**Wprowadzenie**

Nasz projekt oparty jest na analizie 30 opinii hotelu Warsaw Presidential. Opinie zostały zebrane ręcznie ze strony [tripadvisor.com](http://tripadvisor.com), a następnie wklejone do jednego pliku CSV.. W celu analizy treści użyliśmy metody LDA, która wizualizuje słowa o największej informatywności. Naszą analizę rozszerzyliśmy również o Asocjację, która, wykorzystując korelację Pearsona między wektorami słów, znajduje współwystępujące słowa. Metody te pozwalają nam dużą swobodę i dokładną analizę opinii, ponieważ w przypadku metody LDA sami wybieramy ilość tematów, a w przypadku metody Asocjacji sami dobieramy słowa, które poddawane są analizie. Dla czytelnego odczytu utworzyliśmy liczne wykresy i chmurę słów.

**Cele systemu**

Celem naszego projektu jest ocena aktualnego stanu hotelu na podstawie opinii pozostawionych przez klientów. Przeprowadzona analiza ma na celu identyfikację zarówno problemów zgłaszanych przez gości, jak i elementów, które są przez nich szczególnie doceniane. Przykładowo, analizując kontekst występowania słowa „service”, jesteśmy w stanie określić, czy obsługa hotelowa spełnia oczekiwania klientów. Takie podejście pozwala na kompleksowe wskazanie mocnych i słabych stron funkcjonowania obiektu, a wyniki prezentowane za pomocą przejrzystych wykresów ułatwiają przekazanie istotnych informacji zarządowi hotelu.

**Wymagania funkcjonalne**

Wymagania funkcjonalne mówią o tym jakie funkcje stworzony system ma realizować. Stworzony przez nas kod ma wiele wymagań funkcjonalnych. Po pierwsze musi importować i przetwarzać dane, co jest kluczowe do dalszej analizy. (powinien wczytywać plik CSV, budować korpus tekstu i czyścić ten tekst). Następnie powinien analizować częstości (generowanie macierzy DTM). Wreszcie stworzony przez nas kod powinien poprawnie modelować tematy metodą LDA i analizować współwystępowania słów poprzez metodę Asocjacji. Warto również dodać, że kod powinien również generować wykresy i chmurę słów w celu łatwego odczytu danych. Na koniec wyniki kodu powinny móc zostać zapisane w postaci raportu.

**Wymagania niefunkcjonalne**

Wymaganie niefunkcjonalne opisują jak system ma działać. Kod stworzony przez nas musi być wydajny, czas analizy opinii nie powinien trwać dłużej niż 10 sekund. Ważna jest również niezawodność, kod powinien działać stabilnie i powtarzalnie, gwarantując poprawność rezultatów niezależnie od liczby uruchomień czy warunków środowiskowych. Ważnym aspektem jest również skalowalność, kod powinien dać się łatwo skalować do większego zbioru opinii. Użyteczność kodu w tym przypadku jest kluczowa, generowane przez nas wykresy i raporty muszą być maksymalnie czytelne i intuicyjne, aby nawet osoby nietechniczne, takie jak zarząd hotelu, mogły bez trudu wyciągać z nich kluczowe wnioski.

**Interfejsy użytkownika i wymagania dotyczące danych**

**Wejście:**

* Plik tekstowy z 30 opiniami .CSV

**Wyjście:**

* Chmura słów
* Tabela częstości słów
* Wykresy ważności słów dla wybranej liczby tematów w modelu LDA
* Lista korelacji o wybranym poziomie dla wybranych słów
* Wykresy lizakowe z natężeniem korelacji dla wybranych słów

**Wymagania dotyczące danych:**

* Skrypt obsługuje tylko dane tekstowe w języku angielskim
* Skrypt obsługuje dane tekstowe z plików .CSV
* Skrypt nie obsługuje plików o rozmiarze powyżej 100 MB

**Słownictwo dokumentacji**

* **Korpus** - zbiór tekstów poddanych analizie
* **Macierz DTM** - macierz gdzie kolumny to słowa, wiersze to nazwy tekstów a wartości w komórkach to częstości występowania danego słowa w danym dokumencie
* **Macierz TDM** - macierz gdzie kolumny to nazwy tekstów, wiersze to słowa a wartości w komórkach to częstości występowania danego słowa w danym dokumencie
* **LDA (Utajniona alokacja Dirichleta)** – nienadzorowany algorytm uczenia maszynowego pomagający zidentyfikować tematy w zbiorze tekstów
* **Temat** – zbiór słów, które są ze sobą powiązane i często występują razem w tekstach
* **Korelacja** – miara jak często słowa występują w tych samych dokumentach lub w dokumentach o podobnych tematach

**Przypadki użycia (Use cases)**

**Użytkownik:**

**-** Wczytuje plik .csv

- Wybiera słowo do analizy

**-** Uruchamia analizę

**-** Wyświetla wyniki

**-** Generuje wykresy i raport html

**Skrypt:**

- Tworzy korpus

- Przetwarza tekst

- Tworzy macierz TDM

- Zlicza częstości słów w macierzach i wypisuje 12 z największą częstością

- Generuje chmurę słów

- Generuje wykresy ważności słów dla wybranej liczby tematów w modelu LDA

- Znajduje korelacje o wybranym współczynniku dla wybranych słów

- Generuje wykresy lizakowe z natężeniem korelacji dla wybranych słów

**Testowe przypadki użycia:**

- Test z plikiem .CSV zawierający 30 pozytywnych opinii na jeden temat

- Test z plikiem .CSV zawierający 30 negatywnych opinii na jeden temat

- Test z plikiem .CSV zawierający 30 pozytywnych opinii na kilka tematów

- Test z plikiem .CSV zawierający 30 negatywnych opinii na kilka tematów

- Test z plikiem .CSV zawierający 30 pozytywnych opinii dotyczących połowy tematów, podczas gdy reszta tematów ma negatywne opinie

- Test z użyciem słowa bez korelacji z innymi

- Test z użyciem słowa silnie skorelowanego z wieloma innymi słowami

- Test z użyciem słowa silnie skorelowanego tylko z jednym słowem

**Scenariusz użytkownika (User stories)**

**Scenariusz 1: Analiza tematów i powiązań w opiniach klientów o hotelu**

**Jako:** Specjalista do spraw jakości

**Chcę:** Przeanalizować opinie klientów hotelu, dla którego pracuję.

**Aby:** Zidentyfikować, które obszary obsługi hotelowej dobrze funkcjonują, a które wymagają poprawy

**Kryteria akceptacji**

* Użytkownik może załadować plik .CSV z opiniami
* Skrypt przeprowadza analizę tematów
* Skrypt generuje wykresy ważności słów dla tematów
* Skrypt wykrywa korelacje między słowami i pokazuję ja na wykresie
* Użytkownik w łatwy sposób może ocenić czy słowa skorelowane z daną usługa są pozytywne czy negatywne

**Scenariusz 2: Analiza tematów i powiązań w opiniach klientów o hotelu**

**Jako:** Specjalista do spraw doświadczeń klientów

**Chcę:** Przeanalizować opinie pasażerów danej linii lotniczej, aby zidentyfikować głównetematy tych opinii i powiązania między najważniejszymi słowami

**Aby:** Zrozumieć na co największą uwagę zwracają klienci i wiedzieć, które obszary obsługi gorzej funkcjonują

**Kryteria akceptacji**

* Użytkownik może załadować plik .CSV z opiniami
* Skrypt pokazuje jakie słowa są najczęściej poruszane w opiniach
* Skrypt generuje wykresy ważności słów dla tematów
* Skrypt pokazuje silnie skorelowane słowa do danej usługi
* Użytkownik może zobaczyć jakie słowa najczęściej opisują dane usługi