System oceny przydatności pytań na podstawie ich meta danych i treści dla portalu Stack Overflow w celu ułatwienia pracy moderatorom

Dokumentacja projektu

Zespół

- Dański Mateusz
- Pietrzak Mateusz

Spis treści

Motywacja

StackOverflow to platforma, która pozwala użytkownikom rozwiązywać problemy z kodem aplikacji, wyjaśnić zagadnienie informatyczne itp. poprzez zadawanie pytań. Użytkownicy posiadający odpowiednią reputację odpowiadają na pytania, a po znalezieniu najlepszej odpowiedzi pytanie zostaje zamknięte przez pytającego. Niestety większość pytań wciąż pozostaje otwarta, albo zostaje usunięta, ponieważ użytkownicy zadają bardzo ogólnikowe pytanie, nie umieszczają kodu albo sami nie do końca wiedzą co chcą osiągnąć.

Moderatorzy są odpowiedzialni za odsiewanie pytań wysokiej jakości od tych niskiej jakości. Często zdarza się, że najpierw proszą o uzupełnienie pytania i jeśli pytający nie wprowadzi potrzebnych zmian to dopiero wtedy pytanie zostaje zablokowane i usunięte. Przykładowo w 2018 roku platforma miała ponad 100 milionów aktywnych użytkowników, którzy zadali 2 miliony nowych pytań. Nietrudno wyobrazić sobie, że gdyby nie wsparcie społeczności i moderatorów – wolontariuszy platforma mogłaby już zostać zalana bardzo niskiej jakości pytaniami. Problem ten ma pomóc zaprojektowany system do oceny przydatności pytania na podstawie jego tytułu, treści, tagów, daty utworzenia.

Baza danych

System został zaprojektowany na podstawie bazy danych zawierającej 60 tys. wybranych pytań z lat 2016-2020 zadanych na platformie StackOverflow. Każdemu wpisowi została przyporządkowana jedna z trzech kategorii:

- HQ – wysokiej jakości pytania, z wysoką oceną społeczności i bez edycji
- LQ_EDIT – niskiej jakości pytania z negatywnym wynikiem i z licznymi poprawkami społeczności. Pomimo zmian pytanie wciąż pozostaje otwarte
- LQ_CLOSE – niskiej jakości pytania, które zostały zamknięte przez społeczność bez ani jednej poprawki

Baz danych została podzielona na dwa pliki:

- Treningowy zawierający 45 tys. rekordów
- Testowy zawierający 15 tys. rekordów

Dane dotyczące każdego pytanie mają następujące meta dane:

1. **Id**: unikalny numer pytania
2. **Title**: tytuł pytania
3. **Body**: treść pytania, zawiera formatowanie HTML
4. **Tags**: tagi przypisane do pytania, rozdzielone przecinkiem
5. **CreationDate**: data utworzenia pytania

Analiza danych

?

System

Technologia

Projekt został przygotowany w języku Python (logika przetwarzania danych) oraz C# (GUI). Wymaga zainstalowania modułu Python 'pythonnet'. Testowane na .NET Framework 4.7.2.

Struktura

Struktura projektu przedstawia się następująco:

- **Data** – folder zawierający projekty
    - **Stack Overflow DB** – zawiera pliki bazy danych CSV oraz reguły jako skrypty Pythona
        - **evaulator.py** – zawiera reguły
        - **sample.csv** – wycinek train.csv
        - **train.csv** – dane trenujące
        - **valid.csv** – dane testowe
- **Docs** – dokumentacja projektu
- **Logs** – Logi zapisywane podczas działania aplikacji
- **main.pyw** – główny skrypt aplikacji
- **MainWindows.xaml** – układ okna
- **utility.py** – przydatne funkcje do użycia w programie

Aplikacja została przygotowana w taki sposób, że można dołączać inne projekty poprzez utworzenie nowego katalogu w folderze Data, a w środku należy umieścić plik (lub pliki) CSV bazy danych oraz plik (lub pliki) klasyfikatora.

Klasyfikator

Klasyfikator to skrypt napisany w jeżyku Python, który zwiera dwie funkcje:

parseRow(data)

Przygotowuje dane do klasyfikacji poprzez utworzenie nowych kolumn.

data – słownik zawierający dane w postaci <nazwa kolumny, zawartość> odczytane z pliku CSV, nie zawiera kolumny zawierającej kategorię rekordu

Funkcja musi zwracać słownik <nazwa, wartość> danych, które mogą zostać użyte do klasyfikacji.

examineEntry(data)

Klasyfikuje wpis na podstawie jego meta danych.

data – słownik utworzony poprzez metodę **parseRow**

Funkcja musi zwracać kategorię, która została rozpoznana na podstawie przekazanych informacji.

Wynik

Na podstawie przeprowadzonych obliczeń program zwraca skuteczność rozpoznania dla wszystkich rekordów oraz dla poszczególnych kategorii.

Instrukcja

Instalacja

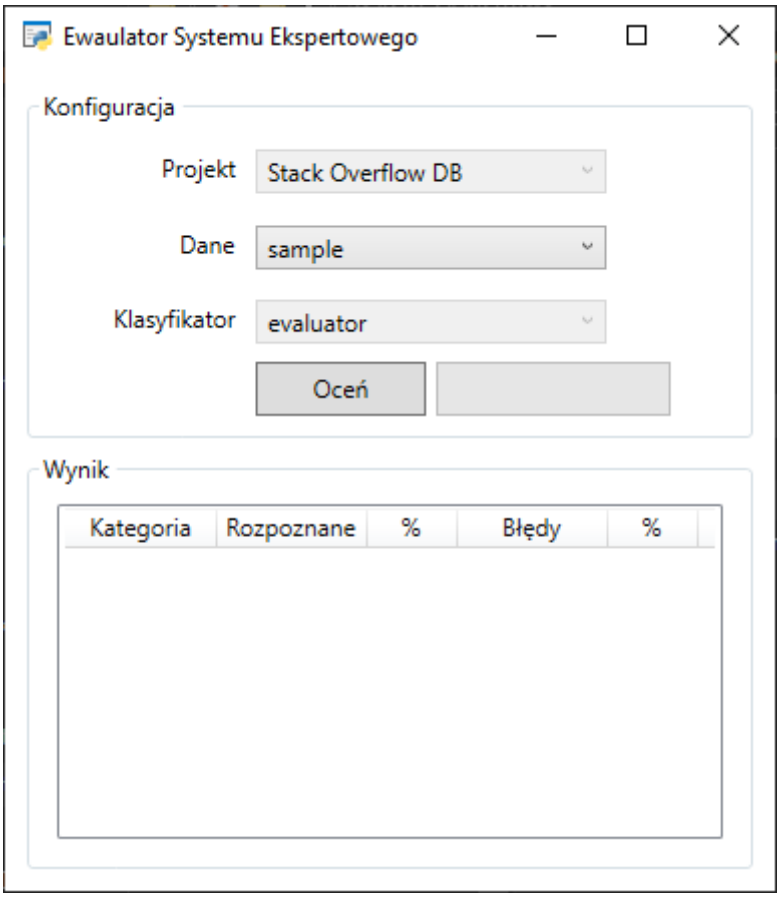
Program nie wymaga instalacji, ale wymaga doinstalowania modułu pythonnet dla Pythona:

```
python –m pip install pythonnet
```

Dodatkowo należy się upewnić, że w systemie zainstalowany został .NET Framework w wersji 4.0+.

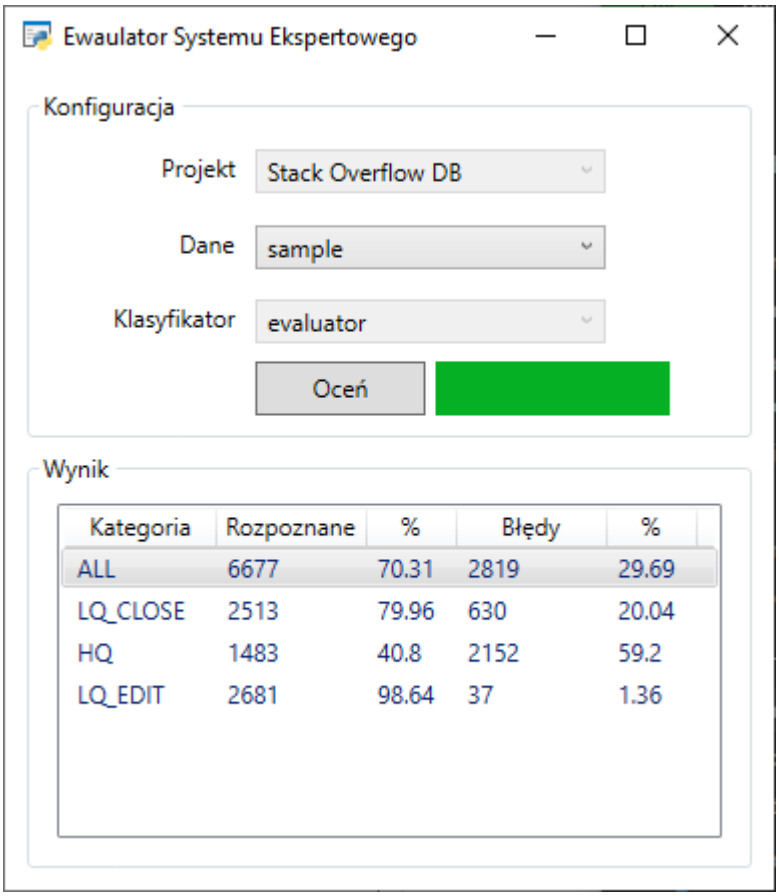
Ocena klasyfikacji

Program uruchamiany poprzez otworzenie skryptu ‘main.py’. Powinno pojawić się następujące okno:



W celu wykonania klasyfikacji kolejno:

1. Wybór projektu (wybór folderu z katalogu ‘Data’)
2. Wybór danych (wybór pliku CSV z folderu projektu)
3. Wybór klasyfikator (wybór pliku klasyfikatora z folderu projektu)
4. Klikamy na przycisk „Oceń”
5. Podczas obliczeń postęp będzie sygnalizowany poprzez pasek postępu obok przycisku
6. Po zakończeniu klasyfikacji wynik zostanie wyświetlony w dolnej części okna:



Pierwszy wiersz zawsze zawiera podsumowanie „ALL” dla ogólnego wyniku klasyfikacji. Pozostałe wiersze to wyniki rozpoznania poszczególnych kategorii. Tutaj można zauważyć, że klasyfikator nie radzi sobie z poro znaniem pytań wysokiej jakości (HQ), ale bardzo dobrze radzi sobie z nie zamkniętymi, niskiej jakości pytaniami LQ_EDIT.