KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Instrukcja współfinansowana przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
w projekcie

*„Innowacyjna dydaktyka bez ograniczeń
– zintegrowany rozwój Politechniki Łódzkiej – zarządzanie Uczelnią,
nowoczesna oferta edukacyjna i wzmacniania zdolności
do zatrudniania osób niepełnosprawnych”*

Instrukcja jest dystrybuowana bezpłatnie

Instrukcja do laboratorium, część 6

dr inż. Artur Klepaczko

Eksploracja danych

Zadanie nr 13 – Studia podyplomowe „Przetwarzanie i analiza obrazów biomedycznych”

Politechnika Łódzka
Instytut Elektroniki

90-924 Łódź, ul. Żeromskiego 116,
tel. 042 631 28 83
www.kapitalludzki.p.lodz.pl

Spis treści

1 Wstęp

Podstawową miarą jakości klasyfikatora jest błąd klasyfikacji mierzony stosunkiem liczby błędnie sklasyfikowanych wektorów cech do całkowitego rozmiaru zbioru danych. Jednakże taka funkcja oceny często pozostaje niewystarczająca. Pełniejszy obraz przydatności algorytmu do określonego zadania klasyfikacji można uzyskać poprzez wyznaczanie przedziału ufności dla uzyskanego oszacowania błędu klasyfikacji lub wykonanie krzywej ROC. Ponadto, dla zwiększenia wiarygodności klasyfikacji i uniezależnienia modelu klasyfikatora od dostępnego na etapie uczenia zbioru danych treningowych, często zamiast pojedynczego algorytmu wykorzystuje się zestawy klasyfikatorów. Tworzą one tzw. komitety, które stosują różnorodne techniki głosowania do wyznaczania najbardziej prawdopodobnych etykiet kategorii wektorów danych. Zagadnieniom tym poświęcona będzie pierwsza część laboratorium, w której wykorzystywany będzie nowy moduł programu Weka — *KnowledgeFlow* (zob. pkt 2).

Zadania umieszczone w drugiej części dotyczą technik przekształcania danych za pomocą metod ekstrakcji cech. W szczególności badane będą dwie metody ekstrakcji — analiza składowych głównych (ang. *principal component analysis*, PCA) oraz transformacja losowa (ang. *random projection*, RP). Ich skuteczność zostanie porównana na przykładzie zadania klasyfikacji obrazów tekstur, a także w odniesieniu do klasyfikacji dokumentów tekstowych (zob. pkt 3).

Oczekiwane efekty kształcenia:

→ Umiejętność pracy z programem Weka za pomocą modułu *KnowledgeFlow*.

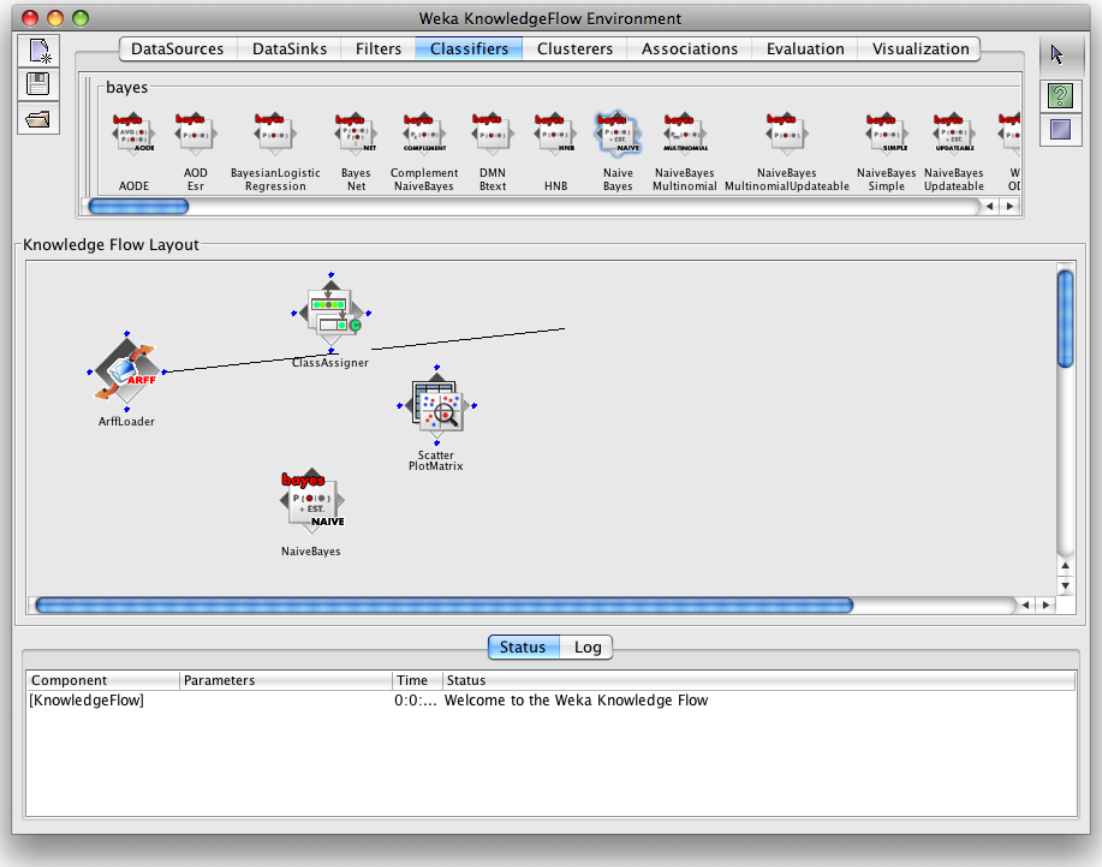
→ Znajomość wybranych metod ekstrakcji cech oraz metod oceny klasyfikacji danych.

2 Moduł *KnowledgeFlow* programu Weka

2.1 Opis modułu *KnowledgeFlow*

Obok modułu *Explorer*, *KnowledgeFlow* stanowi drugi podstawowy moduł programu Weka. Udostępnia on zestaw tych samych algorytmów klasyfikacji, grupowania czy selekcji cech, co *Explorer*, lecz dodatkowo umożliwia połączenie ich w jedno zadanie eksploracji danych. Główną zaletą narzędzia *KnowledgeFlow* jest graficzny interfejs (rys. 1), który pozwala na zaplanowanie w wygodny sposób eksperymentu obejmującego wstępne przetwarzanie danych, właściwą eksplorację, jak również przetwarzanie końcowe wraz z oceną otrzymanych wyników. Przygotowanie eksperymentu polega na umieszczeniu w obszarze roboczym programu (panel *Knowledge Flow Layout*) komponentów reprezentujących określone zadania. Komponenty te, uwidocznione w górnej części okna, zgrupowane są w następujących kategoriach:

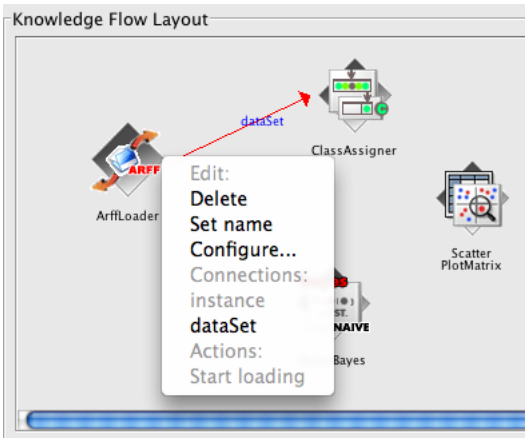
- *DataSources* — źródła danych (m.in. pliki typu `*.arff`, `*.csv`, bazy danych),
- *DataSinks* — lokalizacje docelowe dla danych po przetworzeniu,
- *Filters* — różnego rodzaju filtry (np. standaryzacja, dyskretyzacja, analiza PCA),



Rysunek 1: Widok okna modułu *KnowledgeFlow*

- *Classifiers* — klasyfikatory,
- *Clusterers* — algorytmy grupowania danych,
- *Associations* — algorytmy dla reguł asocjacyjnych,
- *Evaluation* — metody oceny klasyfikacji,
- *Visualization* — narzędzia do wizualizacji wyników przetwarzania danych.

Aby włączyć wybrany komponent do eksperymentu, należy na nim kliknąć lewym klawiszem myszy (obraz kursora zmieni się na znak „plus"), a następnie uczynić to samo w obszarze roboczym programu. Naciśnięcie prawym klawiszem myszy na danym komponencie wywołuje skojarzone z nim menu podręczne (rys. 2), które może zawierać do trzech sekcji: *Edit* — znajdują się tu polecenia pozwalające skonfigurować dane zadanie lub je usunąć z eksperymentu, *Connections* — umożliwia połączenie komponentu z innymi, *Actions* — powoduje uruchomienie określonego działania. Poszczególne komponenty mogą posiadać wszystkie trzy sekcje menu podręcznego lub tylko niektóre z nich. w przypadku próby połączenia danego



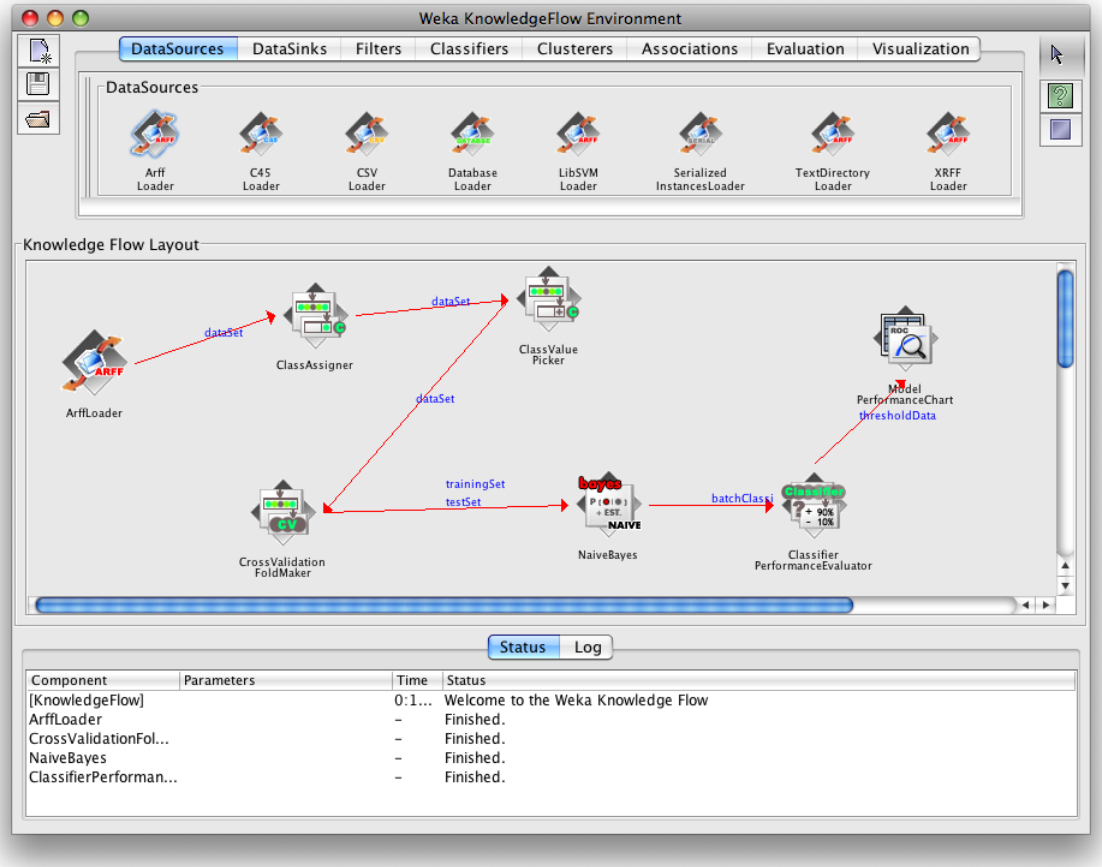
Rysunek 2: Menu podręczne komponentu *ArffLoader*

komponentu z innymi biorącymi udział w eksperymencie program zaznacza te, dla których jest to możliwe. Przykładowo, na rys. 1 komponent *ArffLoader* (reprezentujący plik `*.arff` jako źródło danych) można połączyć jedynie z komponentem *ClassAssigner* (oznaczenie atrybutu klasy) lub *ScatterPlotMatrix*. Po wyborze z menu podręcznego komponentu *ArffLoader* polecenia *dataSet* z sekcji *Connections* w narożnikach odpowiednich komponentów pojawią się niebieskie punkty.

2.2 Wykonanie krzywej ROC

Moduł *KnowledgeFlow* nie stanowi jedynie wygodniejszej alternatywy dla modułu *Explorer*. Umożliwia on wykonanie dodatkowych zadań, takich jak np. ocena klasyfikacji przy pomocy krzywej ROC. Rysunek 3 przedstawia wygląd odpowiedniego eksperymentu. Składa się on z następujących komponentów:

- *ArffLoader* — źródło danych,
- *ClassAssigner* — oznaczenie atrybutu kategorii,
- *ClassValuePicker* — oznaczenie etykiety klasy, która ma być rozpoznawana jako *positive*,
- *CrossValidationFoldMaker* — podział zbioru danych zgodnie z metodą n–krotnej walidacji krzyżowej,
- *NaiveBayes* — naiwny klasyfikator Bayesa,
- *ClassifierPerformanceEvaluator* — wykonanie oceny klasyfikacji,
- *ModelPerformanceChart* — wykonanie krzywej ROC oraz innych wykresów oceny.



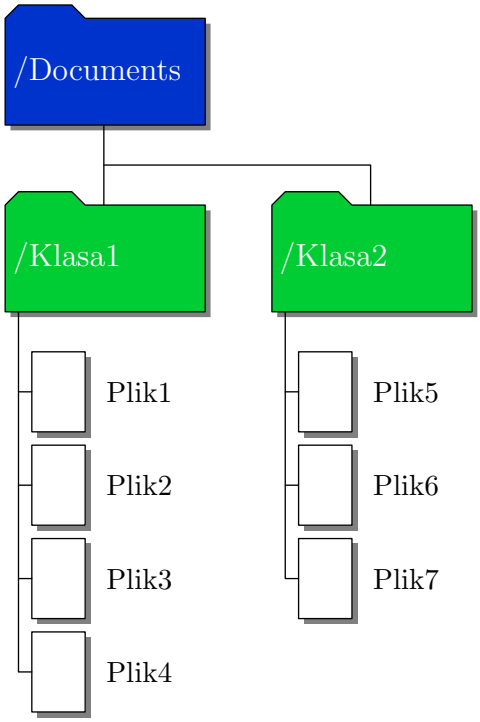
Rysunek 3: Plan eksperymentu do wykonania krzywej ROC

3 Klasyfikacja dokumentów tekstowych

W miarę powiększania się zasobów sieci Internet, a także innych sposobów magazynowania dokumentów w postaci elektronicznej, narasta potrzeba automatycznego rozpoznawania kategorii danego tekstu na podstawie jego zawartości. Zadaniem systemu uczącego się jest m.in. ułatwienie wyszukiwania informacji tekstowej odpowiadającej zadanej sekwencji słów kluczowych. Innym zastosowaniem algorytmów uczenia maszynowego jest identyfikacja dokumentów, wiadomości poczty elektronicznej lub stron internetowych zawierających treści zabronione — lub przeciwnie — najbardziej istotne dla danego użytkownika.

3.1 Przygotowanie katalogu dokumentów

W tych oraz innych przypadkach klasyfikacji dokumentów tekstowych, zasadniczą kwestią jest przygotowanie zbioru danych, czyli opisanie każdego z obiektów biorących udział w uczeniu odpowiednim zestawem atrybutów. W tym celu najczęściej stosuje się cechy określające częstość występowania w dokumencie słów z tzw. słownika haseł kluczowych. Słownik ten



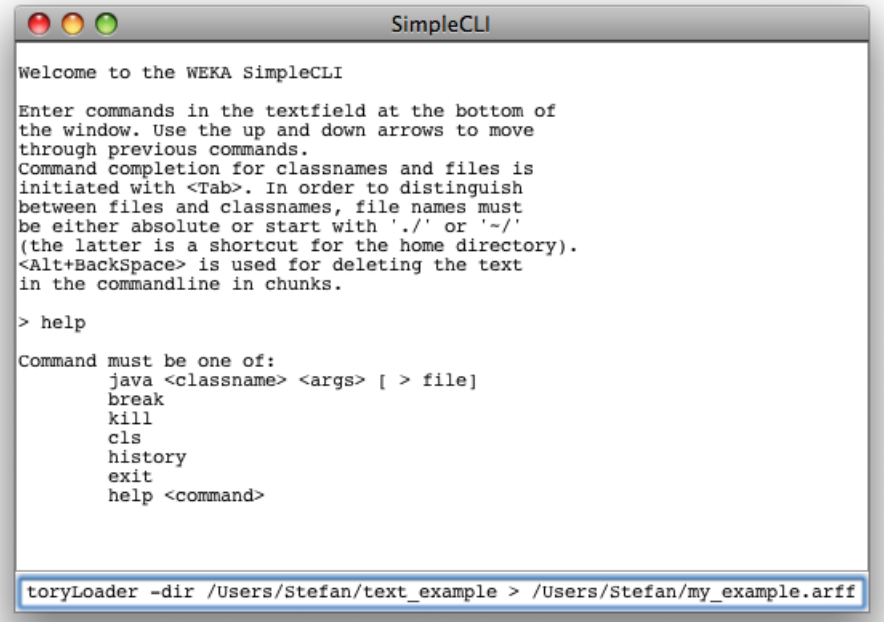
Rysunek 4: Przykładowa struktura katalogów plików tekstowych

należy najpierw utworzyć — automatycznie lub ręcznie. W pierwszym przypadku zadanie to sprowadza się do obliczenia częstości występowania wszystkich wyrazów napotkanych we wszystkich dokumentach ze zbioru treningowego i wyborze spośród nich tylko najbardziej znaczących. W drugim przypadku procedura budowy słownika może być żmudna, ale najprawdopodobniej pozwoli uzyskać bardziej intuicyjny zestaw haseł kluczowych.

Pakiet Weka zawiera specjalne narzędzie przeznaczone do tworzenia zbiorów danych na podstawie dokumentów tekstowych. Jego użycie musi być poprzedzone odpowiednim przygotowaniem plików zawierających klasyfikowane teksty. Każdej klasie powinien odpowiadać odrębny katalog na dysku lokalnym komputera. W nim umieszczone są pliki tekstowe (w dowolnym formacie znakowym, również `*.html`). Katalogi klas dokumentów muszą z kolei znajdować się w jednym wspólnym i nadrzędnym wobec nich katalogu. Przykładowa struktura katologów powinna wyglądać tak, jak pokazano na rys. 4.

3.2 Wstępne przetworzenie zbioru danych tekstowych

Zbiór dokumentów tekstowych zorganizowanych w odpowiednią strukturę katalogową należy w pierwszej kolejności przekonwertować do pliku w formacie *ARFF*. w tym celu można wykorzystać klasę biblioteki Weka o nazwie `TextDirectoryLoader` oraz kolejny z modułów dostępnych w programie Weka — *SimpleCLI*. Moduł ten pozwala korzystać z biblioteki klas Weka w tzw. trybie konsolowym. Rysunek 5 przedstawia okno modułu *SimpleCLI*, które składa się z dwóch zasadniczych fragmentów — linii poleceń umieszczonej u dołu oraz pola raportu, w którym zapisywane są wyniki wykonanych operacji.



Rysunek 5: Interfejs modułu *SimpleCLI*

Zakładając, iż dokumenty znajdują się w katalogu `C:\Users\Stefan\text_example`, konwersja danych tekstowych do pliku zostanie przeprowadzona po wpisaniu polecenia:

```
java weka.core.converters.TextDirectoryLoader -dir
        C\:Users\Stefan\text\_example > C:\Users\Stefan\my_example.arff
```

W powyższym przykładzie nowo utworzony zbiór danych zostanie zapisany do pliku o nazwie `my_example.arff`. Należy zwrócić uwagę, aby miejscem docelowym pliku był katalog, do którego bieżący użytkownik posiada prawa dostępu. W przeciwnym razie zbiór danych zostanie wypisany w obszarze raportu.

Otrzymany zbiór danych zawiera tyle wektorów, ile dokumentów znajduje się w danym katalogu. Każdy wektor opisany jest natomiast jednym arybutem typu *String*, czyli ciągiem znaków reprezentujących słowa bądź fragmenty słów znajdujących się w określonym tekście. Zbiór danych w tej postaci nie może być jeszcze poddany procedurze klasfikacji. Najpierw atrybut *String* musi zostać przekształcony do tzw. wektora słów, zawierającego częstości występowania określonych ciągów literowych w tekście. Aby wykonać to przekształcenie, należy wczytać plik danych tekstowych, np. do modułu *Explorera* i skorzystać z filtru *StringToWordVector*, znajdujący się w grupie filtrów *unsupervised/attribute*. Ponieważ wykonanie wspomnianego filtru prowadzi zazwaczaj do uzyskania wielowymiarowych wektorów cech, w dalszej kolejności należy zredukować wymiar danych, np. za pomocą transformacji PCA lub RP.

4 Zadania do rozwiązania

Uwaga! Zbiory danych, których dotyczą kolejne ćwiczenia umieszczone są na serwerze pod adresem `http://eletel.p.lodz.pl/aklepaczko/open/`.

4.1 Ocena klasyfikacji

Zadanie 1

- Uruchom moduł *KnowledgeFlow* programu Weka.
- W obszarze roboczym programu umieść i połącz w podanej kolejności następujące komponenty (w nawiasach podano kategorie poszczególnych obiektów): *ArffLoader* (źródło danych), *ClassAssigner* (ocena klasyfikacji), *CrossValidationFoldMaker* (ocena klasyfikacji), *IB1* (klasyfikatory), *ClassifierPerformanceEvaluator* (ocena klasyfikacji) oraz *TextViewer* (wizualizacja).
- Skonfiguruj źródło danych tak, aby program wczytywał plik o nazwie `tex_edlab6.arff`. W komponencie *ClassAssigner* wskaż atrybut *category* jako etykietę klasy. Komponent *CrossValidationFoldMaker* należy połączyć z klasyfikatorem dwukrotnie — raz za pomocą połączenia typu *trainingSet*, a raz za pomocą *testSet*. Klasyfikator *IB1* należy połączyć z obiektem oceniającym za pomocą połączenia *batchClassifier*, zaś ten z obiektem tekstowym za pomocą połączenia typu *text*.
- Wykonaj eksperyment wybierając polecenie *Start loading* z menu podręcznego komponentu *ArffLoader*. Sprawdź wynik klasyfikacji wybierając z menu podręcznego komponentu *TextViwer* polecenie *ShowResults*.
- Korzystając z tabeli 1 oraz poniższego równania wyznacz przedział ufności dla uzyskanej poprawności klasyfikacji przy poziomie ufności 98%.

$$s = \left(f + \frac{z^2}{2Q} \pm z\sqrt{\frac{f}{Q} - \frac{f^2}{Q} + \frac{z^2}{4Q^2}} \right) \Big/ \left(1 + \frac{z^2}{Q} \right),$$

  gdzie f oznacza wynik (dokładność) klasyfikacji otrzymany z eksperymentu, Q stanowi liczność zbioru danych, zaś z jest granicą przedziału ufności odczytaną z tabeli dla ustandaryzowanej zmiennej Gaussa.

Zadanie 2

- Zaprojektuj nowy eksperyment w module *KnowledgeFlow* zgodnie ze schematem pokazanym na rys. 3.

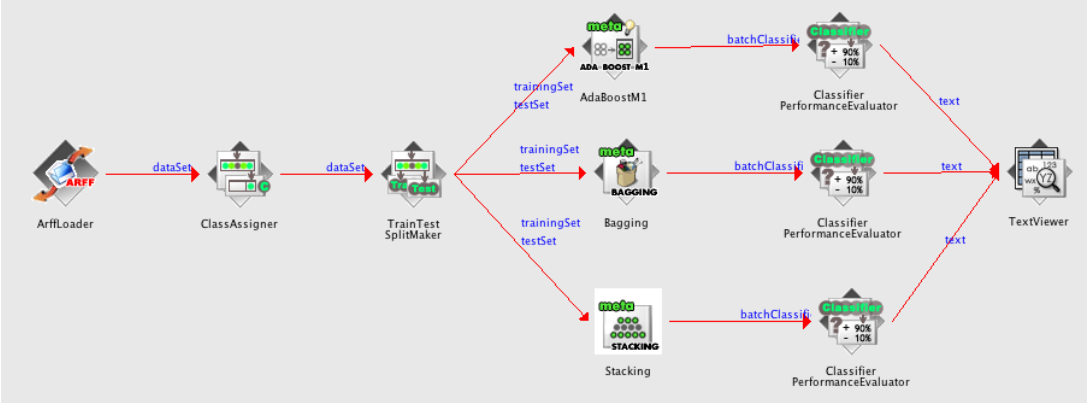
Tabela 1: Granice przedziałów ufności dla rozkładu Gaussa $N(0,1)$

$\Pr[X \geqslant z]$	z
0.1%	3.09
0.5%	2.58
1 %	2.33
5.0%	1.65
10.0%	1.28
20.0%	0.84

- Skonfiguruj źródło danych tak, aby program wczytywał zbiór danych `sonar.arff`. W komponencie *ClassAssigner* wskaż atrybut *Class* jako etykietę klasy, zaś w *ClassValuePicker* wskaż klasę *Rock*. Wykonaj obliczenia wybierając polecenie *Start loading* z menu podręcznego komponentu *ArffLoader*.
- Powtórz eksperyment wykorzystując zamiast naiwnego klasyfikatora Bayesa algorytm klasyfikacji oparty na regresji logistycznej (komponent *Logistic*).
- Otwórz okno z wykonanymi wykresami ROC wybierając polecenie *ShowChart* z menu podręcznego komponentu *ModelPerformanceChart*.
- Co można powiedzieć o porównywanych klasyfikatorach na podstawie otrzymanych wykresów?

Zadanie 3

- Zaprojektuj eksperyment w module *KnowledgeFlow* zgodnie ze schematem pokazanym na rys. 6. Przyjmij wstępnie, że komitety *AdaBoostM1* oraz *Bagging* wykorzystują algorytm wektorów podpierających (*SMO*) z wielomianową funkcją jądra drugiego stopni, a zawierającą wyrażenia niższego rzędu (należy odpowiednio ustawić parametry *exponent* oraz *useLowerOrder*). Z kolei komitet typu *Stacking* niech składa się dodatkowo z klasyfikatorów *IB1*, *NaiveBayes* oraz algorytmu *OneR* jako nadrzędnego metaucznia.
- Ustaw liczbę iteracji w komitetach *AdaBoostM1* oraz *Bagging* na 10 (parametr *numIterations* w obu przypadkach).
- Zachowaj konfigurację komponentów *ArffLoader* oraz *ClassAssigner* taką, jak w poprzednim zadaniu.
- Wykonaj obliczenia i sprawdź uzyskane wyniki. Czy zastosowanie komitetów poprawia jakość klasyfikacji w stosunku do pojedycznych klasyfikatorów?
- Powtórz doświadczenie z innymi, dowolnie wybranymi algorytmami klasyfikacji, a także dla innych zbiorów danych (np. zbiorów tekstur wykorzystywanych w poprzednich ćwiczeniach laboratoryjnych).



Rysunek 6: Plan eksperymentu dla Zadania 3

4.2 Transformacje danych

Zadanie 4

- Zaprojektuj eksperyment w module *KnowledgeFlow* w podobny sposób, jak opisano to w Zadaniu 1. Dodatkowo, między komponentami *ClassAssigner* oraz *CrossValidationFoldMaker*, umieść komponent filtru *AttributeSelection*.
- Skonfiguruj nowy komponent tak, aby wykonywał selekcję za pomocą algorytmu przeszukiwania *LinearForwardSelection* oraz funkcji oceny podzbiorów cech *CfsSubsetEval*. Ponadto zmień wartość parametru *forwardSelectionMethod* w metodzie przeszukiwania na *Floating forward selection*.
- Jako źródło danych ponownie wskaż plik `tex_edlab6.arff`. Wykonaj klasyfikację.
- Powtórz obliczenia zastępując komponent *AttributeSelection* filtrem *PrincipalComponents*, a następnie *RandomProjection*. W obu przypadkach ustaw docelowy wymiar przestrzeni cech na wartość 10 (odpowiednio parametry *maximumAttributes* oraz *numberOfAttributes*).
- Porównaj uzyskane wyniki eksperymentów zarówno pod względem dokładności klasyfikacji, jak i szybkości obliczeń.

Zadanie 5

- Na podstawie zasobów internetowych przygotuj zbiór kilkunastu dokumentów tekstowych zawierających informacje z trzech dowolnie wybranych kategorii (np. sport, polityka, ekonomia). Pogrupuj pliki w katalogi zgodnie ze strukturą opisaną w pkt 3.1.
- Wykonaj konwersję plików tekstowych do zbioru danych w formacie *ARFF* zgodnie z procedurą opisaną w pkt 3.2.

- Przygotuj w programie *KnowledgeFlow* plan prostego eksperymentu klasyfikacyjnego, zawierającego przynajmniej następujące elementy:
  - *ArffLoader*,
  - *ClassAssigner*,
  - jeden z filtrów *PrincipalComponents* lub *RandomProjection*,
  - *CrossValidationFoldMaker*,
  - dowolny klasyfikator (poza algorytmem *ZeroR*),
  - *ClassifierPerformanceEvaluator*,
  - *TextViewer*.
- Sprawdź zdolność wybranego algorytmu klasyfikacji do predykcji kategorii dokumentów tekstowych.

Łódź 2009

Zredagowane w LaTeX-u