Weka

Praktické použití

Antonín Pavelka

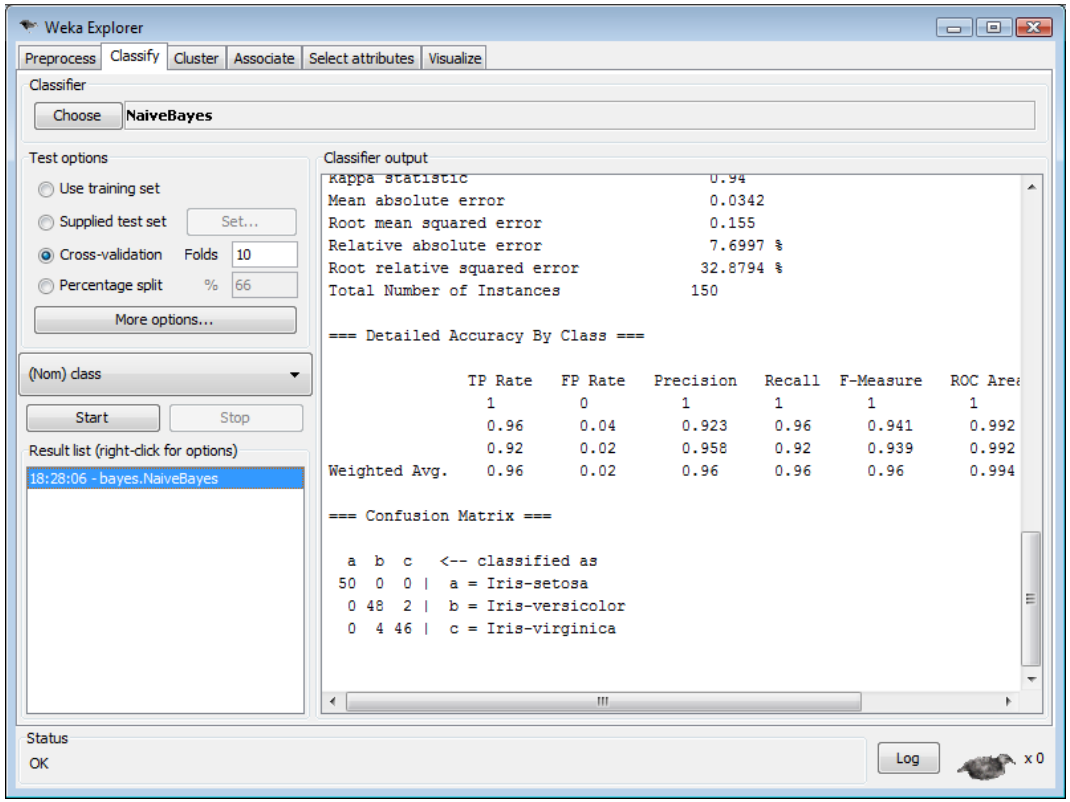
Weka - úvod

- systém pro analýzu dat a prediktivní modelování
- University of Waikato, Nový Zéland
- 1993 TCL/TK, C, Makefiles
- 1997 rozhodnutí přejít na čistou Javu
- integrována
  - RapidMiner
  - Petaho (systém business intelligence)
- GNU General Public License

Ovládání

- grafické rozhraní
  - Explorer – jednotlivé činnosti na kliknutí
  - Experimenter – systematické srovnání
  - Knowledge flow – činnosti jako tok
- příkazový řádek
- Java API

Ukázka – grafické rozhraní …



... příkazový řádek ...

```
java –classpath weka.jar
    weka.classifiers.bayes.NaiveBayes
    –t data/iris.arff
```

... Java API

```
Instances instances = new Instances(
        new BufferedReader(
        new FileReader("iris.arff")));
instances.setClassIndex(instances.numAttributes() - 1);

NaiveBayes c = new NaiveBayes();

Evaluation eval = new Evaluation(instances);

eval.crossValidateModel(c, instances, 10, new Random(1));

System.out.println(eval.toSummaryString());
System.out.println(eval.toMatrixString());
```

1. Attribute-Relation File Format (ARFF)

ARFF soubor

```
@relation spambase
% spam, non-spam
@attribute word_freq_make    real
@attribute 'char_freq_#‘     real
@attribute                   {spam, ham}
@data
0,0.64,0.64,spam
0.21,0.28,0.5,spam
0.06,0,0.71,ham
```

Čas

@ATTRIBUTE timestamp DATE "yyyy-MM-dd HH:mm:ss"
@DATA "2001-04-03 12:12:12" "2001-05-03 12:59:55"

Řídký formát

0, X, 0, Y, "class A" → {1 X, 3 Y, 4 "class A"}
0, 0, W, 0, "class B" → {2 W, 4 "class B"}

Řetězce

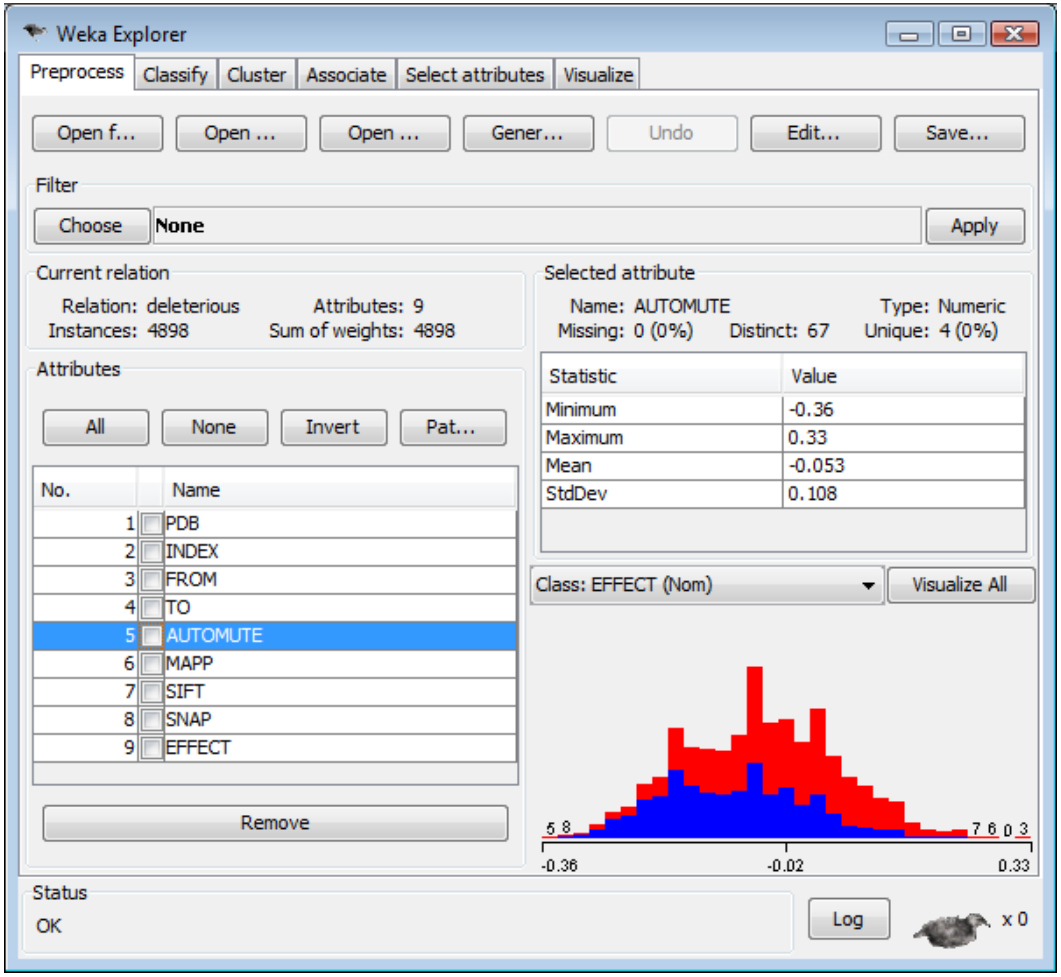
@attribute LCC string
@attribute LCSH string

@data
AG5, 'Encyclopedias and dictionaries.;Twentieth century.,

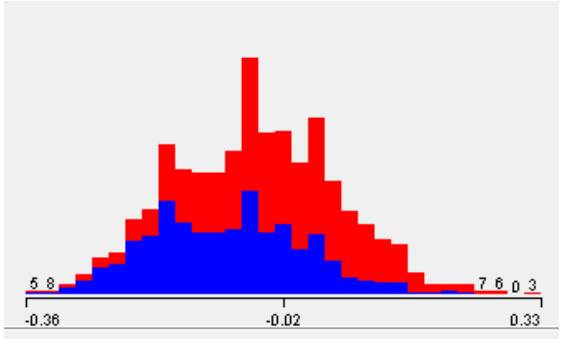
Chybějící hodnoty

4.4,?,1.5,?,Tolkien

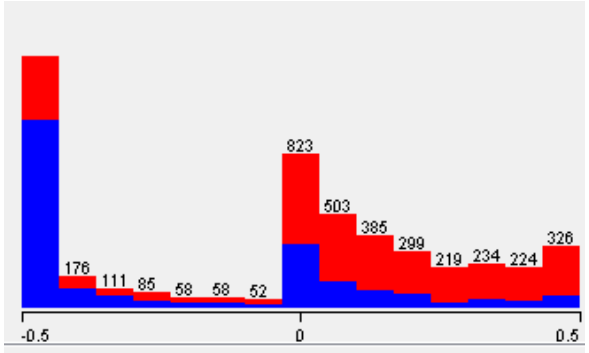
2. Předzpracování dat



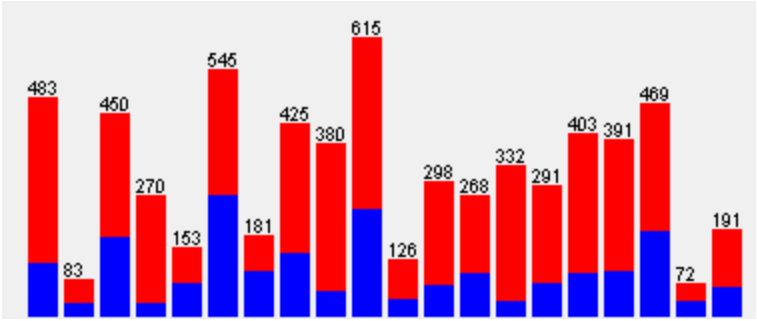
Histogramy



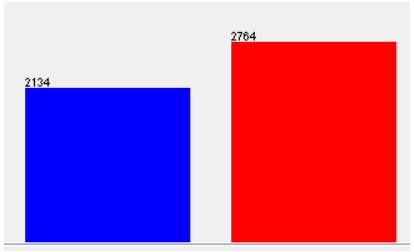
užitečný číselný atribut



podezřelý číselný atribut



20-hodnotový atribut

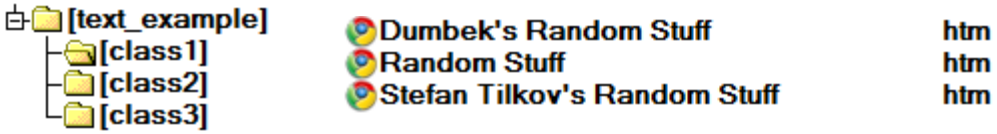


binární cílový atribut

Filtry

- Remove –V –R 1-5,8 (V = inverze, zachovej pouze tyto atributy)
- Discretize
  - některé algoritmy nepracují s čísly
  - urychlení
  - někdy i zvýšení přesnosti
- převzorkování
- doplnění chybějících atributů, odstranění chybějících hodnot
- Obfuscator
- Principal Component Analysis, Partial Least Squares
- AttributeSelection

StringToWordVector




```
TextDirectoryLoader loader = new TextDirectoryLoader();
loader.setDirectory(new File("c:/data/text_example"));
Instances dataRaw = loader.getDataSet();

ArffSaver s1 = new ArffSaver();
s1.setInstances(dataRaw);
s1.setFile(new File("c:/data/text1.arff"));
s1.writeBatch();

StringToWordVector filter = new StringToWordVector();
filter.setInputFormat(dataRaw);
Instances dataFiltered = Filter.useFilter(dataRaw, filter);

ArffSaver s2 = new ArffSaver();
s2.setInstances(dataFiltered);
s2.setFile(new File("c:/data/text2.arff"));
s2.writeBatch();

J48 c = new J48();
c.buildClassifier(dataFiltered);
System.out.println("Classifier model: " + c);
```

```
@attribute text string
@attribute class {class1,class2,class3}

@data
'<html>\n\t<head>\n\t\t<title>Dumbek\'s Rand
'<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.
'<html>\r\n\r\n<head>\r\n<meta name=\"descri
'<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1
```

```
@attribute class {class1,class2,class3}
@attribute ago numeric
@attribute align= numeric
@attribute all numeric
@attribute always numeric
@attribute business numeric
@attribute but numeric
@attribute button numeric

@data
{1 1,3 1,4 1,11 1,12 1,13 1,14 1,15.....
{10 1,34 1,37 1,49 1,50 1,53 1,99 1....
{2 1,5 1,6 1,7 1,8 1,9 1,31 1,32 1,.....
```

Klasifikace – algoritmy 1

- NaiveBayes, BayesNet, Averaged One-Dependence Estimators (AODE)
- SMO, SMOreg, LibSVM

StringKernel

```
@attribute name string
@attribute class {female, male}
@data
Midori,female
Koichi,male
```

- 291 ženských a 385 mužských jmen (odstraněno 13 univerzálních jmen)
- první spuštění: Q_2 = 63 %

Další SVM parametry a jejich optimalizace

- meta.CVParameterSelection –P "C 0.5 50000.0 5.0" ...

```
Cross-validation Parameter: '-C' ranged from 0.5 to 50000.0 with 5.0 steps
Classifier Options: -C 12500.375 ...
```

- bez predikce spolehlivosti

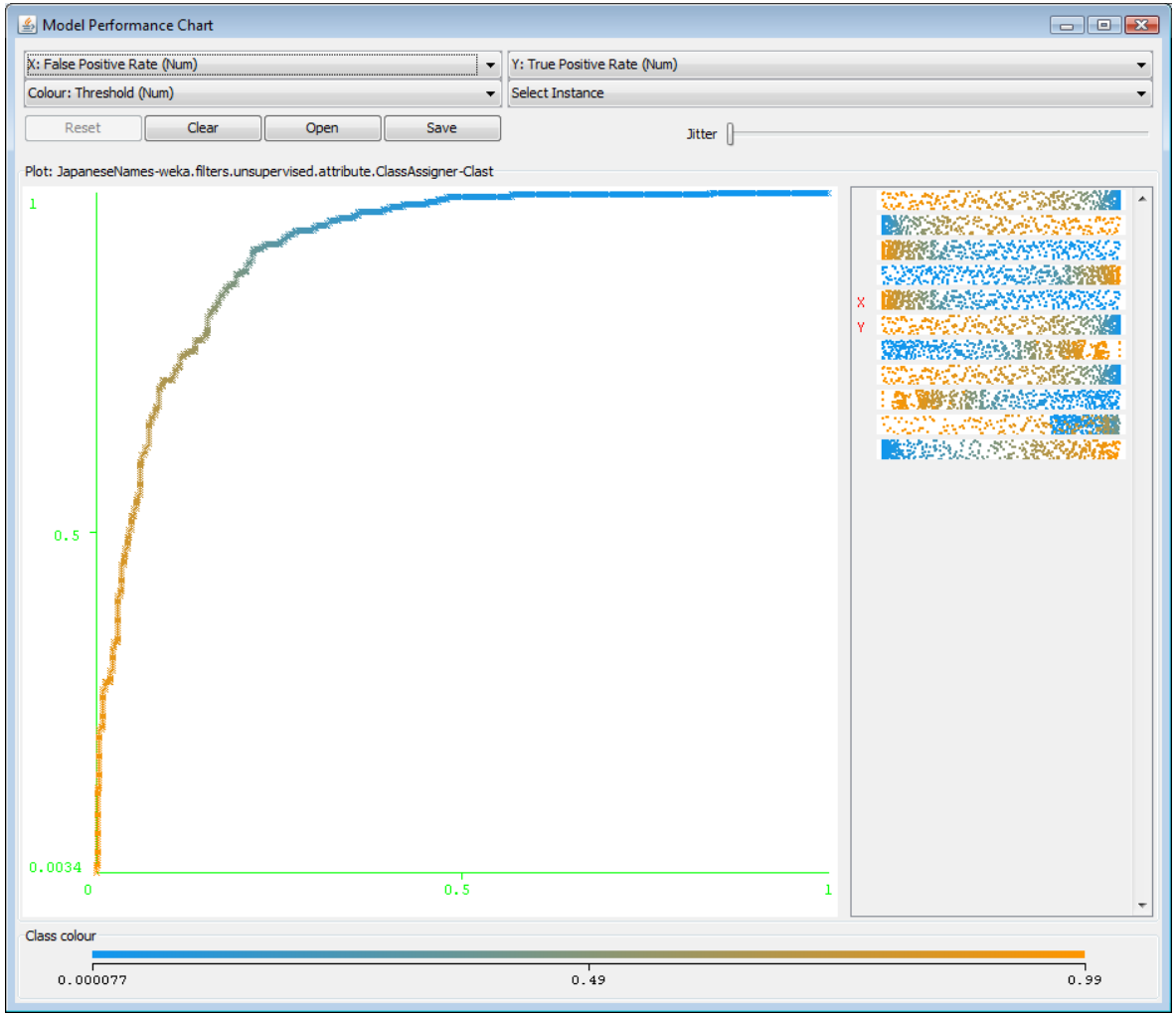
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.887	0.192	0.777	0.887	0.828	0.847	female
	0.808	0.113	0.904	0.808	0.853	0.847	male
Weighted Avg.	0.842	0.147	0.849	0.842	0.842	0.847	

- predikce spolehlivosti logistickou regresí

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.835	0.148	0.81	0.835	0.822	0.921	female
	0.852	0.165	0.872	0.852	0.862	0.921	male
Weighted Avg.	0.845	0.158	0.846	0.845	0.845	0.921	

Predikovaná spolehlivost a ROC křivka

error	prediction
	0.792
	0.61
	0.932
+	0.705
	0.959
	0.941
	0.993
	0.998
+	0.818
	0.848
	0.948
	0.517
	0.964
	0.834
+	0.928
+	0.838
	0.839
	0.963
	0.989
+	0.797



Klasifikace – algoritmy 2

- MultilayerPerceptron
  - validační množina
  - pomalé
- LinearRegression
- PLSClassifier – Partial Least Squares regression
- stromy
  - J48, RandomForest, ...
- meta
  - boosting, bagging, ...
  - ClassifivationViaRegression
  - AttributeSelectedClassifier
  - CostSensitiveClassifier

Vážení chyb

```
TP Rate
0.81
0.915
```

- meta.CostSensitiveClassifier

```
% Rows Columns
2 2
% Matrix elements
0 2
1 0
```

- cena za špatně klasifikovaný P je 2x větší než za N

Výběr atributů

Metoda hodnocení

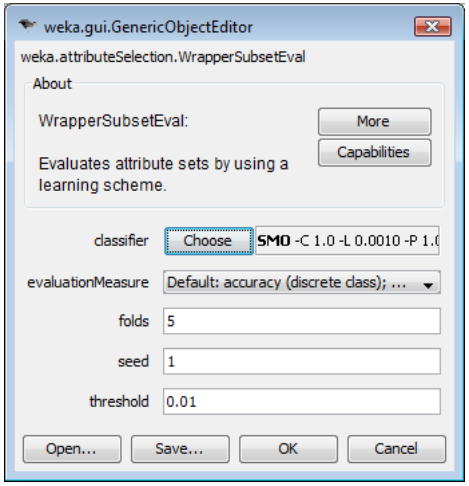
- atributů
  - ChiSquaredAttributeEval
  - SVMAttributeEval
- podmnožiny
  - CfsSubsetEval
  - WrapperSubsetEval, ClassifierSubsetEval

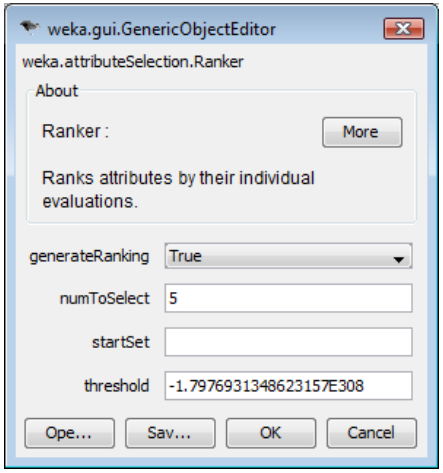
Metoda prohledávání

- pro atributy
  - Ranker
- pro podmnožiny
  - BestFirst
  - GeneticSearch

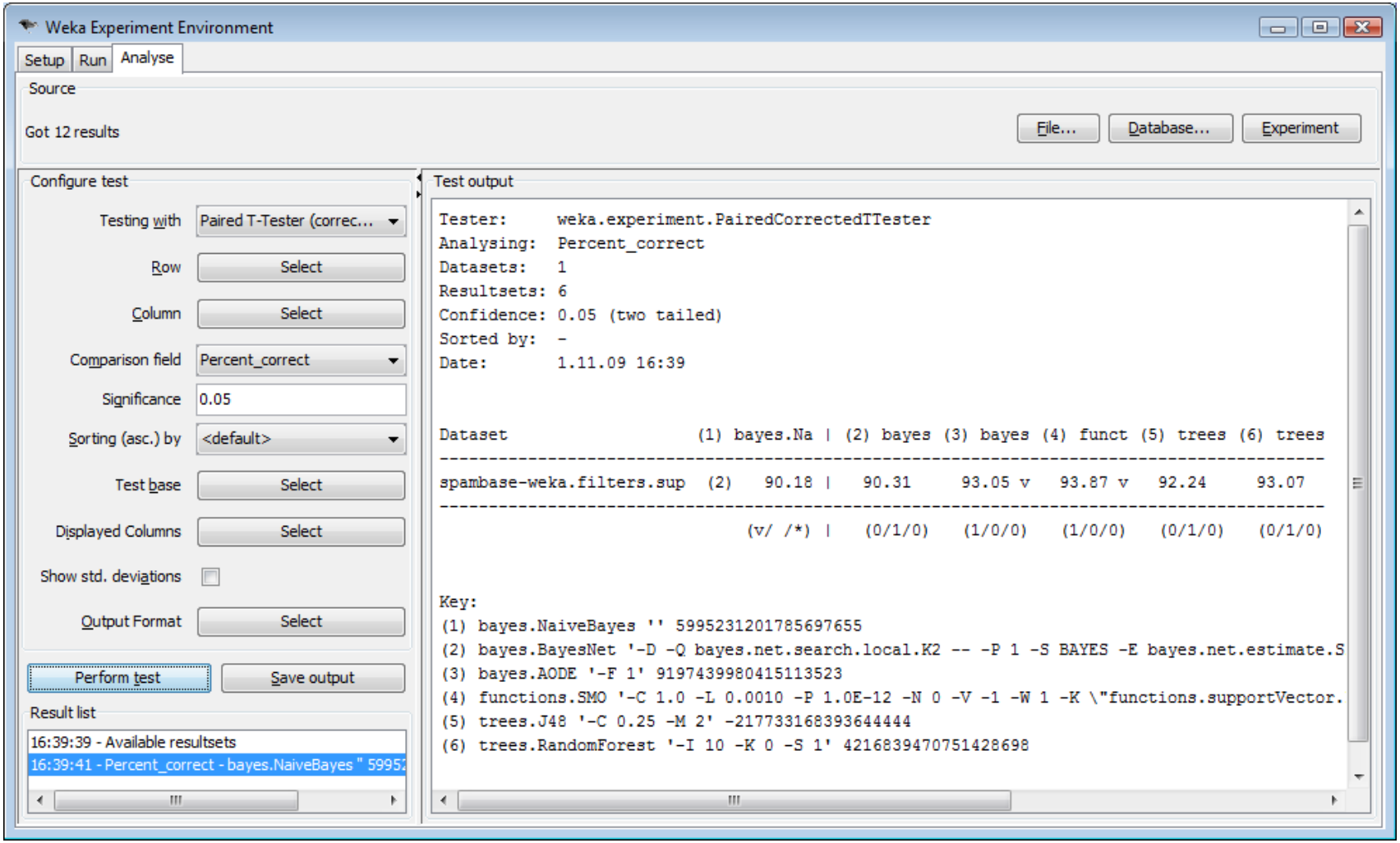
Redukce dimenzí filtrem

- Principal Component Analysis, Partial Least Squares

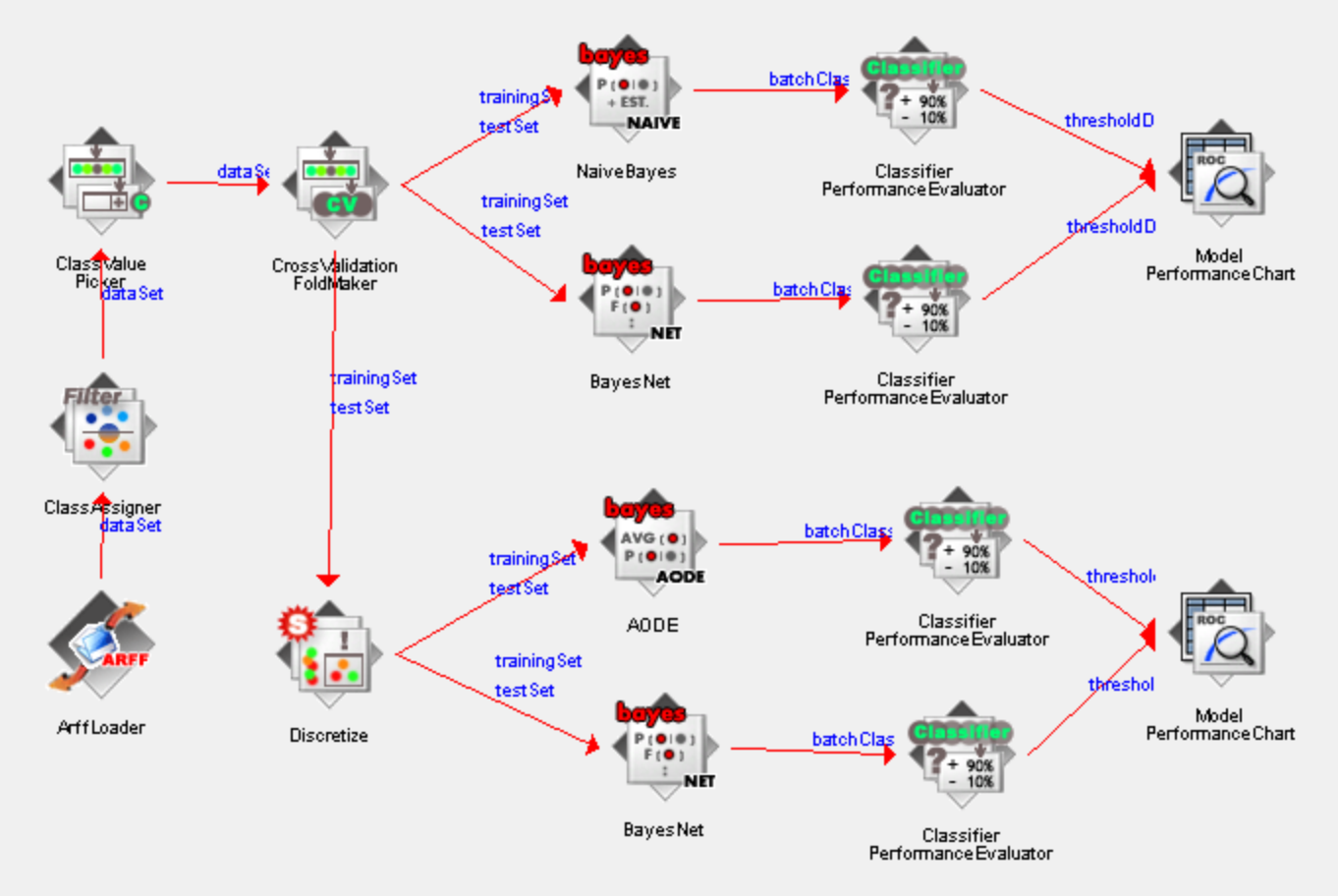




Experimenter



Knowledge Flow



Zdroje

Knihy

WEKA Manual for Version 3-7-0

Data Mining: Practical Machine Learning Tools and Techniques

Web

http://www.cs.waikato.ac.nz/ml/weka/

http://weka.wikispaces.com/

http://wekadocs.com/

http://www.hakank.org/weka/

Spuštění Weky

- ssh –X lethe
- module add java
- vytvořte si pracovní adresář (mkdir <jméno >, cd <jméno>)
- wget loschmidt.chemi.muni.cz/~tonda/w.zip
- unzip w.zip
- java –Xmx256m –jar weka.jar

Úkol 1

Explorer – J48 a SMO

- spusťte 2x Weku a Explorer
- v obou
  - otevřete spambase.arff a běžte do tabu Classify
  - v Test options, More options nastavte Output predictions na Plain text
- v prvním
  - vyberte klasifikátor trees.J48
  - klikněte do políčka vpravo od tlačítka Choose a nastavte
    - useLaplace: True
  - spusťte 10-ti násobné křížové ověření
- v druhém
  - vyberte klasifikátor functions.SMO
  - klikněte do políčka vpravo od tlačítka Choose a nastavte
    - buildLogisticModels: True
    - numFolds: 10
  - spusťte 10-ti násobné křížové ověření
- Srovnejte rychlost a přesnost obou algoritmů. Odhadněte užitečnost predikce důvěryhodnosti výsledku (=== Predictions on test data ===, sloupec prediction).

Úkol 2

Knoledge Flow - ROC křivky

- spusťte Knowledge Flow
- otevřete spam_roc.kf
- nastavte ArffLoader na spambase.arff
- klikněte pravým tlačítkem na ArffLoader, Start loading
- srovnejte ROC křivky NaiveBayese a BayesNetu (klik pravým tlačítkem na horní Model Performance Chart, Show chart)
- srovnejte ROC křivky BayesNetu a AODE.
- Po kliknutí na bod křivky se zobrazí čísla. Kolik procent spamu identifikujeme, pokud jsme ochotní tolerovat, že ve spamovém koši skončí 4 % hamu (spam = class 1, osa X: FPR = FP/N, osa Y: TPR = TP / P)?

Úkol 3

Experimenter - srovnání klasifikátorů

- spusťe Experimenter
- klikněte na tlačítko New
- Result destination: nastavte cestu a zvolte jméno nového ARFF souboru
- přidejte dataset spam_discretized.arff
- přidejte algoritmy bayes.AODE, tree.J48, tree.RandomForest
- spusťte výpočet v tabu Run
- jakmile skončí, přejďěte do tabu Analyse
- klikněte na Experiment a Perform Test
- Je přesnost některé z metod na této sadě statisticky významně lepší na hladině 0.05? Jak je to s Area_under_ROC?