

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
 - c. Proiecția datelor (analiza componentelor principale)

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 absentă)
- b) Valorile absente pot fi tratate în diferite moduri:
 - Se consideră valoarea absentă ca un caz specific (de exemplu “?” în Weka files) – în acest caz simbolul utilizat pentru a marca o valoare absentă 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 similare 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 R, Rattle or Weka) fișierul “[autos.arff](#)” și excludeți toate instanțele pentru care valoarea atributului 25 ([price](#)) este mai mare decât 10000.

(Indicație:

R: citirea unui fișier de tip arff :

```
> library(foreign)
> setwd(" ... ") # set working directory
```

```
> autos <- read.arff("datasets/autos.arff")
> autos10000 <- autos[autos$price<10000,]
```

Weka: se utilizează `Filters->Unsupervised->Instance->SubsetByExpression` specificând expresia `ATT25<=10000`)

- b) Deschideți setul de date cu informații despre pasagerii de pe Titanic ("`trainTitanic.csv`"). Identificați care dintre instanțe au valori absente și eliminați aceste instanțe.

Indicație:

R/Rattle:

```
> titanic <- read.csv("datasets/trainTitanic.csv")
> # remove instances with missing Age
> titanicAge <- titanic[!is.na(titanic$Age,)]
```

sau `Transform -> Cleanup` (în Rattle)

Intrebare: cum sunt specificate valorile absente în cazul atributului corespunzător numărului cabinei? Cum pot fi eliminate instanțele pentru care nu e completat numărul cabinei?

Weka: se utilizează `Filters->Unsupervised->Instance->SubsetByExpression` cu expresia `not ismissing(ATTx)` unde x este numărul atributului care conține valori absente.

- c) Completați valorile absente ale atributului Age în setul de date Titanic utilizând media/mediana/moda

Indicație:

R/Rattle: `Transform->Impute->Mean/Median/Mode`, selectați atributul Age attribute și click pe `Execute`

Intrebare: Care variantă este mai potrivită? Influențează completarea valorilor absente performanța unui clasificator bazat pe arbori de decizie?

2. Transformări

a) Conversii între tipuri

- **Discretizare:** transformarea atributelor care iau valori într-un domeniu continuu (atribute de tip real) în atribute 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 $h=(\max-\min)/r$) 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 atribute binare (sau logice) sau transformarea unui subset de itemi (corespunzători unei tranzacții) într-un vector binar. De exemplu, dacă un atribut A poate lua valori din mulțimea $\{v1, v2, v3\}$ atunci el va fi înlocuit cu 3 atribute binare A1, A2 și A3.

b) Scalare

- Este utilă când atribute 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 $[\min_A, \max_A]$ 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:
- $st(v) = \frac{v - avg(A)}{stdev(A)}$, unde $avg(A)$ reprezintă media valorilor atributului A iar $stdev(A)$ este abaterea standard a valorilor atributului A

Exercițiu 2: Transformați atributul [Age](#) din setul [Titanic](#) (după completarea valorilor lipsă) prin discretizare (utilizând 8 sub-intervale). Analizați impactul discretizării asupra performanței unui clasificator bazat pe arbori de decizie.

Indicație:

Rattle: [Transform](#) -> [Recode](#) -> [EqualWidth](#) -> [Number=8](#)

Exercițiu 3: Deschideți (folosind Rattle sau Weka) fișierul “[car.arff](#)”.

- a) Transformați toate atributele nominale prin binarizare și salvați rezultatul în fișierul [carBinary.arff](#)

Indicație:

Rattle: [Transform](#) -> [Recode](#) -> [Indicator variable](#)

Weka: [Filter](#)->[Unsupervised](#)->[Attribute](#)->[NominalToBinary](#)

- b) Analizați impactul clasificării asupra performanței unui clasificator

Rattle: utilizați [Model](#) -> [Tree](#) și [Evaluate](#) pentru setul initial și pentru cel transformat

Weka: Utilizați Weka-Experimenter pentru a analiza performanța următorilor clasificatori: [ZeroR](#), [OneR](#), [J48](#), [NaiveBayes](#) pentru seturile de date [car.arff](#) și [carBinary.arff](#)

Reminder Lab 1: cum se utilizează Weka-Experimenter

1. Selectați [Experimenter](#) din panoul Weka (Weka chooser)
2. Click pe butonul [New](#) pentru a crea un nou experiment
3. Adaugați seturile de date - [Add datasets](#) (e.g. [car.arff](#) and [carBinary.arff](#))
4. Adaugați 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. Interpretați rezultatele testului statistic (primul algoritm specificat, zeroR , va fi considerat ca metodă de referință)

- 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;

Studiu de caz 1. Predicția prezenței unor tipuri de alge în apă (Algae Bloom - source: [L.Torgo, Data Mining with R. Learning with Case Studies, 2011](#))

Setul de date constă din 200 de instanțe ce conțin caracteristici (în principal de natură chimică) corespunzătoare unor eșantioane de apă colectate din diferite râuri. Fiecare instanță conține, pe lângă caracteristicile râului (dimensiune și viteză a apei) valori medii ale concentrației unor substanțe chimice bazate pe măsurători efectuate pe parcursul a 3 luni (separate pentru fiecare anotimp). Fiecare instanță conține 18 atribute. Trei dintre acestea sunt nominal și specific: anotimpul, dimensiunea râului și viteza apei. Următoarele 8 atribute corespund concentrațiilor unor substanțe chimice:

- valoare maximă pH
- valoare minimă O₂ (oxigen)
- valoare medie Cl (clor)
- valoare medie NO₃⁻ (nitrați)
- valoare medie NH₄⁺ (amoniac)
- valoare medie PO₃⁴⁻ (ortofosfat)
- valoare medie PO₄ (fosfat)
- valoare medie clorofilă

Ultimele 7 atribute sunt valorile țintă și corespund frecvenței de apariție ale unor alge în eșantioanele de apă analizate. Scopul urmărit este de a prezice prezența unor tipuri de alge în funcție de caracteristicile râului și concentrațiile de substanțe chimice.

Pas 1: analizați caracteristicile datelor și decideți ce tipuri de pre-procesări sunt utile

Punct de start: CaseStudy_Algae.R

3. Reducerea dimensiunii datelor

Motivație: anumite atribute sau instanțe pot fi irelevante sau redundante. De exemplu un atribut care are aceeași valoare pe întregul set de date ar trebui ignorat. Două atribute puternic corelate (e.g. $A_1=2*A_2$) sunt redundante și ar fi suficient să se rețină doar unul dintre ele.

- Reducerea numărului de atribute 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 atribute poate fi realizată prin:

- **Selecția atributelor:**
 - Selector de tip “filtru”: selecția se bazează doar pe analiza proprietăților datelor fiind independentă 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 atribute (în cazul a n atribute spațiul de căutare are cardinalul $2^n - 2$ – se ignoră mulțimea vidă și întreg setul de atribute)
- Prin ierarhizarea atributelor în concordanță cu relevanța lor și selectarea celor mai relevante (în cazul a n atribute ierarhizate A_1, A_2, \dots, A_n , sunt $n-1$ variante de analizat: $\{A_1\}$, $\{A_1, A_2\}$, $\{A_1, A_2, A_3\}$... $\{A_1, A_2, \dots, A_{n-1}\}$)

Pentru fiecare dintre variante trebuie specificate:

- O măsură a relevanței unui atribut sau subset de atribute
 - *Cazul nesupervizat*: informația privind clasa atributului nu este folosită pentru a evalua relevanța (analiza se bazează în principal pe corelațiile sau similaritățile dintre atribute)
 - *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 atribute 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 (atributele sunt selectate în ordinea dictată de o ierarhie stabilită anterior)

Exercițiu 4.

R: Implementați o strategie greedy pentru selecția incrementală a atributelor în ordinea crescătoare a indexului Gini respectiv în ordinea descrescătoare a câștigului informațional. Set de date: [weatherNominal.csv](#)

Punct de start: [GiniIndex.r](#) și [InformationalGain.r](#)

Weka:

- a) Aplicați o strategie de selecție a atributelor (e.g. [Select attributes](#) -> [CfsSubsetEval](#), [GreedyStepwise](#)) și salvați setul de date redus (click dreapta pe rezultat și [Save reduced date](#))
- b) Aplicați o metodă de ierarhizare a atributelor (e.g. [Select attributes](#) -> [InfoGainAttributeEval](#), [Ranker](#)). Comparați rezultatul cu cel obținut la punctul a).
- c) Analizați impactul reducerii setului asupra performanței unor clasificatori (utilizând Experimenter 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 atribute și să se păstreze doar acele atribute care conservă 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 atribute atunci matricea va avea dimensiunea $n \times n$); în practică se obișnuiește să se centreze datele înainte de a construi matricea (se scade din fiecare instanță media setului de date) – în felul acesta se ajunge să se calculeze matricea de corelație
- Se calculează valorile și vectorii proprii ai matricii C
- Se ordonează descrescător valorile proprii
- Se selectează $m < n$ vectori proprii (cei corespunzători celor mai mari valori proprii); m se alege astfel încât 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

Exercițiu 5. Aplicați analiza componentelor principale asupra setului de date [iris](#)

Indicație:

R: punct de start pentru implementarea principalilor pași ai metodei: [pca.r](#)

Rattle: utilizați [Explore](#) -> [Principal Components](#) ([eig](#) corespunde variantei bazate pe descompunerea matricii de covarianță pe baza vectorilor proprii).

Weka:

- a) Utilizați [Select attributes](#) -> [Principal Components](#), [Ranker](#) și salvați setul redus (click dreapta pe rezultat și [Save transformed data](#))
- b) Visualizați datele transformate și comparați-le cu cele inițiale.
- c) Analizați impactul transformării PCA asupra performanței unor clasificatori folosind [Experimenter](#) (vezi Ex. 2-3).

Studiu de caz 2. Analiza transformărilor pentru un set de date sintetic. Analiza distribuției distanțelor dintre instanțe și calculul scorului Fisher.

Punct de start: [CaseStudy_SyntheticData.r](#)

Temă.

1. Analizați funcțiile din pachetul R [dprep](#) (pt pre-procesare).
2. Descărcați setul de date *Arrhythmia* de la *UCI Machine Learning Repository*.
 - a. Transformați toate instanțele astfel încât să aibă media 0 și abaterea standard 1.
 - b. Discretizați fiecare atribut numeric în (i) 10 sub-intervale de aceeași dimensiune (ii) 10 sub-intervale ce conțin același număr de valori.
3. Descărcați setul de date *Musk* de la *UCI Machine Learning Repository*. Determinați vectorii și valorile proprii asociate setului de date. Ce informații oferă?