



Weka

Praktické použití

Antonín Pavelka

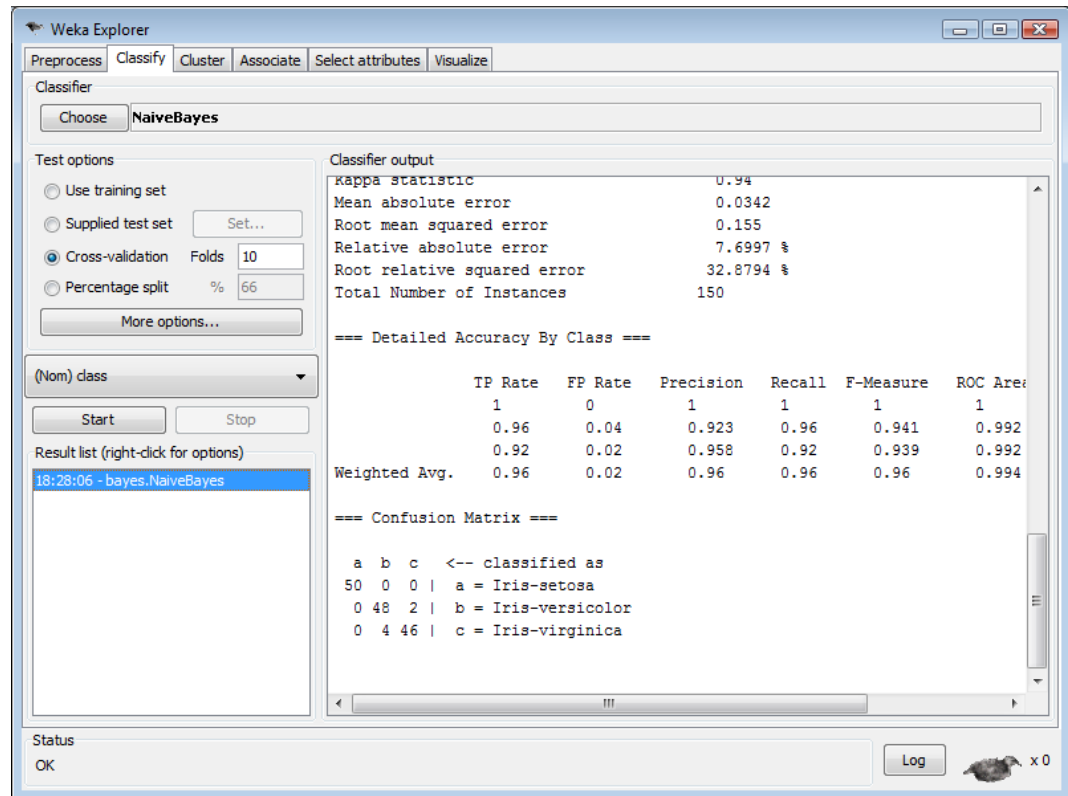
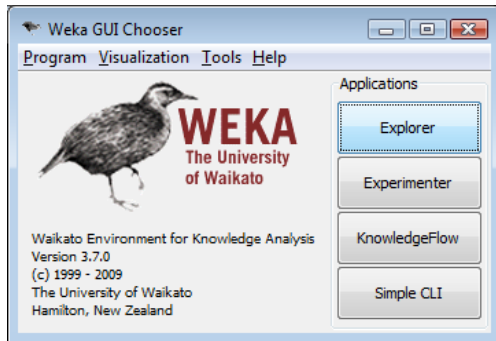
Weka - úvod

- 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

The screenshot shows the Weka Explorer interface with the 'Preprocess' tab selected. The 'Current relation' is 'deleterious' with 4898 instances and 9 attributes. The 'Selected attribute' is 'AUTOMUTE', which is numeric with 67 distinct values and 4 unique values. A histogram for 'AUTOMUTE' is displayed, showing a distribution of values from -0.36 to 0.33. The histogram has two overlapping series: a blue one and a red one. The x-axis is labeled with values -0.36, -0.02, and 0.33. The y-axis has values 5.8, 7.6, and 0.3. The status bar at the bottom shows 'OK' and a 'Log' button.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open f... | Open ... | Open ... | Gener... | Undo | Edit... | Save...

Filter

Choose **None** Apply

Current relation

Relation: deleterious Attributes: 9
Instances: 4898 Sum of weights: 4898

Selected attribute

Name: AUTOMUTE Type: Numeric
Missing: 0 (0%) Distinct: 67 Unique: 4 (0%)

Statistic	Value
Minimum	-0.36
Maximum	0.33
Mean	-0.053
StdDev	0.108

Attributes

All | None | Invert | Pat...

No.	Name
1	<input type="checkbox"/> PDB
2	<input type="checkbox"/> INDEX
3	<input type="checkbox"/> FROM
4	<input type="checkbox"/> TO
5	<input checked="" type="checkbox"/> AUTOMUTE
6	<input type="checkbox"/> MAPP
7	<input type="checkbox"/> SIFT
8	<input type="checkbox"/> SNAP
9	<input type="checkbox"/> EFFECT

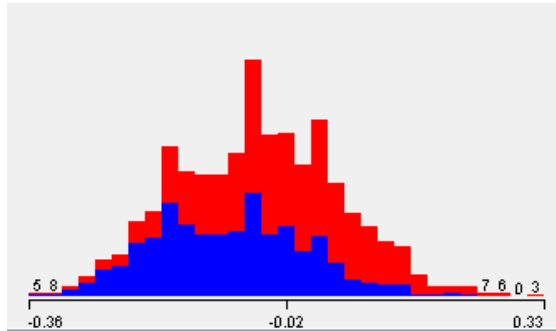
Remove

Class: EFFECT (Nom) Visualize All

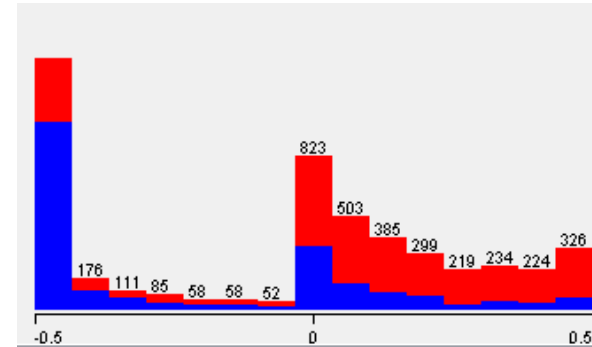
Status

OK Log x 0

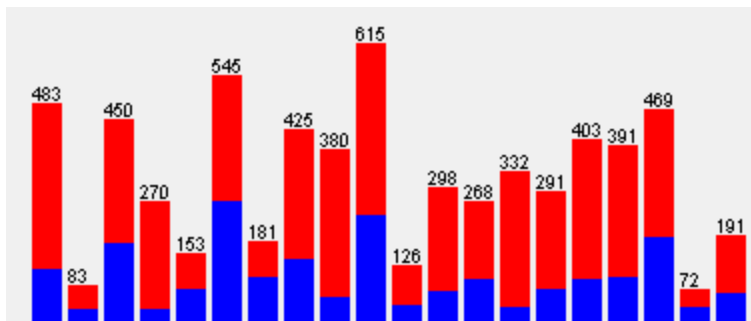
Histogramy



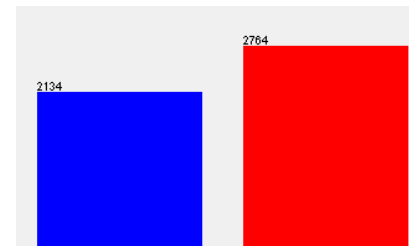
užitečný číselný atribut



podezřelý číselný atribut



20-hodnotový atribut

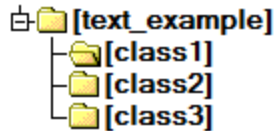


binární cílový atribut

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



- Dumbek's Random Stuff
- Random Stuff
- Stefan Tilkov's Random Stuff

htm
htm
htm

```
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();
```



```
@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
```

```
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();
```



```
@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,.....
```

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

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	

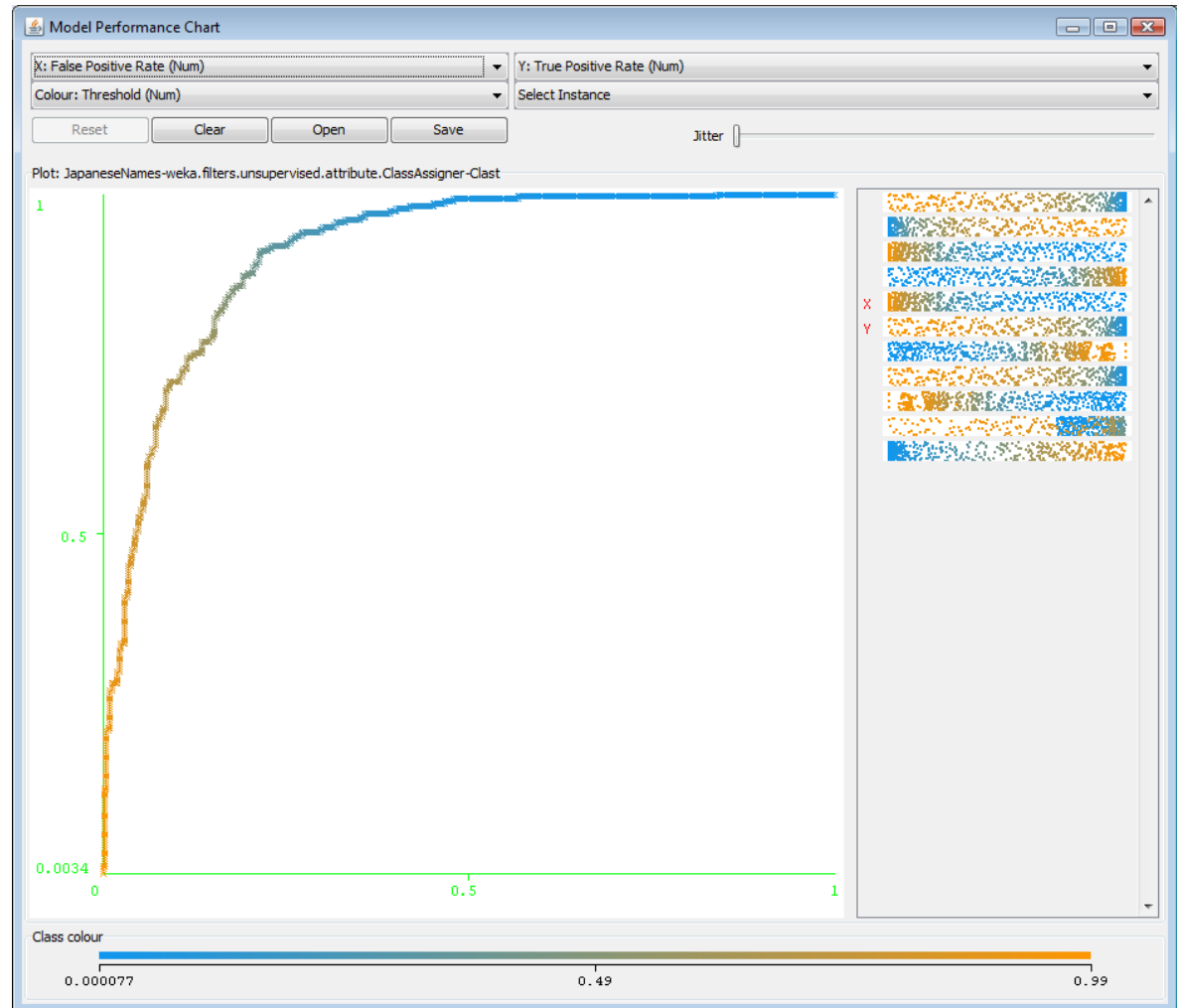
- predikce spolehlivosti logistickou regresí

	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, ...
 - ClassificationViaRegression
 - 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

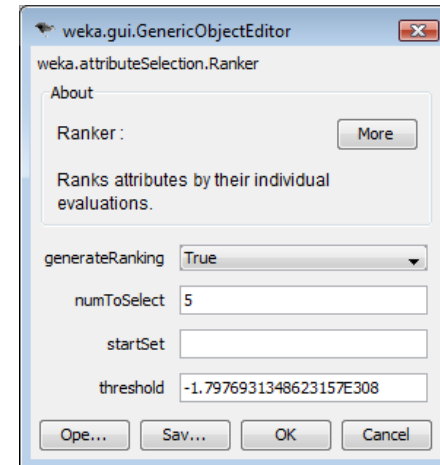
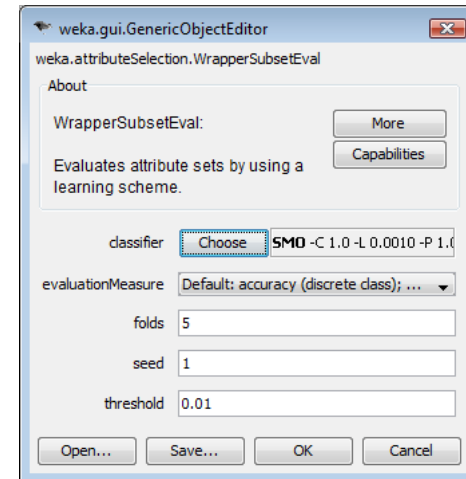
- atributů
 - ChiSquaredAttributeEval
 - SVMAttributeEval
- podmnožiny
 - CfsSubsetEval
 - WrapperSubsetEval, ClassifierSubsetEval

Metoda prohledávání

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

Redukce dimenzí filtrem

- Principal Component Analysis, Partial Least Squares



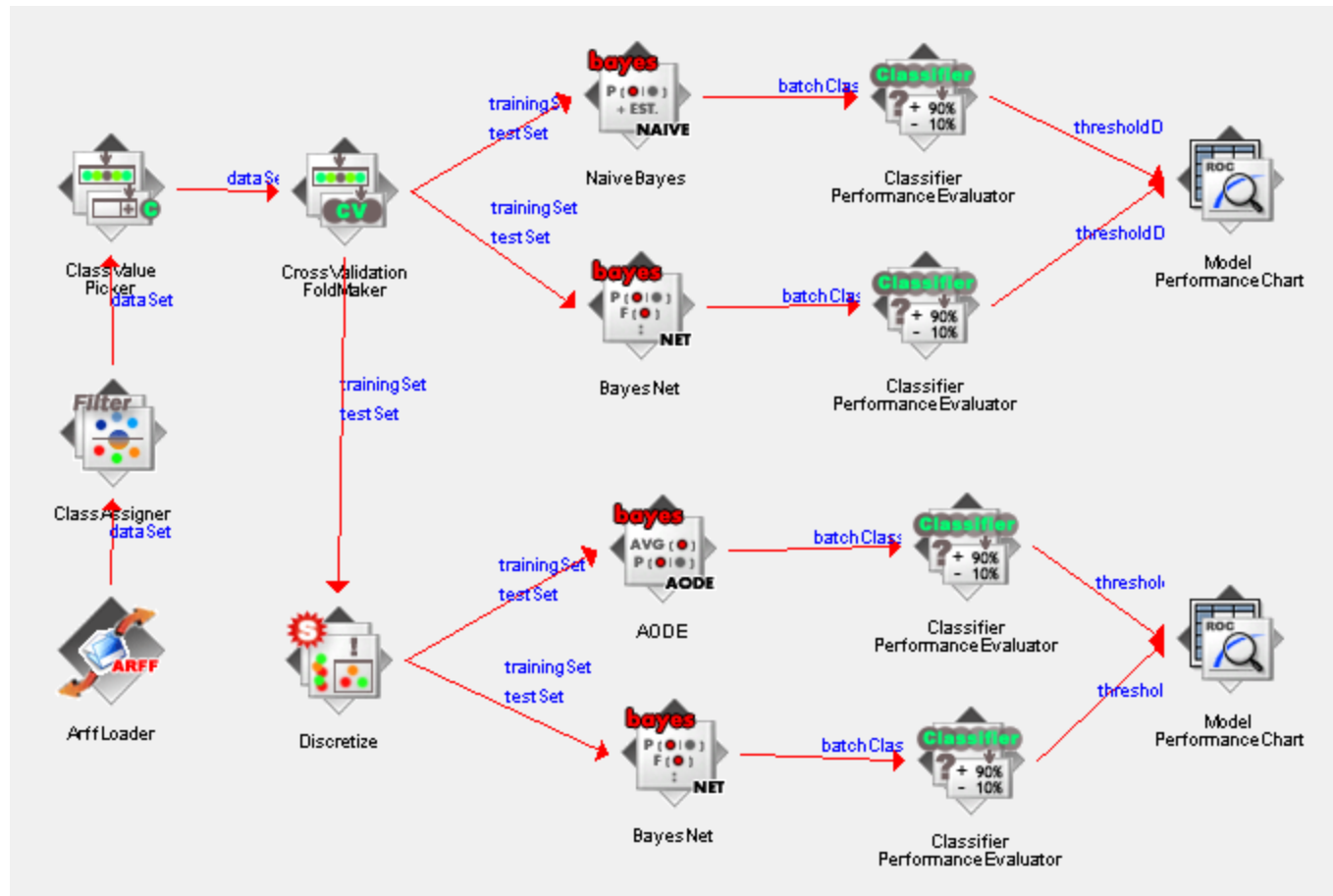
Experimenter

The screenshot displays the Weka Experiment Environment window. The interface is divided into several sections:

- Source:** Shows "Got 12 results" and buttons for "File...", "Database...", and "Experiment".
- Configure test:** A panel on the left with the following settings:
 - Testing with: Paired T-Tester (corrected)
 - Row: Select
 - Column: Select
 - Comparison field: Percent_correct
 - Significance: 0.05
 - Sorting (asc.) by: <default>
 - Test base: Select
 - Displayed Columns: Select
 - Show std. deviations:
 - Output Format: Select
- Test output:** A text area showing the following information:
 - Tester: weka.experiment.PairedCorrectedTTester
 - Analysing: Percent_correct
 - Datasets: 1
 - Resultsets: 6
 - Confidence: 0.05 (two tailed)
 - Sorted by: -
 - Date: 1.11.09 16:39
- Dataset Comparison Table:**

Dataset	(1) bayes.Na	(2) bayes	(3) bayes	(4) funct	(5) trees	(6) trees
spambase-weka.filters.sup	(2) 90.18	90.31	93.05 v	93.87 v	92.24	93.07
	(v/ /*)	(0/1/0)	(1/0/0)	(1/0/0)	(0/1/0)	(0/1/0)
- Key:**
 - (1) bayes.NaiveBayes '' 5995231201785697655
 - (2) bayes.BayesNet '-D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES -E bayes.net.estimate.S
 - (3) bayes.AODE '-F 1' 9197439980415113523
 - (4) functions.SMO '-C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.
 - (5) trees.J48 '-C 0.25 -M 2' -217733168393644444
 - (6) trees.RandomForest '-I 10 -K 0 -S 1' 4216839470751428698
- Buttons:** "Perform test" (highlighted) and "Save output".
- Result list:** A scrollable list showing "16:39:39 - Available resultsets" and "16:39:41 - Percent_correct - bayes.NaiveBayes " 5995231201785697655".

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://wekadoocs.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
- Srovnajte rychlost a přesnost obou algoritmů. Odhadněte užitečnost predikce důvěryhodnosti výsledku (=== Predictions on test data ===, sloupec prediction).

Úkol 2

Knowledge Flow - ROC křivky

- spustíte 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

- spusťte 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, přejděte 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?