Specyfikacja Wymagań Systemowych (SRS)

Projekt: Automatyczna analiza i klasyfikacja recenzji filmowych

Wersja dokumentu: 1.0

Data: 08.06.2025

Autorzy: Filip Lalik, Jakub Grzelak

1. Wprowadzenie

Celem niniejszego dokumentu jest przedstawienie specyfikacji wymagań dla projektu implementowanego w języku R, którego zadaniem jest analiza recenzji filmowych. System przetwarza zbiór opinii, oczyszcza tekst, analizuje częstość i znaczenie słów, wyodrębnia tematy przy pomocy modelu LDA, wizualizuje dane oraz wykorzystuje klasyfikator SVM do przewidywania czy film jest „warty obejrzenia".

2. Cele systemu

- Automatyczne wczytanie i oczyszczenie recenzji filmowych (plik .txt w kodowaniu UTF-8).
- Ogólna normalizacja tekstu (usunięcie znaków specjalnych itp.)
- Redukcja szumu semantycznego za pomocą filtra stopwords.
- Tokenizacja, stemming i uzupełnienie rdzeni słów.
- Analiza częstości słów i ich znaczenia (TF, TF-IDF).
- Modelowanie tematów z użyciem algorytmu LDA i ich wizualizacja.
- Generowanie chmur słów z wyników TF i TF-IDF.
- Budowa klasyfikatora SVM przewidującego wartość recenzji przy podziale na zbiory stratyfikowane.
- Ocena skuteczności modelu przy pomocy metryk Accuracy, Precision, Recall i Specificity.
- Graficzna prezentacja wyników klasyfikacji za pomocą wykresów słupkowych oraz macierzy pomyłek.

3. Wymagania funkcjonalne

• Wczytywanie danych
-Skrypt powinien obsługiwać kodowanie UTF-8.
- Wczytywanie danych z pliku CSV w kodowaniu UTF-8.

• Normalizacja tekstu
- Usunięcie znaków specjalnych, apostrofów, linków, liczb, znaków interpunkcyjnych.
- Usunięcie końcówki „‘s”, często występującej w j. angielskim.
- Zamiana „podwójnego” (grubego) myślnika na spację.
- Tokenizacja i usunięcie stopwords.
- Stemming i uzupełnienie rdzeni słów.

• Analiza sentymentu
- Wyodrębnienie tematów metodą LDA z możliwością określenia liczby tematów.
- Obsługa klasyfikatora SVM i ewaluacja modelu.
- Badanie macierzy pomyłek.

• Wizualizacja wyników
- Generowanie chmur słów (TF i TF-IDF).
- Wykresy słupkowe dla LDA.
- Stworzenie wykresu dla metryki SVM.
- Macierz pomyłek.

4. Wymagania niefunkcjonalne

- Wydajność: analiza do 500 recenzji w czasie do 30 sekund.[1]
- Bezpieczeństwo: walidacja danych wejściowych.
- Niezawodność: obsługa pustych i niekompletnych danych.
- Użyteczność: czytelna prezentacja wyników.

[1] Dla zbiorów do 500 recenzji typowej długości (100–300 słów), analiza powinna zakończyć się w czasie poniżej 30 sekund na standardowym laptopie z 8 GB RAM. Skrypt nie zawiera jednakże mierników czasu działania – szacunki należy weryfikować empirycznie w danym środowisku.

- Kompatybilność: skrypt powinien działać w R w wersji 4.0 bądź nowszej.

5. Interfejsy użytkownika

Wejście:
- Plik recenzje.csv z recenzjami i etykietami.
- Plik wstop.txt z niestandardowymi stopwords.

Wyjście:
- Chmury słów.
- Tabela z częstością słów.
- Wykresy LDA.
- Metryki klasyfikacji.
- Macierz pomyłek.

6. Wymagania dotyczące danych

- Dane muszą być w języku angielskim.
- Plik wejściowy musi mieć format .csv i zawierać etykiety binarne (yes/no).
- Skrypt nie zawiera sztywnego limitu rozmiaru danych, jednak dla zbiorów powyżej kilku tysięcy recenzji może być konieczna optymalizacja (np. usuwanie rzadkich słów, zmniejszenie sparsity (=rzadkość występowania) w DTM).

7. Słownictwo dokumentacji

- Sentyment – ładunek emocjonalny obecny w tekście.
- Stopwords – słowa bez znaczenia semantycznego.
- Stem – uproszczona forma słowa (po sprowadzeniu go do rdzenia).
- Chmura słów – wizualizacja częstości słów.
- Macierz pomyłek – tabela trafności klasyfikatora.
- TF-IDF – Term Frequency-Inverse Document Frequency.
- LDA – Latent Dirichlet Allocation.
- SVM – Support Vector Machine.

8. Przypadki użycia (use cases)

UC1 – Wczytanie i wstępne przetwarzanie recenzji: Użytkownik ładuje dane z pliku CSV, a system wykonuje czyszczenie tekstu.

UC2 – Generowanie chmury słów (TF): System przelicza częstość słów i wyświetla graficzną chmurę.

UC3 – Generowanie chmury słów (TF-IDF): System oblicza TF-IDF i tworzy chmurę słów na tej podstawie.

UC4 – Ekstrakcja tematów metodą LDA: Użytkownik wybiera liczbę tematów, a system wyświetla najbardziej charakterystyczne słowa.

UC5 – Budowa klasyfikatora SVM: System trenuje model klasyfikacyjny do przewidywania etykiety „worth watching".

UC6 – Ewaluacja modelu SVM: System prezentuje metryki skuteczności, wykresy i macierz pomyłek.

9. Scenariusze testowe

Wybrane testy funkcjonalności i odporności systemu.

T1 – Test poprawnego wczytania danych | Wejście: Poprawny plik `recenzje.csv` | Oczekiwane: Brak błędów, dane poprawnie załadowane.

T2 – Test działania czyszczenia tekstu | Wejście: Recenzja z linkiem URL i ze znakami specjalnymi | Oczekiwane: Zwrócony czysty tekst.

T3 – Test działania stemmowania | Wejście: Tekst zawierający „running", „actors" | Oczekiwane: Zredukowane formy: „run", „actor".

T5 – Test LDA (k = 4) | Wejście: Skrypt z `number_of_topics = 4` | Oczekiwane: Wykres z 4 panelami tematycznymi.

T6 – Test klasyfikatora (SVM) | Wejście: Zrównoważone dane | Oczekiwane: Poprawnie wytrenowany model, metryki.

10. Scenariusze użytkownika

Scenariusz 1: Recenzent filmowy

- **Cel:** Automatyczna analiza recenzji z różnych źródeł w celu określenia, czy film warto obejrzeć.
- **Użytkownik:** Krytyk filmowy lub redaktor portalu recenzji.
- **Kroki:**
    1. Wczytanie pliku recenzje.csv zawierającego teksty i oceny.
    2. Przetworzenie tekstu: czyszczenie i stemming.
    3. Uruchomienie modelu SVM do klasyfikacji.
    4. Odczyt metryk skuteczności (Accuracy, F1 Score itp.).
    5. Interpretacja wyników i publikacja syntetycznej oceny filmu.

Scenariusz 2: Badacz języka mediów

- **Cel:** Zbadanie dominujących tematów i stylu językowego w recenzjach filmowych.
- **Użytkownik:** Lingwista, badacz kultury popularnej.
- **Kroki:**
    1. Wczytanie zbioru recenzji bez względu na ocenę.
    2. Przetworzenie i standaryzacja tekstu.
    3. Przeprowadzenie analizy LDA z różnymi parametrami k (liczba tematów).
    4. Wizualizacja najczęstszych słów dla każdego tematu.
    5. Analiza spójności i interpretacja tematów w kontekście trendów filmowych.

Scenariusz 3: Data scientist w branży streamingowej

- **Cel:** Automatyzacja oceny potencjalnego sukcesu filmu na podstawie recenzji.
- **Użytkownik:** Analityk danych w firmie streamingowej.
- **Kroki:**
    1. Wczytanie dużego zbioru recenzji użytkowników platformy.
    2. Przetworzenie danych i trening modelu SVM.
    3. Testowanie modelu na nowych recenzjach (np. przedpremierowych).
    4. Interpretacja wyników i przekazanie informacji zespołowi programowemu (np. decyzje zakupowe oparte na przewidywanej jakości filmu).

Scenariusz 4: Student kierunku Data Science

- **Cel:** Nauka procesu klasyfikacji tekstu oraz eksploracji tematów.
- **Użytkownik:** Osoba ucząca się przetwarzania języka naturalnego w R.
- **Kroki:**
    1. Pobranie danych z recenzjami.
    2. Eksperymentowanie z funkcją top_terms_by_topic_LDA() dla różnych wartości k.
    3. Przeprowadzenie klasyfikacji SVM i ocena wyników.
    4. Modyfikowanie parametrów preprocessingowych (np. własna lista stopwords) i analiza wpływu tych zmian na skuteczność klasyfikatora.