

АтомикХак 3.0

Кейс: ИИ-анализатор журналов событий приложений

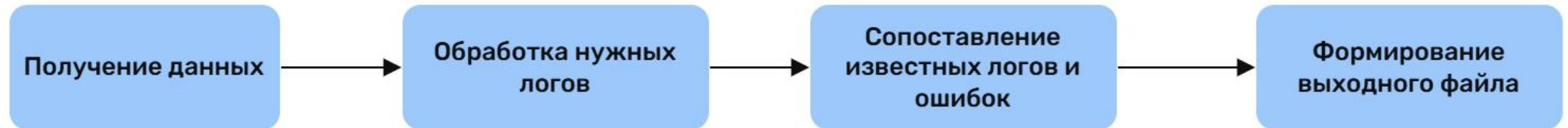
Команда: Black Lotus

Задачи решения

- Сбор и анализ логов со всей инфраструктуры.
- Выявление аномалий и потенциальных проблем.
- Поиск и локализация корневых причин с указанием источника в логах.
- Предоставление отчета в заданном формате.

Первый вариант решения

Первый вариант решения основывался на разработке примитивного алгоритма поиска с использованием словаря.



Разбор первого варианта

Плюсы:

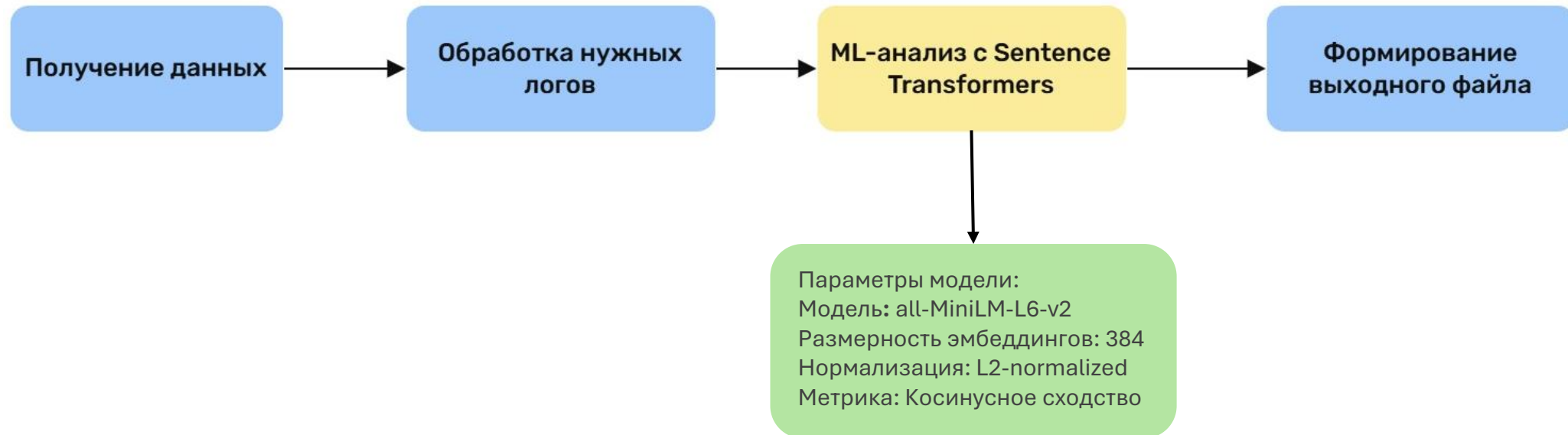
- Быстро работает
- Легок в реализации

Минусы:

- Нет обработки новых проблем и аномалий
- Не устойчив к ошибкам в логах

Второй вариант решения

Во втором варианте использовался ML-анализ с Sentence Transformers.



Разбор второго варианта

Плюсы:

- Устойчивость к вариациям текста
- Находит похожие по смыслу сообщения, даже если формулировка отличается
- Не требует точного совпадения строк

Минусы:

- Работает дольше

На валидационной выборке показатель 32,2 сек, в то время как у первого варианта 3.35

Проверка

Для проверки корректности работы системы было внесено изменение в файл с логами.

*Первый вариант
решения*

2 rows 2 rows x 5 cols							Edit in Data Wrangler				
	ID аномалии	ID проблемы	Файл с проблемой	№ строки	Строка из лога						
0	267	34	web_server_log.txt	27744	Index corruption on table						
1	271	34	web_server_log.txt	27744	Index corruption on table						

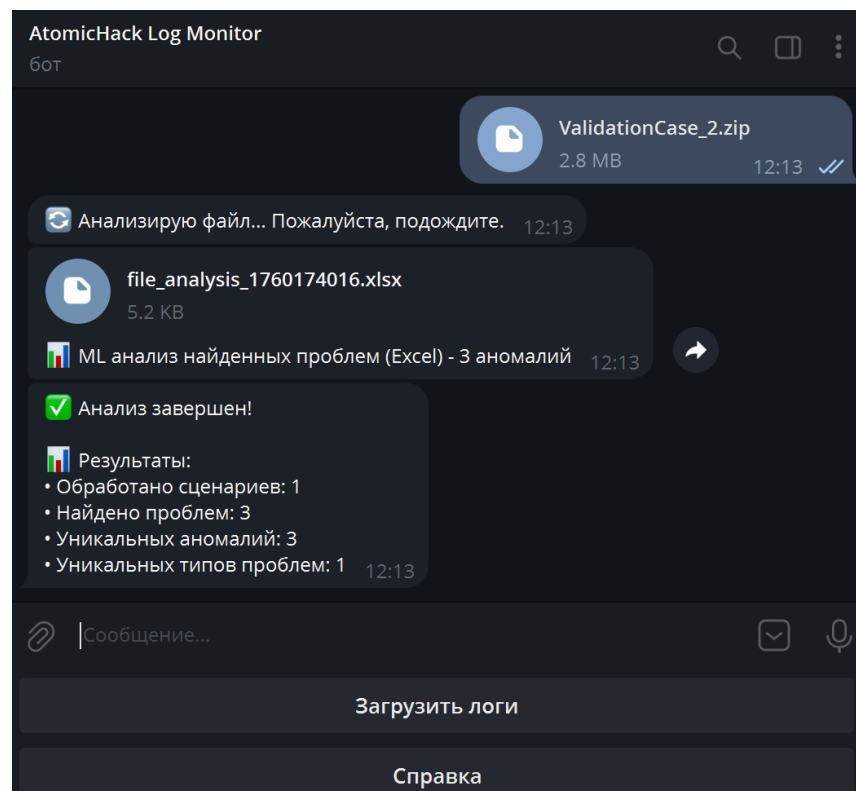
*Второй вариант
решения*

3 rows 3 rows x 7 cols							Edit in Data Wrangler				
	ID аномалии	Аномалия	Уверенность	ID проблемы	Файл с проблемой						
0	267	Slow query from index issue	1.000	34	web_server_log.txt						
1	271	Query timeout warning	1.000	34	web_server_log.txt						
2	274	Application crash on index table	0.931	34	web_server_log.txt						

Как видно первый вариант решения не позволяет найти новую аномалию, а второй справляется с этим.

Взаимодействие с системой

Для взаимодействия с разработанной системой был реализован телеграм-бот.



Docker - интеграция за 3 команды

Преимущества:

- Изоляция окружения,
- Воспроизводимость,
- Легкая миграция,
- Контроль ресурсов

ШАГ 1: Подготовка
git clone repository
echo BOT_TOKEN > .env
Время: 15 секунд

ШАГ 2: Docker Compose
docker-compose up -d
- Сборка образа Python 3.13
- Установка зависимостей
- Загрузка ML модели all-MiniLM-L6-v2
- Настройка volumes
Время: 3-5 минут

ШАГ 3: Готово к работе
Telegram Bot активен
Volumes смонтированы
Автоперезапуск настроен
Общее время: 5-10 минут

Спасибо за внимание

Команда: Black Lotus