

## Data Mining

### Lab 2: Pre-procesarea datelor

---

Sumar: Pregătirea datelor pentru aplicarea tehnicilor de analiză:

1. Curățare (e.g. tratarea erorilor și a valorilor absente)
2. Transformare
  - a. Conversii între diferite tipuri de atribute (e.g. discretizare, binarizare)
  - b. Scalare
  - c. Standardizare
3. Reducerea dimensiunii
  - a. Selecția atributelor
  - b. Selecția instanțelor

#### 1. Curățarea datelor

Datele pot conține valori eronate sau absente cauzate fie de disfuncționalități ale dispozitivelor de măsurare/înregistrare, erori umane, refuzul de a completa anumite informații (în cazul informațiilor colectate pe bază de chestionare). În unele situații erorile pot fi detectate și corectate în mod automat:

- a) Valori care sunt în afara domeniilor de valori valide (de exemplu, vârstă negativă, valori ale temperaturii corporale, tensiunii arteriale etc în afara plajei de valori plauzibile). În astfel de situații există mai multe variante:
  - Eliminarea atributelor ce conțin astfel de valori (ex Weka: [Filters->Unsupervised->attribute->NumericCleaner](#))
  - Eliminarea instanțelor ce conțin valori care nu sunt valide (ex Weka: [Filters->Unsupervised->Instance->SubsetByExpression](#))
  - Eliminarea valorii eronate (și interpretarea ei ca valoare absent)
- b) Valorile absente pot fi tratate în diferite moduri:
  - Se consider valoarea absentă ca un caz specific (de exemplu “?” în Weka files) – în acest caz simbolul utilizat pentru a marca o valoare absent poate fi tratat ca o valoare distinctă și poate să apară în modelele construite pe baza datelor (de exemplu în regulile de clasificare). Aceasta este abordarea implicită în Weka.
  - Eliminarea instanțelor ce conțin valori absente (ex Weka: [Filters->Unsupervised->Instance->SubsetByExpression](#); specificând expresia `not ismissing(ATT2)` se vor elimina toate instanțele în care atributul 2 are valori absente)
  - Completarea valorilor absente cu valori estimate pe baza celor existente în setul de date (de exemplu cu media tuturor valorilor existente pentru atributul corespunzător sau cu media valorilor din instanțele similar celei care conține valoarea absent – similaritatea se măsoară folosind valorile asociate celorlalte atribute). Această abordare este cunoscută sub numele de [tehnica imputării](#).

#### Exercițiul 1:

- a) Deschideți în Weka fișierul “[autos.arff](#)” și eliminați toate instanțele pentru care valoarea atributului 25 ([price](#)) este mai mare decât 10000 (Indicație: utilizați [Filters->Unsupervised->Instance->SubsetByExpression](#) cu expresia `ATT25<=10000`)
- b) Deschideți în Weka fișierul “[breast-w.arff](#)”. Identificați atributele ce conțin valori absente și eliminați instanțele care conțin astfel de valori. Indicație: utilizați [Filters-](#)

I>Unsupervised->Instance->SubsetByExpression și o expresie de forma not ismissing(ATT<nr atribut> pentru a păstra doar instanțele complete.

## 2. Transformări

### a) Conversii între tipuri

- **Discretizare:** transformarea atributelor care iau valori într-un domeniu continuu (atribute de tip real) în attribute care iau valori într-o mulțime discretă (de tip întreg, nominal sau ordinal). Cea mai simplă abordare este de a diviza intervalul de valori [min, max] în r subintervale de aceeași lungime (de exemplu [min, min+h), [min+h, min+2h), ... [min+(r-1)h, max], unde  $h=(\text{max}-\text{min})/r$  și fiecare interval va fi asociat cu o valoare discretă (de exemplu 1, 2, 3, ...r).
- **Binarizare:** permite transformarea unui atribut nominal într-un set de attribute binare (sau logice) sau transformarea unui subset de itemi (corespunzatori unei tranzacții) într-un vector binary. De exemplu, dacă un atribut A poate lua valori din mulțimea {v1,v2,v3} atunci el va fi înlocuit cu 3 attribute binare A1, A2 și A3.

### b) Scalare

- Este utilă când attribute diferite iau valori în domenii care diferă semnificativ (de exemplu un atribut ia valori în [-0.1, 0.2] și altul ia valori în [10000, 20000])
- Cea mai simplă variantă de scalare este cea liniară prin care un atribut A având valoarea v din [min<sub>A</sub>, max<sub>A</sub>] este transformat într-un atribut care ia valori în [0,1]:

$$s(v) = \frac{v - \min_A}{\max_A - \min_A}, \text{ unde } \min \text{ și } \max \text{ corespund valorii minime respectiv celei maxime}$$

### c) Standardizare

- E utilă când interesează măsurarea abaterii valorilor față de medie în unități proporționale cu valoarea abaterii standard a datelor
- Prin standardizare, o valoare v este transformată după cum urmează:
- $st(v) = \frac{v - \text{avg}(A)}{\text{stdev}(A)}$ , unde avg(A) reprezintă media valorilor atributului A iar stdev(A) este abaterea standard a valorilor atributului A

**Exercițiu 2:** Deschideți în Weka fișierul “breast-w.arff”

- Transformați toate attributele numerice prin discretizare în: (i) 10 subintervale; (ii) 5 subintervale și salvați rezultatele în fișierele “breast-w-discrete10.arff” și “breast-w-discrete5.arff”. Indicație: Utilizați Filter->Unsupervised->Attribute->Discretize (setați numărul de subintervale ca valoare a parametrului bins (-B))
- Utilizați Weka-Experimenter pentru a analiza acuratețea următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes pentru seturile de date “breast-w.arff”, “breast-w-discrete10.arff” și “breast-w-discrete5.arff”

**Exercițiu 3:** Deschideți în Weka fișierul “car.arff”.

- Transformați toate attributele nominale prin binarizare (Filter->Unsupervised->Attribute->NominalToBinary) și salvați datele transformate în fișierul carBinary.arff
- Utilizați Weka-Experimenter pentru a analiza acuratețea următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes aplicați datelor din car.arff și carBinary.arff

### Reminder Lab 1: cum se utilizează Weka-Experimenter

1. Selectati [Experimenter](#) din panoul Weka (Weka chooser)
2. Click pe butonul [New](#) pentru a crea un nou experiment
3. Adaugati seturile de date - [Add datasets](#) (e.g. [car.arff](#) and [carBinary.arff](#))
4. Adaugati algoritmi - [Add algorithms](#): [zeroR](#), [oneR](#), [J48](#), [naiveBayes](#) (oneR și zeroR sunt din grupul “Rules”, naiveBayes este din grupul “Bayes” iar J48 este din grupul “Tree”)
5. Rulați experimentul ([Run](#))
6. [Analizați rezultatele \(Analyze\)](#):
  - a. Click pe [Experiment](#)
  - b. Aplicați testul statistic ([Perform the statistical test](#))
  - c. Interpretati rezultatele testului statistic (primul algoritm specificat, zeroR , va fi considerat ca metoda de referinta)
    - i.  $v / *$  se interpretează după cum urmează:  $v$  = număr de cazuri (datasets) pentru care metoda curentă este semnificativ mai bună decât cea de referință;  $/ =$  număr de cazuri (datasets) pentru care metoda curentă este la fel de bună ca cea de referință;  $*$  = număr de cazuri (datasets) pentru care metoda curentă este semnificativ mai slabă decât cea de referință;

### Exercițiu 4. Open in Weka the file “iris.arff” and “breast-w.arff”

- a) Aplicați scalarea asupra atributelor (Indicatie: utilizați [Filters->Unsupervised->Attribute->Normalize](#)) și salvați fișierele ca “iris\_scaled.arff” și “breast-w\_scaled.arff”
- b) Aplicați standardizarea supra atributelor (Indicatie: utilizați [Filters->Unsupervised->Attribute->Standardize](#)) și salvați fișierele ca “iris\_standardized.arff” and “breast-w\_standardized.arff”
- c) Analizați impactul scalării și standardizării asupra următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes, IB1, MultilayerPerceptron (din grupul Functions). Obs: schimbați pe [False](#) opțiunea [NormalizeAttributes](#) de la MultilayerPerceptron.

**Exercițiu 5 (temă).** Inmultiti toate valorile primului atribut din “breast-w.arff” cu 100 (Hint: use [Filters->unsupervised->attribute->MathExpression](#)) dupa care efectuați din nou toate prelucrările specificate la Exercițiul 4.

### 3. Reducerea dimensiunii datelor

Motivație: anumite attribute sau instante pot fi irelevante sau redundante. De exemplu un atribut care are aceeasi valoare pe intregul set de date ar trebui ignorat. Două attribute puternic correlate (e.g.  $A1=2*A2$ ) sunt redundante și ar fi suficient să se rețină doar unul dintre ele.

- Reducerea numărului de attribute poate reduce costul de calcul dar poate să și conducă la o îmbunătățire a performanței modelului construit pe baza datelor (e.g. clasificator)
- Reducerea numărului de instanțe din setul de antrenare poate reduce timpul necesar antrenării (sau clasificării, în cazul clasificatorilor “leneși”) dar poate să și îmbunătățească performanțele modelului (în special în cazul seturilor de date dezechilibrate)

Reducerea numărului de attribute poate fi realizată prin:

- [Selecția atributelor](#):
  - Selector de tip “filtru”: selecția se bazează doar pe analiza proprietăților datelor fiind independent de prelucrarea care va fi ulterior aplicată datelor.

- Selector de tip “wrapper”: selecția se realizează în contextul utilizării datelor într-un proces specific de analiză (calitatea subsetului selectat este evaluată prin prisma performanței unui model de analiză extras din date)
- **Proiecția datelor** pe un spațiu de dimensiune mai mică (e.g. Analiza componentelor principale - Principal Component Analysis)

Reducerea instanțelor poate fi realizată prin:

- Eliminarea unor instanțe (pe baza unor reguli specifice)
- Selecția instanțelor (utilizând selecție aleatoare cu sau fără revenire)

### 3.1. Selector de tip filtru

Selecția atributelor poate fi realizată în două moduri:

- Căutând în spațiul submulțimilor de attribute (în cazul a n attribute spațiul de căutare are cardinalul  $2^n - 2$  – se ignoră mulțimea vidă și întreg setul de attribute)
- Prin ierarhizarea atributelor în concordanță cu relevanța lor și selectarea celor mai relevante (în cazul a n attribute ierarhizate  $A_1, A_2, \dots, A_n$ , sunt n-1 variante de analizat:  $\{A_1\}, \{A_1, A_2\}, \{A_1, A_2, A_3\}, \dots, \{A_1, A_2, \dots, A_{n-1}\}$ )

Pentru fiecare dintre variante trebuie specificate:

- O măsură a relevanței unui atribut sau subset de attribute
  - *Cazul nesupervizat*: informația privind clasa atributului nu este folosită pentru a evalua relevanța (analiza se bazează în principal pe corelațiile sau similaritățile dintre attribute)
  - *Cazul supervizat*: informația privind clasa este utilizată (analiza se bazează pe corelația dintre valorile atributelor și eticheta clasei)
- O tehnică de căutare:
  - Căutare exhaustivă (toate subseturile de attribute sunt analizate)
  - Căutare greedy (forward/backward): se adaugă/elimină cel mai bun/slab atribut identificat la momentul curent
  - Căutare aleatoare
  - Căutare bazată pe o ierarhizare (attributele sunt selectate în ordinea dictată de o ierarhie stabilită anterior)

**Exercițiu 6.** Deschideți în Weka fisierul iris.arff, după care:

- a) Aplicați o tehnică de selecție a atributelor (e.g. **Select attributes** -> **CfsSubsetEval**, **GreedyStepwise**) și salvați setul redus de date (click dreapta și **Save reduced data**)
- b) Aplicați o metodă de ierarhizare a atributelor (e.g. **Select attributes** -> **InfoGainAttributeEval**, **Ranker**). Comparați rezultatul cu cel obținut la a).
- c) Analizați impactul reducerii datelor asupra performanțelor clasificatorilor (utilizând **Experimenter** la fel ca în exercițiile 2-3).

### 3.2. Proiecția datelor – analiza componentelor principale

**Scop:** se transformă datele (schimbând sistemul de axe de coordonate) astfel încât să fie eliminată corelația dintre attribute și se păstrează doar acele attribute care conserve cât mai mult din variabilitatea datelor. Această transformare poate fi realizată prin analiza în componente principale (Principal Component Analysis) parcurgând următoarele etape:

- Se calculează matricea de covarianță C a setului de date (daca datele au n attribute atunci matricea va avea dimesniunea nxn); in practică se obișnuiește să se centreze datele inainte de a construe matricea (se scade din fiecare instanță media setului de date)
- Se calculează valorile și vectorii proprii ai matricii C
- Se ordonează descrescător valorile proprii
- Se selecteaza  $m < n$  vectori proprii (cei corespunzători celor mai mari valori proprii); m se alege astfel incât datele transformate să conserve cât mai mult din variabilitatea datelor inițiale (e.g. 95%)
- Se proiectează datele pe spațiul definit de cei m vectori proprii

**Exercise 7.** Deschideți în Weka fisierul [iris.arff](#)

- a) Aplicați Principal Component Analysis (PCA) asupra datelor (e.g. [Select attributes -> Principal Components, Ranker](#) și salvați setul redus de date (click dreapta pe rezultat și [Save transformed data](#))
- b) Visualizați datele transformate și comparați-le cu cele inițiale
- c) Analizați impactul transformării PCA asupra performanțelor clasificatorilor folosind Experimenter (vexi Exerciții 2-3).

**Exercitiu 8.** Aplicați transformarea PCA asupra datelor “[arrhythmia.arff](#)” și analizați rezultatele.