Specyfikacja Wymagań Systemowych

Projekt: Sentiment Analysis

Autor: Damian Lebiedź

Data: 29.04.2025

Spis treści

1. Wprowadzenie

Niniejszy dokument stanowi Specyfikację Wymagań Systemowych (SRS) dla projektu „*Sentiment Analysis*”. Dokument określa cele, wymagania funkcjonalne i niefunkcjonalne, interfejsy, słownictwo, przypadki użycia oraz scenariusze użytkownika dla systemu przeprowadzającego analizę sentymentu danych z wykorzystaniem technik data science oraz podejścia *Reproducible Research*.

2. Cele systemu

Celem projektu jest stworzenie systemu informatycznego umożliwiającego analizę sentymentu wpisów w wątku r/wallstreetbets w serwisie Reddit w dniach największej zmienności na giełdzie amerykańskiej w danym okresie mierzonej wartością indeksu VIX. W tym celu powinny zostać wykorzystane: analiza danych tekstowych oraz giełdowych, analiza sentymentu, klastrowanie, czy wizualizacja wyników. System powinien umożliwiać:

- Pobieranie i czyszczenie danych tekstowych oraz giełdowych,
- Przeprowadzenie analizy sentymentu danych tekstowych
- Przeprowadzenie klastrowania danych tekstowych
- Wizualizację wyników za pomocą wykresów i chmur słów

3. Wymagania funkcjonalne

- System umożliwia pobranie danych giełdowych z zadanego okresu, oczyszczenie ich oraz zaprezentowanie na wykresach.
- System wyodrębnia dzień o największej zmienności na zadanym rynku
- System umożliwia pobranie danych tekstowych zawartych w wątku w serwisie Reddit z dnia wyodrębnionego jako tego o największej zmienności na zadanym rynku oraz wyczyszczenie tych danych
- System przeprowadza analizę sentymentu pobranych wpisów za pomocą modelu językowego DistilBERT
- System przeprowadza klastrowanie PCA & KMeans pobranych wpisów
- System wizualizuje na wykresie średni sentyment wpisów z danego dnia
- System wizualizuje za pomocą chmur słów najczęściej występujące słowa zarówno z podziałem, jak i bez podziału na klastry.
- System pozwala użytkownikowi na uruchomienie analizy na własnych danych wejściowych za pomocą modułu *controller.py*, który umożliwia zmianę parametrów systemu.

4. Wymagania niefunkcjonalne

- System powinien być zaimplementowany w języku Python i posiadać czytelny, skomentowany kod źródłowy.
- Wszystkie dane wejściowe i wyjściowe powinny być przechowywane w formatach umożliwiających powtórzenie i odtworzenie analizy (Reproducible Research).
- System powinien być możliwy do uruchomienia na standardowym komputerze osobistym z systemem Windows, Linux lub MacOS.
- Projekt oraz dokumentacja powinny być umieszczone w publicznym repozytorium GitHub.
- System powinien umożliwiać łatwą rozbudowę o dodatkowe moduły analizy tekstu lub źródła danych.

5. Interfejsy użytkownika i wymagania dotyczące danych

- System uruchamiany jest z poziomu skryptu Python lub Jupyter Notebook.
- Użytkownik może określić zakres dat i parametry systemu za pośrednictwem modułu *controller.py*.
- Dane wejściowe pobrane automatycznie za pomocą biblioteki yfinance oraz API Reddita.
- Dane wyjściowe: raport HTML wyeksportowany z pliku Jupyter Notebook.
- Interfejs użytkownika ogranicza się do parametrów przekazywanych w module *controller.py*.

6. Słownictwo dokumentacji

- **API** – interfejs programowania aplikacji; zestaw reguł umożliwiających przesyłanie danych między aplikacjami.
- **Sentiment** – klasyfikacja emocjonalna treści (pozytywny/negatywny/neutralny).
- **VIX** – indeks zmienności amerykańskiego rynku akcji.
- **Chmura słów** – graficzna reprezentacja częstości występowania słów w zbiorze tekstów.
- **Reproducible Research** – podejście zapewniające możliwość powtórzenia analizy na tych samych danych i kodzie.
- **DistilBERT** – model uczenia maszynowego do analizy tekstu, wykorzystywany do klasyfikacji sentymentu.
- **Klastrowanie PCA & KMeans** - metody statystyczne używane do grupowania danych.

7. Przypadki użycia (Use Cases)

UC1: Analiza sentymentu wpisów w serwisie Reddit

- Aktor: Użytkownik
- Opis: Użytkownik uruchamia skrypt, który pobiera wpisy w serwisie Reddit, analizuje ich sentyment, przeprowadza klastrowanie oraz generuje chmury słów.

UC2: Analiza zmienności aktywa na giełdzie

- Aktor: Użytkownik
- Opis: System pobiera dane giełdowe z zadanego okresu, wizualizuje ich zmienność oraz znajduje dzień największej zmienności tego aktywa.

UC3: Generowanie chmury słów

- Aktor: Użytkownik
- Opis: System tworzy chmurę słów dla zadanego zestawu danych w danym okresie, z podziałem lub bez podziału na klastry.

8. Scenariusze użytkownika (User Stories)

- **Jako analityk danych** chcę pobrać i przeanalizować wpisy w serwisie Reddit dotyczące giełdy amerykańskiej, aby zidentyfikować zmiany nastrojów inwestorów w okresach wysokiej zmienności.
- **Jako inwestor** chcę sprawdzić korelację między emocjami inwestorów a zmiennością na giełdzie amerykańskiej mierzoną np. za pomocą indeksu VIX.
- **Jako członek zespołu projektowego** chcę, aby kod i dokumentacja były dostępne w repozytorium GitHub oraz aby instalacja i uruchomienie projektu były dobrze opisane w README.md, aby ułatwić zrozumienie kodu oraz jego implementację.