Systemy Uczące Się 2017/18:
trzecie zadanie zaliczeniowe

Robert Suchocki (361086)
login: inpieces
rs361086@students.mimuw.edu.pl

1. Wstęp

Ten raport opisuje sposób rozwiązania przeze mnie trzeciego zadania zaliczeniowego

2. Przetwarzanie

Przy przetwarzaniu danych korzystałem z jednej zewnętrznej biblioteki pythonowej nltk zawierającej PorterStemmer - narzędzie, które konwertuje słowa do formy pierwotnej, co pozwoliło na dokładniejsze ich zliczanie. Moje przetwarzanie składało się z poszczególnych kroków:

- Rozbicie tekstów plików DM2018* umieszczonych w folderze Zbiory danych na osobne pliki do folderów test i train, zapis ich w formacie: liczba wystąpień + znormalizowane słowo
- Dla każdego tagu
  - Zsumowanie wystąpień słów z wszystkich plików z danym tagiem celem znalezienia 50 najpopularniejszych słów dla tagu
  - Zliczenie dla wszystkich plików ilości wystąpień każdego z 50 najpopularniejszych słów tagu w tym pliku, a następnie zapisanie w dwóch plikach csv wyników z wszystkich plików, oddzielnie dla próbki testowej i treningowej, z dodatkową kolumną outcome dla plików treningowych
- Dodatkowo (eksperymentalnie) dla każdego pliku csv - zmniejszenie rozmiaru danych i redukcja ilości zer przez podział 50 plików na 5 grup i przeliczenie wystąpień dla danej grupy, co redukuje 50 kolumn + ew. outcome do 5 + outcome

Efektem było powstanie dwóch plików test_<nazwa_tagu>.csv i train_<nazwa_tagu>.csv (oraz dwóch odpowiedników ze zmniejszonymi danymi) na każdy z 357 tagów. Każdy z nich zawiera w pierwszym wierszu listę 50 najpopularniejszych słów w danym tagu jako nazwy kolumn (+ outcome dla train) oraz 100000 linii, gdzie każda zawiera ilości wystąpień słów z pierwszej linii w kolejnych plikach. Dodatkowo w pliku train, każda taka linia zawiera na końcu 1, jeśli plik ma dany tag, 0 wpp.

3. Klasyfikator

Dla tak utworzonych plików dokonałem klasyfikacji XGBoostem, konkretniej XGBRegressorem z pythonowej biblioteki xgboost. Dla każdego tagu obliczałem kolejno:

- Predykcje kolumny outcome dla danego tagu przez klasyfikator dla próbki testowej w pliku test_<nazwa_tagu>.csv, ćwicząc go na próbce treningowej z pliku train_<nazwa_tagu>.csv
- Indeksy n wierszy z najwyższymi wartościami kolumny outcome (gdzie n = ilość wystąpień tego tagu w próbce treningowej), które są jednocześnie numerami linii, w których ten tag się znajdzie w pliku wynikowym
- Dodatkowo regularnie zliczane dla każdej z linii maksymalne wartości outcome i odpowiadające im tagi

Na końcu do pliku wynikowego wystarczyło zapisać spamiętane wcześniej dla każdej linii tagi, które mają się znaleźć w liniach od 1 do 100000. Dodatkowo do każdej linii dodałem tag, który dla odpowiadającego pliku miał maksymalną wartość outcome spośród wszystkich tagów, co rozwiązało też problem braku tagów w pliku

4. Wynik

Tak przygotowany rezultat uzyskał 0.347 średniego F1-score i w momencie pisania tego raportu znajduje się na 21 miejscu we wstępnym rankingu. Dodatkowo klasyfikacja na eksperymentalnie zmniejszonych danych do 5 grup popularności dla słów wiązała się ze spadkiem wyniku do poziomu 0.1022