Sztuczna inteligencja i inżynieria wiedzy

Laboratorium

Ćwiczenie 4. Podstawy maszynowego uczenia na przykładzie klasyfikacji tekstu

Opracowanie: Przemysław Dolata, Julita Bielaniewicz, Jan Jakubik, Maciej Piasecki, Arkadiusz Janz, Jacek Gruber

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawami maszynowego uczenia. Jako przykład wykorzystamy zadanie klasyfikacji tekstu. Wykorzystać należy dwa podejścia:

- naiwny klasyfikator bayesowski lub drzewo decyzyjne – należy wybrać jedną z metod omówionych na wykładzie, oczekiwane jest pełne zrozumienie tych metod i znajomość materiału wykładowego
- maszynę wektorów nośnych (SVM) – jako przykład bardziej złożonego klasyfikatora, wrażliwego na dostrajanie hiperparametrów algorytmu. Należy samodzielnie zapoznać się z tematyką SVM: nie jest wymagana dokładna znajomość algorytmu uczenia, należy jednak rozumieć podstawową ideę na tyle, aby być w stanie wskazać jakie parametry SVM warto przebadać i dlaczego. Przykładowe wprowadzanie z pełnym matematycznym formalizmem można znaleźć pod adresem:

  https://www.ugpti.org/smartse/resources/downloads/support-vector-machines.pdf

  W Internecie łatwo również znaleźć wyjaśnienia skupiające się bardziej na przystępnych wizualizacjach.

Obiektami podlegającymi klasyfikacji będą streszczenia książek, a celem będzie określenie ich przynależności gatunkowej.

Zadanie

W zadaniu należy wykorzystać zbiór CMU Book Summary Dataset podany w linku poniżej:

http://www.cs.cmu.edu/~dbamman/booksummaries.html

Istotną częścią zadania będzie wstępne przetworzenie zbioru danych. Nie wszystkie pola są w nim istotne (np. autor, rok wydania), nie wszystkie są nawet wypełnione danymi. Pola „gatunek" również dotyczy kilka problemów:

- sformatowany jest jako wewnętrzny JSON, który trzeba zdekodować i rozpakować,
- nie dla wszystkich dokumentów jest w ogóle zdefiniowany,
- większości z nich przypisano naraz wiele gatunków.

Należy zaproponować i przeprowadzić procedurę takiego oczyszczenia danych, by uzyskać podzbiór jednoznacznie opisanych przykładów. Dozwolone jest usuwanie zarówno przykładów (np. nieopisanych, lub o zbyt konfliktujących etykietach – np. naraz *science-fiction* i *fantasy*) jak i całych kategorii (zbyt szerokich, np. *fiction*, albo zbyt wąskich, jak *ergodic literature*). Uzyskany zbiór powinien posiadać minimum 4 kategorie i 5000 przykładów.

Na odpowiednio przetworzonym zbiorze streszczeń będziemy chcieli wyuczyć model przewidujący, do jakiego gatunku należy oryginalny utwór. Na podstawie wskazanych metod klasyfikacji (NB/DT, SVM) oraz przygotowanej reprezentacji tekstu należy przygotować program który będzie podstawą do przeprowadzenia badań skuteczności opracowanych rozwiązań. W opracowanym programie należy wykorzystać istniejące systemy lub pakiety uczenia maszynowego. Sugerowane jest wykorzystanie systemu *Weka* dla korzystających z Javy, *scikit-learn* dla programujących w Pythonie. Oba pakiety umożliwiają selekcję cech, trenowanie klasyfikatorów, dostrajanie parametrów oraz testowanie.

Dane do przeprowadzenia badań należy przetworzyć do postaci zgodnej z oczekiwanym wejściem wybranego systemu oraz podzielić zgodnie z zasadami znanymi z wykładu oraz literatury. W eksperymentach obowiązkowe jest wykorzystanie 10-krotnej walidacji krzyżowej.

Podczas pracy nad ćwiczeniem powinien powstawać w sposób przyrostowy raport.

Realizacja ćwiczenia

1. Zapoznanie się z wykładem oraz rozdziałem 13 z “Introduction to Information Retrieval” (dodatkowa literatura rozszerzająca) – opisana w bibliografii.
2. Zapoznanie się z wybranym systemem lub pakietem do maszynowego uczenia.
3. Zapoznanie się ze strukturą, zawartością i metadanymi zbioru danych.
4. Analiza eksploracyjna i oczyszczenie zbioru danych. Zaproponowanie procedury ujednoznaczniania etykiet i pomiar własności uzyskanego podzbioru danych (np. jakie są typowe długości streszczeń, czy wszystkie klasy są równie częste etc.)
5. Zaprojektowanie zestawu cech generowanych na podstawie treści dokumentów. Cechy te mają wynikać z tekstowej zawartości streszczeń, a nie pól metadanych (autor, rok, itp.).
6. Zaprojektowanie i implementacja programu do wydobywania wartości cech z dokumentów i ich zapisywania w formacie odpowiednim dla wybranego systemu lub pakietu do maszynowego uczenia.
7. Zaprojektowanie i skonfigurowanie systemu do maszynowego uczenia obejmującego selekcję cech, dostrajanie, trenowanie klasyfikatorów oraz testowanie.
8. Podział pozyskanych danych na odpowiednie podzbiory zgodnie z zasadami znanymi, np. z wykładu czy literatury. W każdym układzie 10-krotna walidacja krzyżowa jest obowiązkowa.
9. Zaplanowanie eksperymentów (ze zrozumieniem, przemyśleniem, absolutnie nie należy robić tego mechanicznie) mających na celu wnikliwe zbadanie obu algorytmów klasyfikacji w różnych ich wariantach w ramach postawionego zadania. Plan eksperymentów powinien być przedstawiony i uzasadniony w raporcie. Eksperymenty powinny przynajmniej odpowiedzieć na następujące pytania:
   - Jak wrażliwe na hiperparametry są testowane algorytmy?
   - Jak istotna jest selekcja cech?
   - Co daje nam dostrajanie na wydzielonym podzbiorze walidacyjnym?

Poza powyższymi należy zaproponować **co najmniej trzy** własne eksperymenty.

10. Napisanie raportu z przebiegu ćwiczenia. Raport powinien obejmować opis podjętych decyzji oraz ich uzasadnienie. Ze szczególną uwagą powinny być opisane zaplanowane eksperymenty, osiągnięte rezultaty i wyciągnięte wnioski. Raport nie musi i nie powinien być za długi, a jedynie trafnie i treściwie napisany.

Etapy realizacji i punktacja

Etap 1
1 pkt – Zaprojektowanie reprezentacji danych, ich wczytanie i wygenerowanie reprezentacji w wymaganym formacie.
2 pkt – Analiza i oczyszczenie zbioru danych.

Etap 2
3 pkt – Zbudowanie systemu maszynowego uczenia obejmującego: klasyfikator, selekcję cech, dostrajanie hiperparametrów i testowanie. Przeprowadzenie wstępnych eksperymentów.

Etap 3
2 pkt – Zaplanowanie i przeprowadzenie pełnych eksperymentów.

Etap 4
2 pkt – Ukończone sprawozdanie z prac. Klarowna prezentacja wyników i wyciągniętych wniosków.

Bibliografia

1. Christopher D. Manning. Prabhakar Raghavan. Hinrich Schütze. *Introduction* to. *Information*. *Retrieval*. Cambridge University Press, 2008. (there will be also a copy in the Board):
http://www-nlp.stanford.edu/IR-book
lub
https://archive.org/details/AnIntroductionToInformationRetrieval
lub
http://www-connex.lip6.fr/~gallinar/livres%20-%20fichiers/2007-%20Manning-irbookonlinereading.pdf

2. Dokumentacje
- Weka: http://www.cs.waikato.ac.nz/ml/weka/documentation.html
- Scikit-learn: https://sklearn.org/

3. Paweł Cichosz. Systemy uczące się. Wyd. NT, Warszawa, 2000.