

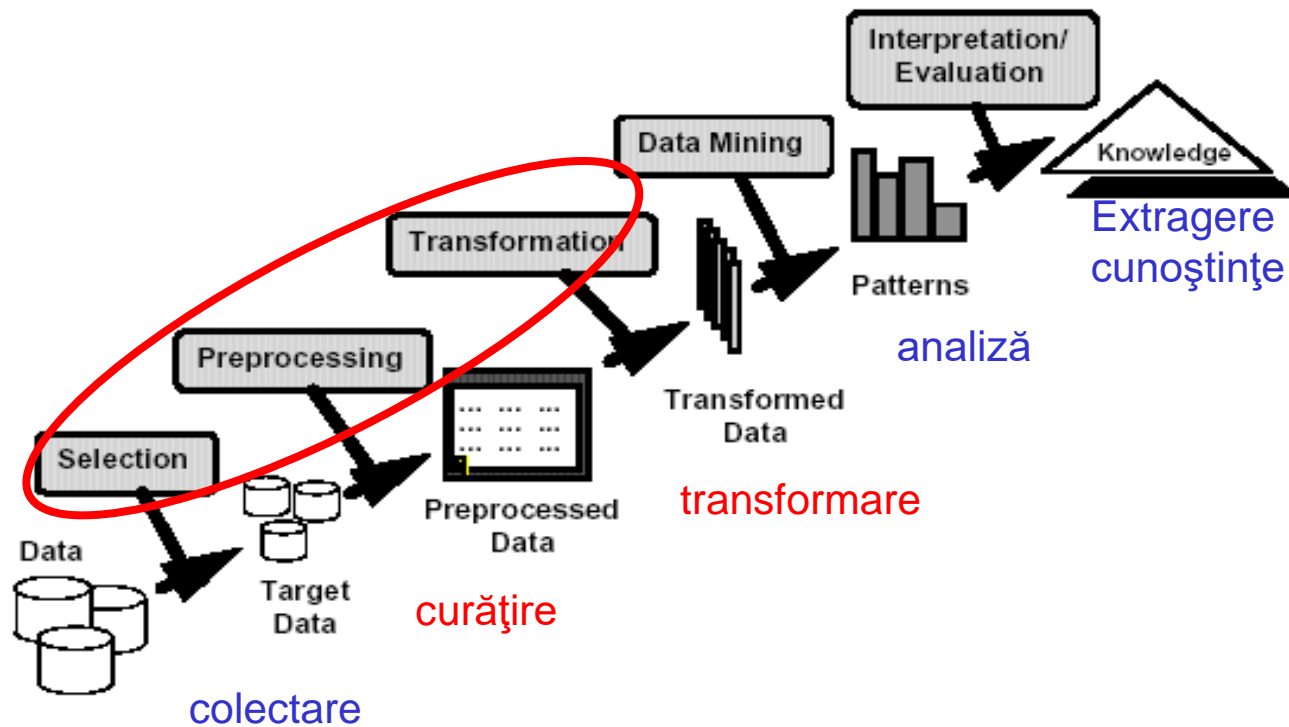
Curs 2:

Pre-procesarea datelor

Structura

- Reminder: etape extragere cunoștințe din date
- Extragerea caracteristicilor (atributelor)
- Tipuri de attribute, conversii între tipuri
- Curățirea datelor
- Reducerea dimensiunii datelor (selecția atributelor)
- Transformarea atributelor

Etape extragere cunoștințe din date



Etape extragere cunoștințe din date

Exemplu: un comerciant care deține un sistem de comerț electronic este interesat să obțină informații referitoare la comportamentul clienților săi cu scopul de a recomanda anumite produse

Surse de date:

- Fișiere de tip log cu informații de conectare

```
98.206.207.157 - - [31/Jul/2013:18:09:38 -0700] "GET
/productA.htm HTTP/1.1" 200 328177 "-" "Mozilla/5.0 (Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0
Mobile/10B329 Safari/8536.25" "retailer.net,,
```

- Informații demografice colectate în procesul de înregistrare al utilizatorilor
- stocate într-o bază de date (ex: e-mail, telefon, oraș, categorie de vârstă, categorie profesională)

Cum ar putea fi folosite aceste informații?

Etape extragere cunoștințe din date

Exemplu: un comerciant care deține un sistem de comerț electronic este interesat să obțină informații referitoare la comportamentul clienților săi cu scopul de a recomanda anumite produse

Cum ar putea fi folosite informațiile referitoare la logare și cele demografice?

Aceasta necesită:

- Stabilirea corespondenței între înregistrările din fișierele cu informații de logare și baza de date cu informații privind clienții (**problema:** datele pot fi incomplete sau chiar conține erori care îngreunează procesul → e necesară **curățirea datelor**)
- Agregarea tuturor informațiilor de logare corespunzătoare unui client (**problema:** nu toate informațiile sunt neapărat utile → ar putea necesita **selecție**)
- Integrarea informațiilor din ambele surse de date (ar putea necesita **transformarea datelor**)

Etape extragere cunoștințe din date

Principalele etape

1. Colectarea datelor (din diferite surse)

2. Pre-procesarea datelor

- **Extragerea caracteristicilor** (specifice problemei de rezolvat)
- **Curățirea datelor** (ex: eliminarea înregistrărilor eronate sau completarea valorilor absente)
- **Selecția atributelor** (ignoră attributele irelevante, redundante sau inconsistente)
- **Transformarea** datelor/ atributelor
 - Transformarea valorilor unui atribut:
 - Numeric → nominal/ordinal (e.g. valoarea vârstei este transformată într-o categorie: foarte tânăr, tânăr, bătrân, foarte bătrân);
 - Nominal → logic/binar (e.g. fiecărei valori posibile a unui atribut nominal i se asociază un atribut binar = **one-hot encoding**)
 - Transformă un set de attribute în alt set de attribute care poartă mai multă informație (e.g. explică mai bine variabilitatea din date)

3. Analiza datelor (extragerea de cunoștințe din date)

Extragerea caracteristicilor

Scop:

- Extragerea caracteristicilor semnificative din datele brute (datele pot proveni din diferite surse)

Particularitate:

- Procesul de extragere depinde de specificul domeniului și necesită expertiză în domeniul respectiv

Exemple: extragerea caracteristicilor din

- imagini
- documente (XML, PDF)
- web logs, date privind traficul în rețea

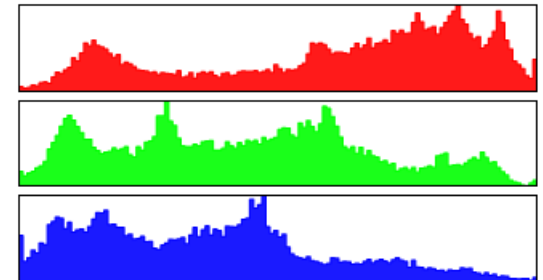
Extragerea caracteristicilor

Exemplu: Extragerea informațiilor privind textura dintr-o imagine:

Abordare bazată pe histogramă:

- Construirea histogramei color (pentru fiecare bandă de culoare și pt fiecare regiune din imagine)
 $H(v)$ =numărul de pixeli care au valoarea v
- Calcul valori statistice:
 - medie
 - varianță
 - energie
 - entropie
 - [alți indicatori statistici (skewness, kurtosis)]

Obs: dacă imaginea este partiționată în K^2 regiuni și pt fiecare regiune și fiecare bandă de culoare sunt calculate 4 mărimi statistice atunci imaginii i se asociază un vector cu $12 K^2$ caracteristici numerice



Extragerea caracteristicilor

Exemplu: Extragerea caracteristicilor de textură dintr-o imagine:

Alte variante (ex: [http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf]):

- Matrici de co-ocurență

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Image example

$i \backslash j$	0	1	2	3
0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

Construction of co-occurrence matrix

Nr pixeli vecini
(pe o direcție
care au
valorile
specificate)

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

$b_{1,0^\circ}$ Vecini pe orizontală

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

$b_{1,90^\circ}$ Vecini pe verticală

Extragerea caracteristicilor

Exemplu: Extragerea caracteristicilor de textură dintr-o imagine:

Alte variante (ex: [http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf]):

- Matrici de co-ocurență

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Image example

$i \backslash j$	0	1	2	3
0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

Construction of co-occurrence matrix

Nr pixeli vecini
(pe o direcție
care au
valorile
specificate)

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

$b_{1,0^\circ}$ Vecini pe orizontală

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

$b_{1,90^\circ}$ Vecini pe verticală

Nr pixeli vecini
pe verticală în
care un pixel
egal cu 2 este
urmat de un
pixel egal cu 3

Extragerea caracteristicilor

Exemplu: Extragerea caracteristicilor dintr-un document:

1. XML - date semistructurate

```
<PersonalData><PersonDescriptors><DemographicDescriptors><Nationality>francaise</Nationality>
</DemographicDescriptors>
<BiologicalDescriptors><DateOfBirth>1978-01-16</DateOfBirth>
<GenderCode>1</GenderCode>
</BiologicalDescriptors>
</PersonDescriptors>
</PersonalData>
```

...

Prin parsare, se pot extrage caracteristicile demografice:

Nationality	Date of birth	Gender
Francaise	1978-01-16	1

Extragerea caracteristicilor

Exemplu: Extragerea caracteristicilor dintr-un document:

2.Fișier text – date nestructurate

Exemplu (abordarea bazată pe bag-of-words):

“In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.”

a) Eliminarea cuvintelor de legătură (stop words)

“In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.”



“document classification bag words sparse vector occurrence counts words
sparse histogram vocabulary computer vision bag visual words vector
occurrence counts vocabulary local image features.”

Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document – fișier text: abordarea de tip bag-of-words

b) Reducerea cuvintelor la rădăcina lor – stemming (algorithm Porter)

“document classification bag words sparse vector occurrence counts words
sparse histogram vocabulary computer vision bag visual words vector
occurrence counts vocabulary local image features”



[<http://textanalysisonline.com/nltk-porter-stemmer>]

“document classif bag word spars vector occur count word spars histogram
vocabulary comput vision bag visual word vector occur count vocabulary local
imag featur”

Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document – fișier text: abordarea de tip bag-of-words

c) Calculul frecvențelor:

“document classif bag word spars vector occur count word spars histogram
vocabulari comput vision bag visual word vector occur count vocabulari local
imag featur”

Caracteristici extrase:

(bag,2), (classif,1), (comput,1), (count,2), (document,1), (featur,1),
(histogram,1), (imag,1), (local,1), (occur,2), (spars,2), (vector,2), (vision,1),
(visual,1), (vocabulari,2), (word,3)

Caracteristici extrase → vector numeric

Extragerea caracteristicilor

Caracteristici extrase → vector numeric

codificarea numerică a textului poate fi realizată și prin modele bazate pe rețele neuronale (text embeddings):

- **word2vec** (2013 - <https://wiki.pathmind.com/word2vec>)
- **GloVe** (2014 - <https://nlp.stanford.edu/projects/glove/>)
- **BERT** (2018 - <https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/>)
- ...
- **Large Language Models**
 - GPT (→ GPT 4) – OpenAI - <https://openai.com/gpt-4>
 - LLAMA - Meta - <https://ai.meta.com/blog/llama-2/>
 - Gemini (2023) – Google - https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf

Tipuri de caracteristici/ attribute

- Numerice (cantitative):

Exemple: vârsta, greutate, preț, cantitate, temperatură etc.

Specific:

- Valorile atributelor cantitative sunt numere (întregi sau reale)
 - Discrete
 - Continue
- Se poate defini o ordine între valori (i.e. se poate calcula: minim, maxim, mediana, cuantile)
- Se pot efectua operații aritmetice:
 - Calcul medie, varianță și alți indicatori statistici
 - Alte operații: adunare, scădere, înmulțire, împărțire etc (e.g. valoare = preț*cantitate)

Obs: un caz particular este reprezentat de valorile de tip dată calendaristică sau oră (ex: 1975-01-16); are sens să se compare sau să se calculeze diferența dintre date dar nu are sens să se înmulțească)

Tipuri de caracteristici/ attribute

- Ordinale (valori discrete aparținând unei mulțimi ordonate)

Exemple:

- Nivele de calitate (e.g: inacceptabil, acceptabil, bun, foarte bun, excelent)
- Nivele ale unei caracteristici (e.g: foarte scăzut, scăzut, mediu, ridicat, foarte ridicat)

Specific:

- Valorile pot fi numere, simboluri, șiruri
- Există relație de ordine pe mulțimea valorilor (i.e. se poate calcula minim, maxim, mediana și se pot ordona valorile)
- Nu are sens să se efectueze operații aritmetice

Tipuri de caracteristici/ attribute

- Nominale/ categoriale (valori discrete aparținând unei mulțimi pe care nu este definită o relație de ordine relevantă pentru semnificația valorilor)

Exemple:

- Gen (e.g: female, male)
- Rasă (e.g. caucaziană, asiatică, africană etc)
- Stare civilă

Specific:

- Valorile unei astfel de caracteristici pot fi simboluri, șiruri de caractere etc
- Nu se pot aplica operații aritmetice sau de ordonare
- Operații posibile:
 - Verificare egalitate
 - Calcul frecvențe

Tipuri de caracteristici/ attribute

- Binar (doar două valori posibile: {0,1} sau {False, True})
 - Se utilizează pentru a codifica absența/prezența unor caracteristici
 - Permite specificarea unor submulțimi (interpretate ca funcții indicator)

Exemplu: set de tranzacții

T1: {lapte, pâine, carne}

T2: {pâine, apă}

T3: {unt, carne}

T4: {apă}

Trans.	pâine	unt	carne	lapte	apă
T1	1	0	1	1	0
T2	1	0	0	0	1
T3	0	1	1	0	0
T4	0	0	0	0	1

- **Obs:** este un exemplu de conversie a datelor (de la nominal la binar)

Conversii între tipuri

Conversia unui atribut numeric continuu într-unul nominal sau numeric discret (discretizare)

- **Motivație:** anumite tehnici de data mining pot fi aplicate doar pt date categoriale sau numerice discrete
- **Idee principală:**
 - Domeniul de valori se împarte în subdomenii
 - Se asignează o valoare fiecărui subdomeniu

Exemplu: considerăm atributul “vârsta” care ia valori în intervalul $[0,100]$; atributul numeric se poate transforma într-unul categorial după cum urmează

Subdomeniu	Valoare
$[0, 10)$	1
$[10,20)$	2
$[20,30)$	3
$[30,40)$	4
...	
$[90,100]$	10

Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

Obs:

- Prin discretizare se pierde o parte din informație
- O discretizare uniformă (ca în ex. anterior) nu e întotdeauna cea mai adecvată (de exemplu intervalul $[90,100]$ conține de regulă mai puține valori decât celelalte intervale).

Alte variante:

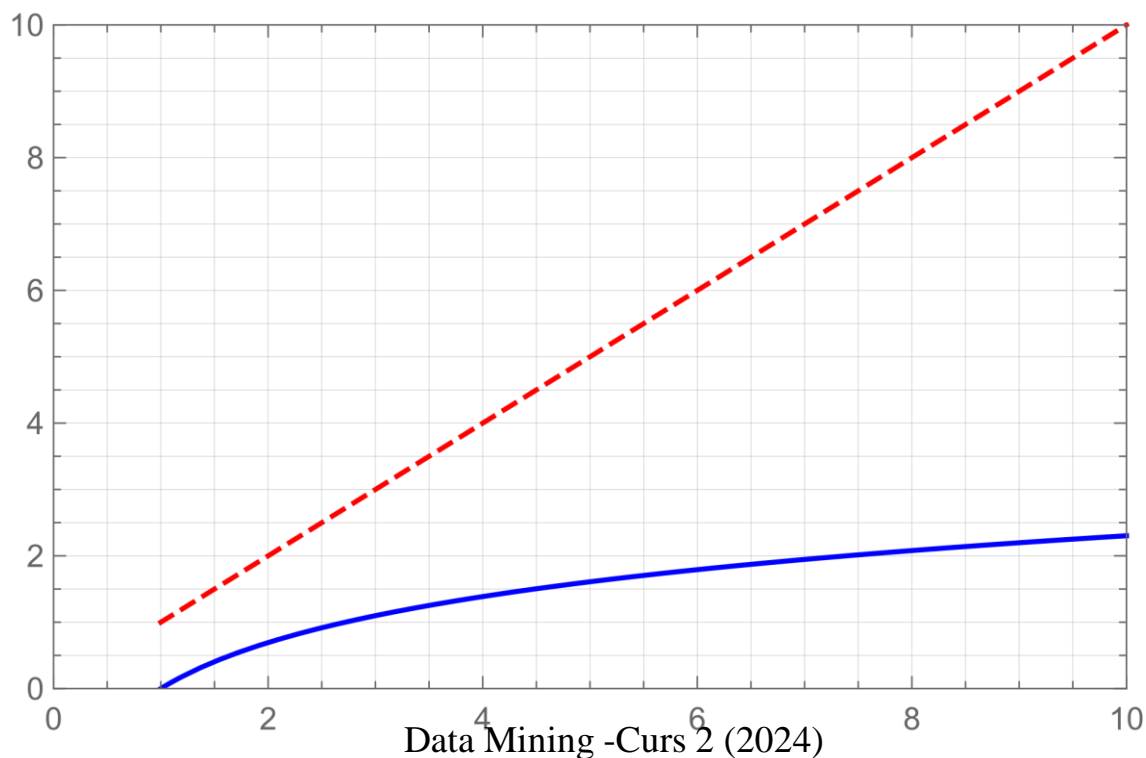
- **Equi-log:** domeniul $[a,b]$ este divizat în K subdomenii $[a_1,b_1), [a_2,b_2), \dots, [a_K,b_K]$ a.î. $\log(b_i) - \log(a_i)$ este constant (în loc de $b_i - a_i$)

Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

Alte variante:

- **Equi-log:** domeniul $[a,b]$ este divizat în K subdomenii $[a_1,b_1), [a_2,b_2), \dots, [a_K,b_K]$ a.î. $\log(b_i) - \log(a_i)$ este constant (în loc de $b_i - a_i$)



Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

- **Equi-depth:** fiecare subdomeniu are același număr de înregistrări
- **Equi-label:** fiecare subdomeniu conține valori care aparțin aceleiași clase (în contextul unei probleme de clasificare pentru care se cunoaște un set de date etichetate)

Exemplu (valorile atributului “vârsta” sunt ordonate crescător):

Vârsta: 15, 16, 16, 20, 20, 20, 25,26,27,30,30,31

Clasa: c1, c2, c2, c1, c1, c1, c2,c2,c1, c2,c2,c1

Equi-depth: [15,18), [18, 22.5), [22.5, 28.5), [28.5, 31)

Equi-label: [15,15.5), [15.5, 18), [18, 22.5), [22.5, 26.5), [26.5, 28.5), [28.5, 30.5), [30.5, 31)

Conversii între tipuri

Conversia atributelor nominale în attribute binare (binarizare)

Motivație: există tehnici de data mining (e.g. rețelele neuronale) care nu pot prelucra direct attribute nominale

Procedura (one-hot encoding): un atribut nominal A care ia valori în mulțimea $\{v_1, v_2, \dots, v_r\}$ este transformat în r attribute binare Av_1, Av_2, \dots, Av_r a.î. într-o instanță dată doar unul dintre attribute va avea valoarea 1 iar toate celelalte vor avea valoarea 0.

Exemplu: considerăm atributul “maintenance price” din setul “car” (UCI Machine Learning @attribute maint {vhigh,high,med,low} și valori corespunzătoare câtorva instanțe:

	vhigh	high	med	low
high	0	1	0	0
medium	0	0	1	0
low	0	0	0	1
Very high	1	0	0	0

Curățirea datelor

Scop: eliminarea erorilor și a inconsistențelor din date

Tipuri de erori:

- Valori eronate
- Valori absente

Cauze ale erorilor:

- Defecte în dispozitivele de înregistrare a datelor (e.g. senzori)
- Erori umane (e.g. completare greșită)
- Absența răspunsurilor (e.g. date confidențiale)

Pacient	Vârsta	Înălțime [cm]	Greutate[kg]
P1	20	170	60
P2	10	1.30	30
P3	22	165	
P4	8	190	80

Curățirea datelor

Scop: eliminarea erorilor și a inconsistențelor din date

Tipuri de erori:

- Valori eronate
- Valori absente

Cauze ale erorilor:

- Defecte în dispozitivele de înregistrare a datelor (e.g. senzori)
- Erori umane (e.g. completare greșită)
- Absența răspunsurilor (e.g. date confidențiale)

Pacient	Vârsta	Înălțime [cm]	Greutate[kg]
P1	20	170	60
P2	10	1.30	30
P3	22	165	?
P4	8	190	80

Valoare eronată

Valoare absentă

Date inconsistente

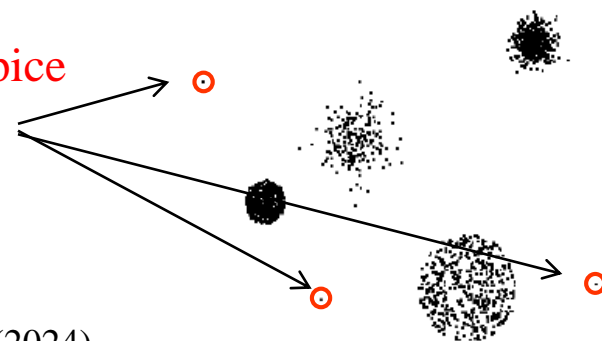
Curățirea datelor – valori eronate

Descoperirea și corecția valorilor eronate:

- Utilizând cunoștințe specifice domeniului (e.g. se pot defini domenii de valori normale)
- Căutând inconsistențe între valorile aceluiași atribut folosind diferite surse de date (e.g. Numele unei persoane poate fi specificat în mai multe moduri, “Ioan Popescu”, “I. Popescu”, “Ioan Popesu”; rasa unei persoane este specificată diferit în diferite instanțe)
- Utilizând o abordare statistică (e.g. se presupune că datele sunt generate de o distribuție normală iar valorile atipice sunt considerate erori)

[Tan, Steinbach, Kumar – Introduction to Data Mining]

Excepții,
valori atipice



Curățirea datelor – valori absente

Cauze ale valorilor absente:

- Omitere în procesul de colectare
- Informații care nu sunt furnizate (e.g. vârsta sau genul într-un chestionar)
- Informații nerelevante în anumite contexte (e.g. valoarea venitului în cazul unor copii)

Curățirea datelor – valori absente

Tratarea valorilor absente:

- **Eliminarea** înregistrărilor care conțin valori absente
- **Asignarea unor valori specifice** (e.g. valoarea absentă este marcată cu 0 iar 0 este considerată o valoare posibilă pt acel caz)
- **Estimarea** valorii absente (o astfel de abordare este denumită **imputare**) utilizând valori corespondente din înregistrări “similare”.

În exemplul anterior s-ar putea folosi 60 (întrucât P1 și P3 sunt similare în raport cu celelalte atribute). Dacă sunt mai multe instanțe “similare” atunci se poate folosi valoarea medie, mediana sau moda atributului din toate instanțele similare.

Obs: Pentru imputarea valorilor numerice se pot folosi modele de regresie.

Selecția atributelor

Scop:

- Reducerea dimensiunii datelor
- Îmbunătățirea modelului de analiză a datelor (prin eliminarea atributelor redundante)

Exemple:

- Atribute irelevante (e.g. ID – identificator; attributele care au aceeași valoare sau attributele care au valori distincte pt toate instanțele)
- Atribute corelate (e.g. $BMI = \text{weight} / \text{height}^2$)

Pacient	Vârsta	Înălțime [m]	Greutate [kg]	BMI	ID	Clasa
P1	20	1.70	60	20.8	111	normal
P2	15	1.30	30	17.8	222	subponderal
P3	22	1.65	100	36.7	333	obez
P4	48	1.90	80	22.2	444	normal

Obs: în practică relația dintre atribute este ascunsă astfel, că nu este evident criteriul de selecție

Selecția atributelor

Scop:

- Reducerea dimensiunii datelor
- Îmbunătățirea modelului de analiză a datelor (prin eliminarea atributelor redundante)

Componente ale unei metode de selecție a atributelor:

- Criteriul de selecție
- Metoda de căutare (în spațiul de submulțimi ale atributelor)

Variante:

- În funcție de datele utilizate în calculul criteriului de selecție
 - Metode **nesupervizate** de selecție (e.g. utilizate în contextul grupării datelor)
 - Metode **supervizate** de selecție (e.g. utilizate în contextul clasificării datelor)
- În funcție de modul de calcul al criteriului de selecție
 - De tip **filtru**
 - De tip **wrapper**

Selecția atributelor

Căutarea în spațiul atributelor:

- Considerăm o matrice de date cu n attribute
- Spațiul de căutare (toate submulțimile posibile de attribute) are dimensiunea 2^n

Abordări posibile:

- **Căutare exhaustivă:** se analizează impactul fiecărei submulțimi de attribute asupra rezultatului; e fezabilă doar dacă n este relativ mic
- **Strategie incrementală (forward):**
 - Se pornește cu un set vid de attribute
 - Se adaugă secvențial câte un nou atribut (se analizează impactul fiecăruia dintre attributele rămase și **se selectează cel mai bun**) – dacă adăugarea unui nou atribut nu îmbunătățește calitatea rezultatului / modelului atunci procesul se oprește

Selecția atributelor

Căutarea în spațiul atributelor:

- Considerăm o matrice de date cu n attribute
- Spațiul de căutare (toate submulțimile posibile de attribute) are dimensiunea 2^n

Abordări posibile:

- **Strategie decrementală (backward):**
 - Se pornește cu întreg setul de attribute
 - Se elimină secvențial câte unul dintre attribute
 - Cel prin a cărui eliminare se obține cea mai mare îmbunătățire a performanței
 - sau
 - Cel prin a cărui eliminare se obține cea mai mică reducere a performanței

Selecție / ierarhizare / ponderare

- În anumite situații este mai util doar să se **ierarhizeze** attributele în ordinea descrescătoare a relevanței și să fie lăsată la latitudinea utilizatorului decizia de a include/exclude un atribut
- Criteriul de ierarhizare este similar celui de selecție (are ca scop să exprime relevanța atributului în contextul problemei de analiză tratate)
- Ierarhizarea poate fi realizată asignând **ponderi** atributelor (o valoare mai mare a ponderii sugerează că atributul este mai important)
 - Estimarea ponderilor conduce de regulă la necesitatea de a rezolva o problemă de optimizare (e.g. determinarea ponderilor care **minimizează pierderea de informație** sau **maximizează calitatea**)
 - Ponderile sunt importante în cazul în care tehnica de analiză se bazează pe calculul unor măsuri de similaritate (e.g. clasificatori de tip “cel mai apropiat vecin”, clustering)

Exemplu: distanța euclidiană ponderată $d_w(x, y) = \sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2}$

Selecția atributelor

Criteriu de selecție - cum se poate evalua un subset de attribute (sau valorile corespunzătoare ale ponderilor)

- **Selecție de tip 'filtru'**
 - Selecția se bazează pe (co)relația dintre:
 - attribute (context nesupervizat)
 - attribute și etichete ale claselor (context supervizat)
- **Selecție 'încorporată' în construirea modelului (wrapper)**
 - Calitatea subsetului de attribute este estimată pe baza performanței clasificatorului sau a modelului de grupare construit pe baza subsetului de attribute

Selecția atributelor

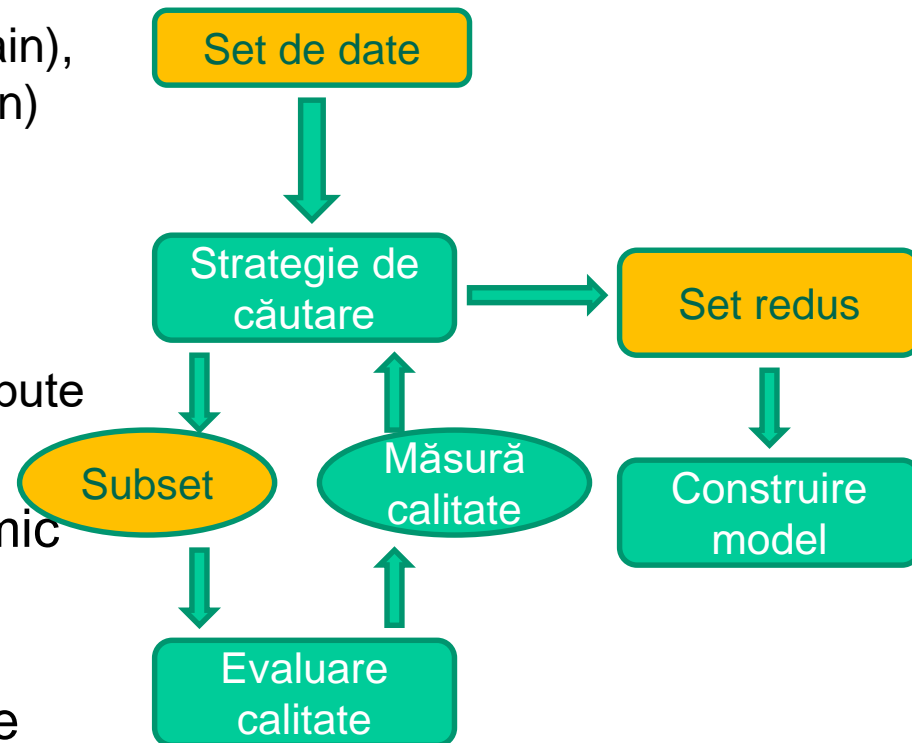
Abordare de tip filtru

Criterii bazate pe date

- Câștig informațional (Informational Gain), Informație mutuală (Mutual Information)
- Gini index
- Compacitate (within-class)
- Separare (between-classes)
- Corelare între etichete de clase și atribute

Avantaj: cost computațional relativ mic

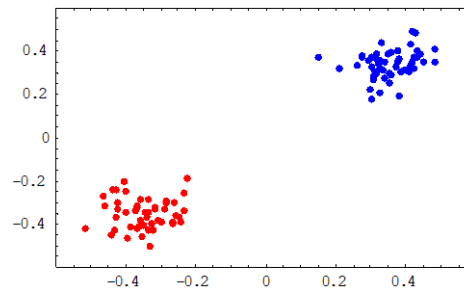
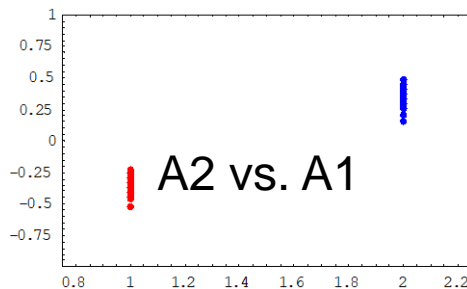
Dezavantaj: ignoră impactul setului redus de date asupra algoritmului de extragere a modelului din date



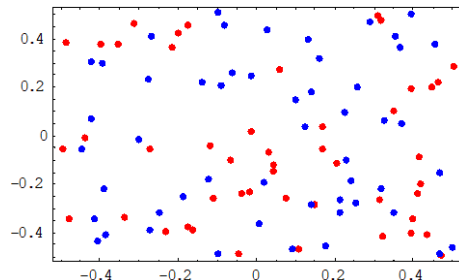
Selecția atributelor

Exemplu: set artificial de date: 10 atribute, 2 clase (clasa 1- roșu, clasa 2 - albastru)

- Atribut 1: identic cu eticheta clasei
- Atribute 2-6:
valori aleatoare cu rep. normală $N(m_1, s_1)$ (clasa 1), $N(m_2, s_2)$ (clasa 2)
- Atribute 7,8: valori constante pentru toate instanțele
- Atribute 9,10: valori aleatoare uniform repartizate ($U(a,b)$) pt toate instanțele



A6 vs. A5



A10 vs. A9

Selecția atributelor

Criteriu supervizat de selecție – attribute cu valori discrete

Câștig informațional: măsoară informația furnizată de atributul analizat (se bazează pe calculul entropiei)

Notății: D – set de date, N = nr instanțe , A_1, A_2, \dots, A_n - attribute, C_1, C_2, \dots, C_K – clase, $v_{i1}, v_{i2}, \dots, v_{ir}$ – valori posibile ale atributului i (se poate utiliza doar pt attribute cu valori discrete; r_i numărul de valori ale atributului A_i)

Entropie set de date:

$$H(D) = - \sum_{k=1}^K p_k \log p_k, \quad p_k = \text{card}(C_k) / \text{card}(D)$$

Entropie set de date care au valoarea v_j pentru atributul A_i :

$$H(D|A_i = v_{ij}) = - \sum_{k=1}^K p_{ijk} \log p_{ijk},$$

$$p_{ijk} = \frac{\text{numar de instante in } C_k \text{ cu } A_i = v_{ij}}{\text{numar de instante cu } A_i = v_{ij}}$$

Selecția atributelor

Criteriu supervizat de selecție – attribute cu valori discrete

Câștig informațional: măsoară informația furnizată pe atributul analizat (se bazează pe calculul entropiei)

Notații: D – set de date, N = nr instanțe, A_1, A_2, \dots, A_n - attribute, C_1, C_2, \dots, C_K – clase, $v_{i1}, v_{i2}, \dots, v_{ir}$ – valori posibile ale atributului i (se poate utiliza doar pt attribute cu valori discrete; r_i numărul de valori ale atributului A_i)

Câștig informațional (Information Gain):

$$IG(A_i) = H(D) - \sum_{j=1}^{r_i} p_{ij} H(D|A_i = v_{ij})$$

$$p_{ij} = \frac{\text{numar de instante din } D \text{ cu } A_i = v_{ij}}{\text{numar de instante din } D}$$

Criteriu selecție: se selectează attributele în **ordinea descrescătoare** a câștigului informațional (un atribut cu IG mare este un atribut cu putere de discriminare mai mare)

Selecția atributelor

Criteriu supervizat de selecție – attribute cu valori discrete

Index Gini: măsoară gradul de “impuritate” a atributelor (un atribut e cu atât mai impur cu cât instanțe cu aceeași valoare a atributului corespund mai multor clase)

Notații: A_1, A_2, \dots, A_n - attribute, C_1, C_2, \dots, C_K – clase, N = nr instanțe
 $v_{i1}, v_{i2}, \dots, v_{ir}$ – valori posibile ale atributului i (se poate utiliza doar pt attribute cu valori discrete; r_i numărul de valori ale atributului A_i)

index Gini pt atributul A_i

$$G(A_i) = \frac{1}{N} \sum_{j=1}^{r_i} n_{ij} G(v_{ij}), \quad G(v_{ij}) = 1 - \sum_{k=1}^K p_{ijk}^2$$

n_{ij} = numărul de instanțe pt care A_i are valoarea v_{ij}

$$p_{ijk} = \frac{\text{numar de instanțe în } C_k \text{ cu } A_i = v_{ij}}{\text{numar de instanțe cu } A_i = v_{ij}}$$

Interpretare: valori mici ale lui $G(A_i)$ sugerează o putere mare de discriminare a lui A_i (impuritate mică)

Selecția atributelor

Criteriu supervizat de selecție – attribute cu valori numerice (continue)

Scor Fisher: măsoară puterea de discriminare a unui atribut

Notații: A_1, A_2, \dots, A_n - attribute, C_1, C_2, \dots, C_K – clase

$$F(A_i) = \frac{\sum_{k=1}^K n_k (\mu_{ik} - \mu_i)^2}{\sum_{k=1}^K n_k \rho_{ik}^2}$$

n_k = număr de instanțe în clasa C_k

μ_{ik} = media valorilor lui A_i corespunzătoare instanțelor din C_k

ρ_{ik}^2 = varianța valorilor atributului A_i calculată pe baza instanțelor din C_k

μ_i = media valorilor atributului A_i

Interpretare: valori mari ale lui $F(A_i)$ sugerează putere mare de discriminare pt A_i

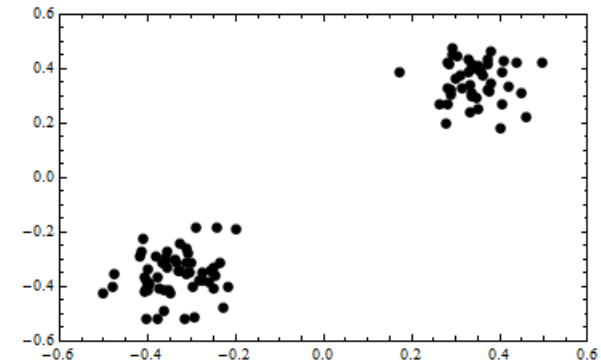
Selecția atributelor

Criteriu nesupervizat de selecție (se bazează doar pe date – fără a se cunoaște etichetele claselor)

Notății:

$D = \{x_1, x_2, \dots, x_N\}$ set de date cu N instanțe, fiecare conținând n atribute

A = set de atribute



Idee:

- Se calculează **similaritățile** între perechile de date din set (pe baza valorilor atributelor) → matrice de similaritate
- Se **discretizează** domeniul valorilor din matricea de similaritate și se calculează frecvența relativă a fiecărei valori → distribuția (p_1, p_2, \dots, p_M)
- Se calculează **entropia** asociată distribuției de mai sus (cu cât este mai mare cu atât este mai puțin probabil să existe o structură în setul de date)
- Se analizează efectul fiecărui atribut asupra valorii entropiei **și se elimină attributele care au cel mai mic impact asupra entropiei**

Selecția atributelor

Criteriu nesupervizat de selecție (se bazează doar pe date – fără a se cunoaște etichetele claselor)

Notatii:

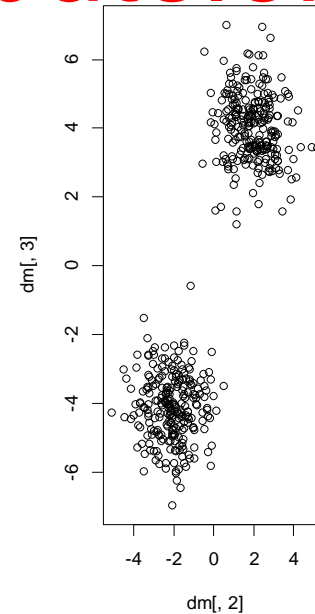
$D = \{x_1, x_2, \dots, x_N\}$ set de date cu N instanțe, fiecare conținând n atribute

A = set de atribute

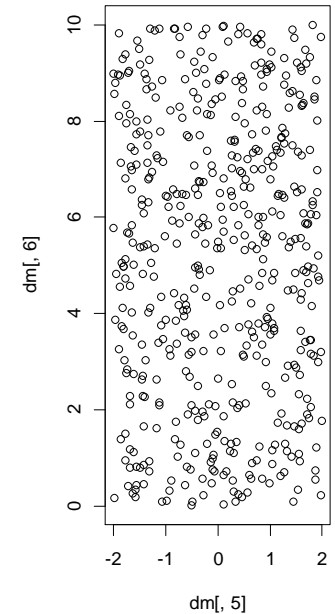
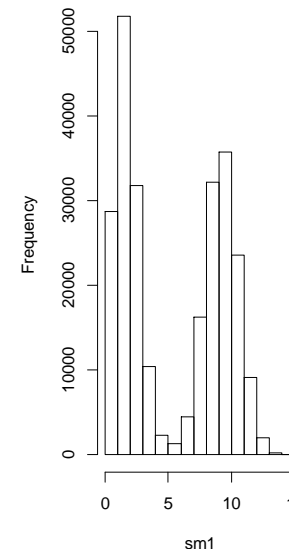
Exemplu

- Două seturi de date
 - Set1: cu structură de clustere
 - Set2: fără structură de clustere
- Distribuția disimilarităților dintre date (distanțe euclidiene)
 - Unimodală – nu există structură de clustere
 - Bimodală (sau multi-modală) – există structură de clustere (există diferență cel puțin între (di)similaritățile intra și inter-clustere

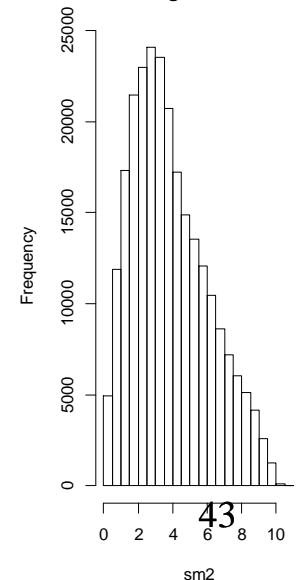
Data Mining -Curs 2 (2024)



Histogram of sm1



Histogram of sm2



Selecția atributelor

Criteriu nesupervizat de selecție

Măsurile de **similaritate** (calculate folosind un set A de atribute – setul de atribute conține n elemente)

Atribute numerice

Domeniul de
valori: (0,1)

$$S_{ij}(A) = \exp(-\alpha d(x_i, x_j)),$$

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

$\alpha = \text{ct. (e.g. 0.5)}$

Atribute nominale/ordinale/binare

$$S_{ij}(A) = \frac{1}{n} \sum_{k=1}^n I(x_{ik}, x_{jk}),$$

$$I(a, b) = 1 \text{ dacă } a = b;$$

$$I(a, b) = 0 \text{ dacă } a \neq b$$

Selecția atributelor

Criteriu nesupervizat de selecție

Entropia

Varianta 1:
$$E(S, A) = - \sum_{i \neq j} S_{ij}(A) \log(S_{ij}(A)) + (1 - S_{ij}(A)) \log(1 - S_{ij}(A))$$

Varianta 2:
$$E(S, A) = - \sum_{m=1}^M p_m(A) \log(p_m(A))$$

Obs:

- p_m este frecvența relativă a valorilor similarității care aparțin intervalului m dintre cele M intervale obținute prin discretizarea domeniului de valori ale similarității
- Varianta 1 a fost propusă în [Dash et al, 1997 – Dimensionality Reduction of Unsupervised Data]

Selecția atributelor

Criteriu nesupervizat de selecție

Algoritm (varianta decrementală)

Pas 1. se pornește cu întreg setul de attribute A

Pas 2. pt fiecare atribut a_i se calculează $E(S, A - \{a_i\})$ și se ierarhizează attributele crescător după valoarea $E(S, A) - E(S, A - \{a_i\})$

Pas 3. se elimină primul atribut din lista ordonată (atributul a căru eliminare a condus la cea mai mică pierdere în entropie) și se repetă Pas 2 – Pas 3 până când rămâne numărul dorit de attribute în A (sau până când reducerea în entropie la eliminarea unui atribut depășește un prag)

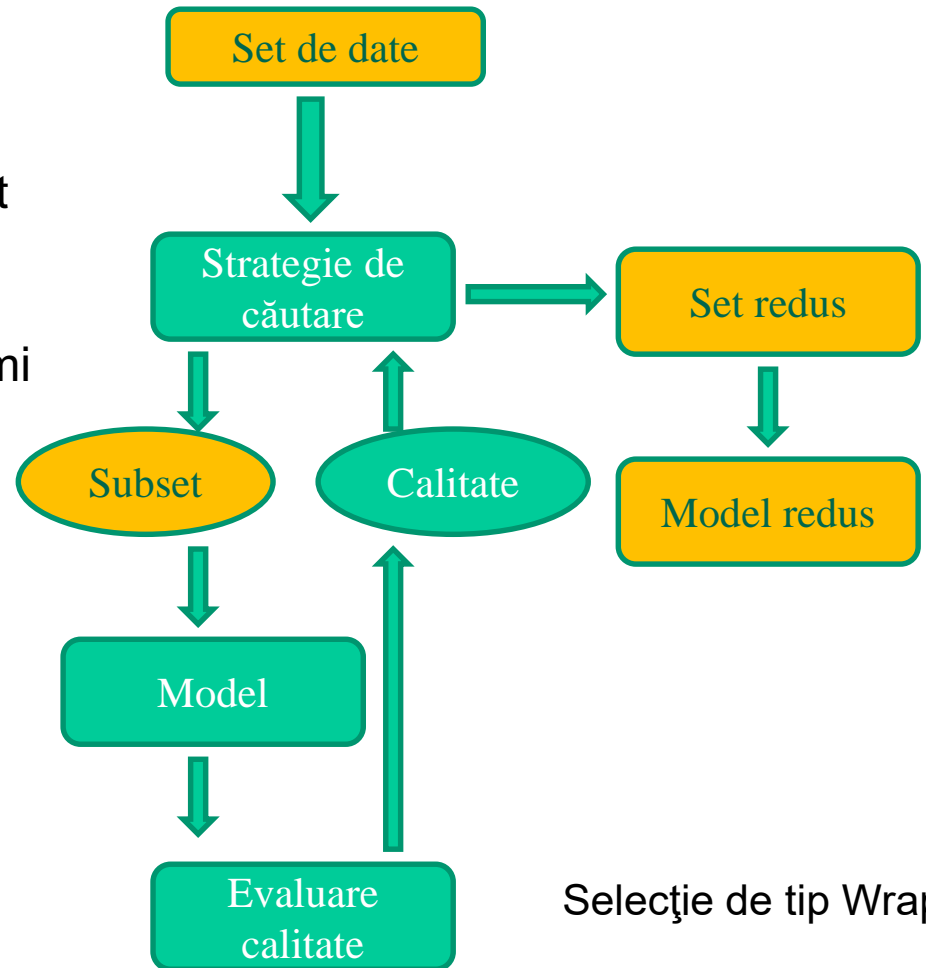
Selecția atributelor

Selecție încorporată în construirea modelului (wrapper-based)

- **Acuratețe** = număr de date corect clasificate/ număr total de date
- Obs: evaluarea fiecărei submulțimi necesită antrenarea întregului model

Avantaj: se folosește de impactul submulțimii de attribute asupra performanței modelului

Dezavantaj: evaluarea este costisitoare



Selecția instanțelor

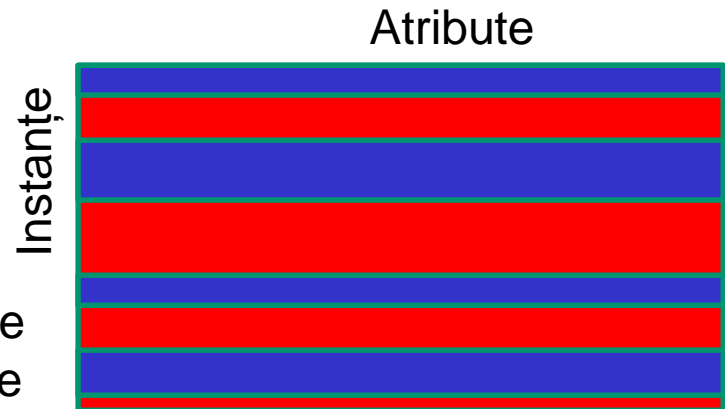
Selecția poate fi aplicată nu doar atributelor ci și instanțelor.

Exemplu (clasificare în 2 clase):

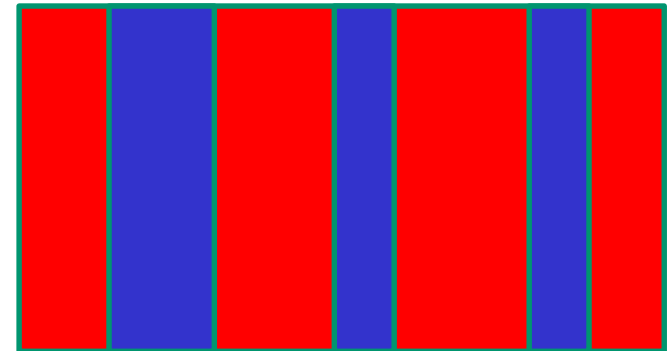
ar fi suficient să se folosească doar datele din vecinătatea frontierei celor două clase

Abordări:

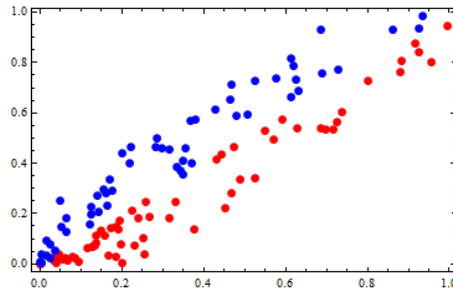
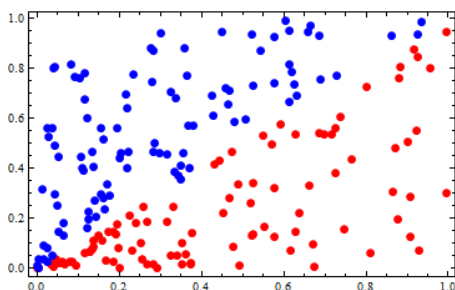
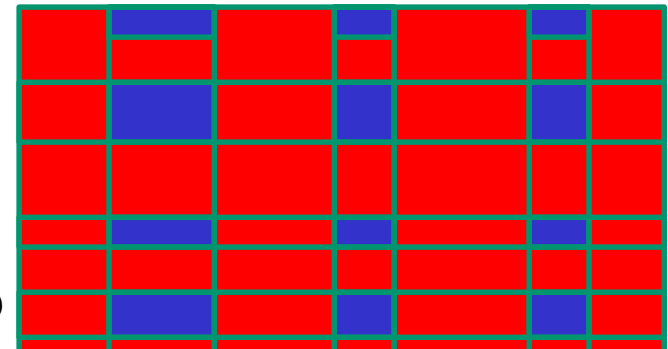
- Selecție aleatoare (cu sau fără revenire)
- Selecție stratificată



Selectie atribute



simultan



Selecția instanțelor

Selecția instanțelor în cazul seturilor de date care sunt **dezechilibrate** (variații mari între numărul de date din diferitele clase – pentru **probleme de clasificare**)

Abordări posibile:

- **down-sampling**: excluderea de elemente din clasa majoritară
 - se selectează aleator elemente din clasa majoritară a.i. setul să devină echilibrat (obs: se ignoră informație disponibilă)
 - se repetă selecția echilibrată de mai multe ori și se construiesc mai multe modele (bootstrapping)
- **up-sampling**: extinderea setului corespunzător clasei minoritare prin:
 - selecție cu revenire (se creează mai multe copii ale aceleiași instanțe)
 - se aplică strategii de modificare a copiilor prin tehnici de imputare specifice celor de la completarea valorilor absente

Selecția instanțelor

Selecția instanțelor în cazul seturilor de date care sunt dezechilibrate (variații mari între numărul de date din diferitele clase – pentru probleme de clasificare)

Abordări posibile:

- **Variantă hibridă** (synthetic minority over-sampling technique - **SMOTE**):
 - Pentru **clasele sub-reprezentate** se sintetizează noi elemente prin procedura:
 - Se alege un element aleator X din clasa sub-reprezentată
 - Se determină cei mai apropiați K vecini ai lui X și pornind de la ei se generează instanțe noi prin mixarea aleatoare a atributelor vecinilor
 - Pentru **clasele supra-reprezentate** se elimină elemente selectate aleator
 - **Parametrii de control pt SMOTE**: proporția de elemente sintetizate, proporția de elemente eliminate, numărul de vecini (ex: $K=5$)

Transformarea atributelor

Scop:

- Îmbunătățirea calității modelului extras din date prin eliminarea influenței induse de scale diferite pentru diferite attribute sau de corelații între attribute

Variante:

- Scalare
- Standardizare
- Normalizare
- Proiecție – analiza componentelor principale (Principal Component Analysis)

Obs: aceste transformări pot fi aplicate **doar atributelor numerice**

Scalare/Standardizare/Normalizare

Scalare :

- Scalare liniară
- Este sensibilă la valorile atipice
- Se aplică la nivel de atribut

Standardizare:

- Se scade media și se împarte la abaterea standard – datele transformate vor avea medie nulă și abatere standard egală cu 1
- Mai robustă decât scalarea liniară
- Se aplică la nivel de atribut

Normalizare euclidiană:

- Se împarte fiecare componentă la normă (e.g. Norma euclidiană)
- Se aplică la nivel de instanță

Scalare liniară:

$$z_i^j = \frac{x_i^j - \min(X^j)}{\max(X^j) - \min(X^j)}, \quad i = \overline{1, n} \quad j = \overline{1, d}$$

Standardizare:

$$z_i^j = \frac{x_i^j - m(X^j)}{s(X^j)}, \quad i = \overline{1, n} \quad j = \overline{1, d}$$

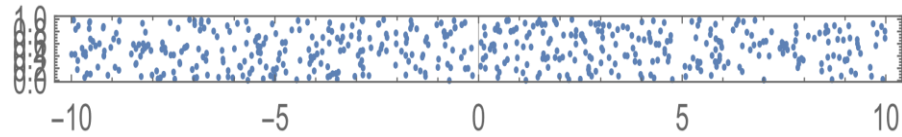
$$m(X^j) = \frac{1}{n} \sum_{i=1}^n x_i^j, \quad s(X^j) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^j - m(X^j))^2}$$

Normalizare:

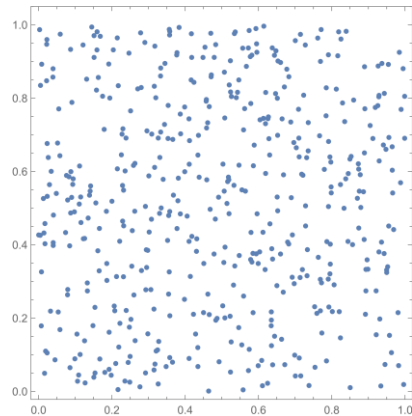
$$Z_i = X_i / \|X\|, \quad \|X\| = \sqrt{\sum_{j=1}^d (x_i^j)^2}, \quad i = \overline{1, n}$$

Scalare/Standardizare/Normalizare

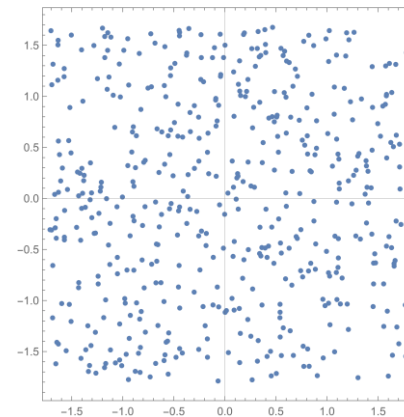
Date inițiale (uniform distribuite în $[-10,10] \times [0,1]$)



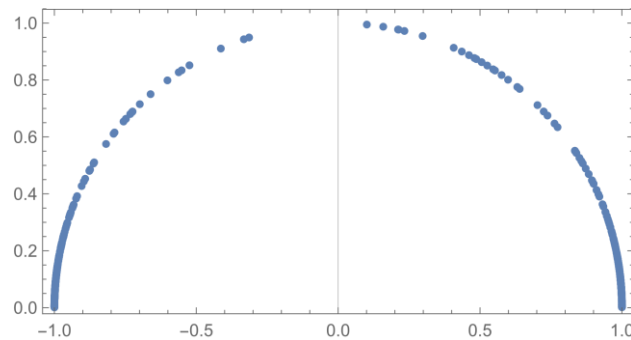
Scalare: $[0,1] \times [0,1]$



Standardizare: medie=0, stdev=1



Normalizare euclidiană:



Analiza componentelor principale

Principal Component Analysis (PCA):

Scop: reducerea numărului de atribute și
decorelarea atributelor

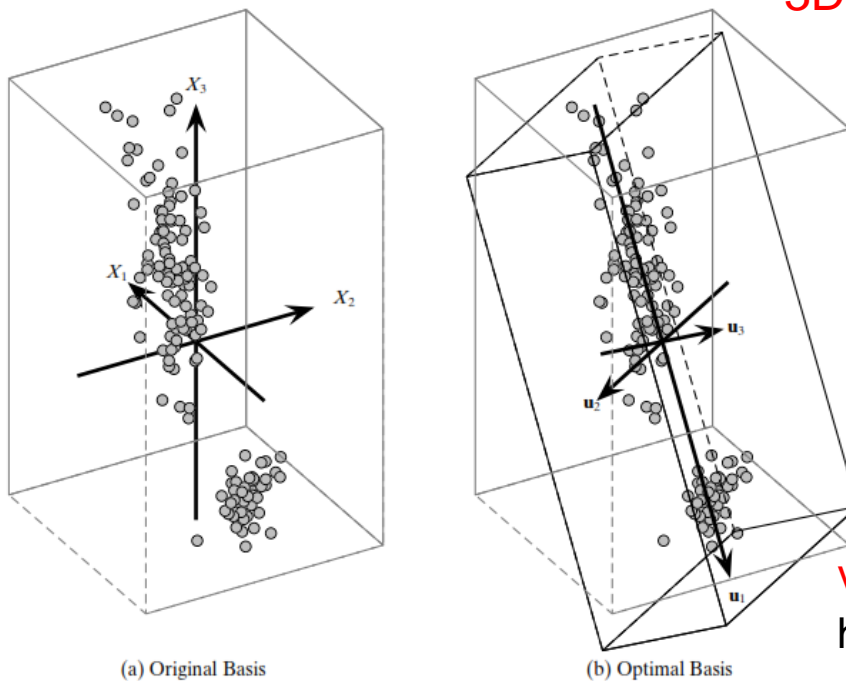
Idee: Se proiectează datele pe direcția de
variabilitate maximă

Analiza componentelor principale

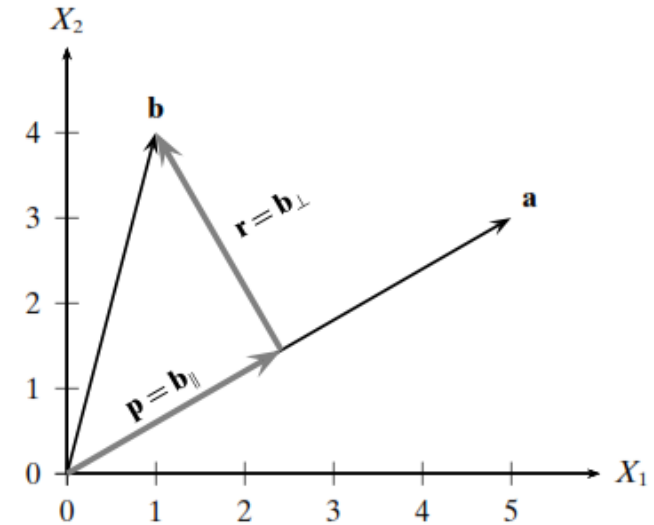
Principal Component Analysis (PCA):

Scop: reducerea numărului de atribute și decorelarea atributelor

Idee: Se proiectează datele pe direcția de variabilitate maximă



Iris dataset – 3D bases [Zaki, 2014]



Orthogonal projection [Zaki, 2014]

Vizualizare PCA:

<http://setosa.io/ev/principal-component-analysis/>

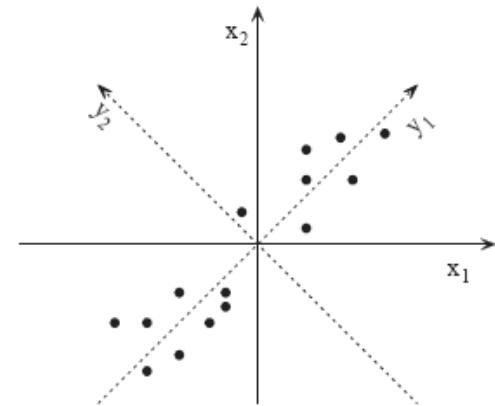
Analiza componentelor principale

Principal Component Analysis (PCA)

Idee: Se proiectează datele pe direcțiile care captează cea mai mare variabilitate din date

Intrare: set de date cu N instanțe având n atribute numerice (matrice de date D cu N linii și n coloane)

Ieșire: matrice de date cu N instanțe având $m < n$ atribute (alese a.î. să fie conservată cât mai mult din variabilitatea datelor inițiale)



Obs:

- PCA concentrează informația privind diferențele dintre instanțe într-un număr mic de atribute
- PCA se utilizează pt a **proiecta** un set de date n -dimensional într-un spațiu m -dimensional astfel încât atributele în noul spațiu sunt **necorelate** și este **conservată cât mai mult din variabilitatea datelor**

Analiza componentelor principale

Principal Component Analysis (PCA)

Intrare: matrice de date D cu N linii și n coloane

Etape principale:

- Se calculează **matricea de covarianță** C (matrice $n \times n$ cu elementele: $C(i,j) = \text{cov}(D(i), D(j))$, unde $D(i)$ este coloana i a matricii de date D);
- Se calculează **valorile proprii** și **vectorii proprii** ai matricii C
- Se **ordonează** vectorii proprii descrescător după valoarea proprie corespunzătoare
- Se selectează vectorii proprii care corespund celor mai mari **m valori proprii**
- Se **proiectează** setul de date D pe hiper-spațiul definit de cei m vectori proprii

Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Matrice de covarianță $C = (c_{ij})_{i=\overline{1,n}, j=\overline{1,n}}$ (n attribute, N instanțe)

$$c_{ij} = \frac{1}{N} \sum_{k=1}^N (x_{ki} - \mu_i)(x_{kj} - \mu_j) = \frac{1}{N} \sum_{k=1}^N x_{ki}x_{kj} - \mu_i\mu_j, \quad i = \overline{1,n}, j = \overline{1,n}$$

μ_i = media valorilor atributului i , c_{ii} = varianța atributului i

C are n vectori proprii v_1, v_2, \dots, v_n corespunzători celor n valori proprii $\lambda_1, \lambda_2, \dots, \lambda_n$

$Cv_i = \lambda_i v_i$ (C transformă vectorii proprii doar prin scalare)

Obs: C este simetrică și pozitiv–semidefinită

$(x^T C x \geq 0 \text{ pt orice vector } x) \Rightarrow$ toate valorile proprii sunt pozitive

Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Proiecția pe spațiul definit de un vector (transformare în date uni-dimensionale)

- dacă v este direcția corespunzătoare unui vector propriu atunci proiecția lui D pe v este Dv (produsul dintre matricea D cu N linii și n coloane și vectorul v cu n elemente)
- Covarianța noului set de date (e de fapt varianța întrucât datele sunt uni-dimensionale)

$$\frac{(Dv)^T(Dv)}{N} - (\mu v)^2 = v^T C v = v^T \lambda v = \lambda \|v\|^2 = \lambda$$

obs: vectorii proprii sunt ortonormati: $v_i^T v_j = 0$, pt $i \neq j$, $\|v\| = 1$

- Varianța proiecției uni-dimensionale pe un vector propriu este egală cu valoarea proprie corespunzătoare, deci pt a capta cât mai multă variabilitate trebuie aleasă cea mai mare valoare proprie

Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică și algebra liniară

Proiecția setului de date pe hiper-spațiul definit de mai mulți vectori proprii

- Rezultat util din algebra liniară (descompunerea după valori proprii):

$$C = P\Lambda P^T,$$

P = matrice ortogonală cu vectorii proprii ai lui C pe coloane ($PP^T = I_{n \times n}$),

$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ = matrice diagonală cu valorile proprii

- Vectorii proprii (fiind ortogonali) definesc un **sistem de axe de coordonate**
Proiecția setului D în noul sistem de coordonate este $D' = DP$
- **Intrebare:** care este matricea de covarianță a noului set D' ?

Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică și algebra liniară

Proiecția setului de date pe hiper-spațiul definit de mai mulți vectori proprii

▪ **Intrebare:** care este matricea de covarianță a noului set D'

$$\begin{aligned} D' &= DP, \quad X'_k = P^T X_k, \quad M = \frac{1}{N} \sum_{k=1}^N X_k \\ C' &= \frac{1}{N} \sum_{k=1}^N (P^T X_k - P^T M)(P^T X_k - P^T M)^T = \\ &= \frac{1}{N} \sum_{k=1}^N P^T (X_k - M)(X_k - M)^T P = \\ &= P^T C P = P^T P \Lambda P^T P = \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \end{aligned}$$

Deci datele din D' **sunt necorelate** (matricea de covarianță este diagonală) și varianța corespunzătoare atributului i este a i -a valoare proprie

Analiza componentelor principale

Proiecția setului de date pe hiper-spațiul definit de mai mulți vectori proprii

- **Intrebare:** ce se întâmplă dacă se păstrează doar m dintre componentele unui vector transformat?
- **Răspuns:** se conservă doar o fracțiune din variabilitatea datelor
- **Ipoteza:** valorile proprii sunt sortate descrescător
- **Procedura:** se calculează raportul varianțelor (R) și se alege m a.î. $R > \text{prag}$ prestabilit (e.g. $R > 0.95$)
- **Rezultat:** noul set de date (cu cele m attribute obținute prin proiecția pe hiper-spațiul definit de vectorii proprii corespunzători celor mai mari m valori proprii) captează cea mai mare parte din variabilitate (e.g. 95%)

$$\text{Proporția varianțelor: } R = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^n \lambda_i}$$

Analiza componentelor principale

Exemplu: setul de date iris

4 attribute numerice:

A1=lungime sepale, A2=lățime sepale, A3=lungime petale, A4=lățime petale,

3 clase

150 instanțe

Matricea de covarianță (calcul în R: `covMatrix <- cov(iris[,1:4])`)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.6856935	-0.0424340	1.2743154	0.5162707
Sepal.Width	-0.0424340	0.1899794	-0.3296564	-0.1216394
Petal.Length	1.2743154	-0.3296564	3.1162779	1.2956094
Petal.Width	0.5162707	-0.1216394	1.2956094	0.5810063

Analiza componentelor principale

Exemplu: setul de date iris

4 attribute numerice:

A1=lungime sepale, A2=lățime sepale, A3=lungime petale, A4=lățime petale,

3 clase 150 instanțe

Valori proprii (calcul în R: `eigen(covMatrix)$values`)

4.23 0.24 0.078 0.02

varianța explicată de primele două componente: 97.7%

Vectori proprii (calcul în R: `eigen(covMatrix)$vectors`)

	[,1]	[,2]	[,3]	[,4]
[1,]	0.36	-0.65	-0.58	0.31
[2,]	-0.08	-0.73	0.59	-0.31
[3,]	0.85	0.17	0.07	-0.47
[4,]	0.35	0.07	0.54	0.75

Noile attribute:

tA1: $0.36 \cdot A1 - 0.08 \cdot A2 + 0.85 \cdot A3 + 0.35 \cdot A4$

tA2: $-0.65 \cdot A1 - 0.73 \cdot A2 + 0.17 \cdot A3 + 0.07 \cdot A4$

Analiza componentelor principale

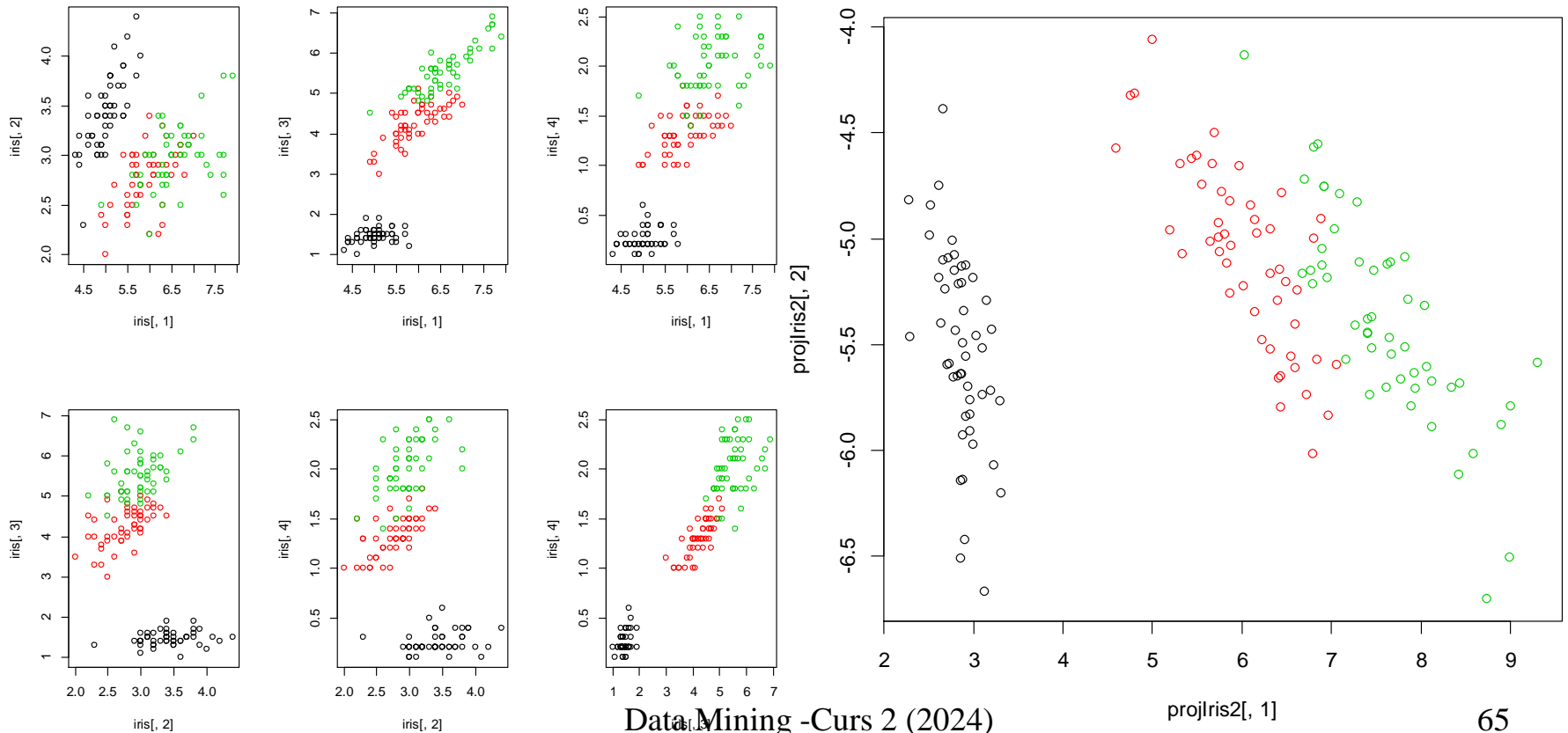
Exemplu: setul de date iris

Date inițiale

Noile attribute (primele două componente principale):

$$tA1: 0.36 \cdot A1 - 0.08 \cdot A2 + 0.85 \cdot A3 + 0.35 \cdot A4$$

$$tA2: -0.65 \cdot A1 - 0.73 \cdot A2 + 0.17 \cdot A3 + 0.07 \cdot A4$$

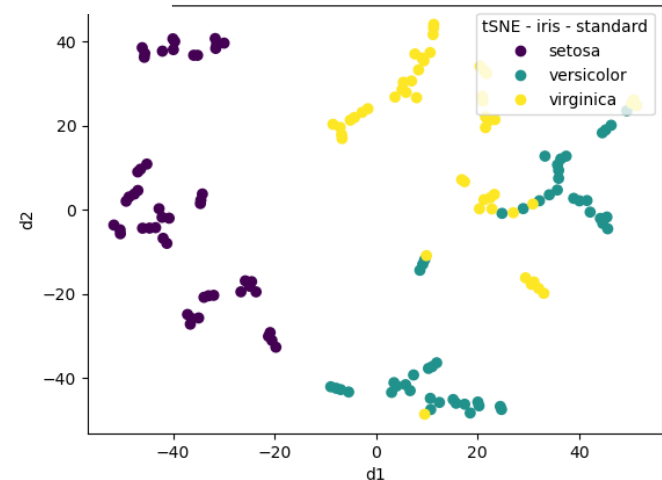
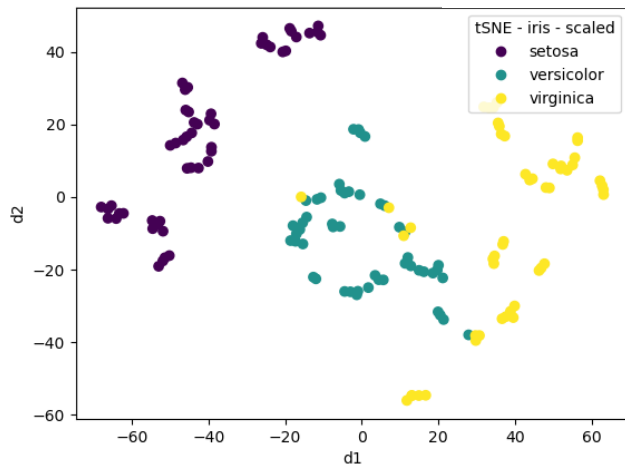
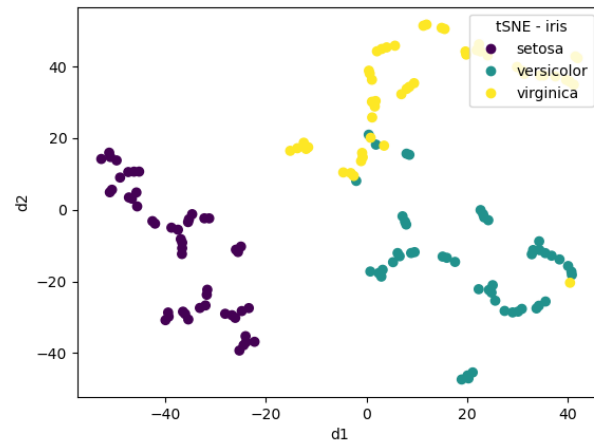


Transformări neliniare

- Metode de tip “manifold learning”
- Idee: identificarea unei transformări neliniare $M: \mathbb{R}^n \rightarrow \mathbb{R}^m$ care conservă “topologia” spațiului de date
- Variante
 - De tip local (tSNE)
 - De tip global (kernelPCA, Autoencoders)
- Exemplu
 - t-SNE: t-Distributed Stochastic Neighbor Embedding [L. van Maaten, G. Hinton, 2008] (<http://lvdmaaten.github.io/tsne/>)
 - distanțele dintre date sunt transformate astfel încât să poată fi interpretate ca distribuții de probabilitate (caracter local: se iau în considerare doar vecinii)
 - maparea este determinată astfel încât să minimizeze divergența Kullback-Leibler dintre distribuția corespunzătoare spațiului sursă și cea corespunzătoare spațiului destinație

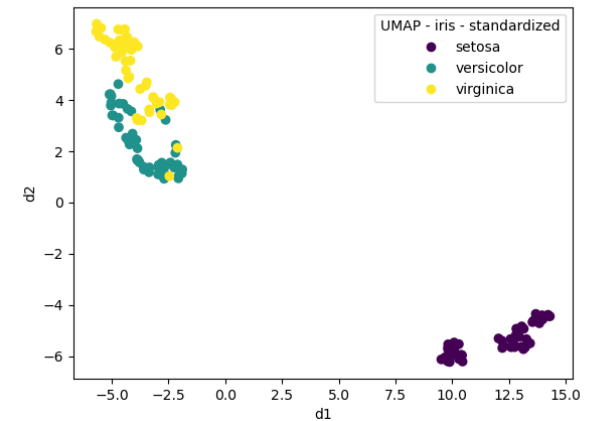
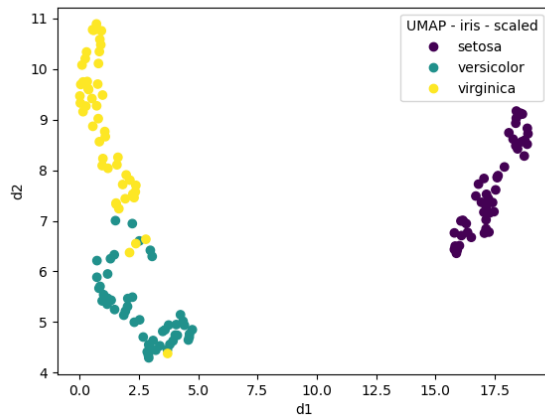
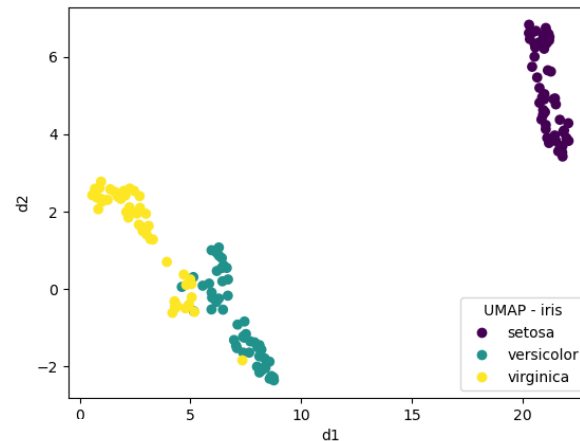
Transformări neliniare

- t-SNE: t-Distributed Stochastic Neighbor Embedding [L. van Maaten, G. Hinton, 2008] (<http://lvdmaaten.github.io/tsne/>)



Transformări neliniare

- UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [L. McInnes, J. Healy, J. Melville, 2020]



Sumar

Tipuri de date: numerice, ordinale, nominale

Pre-procesare date:

- Curățare
 - Identificare valori eronate
 - Tratare valori absente
- Conversie
 - Numeric -> ordinal/nominal
 - Nominal -> numeric (one hot encoding)
- Selecție
 - Atribute
 - Instanțe
- Transformare
 - Scalare
 - Standardizare
 - Normalizare
 - Proiecție

Curs următor

Modele de clasificare:

- Concepte de bază
- Măsuri de evaluare a calității clasificării
- Clasificatori
 - Vot simplu (ZeroR)
 - Arbori de decizie
 - Reguli de decizie