# Dokumentacja Specyfikacji Wymagań (SRS)

**Projekt: Analiza Text Mining wielu plików .txt obejmująca zliczanie częstości słów, analizę sentymentu jednego pliku zbiorczego w oparciu o słowniki NRC, Bing, AFINN oraz GI i QDAP, oraz klastrowanie k-means poprzedzone stemmingiem.**

**Wersja dokumentu: 1.0**

**Data: Czerwiec 2025**

**Autor: Ignacy Brzoza**

1. **Wprowadzenie:**

Dokument zawiera specyfikację wymagań dla skryptu R, który wykonuję analizę text mining oraz analizę sentymentu w oparciu o wiele plików .txt oraz plik zbiorczy, który docelowo powinien zawierać wszystkie pliki .txt zebrane w jedno. System generuje chmury słów, które są wizualizacjami zliczania częstości słów poszczególnych plików, po uprzednim zastosowaniu technik czyszczenia tekstu. System generuje również analizę sentymentu przy użyciu słowników w plikach CSV (NRC, Bing, AFINN) oraz słowników z pakietu SentimentAnalysis (GI, QDAP), wyświelta wykresy zmiany sentymentu w czasie, a także przeprowadza stemming i wykonuje klastrowanie k-means wraz z wizualizacjami wyników.

1. **Cele systemu:**
   * Wczytanie tekstów wejściowych w postaci plików .txt oraz pliku zbiorczego txt., zawierającego wszystkie wgrane pliki zebrane w jedno (wszystkie z odpowiednim kodowaniem UTF-8).
   * Przetwarzanie i oczyszczanie tekstu – usunięcie niepotrzebnych słów stopwords, normalizacja, tokenizacja, a przy klastrowaniu dodatkowo stemming.
   * Zliczenie częstości występowania słów oraz ich wizualizacja w formie chmury.
   * Przeprowadzenie analizy sentymentu pliku zbiorczego z użyciem słowników:
     + w plikach CSV (NRC, Bing, AFINN).
     + wbudowanych w pakiet SentimentAnalysis (GI, QDAP).
   * Przedstawienie wyników w postaci wykresów słupkowych oraz czasowych.
   * Umożliwienie porównania wyników między słownikami.
   * Wykonanie analizy zmiany sentymentu w czasie.
2. **Wymagania funkcjonalne:**
   * **Wczytywanie danych**:
     + Skrypt powinien umożliwiać wczytanie danych tekstowych z lokalnych plików .txt.
     + Skrypt powinien obsługiwać kodowanie UTF-8.
   * **Przetwarzanie i oczyszczanie tekstu:**
     + Skrypt powinien umożliwiać usunięcie znaków interpunkcyjnych.
     + Skrypt powinien umożliwiać usunięcie pustych elementów.
     + Skrypt powinien umożliwiać usunięcie stopwords z pakietów tidytext oraz tm, a także własnoręcznie wpisanych.
     + Skrypt powinien umożliwiać wykonanie stemmingu i uzupełnienia rdzeni słów.
   * **Analiza sentymentu (słowniki w plikach CSV)**:
     + Skrypt powinien umożliwiać wczytanie słowników: nrc.csv, bing.csv, afinn.csv.
     + Skrypt powinien umożliwiać dopasowanie słów do słowników i zliczenie sentymentów.
     + Skrypt powinien umożliwiać filtrowanie słów o sentymencie pozytywnym lub negatywnym.
   * **Analiza sentymentu (słowniki wbudowane w pakiet SentimentAnalysis)**:
     + Skrypt powinien przeprowadzać analizę sentymentu tekstu z wykorzystaniem biblioteki SentimentAnalysis.
     + Skrypt powinien wykorzystywać słowniki GI, QDAP.
     + Skrypt powinien konwertować ciągłe wartości sentymentu na wartości kierunkowe.
   * **Stemming i klastrowanie**:
     + Skrypt powinien umożliwiać wykonanie stemmingu na danych tekstowych.
     + Skrypt powinien umożliwiać klastrowanie k-means.
   * **Wizualizacja danych**:
     + Skrypt powinien umożliwiać wizualizację wyników (wykresy ggplot2).
     + Skrypt powinien dla każdego słownika: tworzyć wykresy skumulowanego sentymentu oraz wykres porównujący sentyment na podstawie różnych słowników.
     + Skrypt powinien tworzyć wykres przedstawiający analizę sentymentu w czasie.
     + Skrypt powinien tworzyć wizualizacje klastrów dokumentów oraz wykresy przypisania dokumentów do klastra.
3. **Wymagania niefunkcjonalne:**
   * **Wydajność**:
     + Analiza pliku o długości 75 000 znaków powinna trwać nie dłużej niż 10 sekund.
   * **Bezpieczeństwo**:
     + System powinien zapewnić poprawność danych wyjściowych.
   * **Niezawodność**:
     + Skrypt powinien poprawnie obsługiwać format .txt danych tekstowych.
   * **Użyteczność**:
     + Wykresy generowane przez skrypt powinny być opisane i czytelne.
     + Skrypt powinien umożliwiać wizualizację wyników z użyciem ggplot2.
     + Skrypt powinien umożliwiać generowanie chmury słów z wykorzystaniem kolorystyki RColorBrewer.
   * **Kompatybilność**:
     + Skrypt powinien być kompatybilny z R w wersji 4.0 lub nowszej.
     + Skrypt powinien korzystać z bibliotek tm, tidyverse, tidytext, ggplot2, ggthemes, wordcloud, RColorBrewer, SentimentAnalysis, SnowballC, cluster, factoextra, dplyr, ggrepel, DT.

1. **Interfejsy użytkownika:**
   * **Wejście**:
     + Pliki tekstowe .txt.
     + Folder textfolder zawierający pliki tekstowe .txt.
     + Pliki słowników w formacie .csv.
   * **Wyjście**:
     + Tabele z częstościami występowania słów.
     + Chmury słów.
     + Wykresy słupkowe rodzaju sentymentu wg słowników (NRC, Bing, AFINN, GI, QDAP).
     + Liniowe i wygładzone wykresy zmian sentymentu w czasie.

1. **Wymagania dotyczące danych:**
   * Skrypt zakłada, że dane tekstowe są w języku angielskim.
   * Skrypt nie obsługuje analizy sentymentu dla innych języków.
   * Skrypt wykorzystuje słowniki sentymentów dostępne w plikach .CSV oraz w pakiecie SentimentAnalysis.
   * Skrypt nie obsługuje analizy sentymentu dla danych tekstowych z innych źródeł niż pliki .txt.
   * Skrypt nie obsługuje plików o rozmiarze powyżej 100 MB.

**Słownictwo dokumentacji:**

* + **Token**: pojedynczy element tekstu (słowo).
  + **Stopwords**: słowa niewnoszące wartości semantycznej do analizy.
  + **Sentyment**: emocjonalne nastawienie w tekście.
  + **Słownik sentymentów**: lista słów i ich ocen wg sentymentu.
  + **Skumulowany sentyment**: suma ocen sentymentu dla całego tekstu.
  + **Wartości kierunkowe**: konwersja ciągłych wartości sentymentu na kategorie (np. pozytywny, negatywny, neutralny).
  + **Stem**: forma słowa po sprowadzeniu go do rdzenia.
  + **Klaster**: grupa przyporządkowanych plików tekstowych wg określonych kryteriów.
  + **Stem Completion**: uzupełnienie rdzenia słowa po stemmingu.

# Przypadki użycia (use cases)

* Użytkownik:

- wczytuje pliki .txt.

- uruchamia analizę.

- wyświetla wyniki.

- generuje wykresy i raport HTML.

* Skrypt:

- przetwarza i oczyszcza teksty.

- tworzy chmury słów dla poszczególnych plików.

- wykonuje analizę sentymentu wg słowników

- wykonuje wizualizacje analizy za pomocą wykresów słupkowych i liniowych.

- wykonuje stemming.

- zlicza częstość słów.

- wykonuje i wizualizuje klastrowanie k-means.

**Testowe przypadki użycia:**

* + Test z plikami .txt o pozytywnych sentymentach.
  + Test z plikami .txt o negatywnych sentymentach.
  + Test z plikami .txt o neutralnych sentymentach.
  + Test z plikami .txt o mieszanych sentymentach.
  + Test z plikami .txt zawierającymi brakujące wartości.
  + Test z plikami .txt zawierającymi znaki specjalne

# Scenariusze użytkownika (user stories)

**Scenariusz 1: Analiza wielu przemówień tego samego polityka**

* **Jako:** Specjalista ds. wizerunku
* **Chcę:** Przeanalizować najczęściej powtarzane słowa oraz sentyment wypowiedzi
* **Aby:** Sprawdzić, jakie sformułowania spotykają się z pozytywnym odbiorem wyborców

**Kryteria akceptacji:**

* Użytkownik może wczytać pliki tekstowe z wypowiedziami polityka.
* Skrypt zlicza częstość słów oraz przeprowadza analizę sentymentu.
* Skrypt tworzy wykresy skumulowanego sentymentu i porównuje wyniki z różnych słowników.
* Użytkownik może przeanalizować różnorodność wypowiedzi i wyciągnąć wnioski, po których wypowiedziach otrzymano najlepszy odbiór.

**Scenariusz 2: Analiza tekstów kultury**

* **Jako:** Redaktor
* **Chcę:** Przeanalizować teksty kultury w zakresie częstości słów oraz analizy sentymentu
* **Aby:** sprawdzić jakiego typu teksty odnoszą największy sukces i cieszą się największą popularnością

**Kryteria akceptacji:**

* Użytkownik może wczytać pliki tekstowe z tekstami kultury.
* Skrypt zlicza częstość słów oraz przeprowadza analizę sentymentu.
* Skrypt tworzy wykresy skumulowanego sentymentu i porównuje wyniki z różnych słowników.
* Użytkownik może zidentyfikować, jakie teksty są najczęściej wskazywane przez czytelników jako interesujące, aby nakierować przyszłe wydania w tym kierunku.