance": 12,
  "lateCount": 0
}
]
```

Главное поле для анализа:

1 rows

---

#### Шаг 4 Добавление узла HTTP Request

- 1 После Webhook нажмите Add node
  - 2 Найдите узел HTTP Request
  - 3 Добавьте его в workflow
  - 4 Соедините Webhook с HTTP Request
- 

#### Шаг 5 Настройка HTTP Request для Ollama

##### Основные параметры

Укажите:

- 1 Method — POST
- 2 URL — `http://localhost:11434/api/generate`

##### Формат отправки

- 1 включите отправку тела запроса
- 2 выберите JSON

##### Тело запроса

В body нужно передать модель, промпт и данные.

Используйте такой JSON:

```
{
  "model": "llama3.1",
  "prompt": "Ты анализируешь учебный набор данных и ищешь аномальные записи.\n\nПравила:\n1. Найди строки, которые резко выбиваются из общей картины\n2. Учитывай нелогичные сочетания значений\n3. Не помечай строки без причины\n4. Для каждой найденной аномалии укажи id и краткое объяснение\n5. Верни ответ строго в JSON\n6. Не добавляй никакой текст до и после JSON\n\nФормат ответа:\n{\n  \"aiAnomalies\": [\n    {\n      \"id\": 123,\n      \"reason\": \"краткое объяснение\"\n    }\n  ]\n}\n\nДанные:\n${json.body.rows}"
```

```
"stream": false,
"format": {
  "type": "object",
  "properties": {
    "aiAnomalies": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "integer" },
          "reason": { "type": "string" }
        },
        "required": ["id", "reason"]
      }
    }
  },
  "required": ["aiAnomalies"]
}
```

---

#### Что делает этот запрос

- 1 отправляет данные в локальную модель
  - 2 просит её найти аномалии
  - 3 требует вернуть ответ строго в JSON
  - 4 дополнительно ограничивает формат через JSON-схему
- 

#### Шаг 6 Проверка ответа Ollama

Ollama должна вернуть объект, где есть поле с результатом генерации.

Обычно внутри ответа будет текст, содержащий JSON.

Пример того, что модель должна выдать:

```
{
  "aiAnomalies": [
    {
      "id": 2,
      "reason": "Слишком высокий балл при крайне низкой посещаемости"
    },
    {
      "id": 7,
      "reason": "Нереалистично большое число опозданий"
    }
  ]
}
```

---

#### Шаг 7 Добавление узла Code

- 1 После HTTP Request добавьте узел Code
- 2 Соедините HTTP Request с Code

Этот узел нужен для того, чтобы:

- 1 взять текст ответа модели
- 2 попытаться преобразовать его в JSON
- 3 передать дальше уже готовый объект

### Код для узла Code

```
const raw =
  $json.response ||
  $json.text ||
  $json.output ||
  "";

let parsed;

try {
  parsed = typeof raw === 'string' ? JSON.parse(raw) : raw;
} catch (e) {
  parsed = { aiAnomalies: [] };
}

return [
  {
    json: parsed
  }
];
```

---

### Шаг 8 Добавление узла Respond to Webhook

- 1 После Code добавьте узел Respond to Webhook
- 2 Соедините Code с Respond to Webhook

#### Что должен вернуть узел

Он должен вернуть JSON из Code на сайт.

Итоговый ответ должен иметь вид:

```
{
  "aiAnomalies": [
    {
      "id": 2,
      "reason": "Слишком высокий балл при крайне низкой посещаемости"
    }
  ]
}
```

---

### Шаг 9 Проверка структуры workflow

В итоге у вас должна быть такая цепочка:

Webhook -> HTTP Request -> Code -> Respond to Webhook

---

### **Часть 3 Подключение workflow к сайту**

#### **Шаг 1 Запуск workflow**

- 1 Сохраните workflow
- 2 Активируйте его
- 3 Убедитесь, что webhook доступен
- 4 Скопируйте webhook URL

#### **Шаг 2 Вставка webhook в сайт**

- 1 Перейдите в сайт
- 2 Найдите поле для webhook
- 3 Вставьте туда адрес из п8п
- 4 Убедитесь, что адрес вставлен без ошибок

#### **Шаг 3 Отправка данных в ИИ**

- 1 Нажмите кнопку отправки данных в ИИ
  - 2 Сайт отправит таблицу в п8п
  - 3 п8п отправит запрос в Ollama
  - 4 Ollama проанализирует данные
  - 5 Ответ вернётся обратно на сайт
- 

### **Часть 4 Анализ результата**

После успешного выполнения будут доступны три результата:

#### **1 Результат студента**

Это строки, которые студент отметил вручную

#### **2 Результат ИИ**

Это строки, которые выделила Ollama

#### **3 Правильный ответ**

Это эталонные аномалии, заранее заложенные в варианте

---

### **Что нужно сравнить**

После открытия сравнения проанализируйте:

- 1 какие строки нашёл студент
- 2 какие строки нашёл ИИ
- 3 какие строки были действительно аномальными
- 4 что студент пропустил
- 5 где ИИ ошибся
- 6 кто оказался точнее

---

## Как оформить вывод

Пример вывода:

В ходе практической работы был выполнен ручной поиск аномалий в наборе данных. После этого в системе n8n был собран workflow, который принимал данные с сайта через Webhook, отправлял их в локальную модель Ollama через HTTP Request, обрабатывал ответ и возвращал результат обратно. После сравнения ручного анализа, ответа ИИ и эталонного ответа было установлено, что часть аномалий была найдена одинаково, однако некоторые записи были пропущены или отмечены ошибочно. В результате работы были закреплены навыки анализа данных, визуального программирования и использования локального ИИ для автоматизации поиска аномалий.

---

## Что студент должен сдать

- 1 рабочий workflow в n8n
- 2 рабочий webhook
- 3 заполненный сайт с ручным анализом
- 4 результат ответа ИИ
- 5 итоговый вывод

Если нужен отчёт, добавить:

- 1 тему работы
  - 2 цель работы
  - 3 краткое описание выполненных действий
  - 4 скрин сайта до отправки в ИИ
  - 5 скрин workflow в n8n
  - 6 скрин результата сравнения
  - 7 вывод
- 

## Типовые ошибки

### Ошибка 1

Отмечено слишком много строк

Причина: студент помечает почти всё подряд

Что делать: искать только реальные выбросы и нелогичные сочетания

### Ошибка 2

Пояснения слишком короткие

Причина: студент пишет только одно слово

Что делать: объяснять, почему строка подозрительна

### Ошибка 3

Webhook настроен не на POST

Что делать: проверить метод в Webhook

### Ошибка 4

Неправильный URL в HTTP Request

Что делать: проверить адрес `http://localhost:11434/api/generate`

#### **Ошибка 5**

Ollama не запущена

Что делать: убедиться, что модель доступна локально

#### **Ошибка 6**

Модель отвечает обычным текстом, а не JSON

Что делать:

- 1 сделать промпт жёстче
- 2 оставить `stream: false`
- 3 использовать `format`
- 4 оставить узел `Code`

#### **Ошибка 7**

Сайт не получает ответ

Что делать:

- 1 проверить `webhook URL`
  - 2 проверить активность `workflow`
  - 3 проверить корректность JSON
  - 4 посмотреть, нет ли ошибки в `Code node`
- 

#### **Контрольные вопросы**

- 1 Что такое аномалия в данных
- 2 По каким признакам можно обнаружить аномальную строку
- 3 Почему перед автоматическим анализом полезно выполнить ручной поиск
- 4 Для чего нужен узел `Webhook`
- 5 Для чего нужен узел `HTTP Request`
- 6 Почему Ollama вызывается по локальному адресу
- 7 Зачем нужен узел `Code`
- 8 Почему ответ ИИ должен быть в JSON
- 9 Чем отличаются результат студента, результат ИИ и эталон
- 10 Какие преимущества даёт автоматизированный поиск аномалий