Eksploracja Danych Tekstowych

YouTube Movie Sentiment Analysis

Analiza sentymentu komentarzy YouTube

Projekt wykonali:	D/XXXXXX Marcin Królczyk, D/YYYYYY Łukasz Cichoń
Grupa laboratoryjna:	DS1
Rok studiów:	II
Semestr:	III
Specjalność:	Data Science
Data wykonania:	03.01.2026

Studia Magisterskie
2026, PK, WIiM

Spis treści

1 Wstęp

Niniejsze sprawozdanie przedstawia projekt realizowany w ramach przedmiotu **Eksploracja Danych Tekstowych** na studiach magisterskich kierunku Data Science. Projekt dotyczy analizy sentymentu komentarzy pod filmami na platformie YouTube, ze szczególnym uwzględnieniem wsparcia wielojęzycznego dla języków polskiego i angielskiego.

Głównym celem projektu było stworzenie aplikacji webowej umożliwiającej:

- Pobieranie komentarzy z dowolnego filmu YouTube za pomocą oficjalnego API
- Automatyczne wykrywanie języka komentarzy (polski/angielski)
- Przeprowadzanie analizy sentymentu z wykorzystaniem modeli transformer
- Wizualizację wyników w postaci interaktywnych wykresów i chmur słów
- Eksplorację zaawansowanych metryk tekstowych (TF-IDF, n-gramy, modelowanie tematów LDA)

Projekt łączy techniki przetwarzania języka naturalnego (NLP) z nowoczesnym interfejsem webowym, demonstrując praktyczne zastosowanie metod eksploracji danych tekstowych w kontekście analizy mediów społecznościowych.

2 Opis projektu

2.1 Funkcjonalności aplikacji

Aplikacja oferuje szeroki zakres funkcjonalności związanych z analizą danych tekstowych:

2.1.1 Wsparcie wielojęzyczne

- **Automatyczna detekcja języka** – system automatycznie rozpoznaje czy komentarz jest w języku polskim czy angielskim
- **Dedykowane przetwarzanie** – dla każdego języka stosowane są odpowiednie modele spaCy do tokenizacji i lematyzacji
- **Wielojęzyczna analiza sentymentu** – wykorzystanie modelu BERT przeszkolonego na danych wielojęzycznych

2.1.2 Analiza sentymentu wspomagana AI

- Wykorzystanie modelu `nlptown/bert-base-multilingual-uncased-sentiment` z biblioteki HuggingFace Transformers
- Klasyfikacja sentymentu na kategorie: pozytywny, neutralny, negatywny
- Obliczanie współczynnika pewności (confidence score) dla każdej klasyfikacji
- Agregacja statystyk sentymentu dla całego zbioru komentarzy

2.1.3 Wizualizacje graficzne

- Interaktywne wykresy Plotly (histogram sentymentu, wykres kołowy rozkładu, scatter plot)
- Chmury słów dla różnych kategorii sentymentu
- Porównawcze chmury słów dla języków polskiego i angielskiego (jeżeli pod danym filmem znajduje się wystarczająca ilość komentarzy w języku polskim oraz angielskim)
- Wykresy słupkowe dla tematów LDA i najpopularniejszych emoji

2.1.4 Eksploracja komentarzy

- Filtrowanie po sentymencie i języku
- Sortowanie po wyniku sentymentu, liczbie polubień lub długości
- Eksport wyników do formatu CSV

2.1.5 Zaawansowana analityka tekstowa

- **Słowa kluczowe** – ekstrakcja z wykorzystaniem TF-IDF
- **N-gramy** – analiza bigramów i trigramów
- **Modelowanie tematów** – LDA (Latent Dirichlet Allocation) z biblioteki scikit-learn
- **Analiza emoji** – statystyki użycia emotikon w komentarzach

2.2 Wymagania systemowe

- Python 3.11.XX (testowane z wersją 3.11.9)
- Klucz API YouTube Data API v3 (dostępny bezpłatnie w Google Cloud Console)
- Minimum 4GB RAM (zalecane 8GB ze względu na modele transformer)
- Około 2GB miejsca na dysku (modele językowe)

2.3 Instrukcja uruchomienia – metoda ręczna

2.3.1 Krok 1: Klonowanie repozytorium

```
git clone <repository-url>
cd youtube-movie-sentiment-analysis-project
```

2.3.2 Krok 2: Utworzenie i aktywacja wirtualnego środowiska

Windows (PowerShell):

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
```

Linux/macOS:

```
python -m venv .venv
source .venv/bin/activate
```

2.3.3 Krok 3: Instalacja zależności

```
pip install -r requirements.txt
```

2.3.4 Krok 4: Pobranie modeli językowych spaCy

```
python -m spacy download en_core_web_md
python -m spacy download pl_core_news_md
```

2.3.5 Krok 5: Uruchomienie aplikacji

```
streamlit run dashboard/app.py
```

Aplikacja będzie dostępna pod adresem http://localhost:8501

2.4 Instrukcja uruchomienia z użyciem narzędzia Docker

Projekt zawiera kompletną konfigurację Docker z multi-stage build, co umożliwia szybkie wdrożenie bez konieczności ręcznej instalacji zależności.

2.4.1 Tryb deweloperski (z hot reload)

```
docker compose -f docker-compose.yml -f docker-compose.dev.yml up --
    build
```

2.4.2 Tryb produkcyjny

```
docker compose -f docker-compose.yml -f docker-compose.prod.yml up -d --
    build
```

2.4.3 Użycie Makefile (zalecane)

Projekt zawiera plik Makefile z wygodnymi skrótami:

```
make dev      # Uruchomienie w trybie deweloperskim
make prod     # Uruchomienie w trybie produkcyjnym
make stop     # Zatrzymanie kontenerow
make logs     # Podglad logow
make shell    # Otwarcie powloki w kontenerze
make clean    # Czyszczenie zasobow Docker
```

2.4.4 Zmienne środowiskowe

Zmienna	Opis	Wartość domyślna
`APP_PORT`	Port aplikacji	8501
`YOUTUBE_API_KEY`	Klucz API YouTube	–
`ENV`	Środowisko (development/production)	development

Tabela 1: Zmienne środowiskowe aplikacji

2.4.5 Architektura Docker

- **Multi-stage build** – optymalizacja rozmiaru obrazu (2GB z modelami)
- **Pre-downloaded models** – modele spaCy i NLTK zawarte w obrazie
- **Health checks** – wbudowany endpoint do monitoringu
- **Non-root user** – uruchomienie jako użytkownik bez uprawnień root
- **Volume caching** – cache modeli HuggingFace między restartami

2.5 Konfiguracja klucza API YouTube

1. Przejdź do Google Cloud Console
2. Utwórz nowy projekt
3. Włącz usługę “YouTube Data API v3”
4. Utwórz credentials → API Key
5. Wprowadź klucz w pasku bocznym aplikacji

2.6 Struktura projektu

```
youtube-movie-sentiment-analysis-project/
|-- src/                          # Moduly podstawowe
|   |-- config.py                 # Konfiguracja i stale (zmienne const)
|   |-- youtube_client.py         # Integracja z YouTube API
|   |-- language_detector.py      # Detekcja jezyka
|   |-- text_preprocessor.py      # Czyszczenie i tokenizacja tekstu
|   |-- sentiment_analyzer.py     # Analiza sentymentu (transformers)
|   `-- text_analytics.py         # Chmury slow, tematy, n-gramy
|-- dashboard/                    # Aplikacja Streamlit
|   |-- app.py                    # Glowna aplikacja
|   |-- components/               # Komponenty UI
|   |   |-- sidebar.py            # Panel boczny
|   |   |-- metrics.py            # Metryki
|   |   |-- charts.py             # Wizualizacje Plotly
|   |   `-- wordcloud.py          # Wyswietlanie wykresow typu wordcloud
|   `-- styles/
|       `-- custom.css            # Niestandardowe style CSS
|-- scripts/                      # Skrypty Docker
|   |-- entrypoint.sh             # Uruchomienie kontenera
|   `-- healthcheck.py            # Endpoint health check
|-- Dockerfile                    # Obraz Dockerowy
|-- docker-compose.yml            # Bazowa konfiguracja dla Docker Compose
|-- docker-compose.dev.yml        # Override bazowej konfiguracji dla srodowiska DEV
```

```
|-- docker-compose.prod.yml       # Override bazowej konfiguracji dla srodowiska PROD
|-- Makefile                      # Skroty polecen
|-- requirements.txt              # Moduly/biblioteki Python
`-- README.md                     # Dokumentacja projektu
```

3 Technologie

Projekt wykorzystuje szereg współczesnych technologii i bibliotek do przetwarzania języka naturalnego oraz tworzenia interaktywnych wizualizacji.

3.1 Główne technologie

Komponent	**Technologia**
Język programowania	Python 3.11
Framework webowy	Streamlit 1.28+
Infrastruktura	Docker, Docker Compose

Tabela 2: Podstawowe technologie

3.2 Przetwarzanie języka naturalnego (NLP)

Biblioteka	**Zastosowanie**
HuggingFace Transformers	Analiza sentymentu z wykorzystaniem wielojęzycznego modelu BERT (`nlptown/bert-base-multilingual-uncased-sentiment`)
PyTorch	Backend dla modeli transformer
spaCy	Tokenizacja, lematyzacja, modele językowe (en_core_web_md, pl_core_news_md)
NLTK	Tokenizacja, stopwords, stemming
langdetect	Automatyczna detekcja języka komentarzy
stopwordsiso	Profesjonalne listy stop-words zgodne z ISO

Tabela 3: Biblioteki NLP

3.3 Analityka tekstowa

Biblioteka	**Zastosowanie**
scikit-learn	TF-IDF, modelowanie tematów LDA, przetwarzanie wektorowe
gensim	Dodatkowe narzędzia do topic modeling
wordcloud	Generowanie chmur słów

Tabela 4: Biblioteki analityki tekstowej

3.4 Wizualizacja danych

Biblioteka	Zastosowanie
Plotly	Interaktywne wykresy (histogram, kołowy, scatter, słupkowy)
Matplotlib	Rendering chmur słów

Tabela 5: Biblioteki wizualizacji

3.5 Integracje zewnętrzne

Usługa	Zastosowanie
Google YouTube Data API v3	Pobieranie komentarzy z filmów YouTube

Tabela 6: Integracje API

3.6 Struktura danych i narzędzia

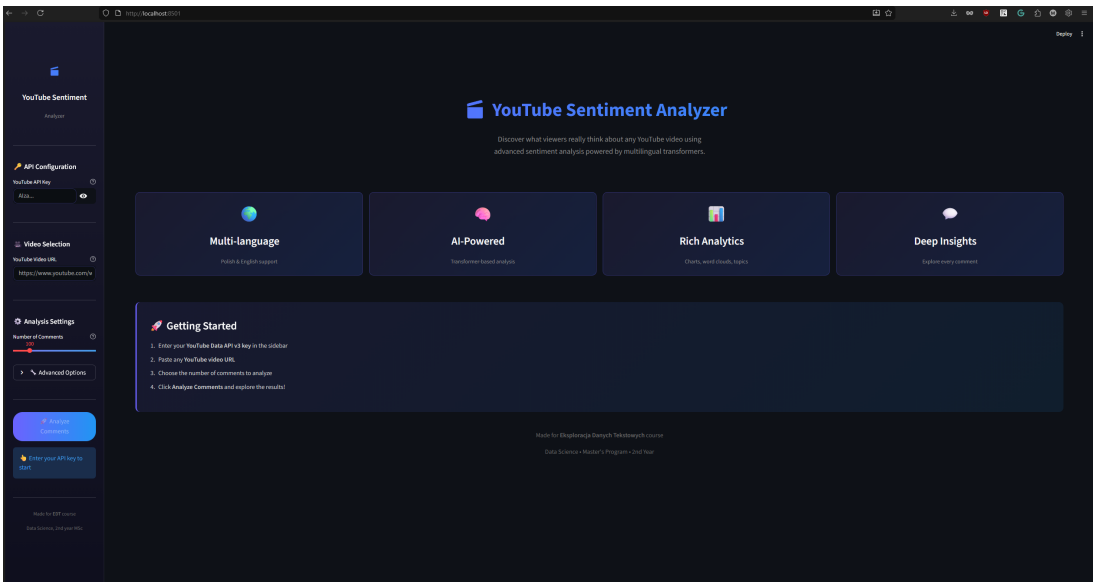
Biblioteka	Zastosowanie
pandas	Manipulacja i analiza danych tabelarycznych
numpy	Operacje numeryczne na tablicach
python-dotenv	Zarządzanie zmiennymi środowiskowymi

Tabela 7: Biblioteki pomocnicze

4 Przegląd interfejsu użytkownika

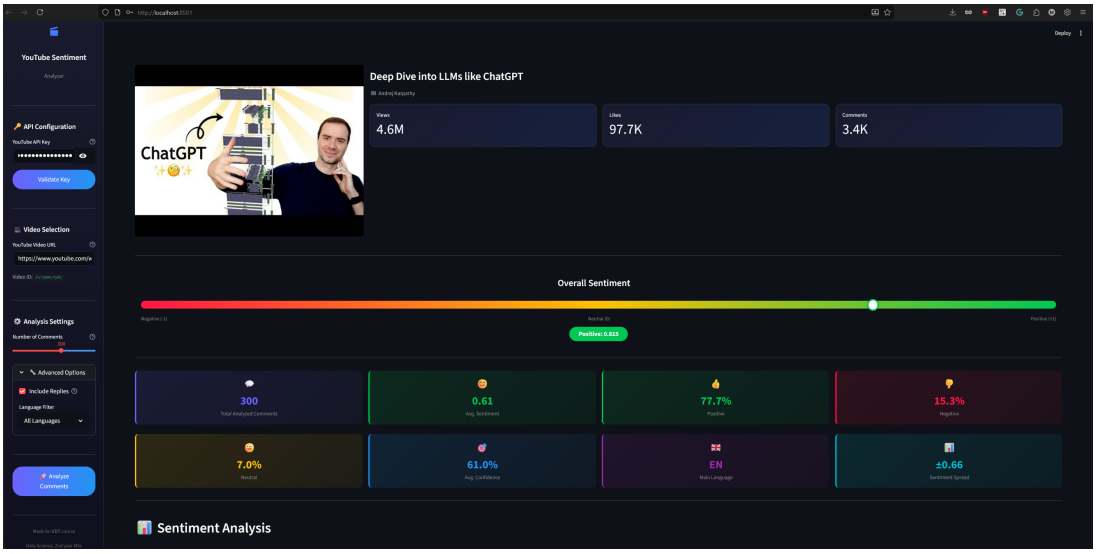
W tej sekcji przedstawiono zrzuty ekranu prezentujące główne widoki i funkcjonalności aplikacji.

4.1 Ekran startowy



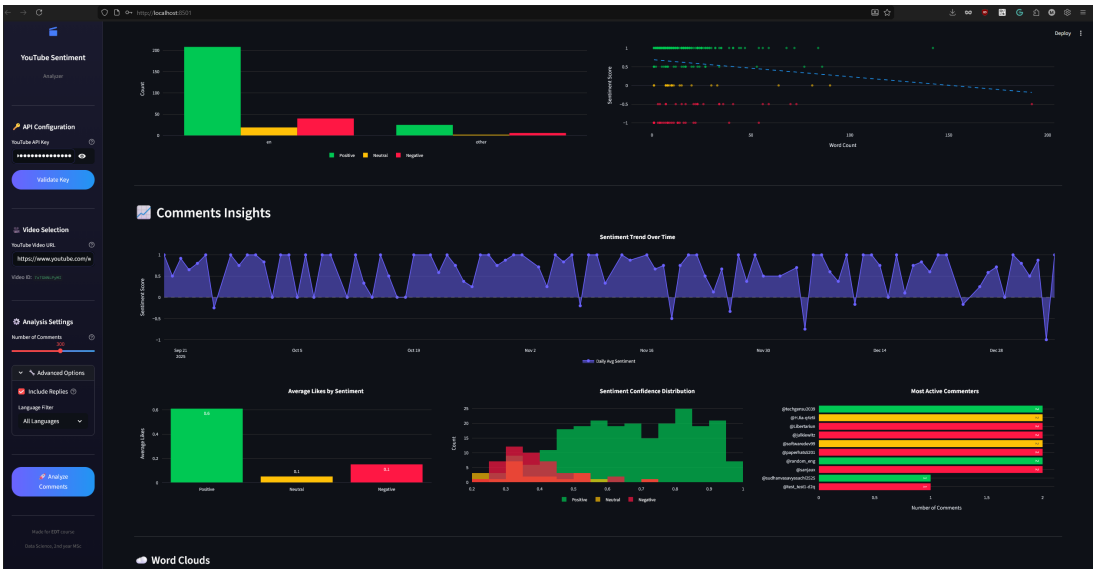
Rysunek 1: Ekran startowy aplikacji z panelem bocznym do wprowadzania danych

4.2 Wyniki analizy – przegląd sentymentu

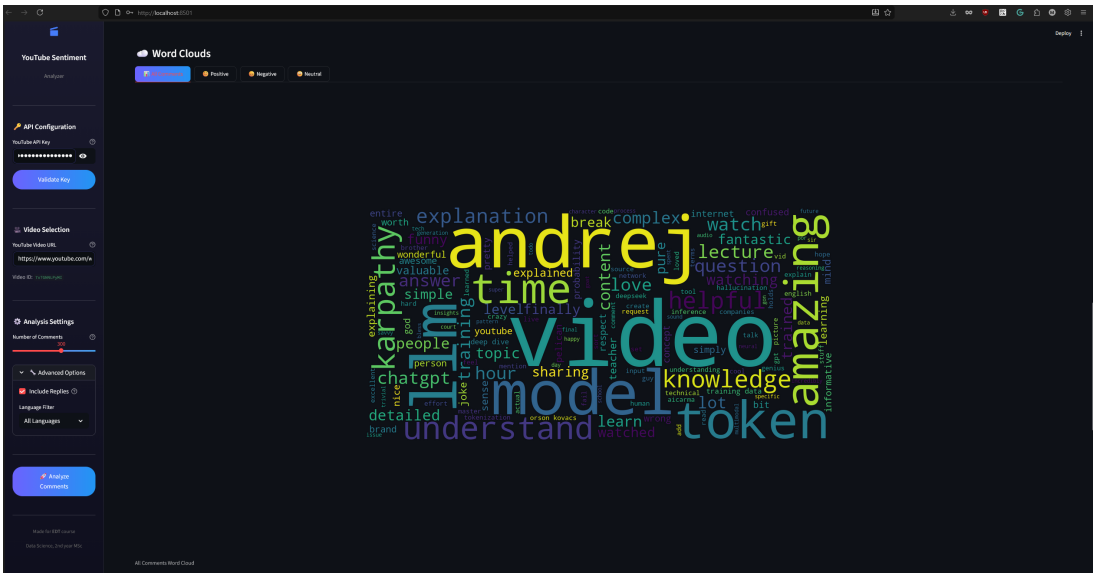


Rysunek 2: Przegląd sentymentu fragment nr. 1

Rysunek 3: Przegląd sentymentu fragment nr. 2

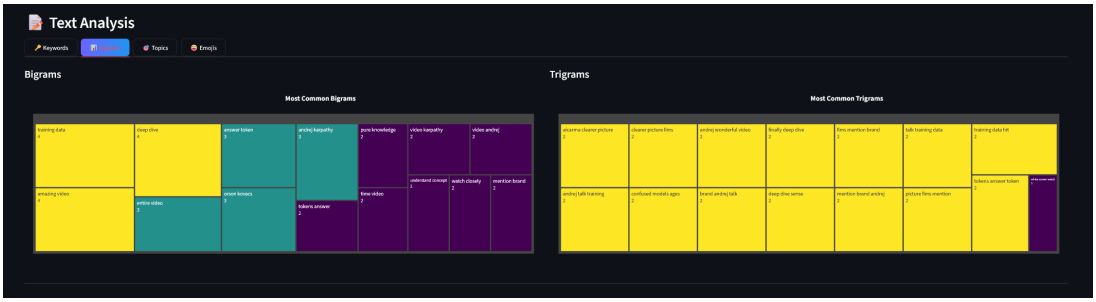


Rysunek 4: Przegląd sentymentu fragment nr. 3

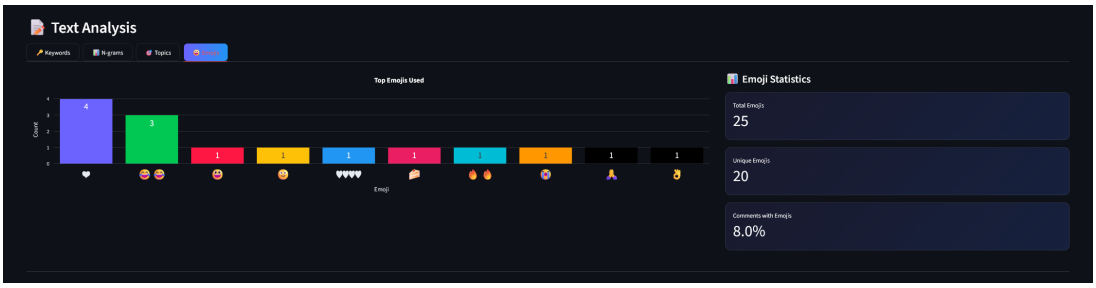


Rysunek 5: Przegląd sentymentu fragment nr. 4

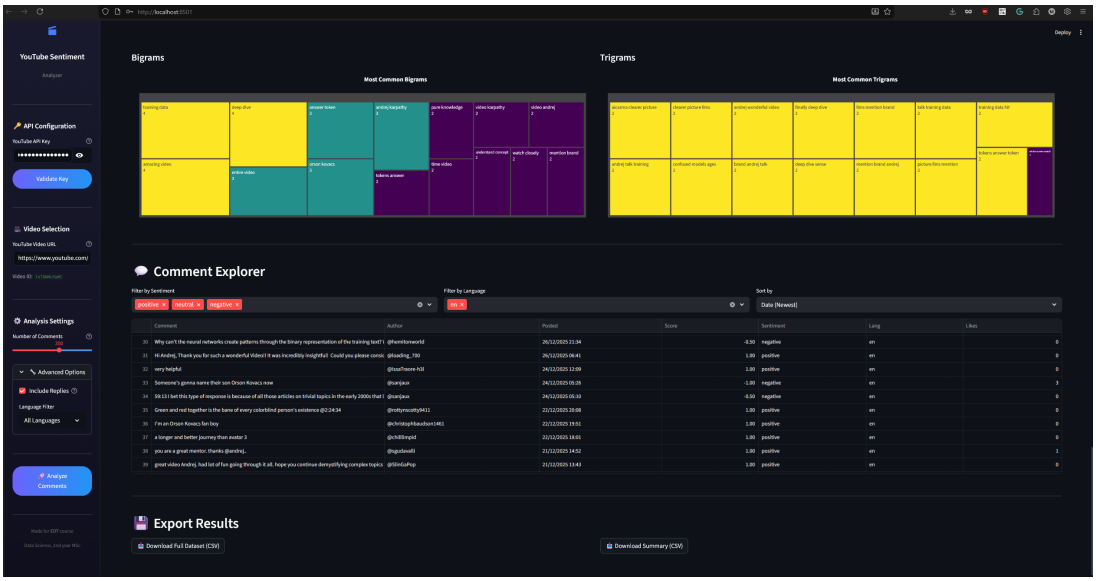
Rysunek 6: Przegląd sentymentu fragment nr. 5



Rysunek 7: Przegląd sentymentu fragment nr. 6



Rysunek 8: Przegląd sentymentu fragment nr. 7



Rysunek 9: Przegląd sentymentu fragment nr. 8

5 Wnioski i podsumowanie

5.1 Osiągnięte cele

Projekt został zrealizowany zgodnie z założeniami i obejmuje:

1. **Kompletną aplikację webową** – interaktywny dashboard Streamlit umożliwiający analizę komentarzy YouTube
2. **Wielojęzyczne przetwarzanie NLP** – automatyczna detekcja języka i dedykowane pipeline'y (wsparcie dla języków: polskiego i angielskiego)
3. **Zaawansowaną analizę sentymentu** – wykorzystanie state-of-the-art modelu BERT przeszkolonego na danych wielojęzycznych
4. **Bogate wizualizacje** – interaktywne wykresy Plotly, chmury słów z podziałem na kategorie sentymentu
5. **Analitykę tekstową** – ekstrakcja słów kluczowych TF-IDF, analiza n-gramów, modelowanie tematów LDA
6. **Konteneryzację aplikacji** – kompletna konfiguracja Docker z multi-stage build oraz health checks, umożliwiająca łatwy deploy aplikacji w przyszłości

5.2 Napotkane wyzwania

Podczas realizacji projektu napotkano następujące wyzwania techniczne:

- **Limity API YouTube** – dzienny limit 10,000 jednostek (dla darmowej wersji)
- **Rozmiar modeli transformer** – pierwsze uruchomienie wymaga pobrania 500MB modeli
- **Wydajność** – przetwarzanie dużej liczby komentarzy z wykorzystaniem modeli BERT wymaga optymalizacji batch processing
- **Wielojęzyczność** – konieczność obsługi specyfiki języka polskiego (odmiany, stopwords)

5.3 Możliwości rozwoju

Projekt może być rozbudowany o następujące funkcjonalności:

- Wsparcie dla dodatkowych języków (niemiecki, hiszpański, etc.)
- Analiza sentymentu w czasie – śledzenie zmian nastrojów komentarzy w momencie publikacji
- Porównywanie wielu filmów jednocześnie
- Automatyczne generowanie raportów PDF
- Integracja z dodatkowymi platformami (Twitter/X, Reddit)
- Fine-tuning modelu BERT na danych specyficznych dla recenzji filmowych

5.4 Podsumowanie

Aplikacja demonstruje praktyczne zastosowanie technik eksploracji danych tekstowych w kontekście analizy mediów społecznościowych. Połączenie nowoczesnych modeli NLP (transformers) z interaktywnym interfejsem webowym (Streamlit) pozwala na efektywną i intuicyjną analizę dużych zbiorów komentarzy tekstowych. Wielojęzyczne wsparcie dla języka polskiego i angielskiego czyni narzędzie użytecznym zarówno dla polskich, jak i zagranicznych treści na platformie YouTube.