AKADEMIA INNOWACYJNYCH ZASTOSOWAŃ TECHNOLOGII CYFROWYCH (AI TECH)

„Uczenie maszynowe" – laboratorium

Laboratorium 5

Zespoły klasyfikatorów

data aktualizacji: 26.05.2022

Cel ćwiczenia

Celem ćwiczenia laboratoryjnego jest zapoznanie się z 3 metodami budującymi rodziny klasyfikatorów: *bagging, boosting i RandomForest*. Każda z metod opiera się na pewnym podstawowym modelu klasyfikacji, który w wyniku dodatkowych mechanizmów manipulacji danych buduje finalny klasyfikator.

Celem zadania jest zapoznanie się z 3 rodzinami zespołów klasyfikatorów: *bagging, boosting i RandomForest*, uruchomienie bibliotek dedykowanych (w Pythonie), przeprowadzenie badań na 3 zbiorach z użyciem różnych konfiguracji algorytmów. Porównanie skuteczności algorytmów i prezentacja wyników.

Użyte zbiory

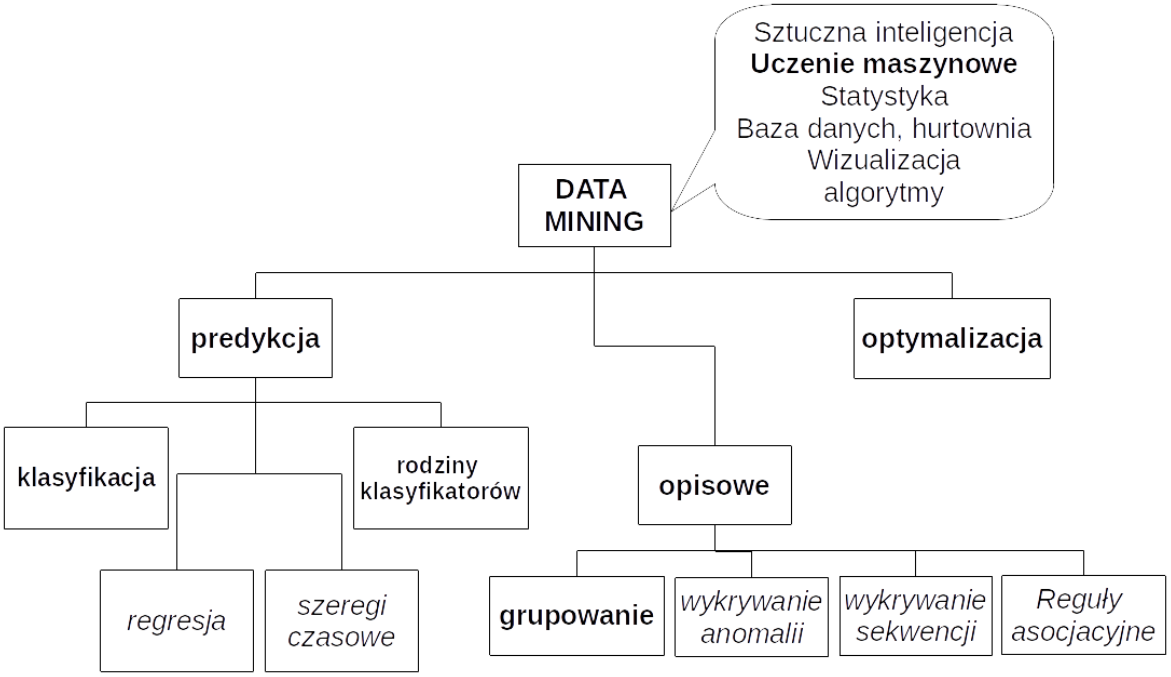
W ćwiczeniu użyte będą powszechnie używane zbiory danych:

- IRIS (*tylko do wstępnych testów*) – https://archive.ics.uci.edu/ml/datasets/iris
- GLASS – https://archive.ics.uci.edu/ml/datasets/glass+identification
- WINE - https://archive.ics.uci.edu/ml/datasets/wine
- SEEDS - https://archive.ics.uci.edu/ml/datasets/seeds

Wprowadzenie

Przy budowie modelu prostego klasyfikatora w poprzednich zbiór danych był niezmienny, a jedyną czynnością wykonaną na zbiorze był podział przy weryfikacji zbioru podczas walidacji krzyżowej (też stratyfikowanej). Dokonywaliśmy podziału zbioru na podzbiory i na ich podstawie tworzyliśmy modele.

W niniejszym ćwiczeniu badane są modele klasyfikacji, które mogą dokonywać selekcji danych, atrybutów lub/i zmieniać ich wagę w zbiorze w zależności od jakości procesu uczenia.



Rys 1. Schemat zadań Data Mining i lokalizacja rodzin klasyfikatorów

Kluczowym w budowie klasyfikatorów jest pojęcie słabego klasyfikatora i na podstawie zestawienia wielu z nich zbudowanie silnego klasyfikatora. W pracy [3] (rozdziały 4.5 i 5.6) jest to bardzo dokładnie opisane z punktu teoretycznego oraz praktycznego.

Sugerowane narzędzia

Zadanie może być realizowane przy pomocy języka Python i bibliotek użytecznych przy manipulacji danymi oraz wizualizacji wyników (tabele/wykresy).

Wskazane narzędzia (python): jupyter, numpy, matplotlib, seaborn itp.

Przy implementacji w Pythonie można użyć gotowych implementacji algorytmów (przegląd i dobry opis działania algorytmów przedstawiono w [4]). Do realizacji zadania można użyć:

- Boosting – np. AdaBoost https://scikit-learn.org/stable/modules/ensemble.html#adaboost

  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier

- RandomForest: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- Bagging Classifier:

  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html

Przebieg ćwiczenia

1. Pobranie danych, sprawdzenie skuteczności modelu bazowego dla 3 zbiorów danych.
2. Algorytm bagging: uruchomienie, wstępna konfiguracja, sprawdzenie skuteczności dla 3 zbiorów. Sugerowane badane parametry (4):
   - liczba klasyfikatorów
   - liczba próbek (*samples*)
   - liczba atrybutów (cech, *features)*
   - *bootstrap*
3. Algorytm boosting: uruchomienie, wstępna konfiguracja, sprawdzenie skuteczności dla 3 zbiorów. Sugerowane badane parametry (2):
   - liczba klasyfikatorów
   - współczynnik uczenia (learning_rate)
4. Algorytm RandomForest: uruchomienie, wstępna konfiguracja, sprawdzenie skuteczności dla 3 zbiorów. Sugerowane badane parametry (4):

- liczba próbek (*samples*)
- liczba atrybutów (cech, *features*)
- liczba drzew
- głębokość drzewa

5. Zastanowienie czy walidacja krzyżowa jest potrzebna?
6. Zestawienie wyników działania 3 zbadanych algorytmów
7. Porównanie wyników z klasyfikatorami bazowymi. Tabele i wykresy, podsumowanie mile widziane.

Punktacja

Przy realizacji zadania można otrzymać **max 8 punktów** wedle poniższej tabeli.

1	Wczytanie danych, wybór modelu bazowego i sprawdzenie jakości klasyfikacji na tego algorytmu.
1	Uruchomienie, konfiguracja (3 parametry) i sprawdzenie jakości klasyfikacji dla algorytmu boosting dla 3 zbiorów danych. Wizualizacja wyników.
2	Uruchomienie, konfiguracja (3 parametry) i sprawdzenie jakości klasyfikacji dla algorytmu bagging dla 3 zbiorów danych. Wizualizacja wyników.
2	Uruchomienie, konfiguracja (3 parametry) i sprawdzenie jakości klasyfikacji dla algorytmu randomForest dla 3 zbiorów danych. Wizualizacja wyników.
2	Zestawienie wyników dla najlepszych konfiguracji na 3 algorytmów na 3 zbiorach danych. Wizualizacja wyników. Wnioski i podsumowanie.

Przy realizacji tego zadania wystarczy prosty raport PDF utworzony przy użyciu Jupyter. Wyniki badań należy skomentować, podać wnioski i podsumowanie.

Warto użyć wykresów, szczególnie polecane są wykresy typu boxplot.

Pytania pomocnicze

1. Czy w modelach klasyfikacji podanych w ćwiczeniu potrzebne (wskazane?) jest użycie walidacji krzyżowej?
2. Czym różnią się algorytmy badane w zadaniu?
3. Który z algorytmów jest łatwiejszy do zrozumienia?
4. Który z algorytmów łatwiej dostroić?
5. Który z algorytmów ma najlepszą skuteczność a który z nich ma najlepszą efektywność (zdefiniowaną roboczo jako „czas przetwarzania")? Kiedy to ma znaczenie?
6. Z czego składa się finalny model klasyfikacji dla poszczególnych algorytmów?

Literatura

1. materiały z wykładów
2. Cichosz P. "Systemy uczące się", WNT Warszawa
3. Koronacki J., Ćwik J., „Statystyczne systemy uczące się", WNT Warszawa
4. Ensemble Classifiers, https://scikit-learn.org/stable/modules/ensemble.html
5. Zasoby Internetu: uczenie maszynowe (*machine learning*), *data mining*, zespoły klasyfikatorów, *ensemble classifiers*, *boosting*, *bagging, random Forest*