Uczenie maszynowe w bezpieczeństwie

Projekt 1

Michał Łaskawski

Zadanie 1

Pobrać, rozpakować i przeanalizować strukturę plików i katalogów archiwum zawierającego wiadomości poczty elektronicznej. Dane te dostępne są pod adresem:

`https://plg.uwaterloo.ca/~gvcormac/treccorpus07/`

Uwaga. Nie należy otwierać plików z archiwum ani w przeglądarce `HTML` ani w programie pocztowym!

Zadanie 2

Wykorzystując informacje z wykładu oraz stosując technikę zakazanych słów kluczowych (`blacklist`), dokonać klasyfikacji binarnej wiadomości z archiwum z podziałem na: `spam` (wiadomości typu spam) oraz `ham` (wiadomości pożądane).

Uwagi:

1. Przed przystąpieniem do procesu klasyfikacji usunąć z wiadomości `stopping words` (np. `the`, `is`, `are`, ...), dokonać stemizacji słów w wiadomościach oraz ekstrakcji tokenów.
2. Do realizacji zadania użyć języka `Python` oraz bibliotek: `string`, `email`, `NLTK`, `os`.
3. Zbiór zakazanych słów kluczowych powinien być wygenerowany na podstawie danych z podzbioru treningowego, natomiast ewaluacja danych uzyskanych z podzbioru testowego.
4. Wynikiem ewaluacji powinna być macierz konfuzji (procentowa) oraz wartość wskaźnika `accuracy`, również w postaci procentowej.

Zadanie 3

Zweryfikować wpływ stemizacji na pracę algorytmu zadania drugiego a następnie porównać uzyskane wyniki.

Zadanie 4

Dokonać klasyfikacji binarnej wiadomości z archiwum (zadanie 1) na `spam` i `ham`, stosując algorytmy rozmytego haszowania.

Uwagi:

1. Do tego celu użyć algorytmu `LSH` (`MinHash`, `MinHashLSH`) z biblioteki `datasketch`.
2. Wyniki pracy algorytmu przedstawić przy pomocy procentowej macierzy konfuzji i wskaźnika `accuracy`.
3. Sprawdzić pracę programu dla różnych wartości parametru `threshold` funkcji `MinHashLSH`.
4. Porównać uzyskane wyniki z wynikami z poprzednich zadań.

Zadanie 5

Dokonać klasyfikacji binarnej wiadomości z archiwum (zadanie 1) na `spam` i `ham`, stosując algorytm `Naive Bayes`.

Uwagi:

1. Do realizacji zadania należy użyć implementacji algorytmu z biblioteki `Scikit-learn`. Algorytm dostępny jest poprzez obiekt `MultinomialNB`.
2. Porównać działanie algorytmu dla przypadków:
   - algorytm pracuje na całych tematach i ciele wiadomości w postaci zwykłego tekstu bez usuwania słów przestankowych i stemizacji przy pomocy narzędzi z biblioteki `NLTK`.
   - algorytm pracuje na bazie stemizowanych danych z usuniętymi słowami przestankowymi.
3. Uzyskane wyniki przedstawić przy pomocy macierzy konfuzji i wskaźnika `accuracy`.
4. Porównać uzyskane wyniki do wyników uzyskanych przy zastosowaniu metod z poprzednich zadań.

Zadanie 6

Dokonać klasyfikacji binarnej wiadomości z archiwum (zadanie 1) na `spam` i `ham`, stosując model gęsto łączonej głębokiej sieci neuronowej i technikę uczenia nadzorowanego.

Uwagi:

1. Zaproponować sposób translacji danych wejściowych do postaci akceptowanego przez sieć tensora wejściowego.
2. Zaproponować liczbę warstw ukrytych oraz liczbę węzłów w poszczególnych warstwach.
3. Zaproponować funkcje aktywacji dla węzłów w warstwach ukrytych oraz w warstwie wyjściowej.
4. Zaproponować metrykę dokładności.
5. Zaproponować optymalizator.
6. Do realizacji zadania zastosować narzędzia z biblioteki `TensorFLow`.
7. W wyniku realizacji zadania wygenerować macierz konfuzji oraz wartość wskaźnika `accuracy`.
8. Porównać uzyskane wyniki dla różnych modeli (to znaczy: ilości warstw ukrytych, ilości węzłów w warstwach, funkcji aktywacji).
9. Porównać uzyskane wyniki z wynikami uzyskanym w ramach realizacji poprzednich zadań.