ML архитектура

Архитектура AI pipeline

1. Сбор данных

- Метрики сна со смарт-часов/трекера (длительность, эффективность, пробуждения, пульс и т.д.)

2. Первичный анализ (Gemini 2.5 Flash)

- Роль: “первый эксперт-сомнолог”
- Задача: интерпретировать конкретные метрики, сравнить с предыдущей ночью/периодом, выделить главную проблему и ключевые темы.

3. Извлечение ключевых слов/тем

- Из ответа Gemini вытаскиваются “якоря” для поиска: термины про фрагментацию, эффективность сна, латентность, стресс/пульс и т.д.

4. RAG (Retrieval-Augmented Generation) по базе статей

- По ключевым словам ищем в базе знаний по сну:
  - определения терминов
  - причины/корреляции
  - рекомендации по гигиеная сна
  - выдержки из статей/гайдов
- На выходе: набор найденных фрагментов (top-k) + ссылки/названия

5. Финальная генерация (Mistral)

- На вход:
  - исходные метрики пользователя
  - краткий анализ Gemini
  - RAG-контекст (фрагменты статей/определений)
  - финальный системный промпт с ограничениями
- На выход: финальная рекомендация в нужном формате

6. Post-processing результатов

- Проверка требований формата:
  - без Markdown
  - нужное число абзацев/структура
- Safety/constraints:
  - не ставить диагнозы
  - не назначать лекарства
  - не выдумывать значения (только входные метрики)
- Очистка артефактов (лишние заголовки, списки, маркеры)

Промпт-дизайн и few-shot examples

Используются строгие ограничения:

- запрет приветствий/представлений
- запрет Markdown
- привязка рекомендаций к измерениям
- фиксированный формат ответа