Инференс моделей

Теория и фреймворки

Евгений Лагуткин

Руководитель отдела «Антифрод Аналитика»

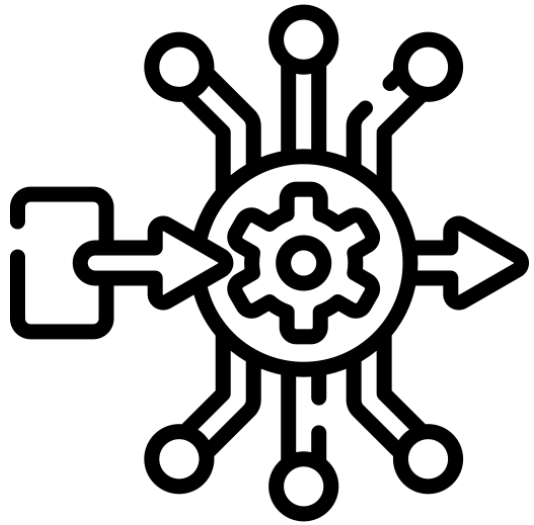
Дирекция «Машинное отделение»

Что будет

1. Общая информация
2. Основные типы
3. Сравнение типов инференса
4. Фреймворки

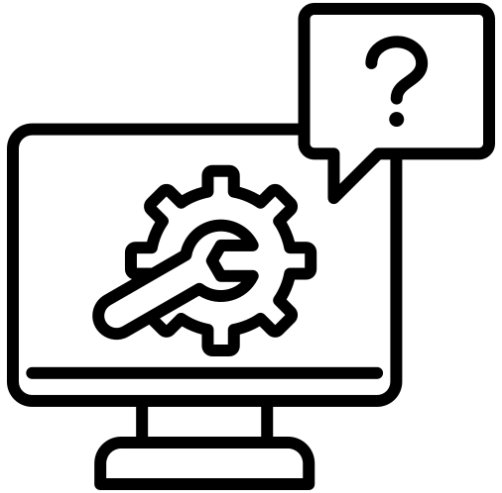
Определение

Инференс (или вывод) в контексте моделей машинного обучения (ML) относится к процессу применения обученной модели к новым данным для получения предсказаний или выводов.



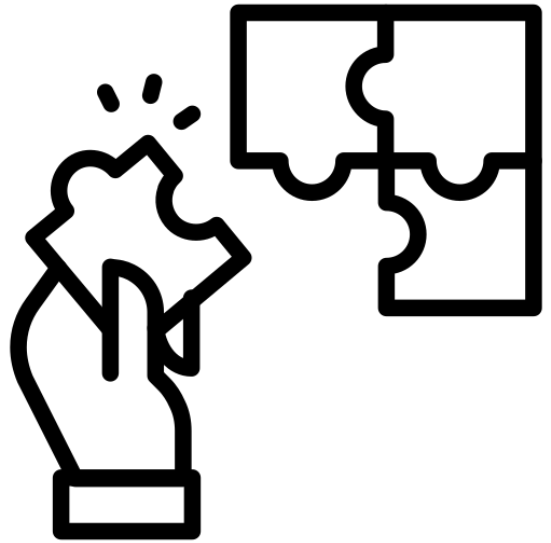
От чего зависит

- Как построена модель
- Железо
- Оптимизация



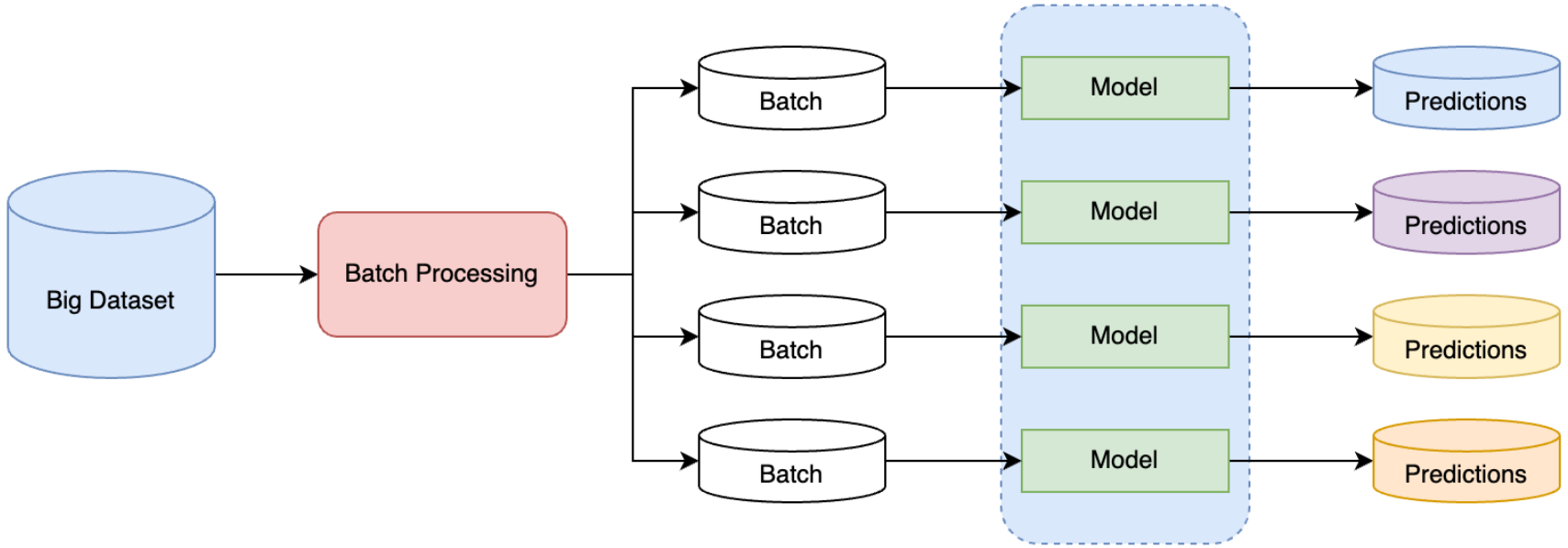
Определение. Итог

Инференс - ключевой результат внедрения модели



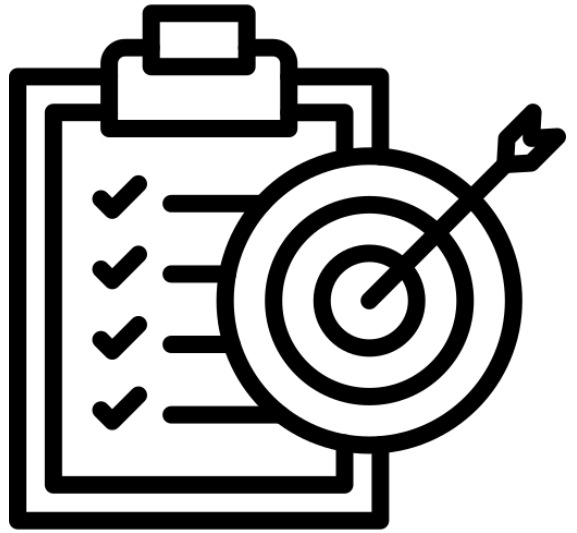
Пакетный инференс

Batch Inference — это процесс, при котором модель машинного обучения применяется к большому объему накопленных данных за один раз. В отличие от инференса в реальном времени, пакетный инференс не обрабатывает запросы мгновенно, а скорее выполняет обработку данных в фоновом режиме.



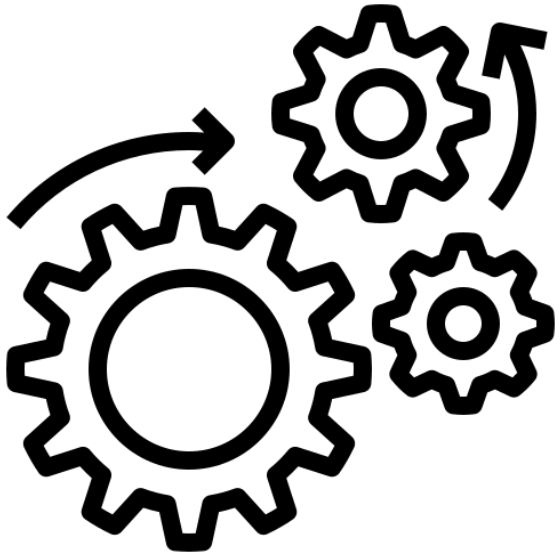
Пакетный инференс. Цели

- Эффективность
- Масштабируемость
- Снижение Стоимости



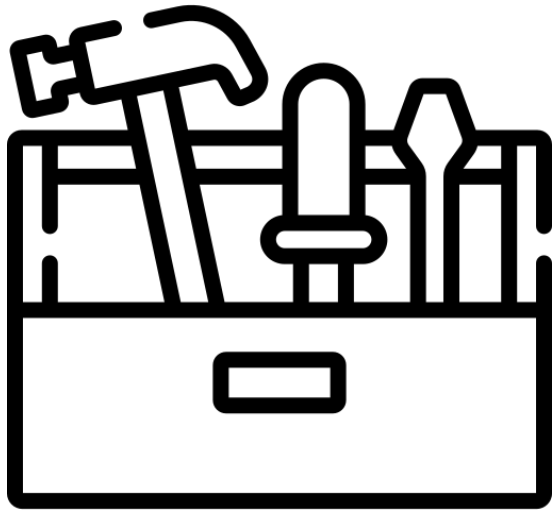
Пакетный инференс. Процесс

- Сбор Данных
- Предобработка
- Загрузка Модели
- Инференс
- Постобработка
- Хранение Результатов

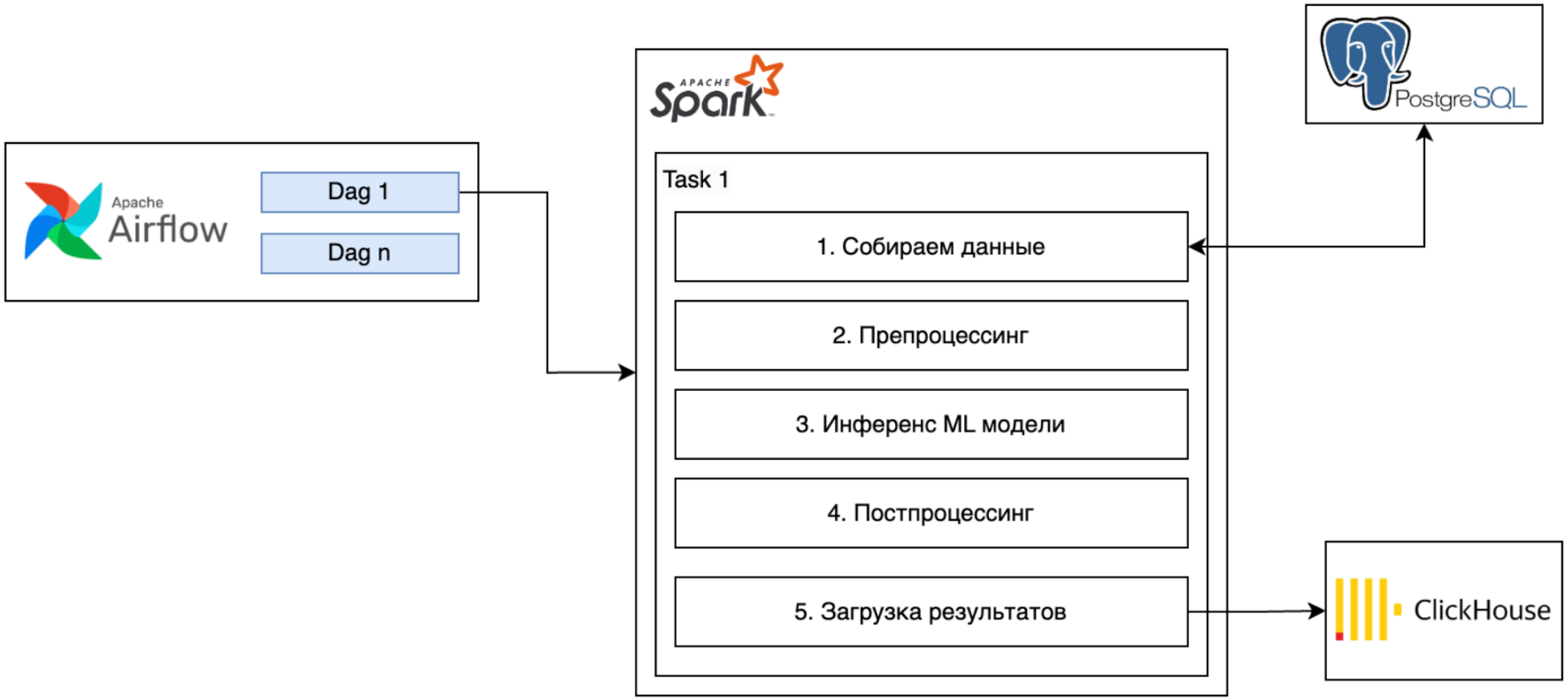


Пакетный инференс. Инструменты

- Apache Spark
- Amazon S3 и AWS Batch
- Ray Tasks
- Python Libraries (Pandas, NumPy)

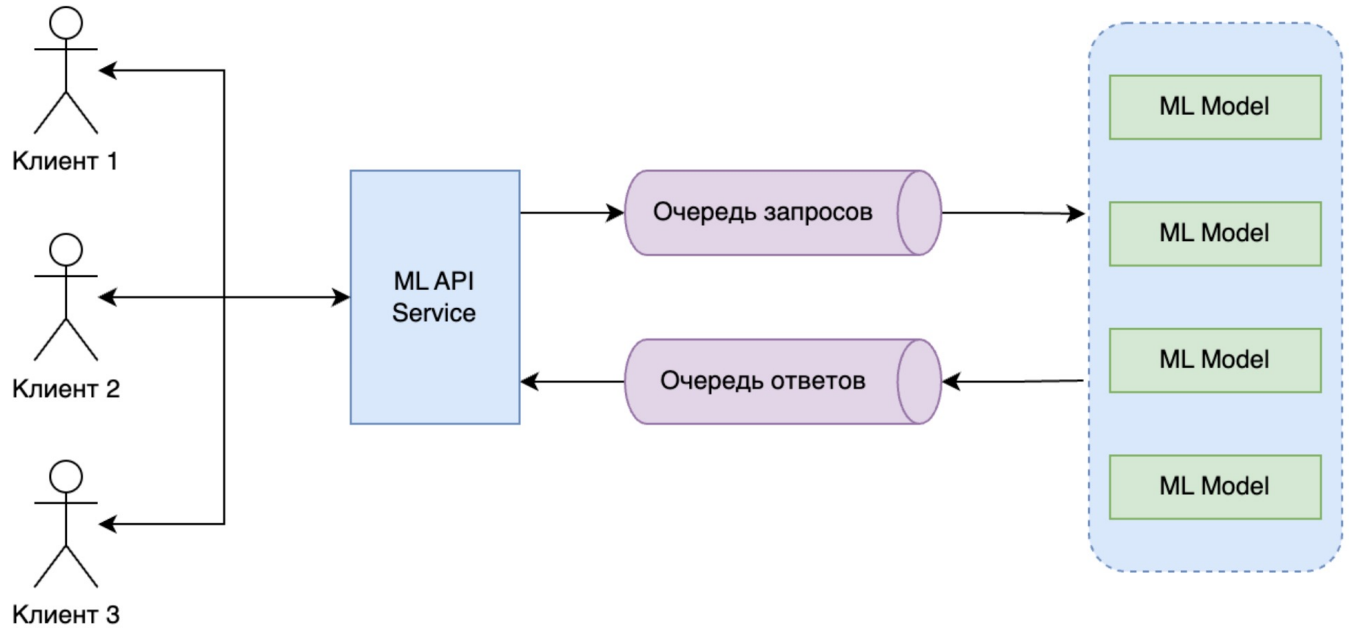


Пакетный инференс. Пример



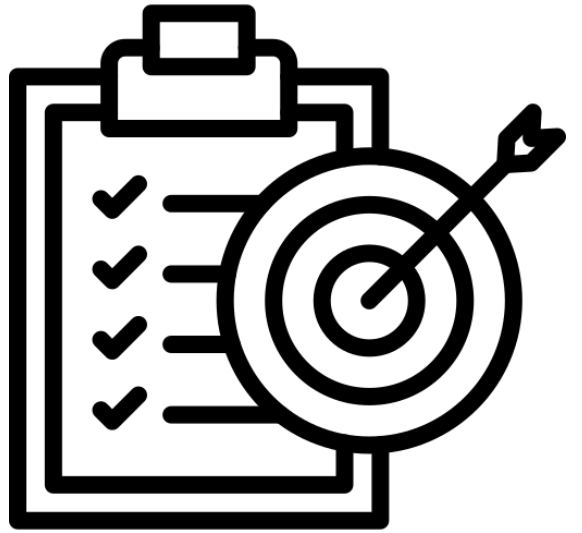
Асинхронный инференс

Asynchronous Inference (Асинхронный Инференс) в контексте моделей машинного обучения (ML) представляет собой подход, при котором запросы на инференс обрабатываются независимо и не требуют мгновенного ответа. Это позволяет системе обрабатывать другие задачи во время ожидания результата инференса.



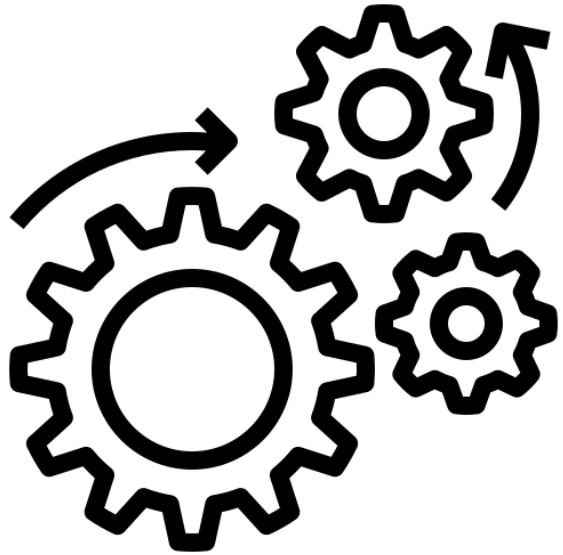
Асинхронный инференс. Цели

Минимизирует затраты, поскольку не требует постоянно активных вычислительных ресурсов



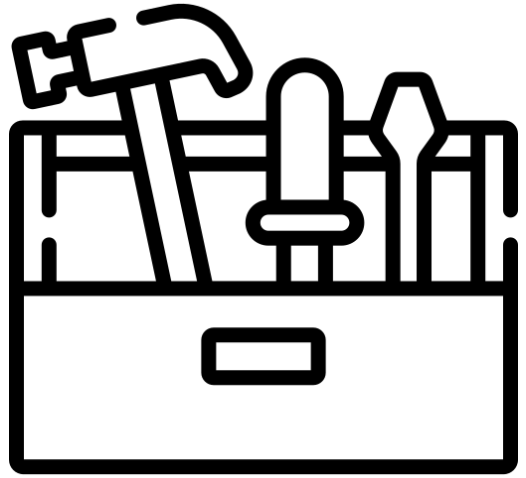
Асинхронный инференс. Процесс

- Отправка запроса
- Обработка в Фоне
- Возврат результат

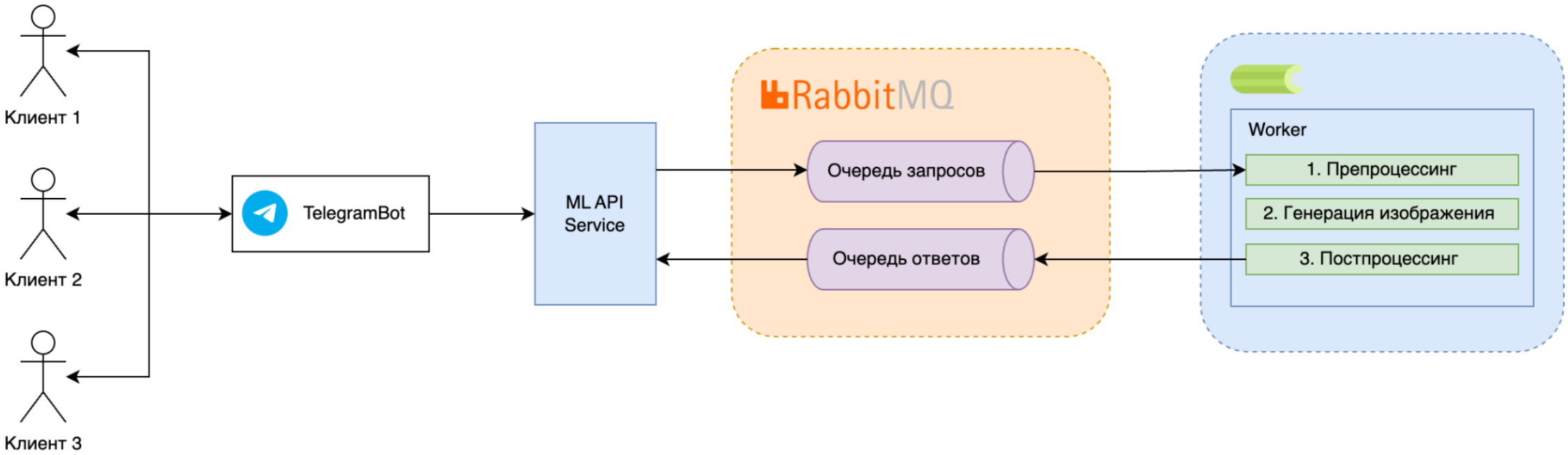


Асинхронный инференс. Инструменты

- Системы Очередей Сообщений Amazon S3 и AWS Batch
- Фреймворки для Asynchronous Processing
- Облачные сервисы

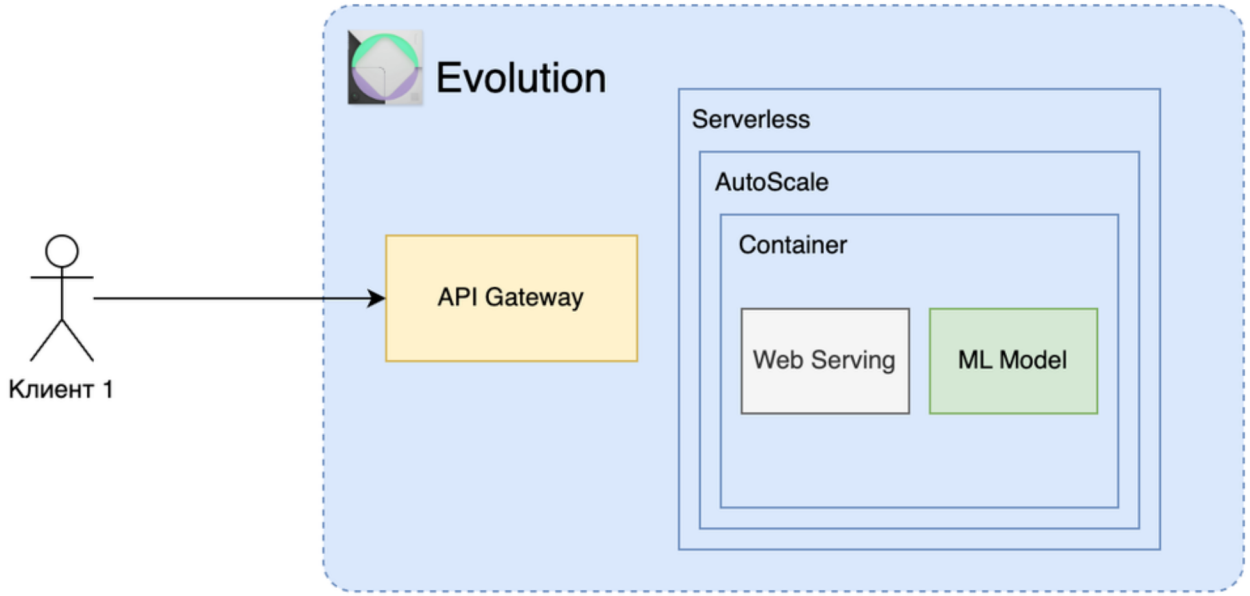


Асинхронный инференс. Пример



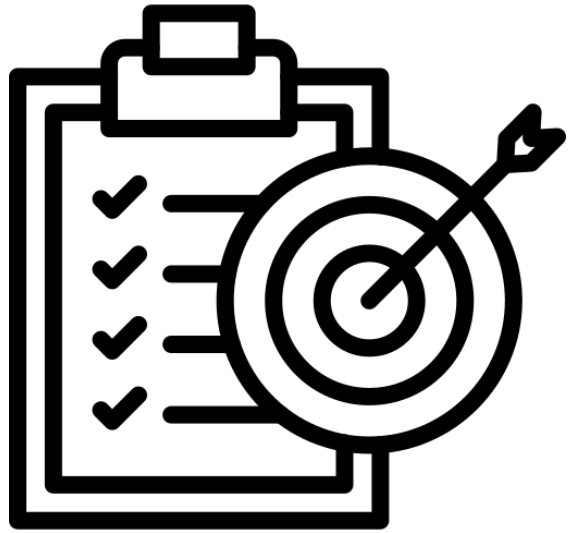
Бессерверный Инференс

Serverless Inference (Бессерверный Инференс) в контексте моделей машинного обучения (ML) относится к методу выполнения инференса, при котором не требуется постоянно работающий сервер. Вместо этого вычислительные ресурсы выделяются динамически для обработки каждого запроса на инференс



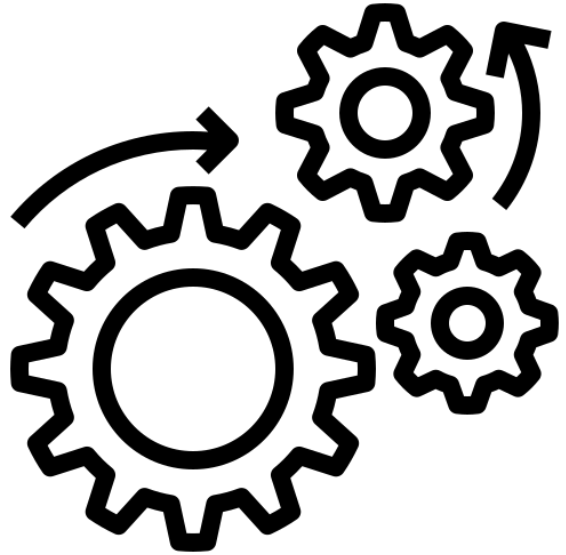
Бессерверный Инференс. Цели

- Уменьшение Затрат
- Эластичность и Масштабируемость
- Простота Развертывания



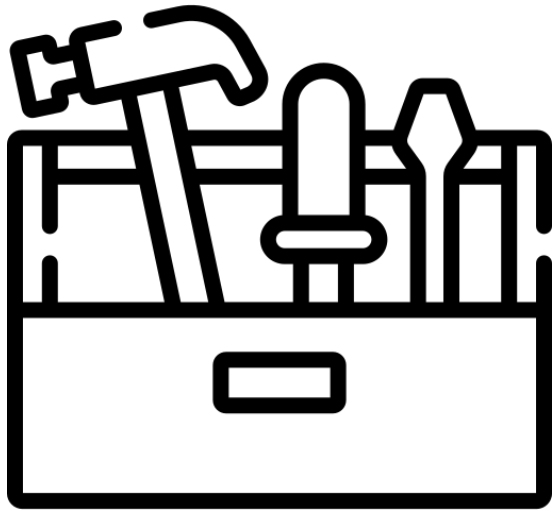
Бессерверный Инференс. Процесс

- Загрузка Модели
- Настройка Триггеров
- Автоматическое Выполнение
- Возврат Результата

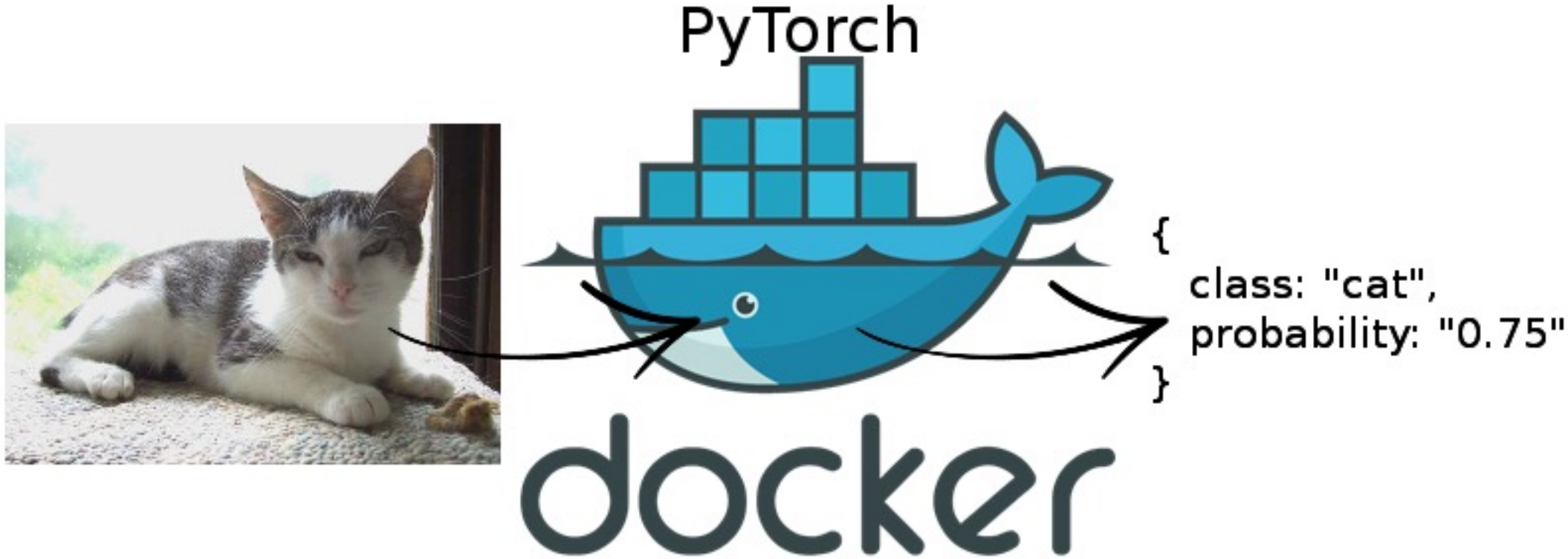


Бессерверный Инференс. Инструменты

- AWS Lambda
- Google Cloud Functions
- Azure Functions
- Serverless Evolution Cloud.ru
- Docker

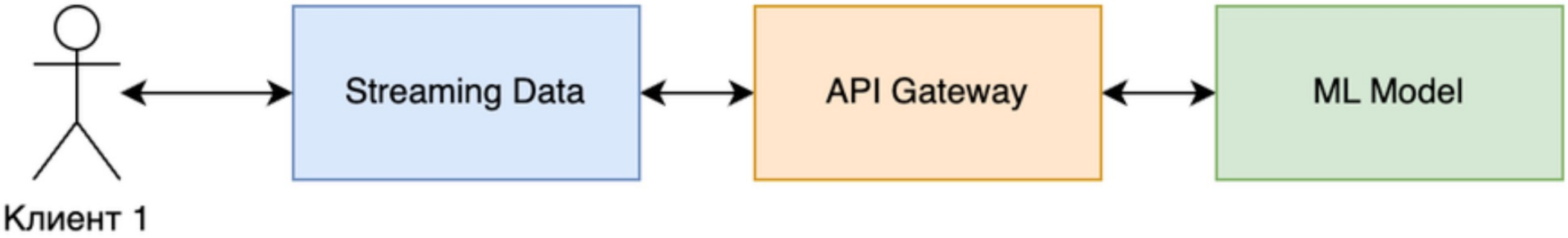


Бессерверный Инференс. Пример



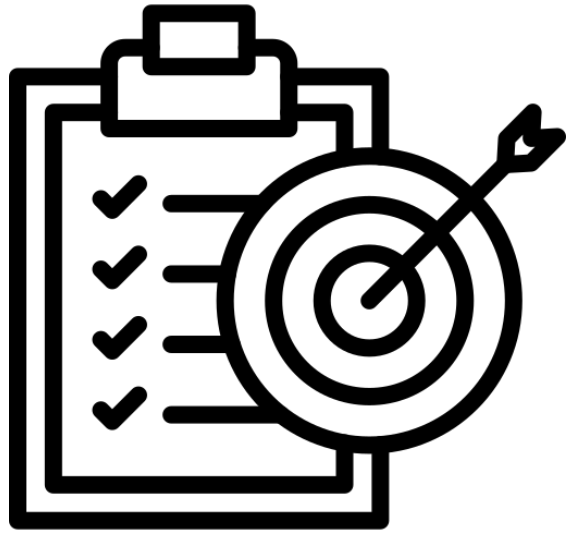
Инференс в Реальном Времени

Real-Time Inference (Инференс в Реальном Времени) в контексте моделей машинного обучения (ML) относится к быстрой обработке данных и предоставлению результатов в режиме реального времени, что критически важно во многих приложениях, требующих немедленного реагирования



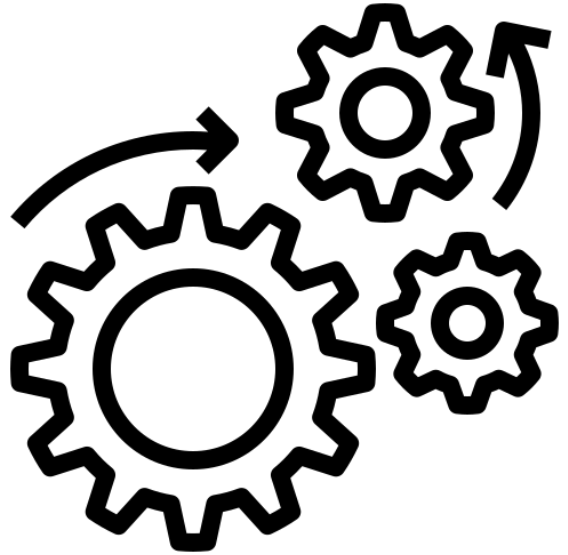
Инференс в Реальном Времени. Цели

- Минимизация Задержек
- Высокая Производительность
- Точность и Надежность



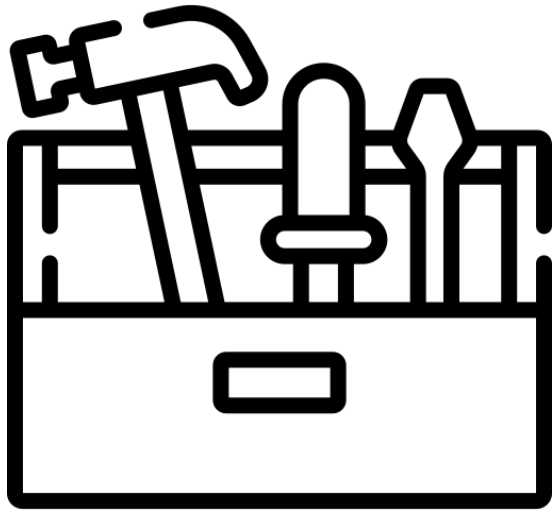
Инференс в Реальном Времени. Процесс

- Прием Данных
- Обработка Данных
- Загрузка Модели
- Выполнение Инференса
- Возврат Результата

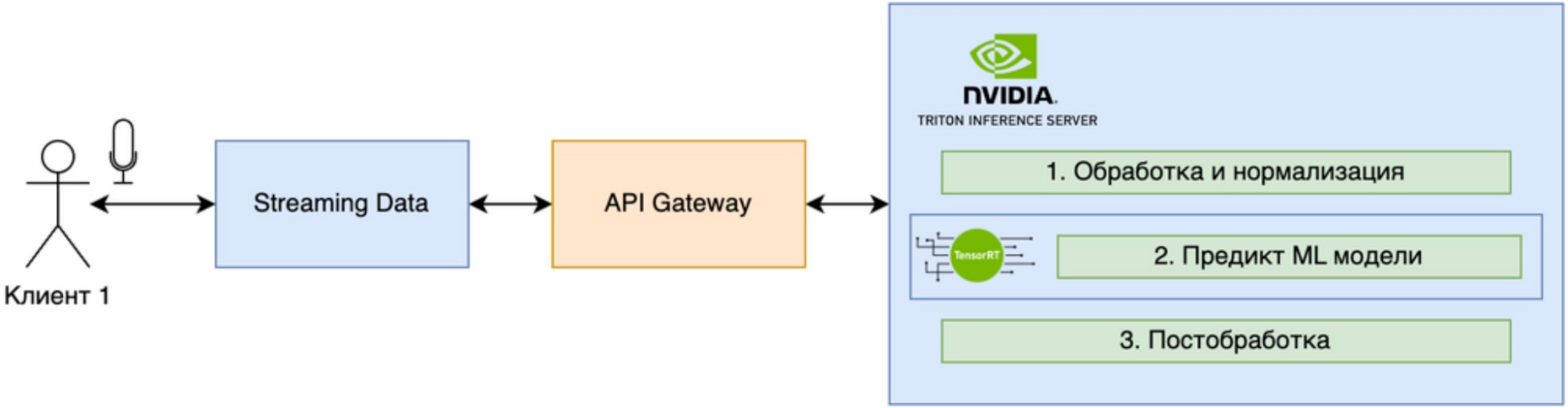


Инференс в Реальном Времени. Инструменты

- Фреймворки ML
- Серверы Инференса
- Средства Мониторинга и Оркестрации



Инференс в Реальном Времени. Пример



Сравнение типов инференса

Тип инференса	Плюсы	Минусы	Цели
Batch Inference	Экономичность, эффективная обработка больших данных.	Высокая задержка, не подходит для реального времени.	Обработка больших наборов данных в нереальном времени.
Asynchronous Inference	Улучшение производительности, разгрузка основного потока.	Сложность управления, зависимость от очередей сообщений.	Эффективная обработка множества запросов без блокировки.
Serverless Inference	Минимальные затраты на инфраструктуру, эластичность.	Ограничения облачных платформ, холодный старт.	Гибкое и масштабируемое развертывание без управления серверами.
Real-Time Inference	Мгновенный ответ, подходит для интерактивных приложений.	Требует значительных вычислительных ресурсов.	Обработка и реагирование на данные в реальном времени.

Фреймворки

- Быстрое прототипирование
- Простота использования
- Демонстрация моделей
- Интерактивность

Gradio

Gradio – это open-source Python-библиотека, которая позволяет быстро создавать веб-интерфейсы для машинного обучения и других Python-функций.

Основные возможности:

- Позволяет превратить Python-скрипт в интерактивное веб-приложение за несколько строк кода.
- Поддерживает различные типы входных и выходных данных
- Интегрируется с популярными ML-фреймворками
- Можно использовать для демонстрации моделей

Gradio

Для кого:
- Data Scientists и ML-инженеры
- Исследователи
- Преподаватели
- Разработчики

Преимущества:
- Быстрота
- Простота
- Доступность
- Интерактивность

Streamlit

Streamlit — это Python-фреймворк для создания веб-приложений и дашбордов с упором на анализ данных и ML

Подходит для:

- Демонстрации ML-моделей
- Визуализации данных (графики, таблицы)
- Создания дашбордов и аналитических инструментов

Streamlit vs Gradio

Критерий	Streamlit	Gradio
Основное назначение	Дашборды, аналитика, полноценные веб-приложения	Быстрые демо ML-моделей
Интерфейс	Гибкий, с поддержкой сложных макетов (колонки, вкладки)	Простые формы с авто-генерацией UI
Интерактивность	Расширенные виджеты (таблицы, графики, кастомные компоненты)	Базовые элементы (слайдеры, кнопки)
Интеграция с данными	Прямая работа с Pandas, Plotly, Altair	Ориентирован на ввод/вывод моделей
Деплой	Streamlit Cloud, серверные решения	Hugging Face Spaces, локальный запуск
Сложность	Требует больше кода для сложных UI	Минималистичный (интерфейс за 5 строк)

Немного практики

Использование Gradio и Streamlit

Полезные ссылки

- Инференс (ML System Design)
- Gradio
- Streamlit