**Wprowadzenie**

Projekt ma na celu stworzenie narzędzia analitycznego służącego do automatycznej analizy opinii tekstowych na temat hoteli. Analiza przeprowadzana jest za pomocą języka R, z wykorzystaniem technik przetwarzania języka naturalnego (NLP), takich jak TF-IDF, oraz metod eksploracji danych, w tym klastrowania z użyciem algorytmu k-średnich. Końcowym celem systemu jest pogrupowanie opinii w logiczne klastry tematyczne oraz ich wizualizacja i interpretacja, co może stanowić wsparcie dla firm turystycznych i klientów indywidualnych.

**Cele systemu**

1. **Zautomatyzowane przetwarzanie opinii tekstowych** – użytkownik może dostarczyć zbiór opinii w formacie .txt, które system połączy i przetworzy.
2. **Czyszczenie i normalizacja tekstu** – usunięcie znaków specjalnych, małe litery, usunięcie stopwords, itp.
3. **Wydobycie cech z tekstu** – zamiana opinii na macierz TF-IDF, dzięki czemu możliwe jest porównywanie treści między dokumentami.
4. **Klastrowanie opinii** – przypisanie każdej opinii do klastra tematycznego metodą k-średnich.
5. **Wizualizacja wyników** – wykresy i chmury słów dla każdego klastra.
6. **Dobór liczby klastrów** – z wykorzystaniem metody silhouette w celu zapewnienia najlepszego podziału danych.

**Wymagania funkcjonalne**

System analizy opinii o biurze podróży, opracowany w języku R, powinien realizować następujące funkcje: Import danych wejściowych System umożliwia użytkownikowi wczytanie jednego lub wielu plików zawierających opinie klientów w formacie .txt lub .csv. Wstępne przetwarzanie danych tekstowych W ramach przygotowania danych system dokonuje: usunięcia znaków interpunkcyjnych i cyfr, konwersji do małych liter, usunięcia tzw. "stop words", lematyzacji lub stemmingu (jeśli zastosowano), tokenizacji. Reprezentacja tekstu przy użyciu algorytmu TF-IDF System przekształca dane tekstowe na macierz cech przy użyciu TF-IDF, umożliwiając analizę istotności słów w opiniach. Analiza skupień (klastrowanie) System grupuje opinie w klastry tematyczne na podstawie podobieństwa cech (np. hierarchiczne klastrowanie lub k-means). Wizualizacja wyników analizy Aplikacja generuje wizualizacje danych, takie jak: chmury słów, wykresy słów o najwyższym TF-IDF, dendrogramy lub inne wykresy klastrów. Generowanie raportu HTML Użytkownik może wygenerować pełny raport HTML z analizy (przy użyciu R Markdown), zawierający wyniki, wykresy oraz interpretację. Eksport danych wynikowych Użytkownik ma możliwość zapisania wyników (np. klastrów, statystyk TF-IDF) do pliku .csv. Obsługa błędów System obsługuje sytuacje wyjątkowe, takie jak: brak danych, błędny format pliku czy puste opinie.

**Wymagania niefunkcjonalne**

Reprodukowalność analizy (Reproducible Research) System powinien spełniać założenia podejścia Reproducible Research – kod źródłowy oraz dane wejściowe muszą być dostępne w formie umożliwiającej powtórzenie eksperymentu. Czytelność i dokumentacja kodu Skrypt R powinien być podzielony na logiczne sekcje z komentarzami. Wszystkie funkcje i działania powinny być opisane. Zgodność techniczna Projekt musi być w pełni zgodny z R (wersja 4.x lub nowsza) i wykorzystywać biblioteki dostępne w CRAN (np. tm, tidytext, ggplot2, cluster). Wydajność System powinien analizować dane (do kilkuset opinii) w czasie nie dłuższym niż 2 minuty, bez zauważalnych opóźnień. Intuicyjna interpretacja wyników Wizualizacje powinny być przejrzyste, podpisane i zawierać czytelne legendy. Nie jest wymagane doświadczenie użytkownika w analizie danych. Działanie lokalne i offline System nie wymaga dostępu do internetu – wszystkie dane i narzędzia są przetwarzane lokalnie.

**Interfejsy użytkownika i wymagania dotyczące danych**

**Interfejs użytkownika:**

System działa jako skrypt w środowisku RStudio, w formie pliku .R lub .Rmd (R Markdown). Nie posiada graficznego interfejsu – użytkownik korzysta z kodu źródłowego, modyfikując:

* lokalizację plików .txt,
* working directory
* liczbę klastrów,
* parametry analizy i czyszczenia tekstu.

**Dane wejściowe:**

* **Format**: Pliki .txt z pojedynczymi opiniami umieszczone w folderze (np. opinie/).
* **Zawartość**: Każdy plik zawiera jedną opinię tekstową.
* **Tworzenie pliku CSV**: System automatycznie łączy pliki .txt i zapisuje je do pliku opinie\_hoteli.csv z kolumną Review\_Text.
* **Wymagania techniczne**:
  + Kodowanie UTF-8,
  + Brak błędów formatowania w plikach .txt,
  + Co najmniej kilka opinii (np. 10) do skutecznego klastrowania.

**Słownictwo dokumentacji**

* Klaster opinii: Grupa opinii klientów biura podróży Contiki, zidentyfikowana za pomocą algorytmu k-means na podstawie podobieństwa ich treści.
* TF-IDF (Term Frequency-Inverse Document Frequency): Metoda ważenia słów w dokumentach, która podkreśla unikalne i istotne terminy w opiniach, eliminując znaczenie powszechnych słów (np. "the", "and").
* Chmura słów: Wizualizacja najczęściej występujących słów w danym klastrze opinii, gdzie rozmiar słowa odzwierciedla jego częstotliwość lub znaczenie (na podstawie TF-IDF).
* Document-Term Matrix (DTM): Macierz, w której wiersze reprezentują dokumenty (opinie), a kolumny reprezentują terminy (słowa), z wartościami wskazującymi wagę słów (tutaj: TF-IDF).

**Przypadki użycia (use cases)**

Analiza sentymentu opinii klientów

* Aktor: Manager biura podróży Contiki
* Cel: Zidentyfikowanie głównych tematów w opiniach klientów, aby poprawić jakość usług.
* Przepływ:

1. Manager importuje plik CSV z opiniami klientów (opinie\_hoteli.csv).
2. System przetwarza dane, tworząc macierz TF-IDF i wykonując klastrowanie k-means (k=2 klastry).
3. System generuje chmurę słów dla każdego klastra, pokazując kluczowe terminy.
4. Manager przegląda wyniki i podejmuje decyzje o poprawie usług (np. szkolenie personelu w odpowiedzi na negatywne opinie).

**Scenariusze użytkownika (user stories)**

"Jako manager biura podróży Contiki, chcę zobaczyć chmurę słów dla każdego klastra opinii, aby szybko zidentyfikować najczęstsze tematy w feedbacku klientów."

* Kryteria akceptacji: Chmura słów wyświetla maksymalnie 15 najczęściej występujących słów w każdym klastrze. "Jako analityk danych w Contiki, chcę przypisać opinie do klastrów za pomocą k-means, aby oddzielić pozytywne i negatywne opinie w celu dalszej analizy."
* Kryteria akceptacji: System przypisuje każdą opinię do jednego z dwóch klastrów i wyświetla wyniki w tabeli.