Анализ трендов YouTube и новостей с использованием API

Как превратить разбросанные данные в красивые дашборды без лишних затрат

Постановка задачи

Проблема:

- Социальные платформы, такие как YouTube, постоянно создают тренды.
- Новостные платформы часто усиливают эти тренды.
- Найти значимые связи между ними сложно без автоматизации.

Цель:

- Построить пайплайн для извлечения, обогащения и анализа данных с платформ YouTube и новостей.
- Визуализировать инсайты в понятной и интерактивной форме.

Обзор архитектуры

Основные элементы пайплайна:

1. **Источники данных:** YouTube API и News API.
2. **Обработка:** Python-скрипты для извлечения, обогащения данных и расчета метрик.
3. **Хранилище:** PostgreSQL как надежная и структурированная база данных.
4. **Визуализация:** Superset для интерактивных дашбордов.
5. **Автоматизация:** Cron для планирования выполнения. Минимум затрат, максимум эффективности.

Почему эти инструменты?

- **PostgreSQL:** "*Потому что реляционные базы данных — как хорошие друзья: надежные и универсальные.*"
- **Python:** "*Швейцарский нож программирования. Почему бы и нет?*"
- **NLTK:** "*Совпадение ключевых слов умнее, чем мой последний запрос в Google.*"
- **Superset:** "*Бесплатно, красиво и не требует докторской степени для использования.*"
- **Cron:** "*Airflow — это круто, но у нас один сервер. Дешево и сердито.*"
- **Kafka:** "*Отличный инструмент, но не для этой задачи. Нам не нужен стриминг в реальном времени.*"

Шаги пайплайна

Шаг 1: Извлечение данных

- YouTube API: Топовые трендовые видео (по категориям и регионам).
- News API: Статьи, соответствующие темам видео на YouTube.

Шаг 2: Очистка и обогащение данных

- Сопоставление ключевых слов с использованием NLP (NLTK).
- Обогащение статей новостей идентификаторами видео YouTube.

Шаг 3: Расчет метрик

- Процент статей, связанных с видео.
- Количество видео с хотя бы одной связанной статьей.

Шаг 4: Хранение и визуализация

- Загрузка данных в PostgreSQL.
- Построение дашбордов в Superset.

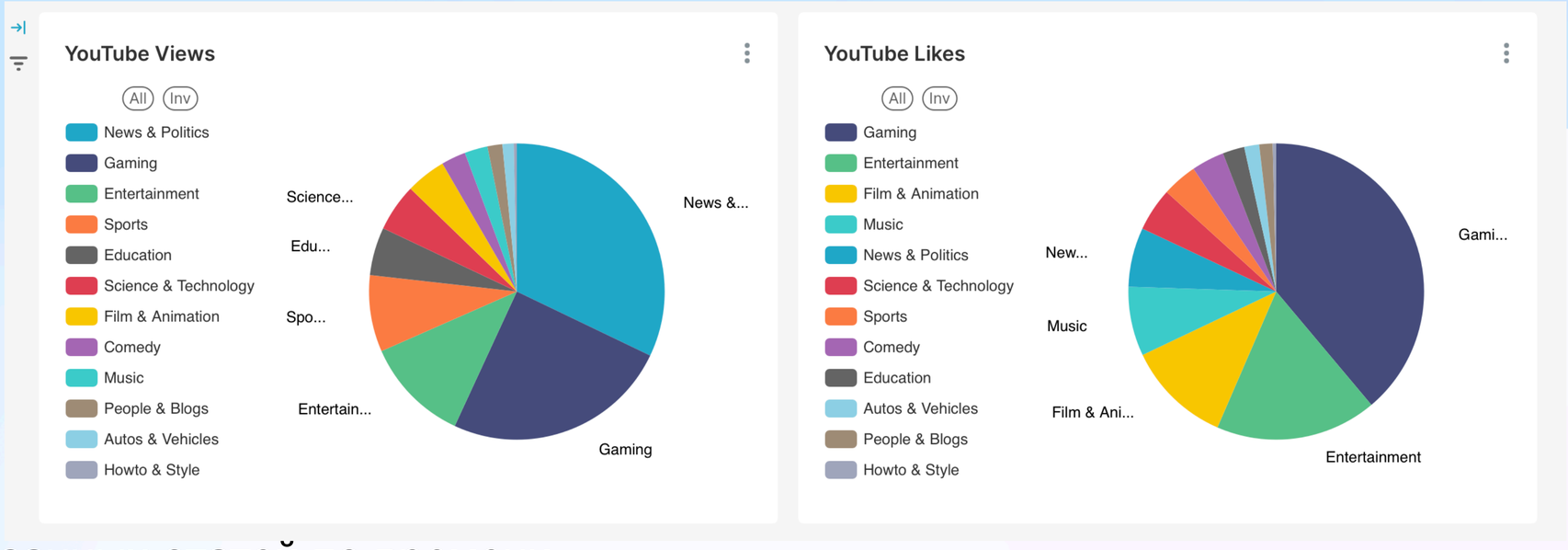
Шаг 5: Автоматизация

- Python-скрипты, запускаемые по расписанию через Cron.

Рассчитанные метрики и Основные дашборды

Метрики:

- **Всего статей:** Общее количество загруженных статей новостей.
- **Связанные статьи:** Статьи, сопоставленные с видео на YouTube.
- **Процент совпадений:** Доля связанных статей от общего числа.
- **Видео с совпадениями:** Видео, имеющие хотя бы одну связанную статью.



Примеры визуализаций:

1. **Линейный график:** Доля связанных статей по времени.
2. **Круговая диаграмма:** Популярные категории видео по просмотрам.
3. **Гистограмма:** Количество статей по источникам новостей.
4. **Таблица:** Детализированный список статей, связанных с видео.

Проблемы и решения

Проблема:

- Низкий процент совпадений на начальных этапах.

Решение:

- Улучшение совпадений за счет извлечения ключевых слов с использованием NLTK.

Проблема:

- Баланс между сложностью и бюджетом.

Решение:

- Избежали использования инструментов, размещение которых на одном сервере заметно замедляет работу всей системы, таких как Kafka и Airflow.
  Остановились на Cron и PostgreSQL.

Что дальше? Варианты развития

- **Потенциальные улучшения:**
  - Добавить больше API (например, TikTok, Google Trends).
  - Интегрировать стриминг данных в реальном времени (Kafka, RabbitMQ).
  - Расширить возможности NLP (например, анализ тональности).

- **Долгосрочные цели:**
  - Построение моделей для прогнозирования трендов.

Заключение

- Построен полноценный пайплайн, который эффективен и прост в обслуживании.
- Визуализированы ценные инсайты из трендов YouTube и новостей.
- Доказано, что для создания полезных решений не требуется огромный бюджет.

Спасибо!

- *Вопросы? Давайте обсудим.*