

Data Mining

Lab 1:

Seturi de date: caracteristici, formate, colecții Introducere în Weka

I. Seturi de date

I.1. Caracteristici și formate

Datele de procesat pot fi de diferite tipuri

- **structurate** (e.g. matrice de date, tabele din baze de date relaționale)
- semi-structurate (e.g. fișiere XML, fișiere de tip log)
- nestructurate (e.g. documente text)

Date structurate:

- seturi de instanțe (înregistrări)
- fiecare instanță conține valori corespunzătoare unor atribute (caracteristici)
- atributele pot fi de diferite tipuri:
 - calitative (valorile lor sunt obiecte simbolice, e.g. simboluri sau șiruri de caractere):
 - nominale (e.g. naționalitate, gen, religie, stare civilă etc)
 - logice/ binare (e.g. prezența sau absența unei caracteristici specifice)
 - ordinale (e.g. nivel de satisfacție (“scăzut”, “mediu”, “ridicat”), calificativ (“insuficient”, “suficient”, ”bine”, “foarte bine”, “excelent”))
 - cantitative (valorile acestora sunt numere care aparțin unor mulțimi discrete sau unor intervale de valori continue):
 - întregi (e.g. număr de copii, vârsta în ani, număr de accesări ale unei pagini web)
 - reale (e.g. temperatura, înălțime, greutate)
- operații posibile asupra valorilor atributelor:
 - verificarea egalității, numărul de apariții (frecvența): toate tipurile de atribute (nominale, logice, binare, ordinale, întregi, reale)
 - comparare, ierarhizare: ordinale, întregi, reale
 - comparare, ierarhizare, adunare, scădere: întregi, reale
 - comparare, ierarhizare, adunare, scădere: întregi, reale, înmulțire, împărțire: reale

I.2. Colecții de date

UCI Machine Learning repository (<http://archive.ics.uci.edu/ml/>)

- cca 350 seturi de date grupate pe categorii
- un set de date conține de obicei un fișier descriptiv (“names”) și fișiere de tip csv care conțin datele propriu-zise (“data”)

KDD competitions (<http://www.kdd.org/kdd-cup>)

- date aferente competițiilor anuale din domeniul Data Mining and Knowledge Discovery organizate de către ACM

Exercițiul 1:

1. Descărcarea unui set de date:
 - a. Număr mic de atribute și de instanțe (e.g. Iris dataset)
 - b. Număr mic de atribute și mare de instanțe (e.g. DBWorld emails dataset)
 - c. Număr mare de atribute și de instanțe
2. Descărcarea unui set de date specific unei anumite prelucrări:
 - a. Clasificare
 - b. Grupare
 - c. Regresie

II. Introducere în Weka <http://www.cs.waikato.ac.nz/ml/weka/>

II.1. Ce este Weka?

WEKA = Waikato Environment for Knowledge Analysis

Pachet open-source dezvoltate la Waikato University ce încorporează o serie de algoritmi folosiți în data mining:

- Vizualizarea datelor
- Pre-procesarea datelor (75 algoritmi implementați)
- Selecția atributelor (cca 25 algoritmi implementați)
- Clasificare (mai mult de 100 algoritmi implementați)
- Grupare (cca 20 algoritmi implementați)
- Extragerea regulilor de asociere

Weka este implementat în Java și rulează pe: Windows, Linux, Mac

II.2. Cum poate fi instalat?

Descărcare de la <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>

Ultima versiune: 3.8

II.3. Care sunt principalele componente ale Weka?

- **Graphical User Interface:**
 - **Explorer** – utilizat pt a aplica prelucrări specific
 - **Experimenter** – utilizat pt a efectua analize comparative între mai multe metode pe mai multe seturi de date
 - **KnowledgeFlow** – interfață grafică ce permite efectuarea unui flux de prelucrări
 - **Workbench**
- **Simple CLI** - utilizare prin linia de comandă
- **Java API**

II.3.1. Explorer:

- Datele de prelucrat pot fi încărcate:
 - Din fișier (**Open File**) – formate uzuale: **arff** (Weka format), **csv** (comma separated values)
 - De la o adresă web (**Open URL**) dintr-o bază de date (**Open DB**)
- Pot fi generate date aleator folosind **Generate**

- Setul de date poate fi vizualizat și editat folosind [Edit](#)
- Categoriile de prelucrări:
 - Vizualizare
 - Preprocesarea datelor
 - Selecția atributelor
 - Clasificare
 - Grupare
 - Reguli de asociere
 - Predicție

II.3.2. Experimenter:

- Permite compararea a mai multor metode asociate aceluiași tip de prelucrare aplicat unuia sau mai multor seturi de date (e.g. classification)
- Analiza statistică se bazează pe testul Student (cu corecții pentru comparații multiple)

II.3.3. Knowledge Flow:

- **Workflow tipic:** “data source” ->”filter”->”classifier”->”evaluator”
- **Exemplu**
 - **Data sources: Arff loader** - [data set](#) connection to
 - **Evaluation: Cross validation foldMaker** – [training set](#), [test set](#) connection to
 - **Classifiers, Bayes, Naïve Bayes** – [batch classifier](#) connection to
 - **Classifier Performance Evaluation** – [text](#) connection to
 - **Text viewer**
- **Remark:** conexiunile se selectează prin click dreapta pe icoana cu prelucrarea
- **Activare flux:**
 - Click buton dreapta pe Arff și [Start loading](#)

II.3.4. Command line interface (exemplu pentru Weka-3-6):

- **Setare variabile de mediu pt java**
 - setenv WEKAHOME c:\Program Files\Weka-3-6
 - setenv CLASSPATH \$WEKAHOME/weka.jar:\$CLASSPATH
- **use a weka function**
 - java weka.classifiers.j48.J48 -t \$WEKAHOME/data/iris.arff

II.3.5. Java API:

- Add weka.jar to class path
- Example of using the J48 classifier


```
import weka.core.Instances;
import weka.classifiers.trees.J48;
...
Instances data = ... // from somewhere
String[] options = new String[1];
options[0] = "-U"; // unpruned tree
J48 tree = new J48(); // new instance of tree
tree.setOptions(options); // set the options
tree.buildClassifier(data); // build classifier
```

II.3.6. Formatul ARFF – Attribute Relation File Format (formatul standard al seturilor de date in Weka)

- Antet:

- Comentarii (%)
- Identificator set de date: @relation dataset_name
- Lista de atribute (fiecare atribut este caracterizat prin nume si tip): @attribute attr_name type
 Obs: în cazul atributelor discrete (care nu au multe valori) tipul este chiar setul de valori posibile; în cazul atributelor numerice discrete tipul se specifică prin integer; în cazul atributelor numerice continue tipul se specifică prin real
- Matricea de date:
 - Specificate prin @data
 - Fiecare linie corespunde unei instanțe; valorile atributelor sunt separate prin virgule

Exemplu 1:

```
@relation car
@attribute buying {vhigh,high,med,low}
@attribute maint {vhigh,high,med,low}
@attribute doors {2,3,4,5more}
@attribute persons {2,4,more}
@attribute lug_boot {small,med,big}
@attribute safety {low,med,high}
@attribute class {unacc,acc,good,vgood}
@data
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc
vhigh,vhigh,2,2,med,med,unacc
```

Exemplu 2:

```
@relation 'autoPrice.names'
@attribute symboling real
@attribute normalized-losses real
@attribute wheel-base real
@attribute length real
@attribute width real
@attribute height real
@attribute curb-weight real
@attribute engine-size real
@attribute bore real
@attribute stroke real
@attribute compression-ratio real
@attribute horsepower real
@attribute peak-rpm real
@attribute city-mpg real
@attribute highway-mpg real
@attribute class real
@data
2,164,99.8,176.6,66.2,54.3,2337,109,3.19,3.4,10,102,5500,24,30,13950
2,164,99.4,176.6,66.4,54.3,2824,136,3.19,3.4,8,115,5500,18,22,17450
1,158,105.8,192.7,71.4,55.7,2844,136,3.19,3.4,8.5,110,5500,19,25,17710
1,158,105.8,192.7,71.4,55.9,3086,131,3.13,3.4,8.3,140,5500,17,20,23875
```

Exercițiul 2:

1. Se încarcă setul [Iris](#) în Weka (click pe [Open file](#)) și se analizează sumarul:
 - a. Număr de instanțe
 - b. Număr de atribute
 - c. Pentru fiecare atribut: valori posibile și număr de apariții
2. Folosiți [Edit](#) pentru a vedea matricea de date și pentru a efectua modificări asupra datelor.
3. Utilizați [Visualize All](#) pt a vedea distribuția valorilor fiecărui atribut în fiecare dintre clase (ultimul atribut corespunde implicit clasei). Identificați perechea de atribute care ar permite separarea datelor pe clase (e.g. (sepal width, petal length)).
4. Utilizați filtrul adecvat din Weka pentru a elimina atributele care nu sunt considerate relevante pentru clasificare.

Exercițiul 3:

1. Parcurgeți aceleași etape pentru setul de date din fișierul [car.arff](#)
Obs: Principala diferență între seturile iris și car este faptul că primul conține doar atribute numerice (cu excepția celui de clasă) pe când al doilea conține atribute nominale/ordinale.
2. Încărcați fișierul [autoMpg.arff](#) (conține informații privind consumul de combustibil (mpg=miles per gallon)) și analizați conținutul folosind [Edit](#).
Obs 1: Majoritatea atributelor (incluzând cel referitor la clasa) sunt numerice; dacă atributul de clasă este numeric atunci datele sunt adecvate pentru sarcini de predicție (estimează valoarea “miles per gallon” în funcție de caracteristicile mașinii).
Obs 2: Câmpurile vide (gray) corespund unor valori absente.
 - a) Utilizați [Visualize All](#) pentru a vedea care dintre atribute sunt corelate cu valoarea mpg value
3. Încărcați fișierul [supermarket.arff](#) (conține date pentru analiza coșului de cumpărături)

Exercițiul 4: comparați performanțele mai multor tipuri de clasificatori pe două seturi de date (mai multe detalii privind metodele de clasificare vor fi prezentate în cursul 3 și lab 3)

1. Selectează [Experimenter](#) din panoul Weka
2. Click [New](#) pt a crea un nou experiment
3. Adaugă seturile de date: [iris.arff](#) și [breast-cancer.arff](#)
4. Adaugă algoritmi: [zeroR](#), [oneR](#), [naiveBayes](#), [J48](#) (oneR și zeroR sunt din grupul “Rules”, naiveBayes este din grupul “Bayes” și J48 este din grupul “Tree”)
5. [Run](#) (rulează experimentul)
6. [Analyze](#) (analizează rezultatele):
 - a. Click pe [Experiment](#)
 - b. [Perform](#) (efectuează testul statistic)
 - c. Interpretează testului statistic (folosind zeroR ca metodă de referință)
 - i. $v/*$ semnifică: v = nr cazuri (seturi de date=) pe care metoda curentă e mai bună ca cea de referință; $/$ = nr cazuri (seturi de date=) pe care metoda curentă nu diferă semnificativ de cea de referință;; $*$ = nr cazuri (seturi de date=) pe care metoda curentă e mai puțin bună ca cea de referință;