TEHNOLOGII AVANSATE DE PROCESARE SI ANALIZA A VOLUMELOR MARI DE DATE

Editor

Mihaela TINCA UDRISTIOIU

Traducere

Mihaela TINCA UDRISTIOIU
Iulian PETRIȘOR
Silvia PUIU
Ion BULIGIU

Autori

Mihaela Tinca UDRIȘTIOIU
Adam DUDÁŠ
Alžbeta MICHALÍKOVÁ
Fatih KILIC
Onder TUTSOY
Jarmila ŠKRINÁROVÁ
Silvia PUIU
Slaveya PETROVA

Acest material a fost finanțat de Comisia Europeană, în cadrul proiectului
Erasmus + Aplicarea unor tehnologii avansate de predare și cercetare, în legătură cu poluarea aerului
Cod proiect 2021-1-RO01-KA220-HED-000030286

Finanțat de Uniunea Europeană

Universitatea din Craiova

Universitatea din Plovdiv Paisii Hilendarski

Universitatea de Științe și Tehnologie Adana Alparslan Türkeş

Universitatea Matej Bel, Banská Bystrica

ISBN
978-625-399-463-1

Book Title
Tehnologii avansate de procesare si analiza a volumelor mari de date

Editör and Project manager
Mihaela Tinca UDRISTIOIU
ORCID iD: 0000-0002-5811-5930

Publishing Coordinator
Yasin DİLMEN

Page and Cover Design
Akademisyen Dizgi Ünitesi

Publisher Certificate Number
47518

Printing and Binding
Vadi Matbaacılık

Bisac Code
BUS070030

DOI
10.37609/akya.2890

Library ID Card
Tınca Udrıstıoıu, Mihaela and others.
Tehnologii avansate de procesare si analiza a volumelor mari de date / Mihaela Tınca Udrıstıoıu, Adam Dudas, Alžbeta Michalikova [and others] ; editor : Mihaela Tınca Udrıstıoıu.
Ankara : Akademisyen Yayınevi Kitabevi, 2023.
175 page. : figure, table, 195x275 mm.
Includes Bibliography.
ISBN 9786253994631
1. Information Technology.

GENERAL DISTRIBUTION
Akademisyen Kitabevi A.Ş.
Halk Sokak 5 / A Yenişehir / Ankara
Tel: 0312 431 16 33
siparis@akademisyen.com

www.akademisyen.com

CUPRINS

INTRODUCERE

Acest manual este unul dintre rezultatele realizate în cadrul proiectului Erasmus+ nr. 2021-1-RO01-KA220-HED-000030286, intitulat "Aplicarea unor tehnologii avansate în predare și cercetare, în legătură cu poluarea aerului". A fost o colaborare între cei patru parteneri implicați în proiect: Matej Bel University din Banská Bystrica, Slovacia; Universitatea din Craiova, România, Universitatea Paisii Hilendarski din Plovdiv, Bulgaria; și Universitatea Adana de Științe și Tehnologie din Adana, Turcia. Autorii manualului și-au propus să vină în sprijinul profesorilor STEM și să îmbunătățească abilitățile studenților pregătiți de aceștia, în ceea ce privește lucrul cu diferite tipuri de date.

Fiecare dintre noi este copleșit de cantitatea de informație care vine din toate direcțiile. Procesarea și extragerea informației esențiale este vitală. În zilele noastre, este necesar ca studenții să cunoască modalități de procesare a datelor pentru a putea extrage informațiile relevante scopului urmărit. În fiecare secundă, computerele, rețelele de senzori și sateliții adună milioane de valori pentru diverse mărimi și parametri fizici. Bazele de date stochează și organizează datele și informațiile, îmbunătățind calitatea datelor brute. Mai mult ca niciodată, informația este putere; din acest punct de vedere, este necesar ca studenții STEM să învețe să lucreze cu seturi de date.

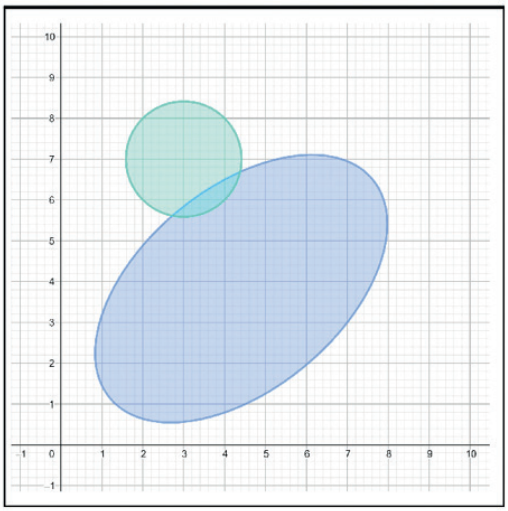
Companiile solicită instituțiilor de învățământ superior să furnizeze pieței muncii absolvenți cu calificare înaltă, capabili să rezolve probleme, pe baza informațiilor oferite de bazele de date sau folosind programe sau algoritmi de specialitate. În universități, este necesar ca studenții STEM să studieze modul în care sunt colectate, analizate și interpretate seturile de date. De asemenea, este necesar ca aceștia să înțeleagă cum pot face clasificări, aproximări și estimări ale datelor. Nu în cele din urmă, piața muncii cere absolvenților STEM să fie capabili să facă predicții legate de modul în care procesele evoluează în spațiu și timp sau să ia decizii. Învățarea automată și inteligența artificială au devenit termeni standard în vocabularul de zi cu zi al studenților și cadrelor didactice și necesită formare continuă pentru aceștia.

Manualul cuprinde zece secțiuni, anexe și note bibliografice. Prima parte a manualului este despre diferite tipuri de date și proprietățile acestora, metodele de eșantionare a datelor și modul de procesare și analiză a datelor. Următoarele secțiuni abordează una dintre cele mai semnificative probleme legate de seturile de date mari, analiza datelor. În analiza Big data, este necesară cunoașterea modului de utilizare a unor metode potrivite de analiză statistică, vizualizare a datelor precum și alte metode exploratorii, predictive și estimative. De asemenea, manualul conține secțiuni diferite axate pe abordări moderne precum inteligența artificială, învățarea automată și rețelele neuronale. Au fost incluse anexe care conțin informații legate de descrierea setului de date Iris, exemple de soluții la unele probleme propuse în manual, seturi de date privind schimbările climatice sau poluarea aerului și informații despre impactul poluării aerului asupra sănătății umane. Un exemplu de curriculum pentru un curs de „Tehnologii avansate de procesare și analiză a datelor Big data" încheie acest manual.

CAPITOLUL 1

DATE ȘI PROPRIETĂȚILE ACESTORA

Această parte a manualului a fost scrisă de Adam Dudáš, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.



	A	B	C	D
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1
14	4.3	3.0	1.1	0.1
15	5.8	4.0	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4
18	5.1	3.5	1.4	0.3

48° 44' 10.597" N 19° 8' 46.291" E

Figura 1. Exemplu de set de date

Acest manual prezintă metode simple de analiză a datelor folosind tehnici din informatică, precum inteligența artificială, învățarea automată sau rețelele neuronale. În următoarea secțiune a lucrării vor fi prezentați termeni și concepte de bază din domeniul datelor, proprietățile acestora, prelucrarea și analiza datelor.

Datele sunt mesaje sau informații tehnice, statistice, economice sau de alt tip care pot fi prelucrate cu ajutorul mijloacelor tehnice – în cazul nostru aceste mijloace tehnice fiind reprezentate de calculatoare. Datele sunt abordate ca fiind obiecte care sunt integrate și partajate în sistem:

- **Integrarea datelor** – datele pot fi plasate în mai multe fișiere, astfel încât duplicarea să fie minimizată iar datele din mai multe fișiere să poată fi accesate simultan.
- **Partajarea datelor** – fiecare obiect dată poate fi partajat de mai mulți utilizatori (în mod repetat și simultan).

Cea mai importantă proprietate a datelor este **persistența** - datele persistente sunt date care există chiar și după terminarea programului. Datele de intrare (input data) pot fi transformate în persistente. Datele de ieșire pot fi transformate din date persistente, în date de intrare sau derivate din acestea. Datele derivate din alte date nu ar trebui să fie persistente (cresc costurile legate de funcționarea sistemului) însă uneori este necesar să facem acest lucru.

Este important să se decidă cum vor fi reprezentate datele înregistrate în funcție de tipurile specificate (ținând cont de cea mai eficientă stocare posibilă). Cele mai obișnuite **tipuri de date** sunt:

- **Date numerice** - pot fi stocate în diverse moduri (binar, caracter, formă semi-logaritmică, ...), este adesea necesară definirea numărului de biți/octeți necesari pentru număr.
- **Stringuri (Șiruri de caractere)** - pot fi stocate în diferite seturi de caractere (ACII, UNICODE, EBDIC, ...).
- **Enumerări** - folosind coduri de caractere în loc de șiruri de caractere (de exemplu, A în loc de excelent, ...).
- **Unități** - trebuie adaptate la situația specifică (nonsens: distanța de zbor pentru avion măsurată în milimetri).

Aceste implementări ale tipurilor de date nu sunt foarte interesante pentru noi în contextul analizei datelor. În general, se vorbește despre două tipuri de date, care se disting prin conținutul lor:

- **Date cantitative constând în valori numerice** (înălțime, distanță, număr, ...). Astfel de date pot fi utilizate în mod direct în modele matematice, ceea ce este critic din punctul de vedere al analizei datelor bazate pe metode de învățare automată.
- **Datele categorice formate din denumiri lingvistice** ale proprietăților (gen, culoare, specie, ...), ceea ce implică necesitatea unor metode specifice de analiză a datelor. Unele date categorice pot fi codificate în cantitative, dar o astfel de operațiune nu este întotdeauna semnificativă. Un exemplu de codificare ar putea fi ceva de genul: *IF gen = masculin THEN gen = 1, IF gen = feminin THEN gen = 2* și așa mai departe. Acest lucru are sens într-un fel, dar există câteva întrebări care descurajează o astfel de codificare:

Deoarece 2 - 1 = 1, este feminin - masculin = masculin?
Care este valoarea maximă a genului?

Cel mai important lucru înainte de a realiza analiza oricărui tip de date este ca acestea să fie structurate corespunzător. Din punct de vedere al **structurii** distingem trei tipuri de date:

- **Date structurate** – date stocate sub formă de tabele sau fișier în care este posibil să se identifice aceleași proprietăți în aceeași ordine (coloane) pentru fiecare obiect (rând) înregistrat. Cele mai frecvent utilizate formate de date structurate sunt csv, fișier excel, fișier text simplu sau baze de date SQL.
- **Date semi-structurate** - datele sunt structurate, dar forma acestora nu este fixă. Prin urmare, putem determina aceleași proprietăți pentru fiecare obiect (linie), dar este posibil ca aceste proprietăți să nu fie toate înregistrate pentru fiecare înregistrare sau să nu fie în aceeași ordine pentru toate înregis-

trările (coloanele nu pot fi identificate). Acest lucru este tipic pentru datele dintr-un set de senzori, din aplicații mobile și altele asemenea. Formatele folosite pentru acest tip de date sunt XML, JSON sau MongoDB.

- **Date multi-structurate/nestructurate** – date brute în diferite formate. De exemplu, date brute de la un senzor, jurnale web, date din rețele sociale, audio, video, imagini, modele 3D, coordonate și altele asemenea.

  În timp ce lucrăm cu date nestructurate, folosim în mod obișnuit următorul flux de lucru:

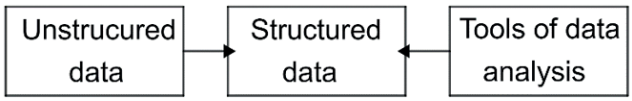


Figura 2. Ilustrarea fluxului de lucru

Exemplu: Conversia de la date nestructurate la date structurate (tabel). Fie o colecție de informații de bază despre trei studenți:

1. *Martin, bărbat, 28.6.1983, anul de studii 2, 40 ani*
2. *Jane, 1994-9-13., F, 1. anul de st., 29 a.*
3. *Miriam, femeie, 5 aprilie 1992, anul de studii, vârsta: 31*

Informațiile sunt într-o formă inconsistentă - ordinea caracteristicilor individuale ale oamenilor este diferită pentru fiecare dintre studenți, iar formatul proprietăților individuale este, de asemenea, diferit. Datele conțin și date ușor de calculat, care nu trebuie stocate (vârstă). Prin urmare, atunci când stocăm datele într-o formă structurată (tabel), este necesar să unificăm ordinea și formatul tuturor proprietăților (cum ar fi spre exemplu data în formatul YYYY-MM-DD).

Tabel 1. Exemplu de date tabelare cu unificarea proprietăților

Numele	Data nașterii	Anul de studii	Sexul
Martin	*1983-6-28*	*2*	*M*
Jane	*1994-9-13*	*1*	*F*
Miriam	*1992-4-5*	*2*	*F*

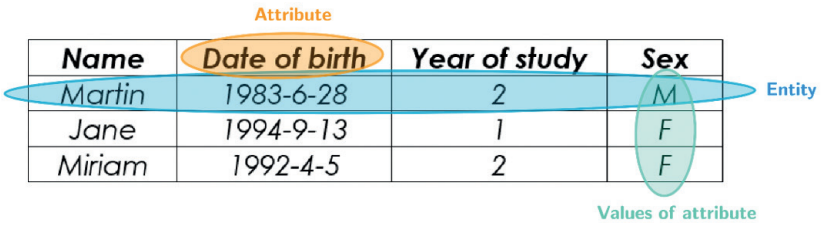
În contextul acestui manual, vom folosi titluri diferite pentru aceleași obiecte din seturile de date. Prin urmare, prezentăm o scurtă explicație a termenilor de mai jos.

Entitatea, obiectul sau înregistrarea este un obiect din lumea reală care este capabil să existe independent și este în mod clar distinct de alte obiecte.

Atributul sau proprietatea este o funcție care atribuie o valoare unei entități, care determină o proprietate esențială a entității (de exemplu, înălțimea, vârsta, ...).

Setul de date sau tabelul este un set de entități constând din același set de atribute.

Structurarea datelor colectate este o metodă de bază de prelucrare a datelor (vezi Secțiunea 2).



Name	Date of birth	Year of study	Sex
Martin	1983-6-28	2	M
Jane	1994-9-13	1	F
Miriam	1992-4-5	2	F

Tabel 2. Exemplu de structurare a datelor

1.1 CÂTEVA CUVINTE DESPRE BIG DATA

În principiu, este întotdeauna mai bine să avem mai multe date decât să avem prea puține (putem oricând să înlăturăm unele dintre înregistrări). Putem utiliza denumirea Big data dacă procesarea și analiza acestor date nu este posibilă cu instrumente convenționale într-un timp practic. Desigur, pot apărea o serie de întrebări precum: *Ce este timpul practic? Ce este un instrument convențional?* Prin urmare, folosim definiția datelor mari prin enumerarea proprietăților acestora.

Datele sunt numite în mod obișnuit Big data în cazul în care acestea au acele proprietăți denumite 3V (numărul acestor „V"-uri crește în timp, în unele articole și cărți, 5V este cel mai utilizat model de vizualizare a acestui tip de date):

- **Volumul de date** – Cantitatea de date care poate fi obținută dintr-un anumit număr de surse face de neconceput utilizarea unor modele simple de baze de date relaționale. Nu putem reprezenta datele într-un tabel simplu sau un set de tabele și să lucrăm pe o singură mașină de calcul. Prin urmare, crește în importanță nevoia de a dezvolta o infrastructură de calcul mai sofisticată și de a implementa algoritmi optimizați. Este necesar ca în acest sistem să se implementeze principii de distribuire și de calcul de înaltă performanță în cloud, și baze de date non-relaționale care pot stoca date omogene și interconectate puternic și modele de inteligență artificială atât pentru procesarea, cât și pentru analiza datelor.
- **Varietatea de date** – Deoarece setul de date care este luat în considerare în cazul problemelor reale de Big data nu este omogen, este necesar să putem lucra cu un număr mare de tipuri și formate de fișiere, cum ar fi de exemplu documente text simple, fișiere audio, fișiere video, coordonate sau modele de computer care se referă la mai mult de două dimensiuni. Cele mai multe dintre aceste tipuri de date nu pot fi stocate în baze de date relaționale și necesită putere de calcul și spațiu de stocare mari pentru o stocare și procesare de succes și în mod convenabil, pentru o utilizare ulterioară.
- **Viteza datelor** – Seturile mari de date sunt cel mai adesea acele seturi de date live (sau dinamice) – seturi de date care se modifică în timp. Această modificare a datelor în timp are loc în toate sistemele care implementează așa-numitele surse de date ambientale – surse care sunt întotdeauna active și colectează date, cum ar fi senzorii Internet of Thing și care măsoară progresul aceluiași set de valori. Durata datelor, procesarea, stocarea și analiza acestora conduc la crearea unor fluxuri de date care intră continuu în sistem. Acest aspect subliniază una dintre cele mai importante cerințe ale sistemelor de date mari – capacitatea de a colecta, stoca, procesa și analiza datele în (aproape) timp real.

Pe lângă această problemă a datelor în timp real, se poate identifica și problema combinației dintre partea dinamică și cea statică a setului de date. Această problemă conduce la multe evenimente problematice în sistem.

În afară de volumul, varietatea și viteza datelor, în literatură apar alte două proprietăți, veridicitate și valoare:

- **Veridicitatea datelor** – Deoarece datele mari sunt folosite adesea în luarea deciziilor relative la un anumit număr de entități reale, există o nevoie importantă de date de încredere și sigure. Este necesară luarea în considerare a metricii, aceasta fiind cea care măsoară încrederea utilizatorului în date. Acest lucru este important nu numai în cazul luării deciziilor, ci și în cazul realității datelor – generarea de date mari nu este dificilă și, prin urmare, poate fi utilizată cu ușurință ca parte a atacurilor care vizează supraîncărcarea sistemului țintă.
- **Valoarea datelor** – așa cum s-a menționat mai sus, datele mari pot fi (și sunt) folosite pentru a lua decizii. Valoarea datelor crește odată cu creșterea cantității datelor referitoare la subiectul studiat sau precizia potențialului de estimare/predicție aferent datelor. Această valoare poate avea o semnificație monetară, comercială, umană, de cercetare sau de altă natură.

Aceste proprietăți ale datelor mari scot la iveală multe **probleme legate de procesarea și analiza lor.**

Prima dintre aceste probleme este legată de **dimensiunea datelor în sine**. Dimensiunea lor este importantă nu numai în contextul spațiului de memorie, care este necesar pentru stocarea datelor în sine, ci și din punct de vedere al căutării în date și al analizei acestora. Atunci când se lucrează cu astfel de date, este necesară folosirea unor metode distribuite sau cloud computing de performanță înaltă, în combinație cu algoritmi de inteligență artificială din învățarea automată, sisteme de inferență fuzzy și rețele neuronale pentru a obține informație de calitate din acest tip de date.

Pe lângă faptul că aceste date sunt mari, ele sunt adesea compuse din **partiții eterogene care pot să difere în mai multe aspecte** – dimensionalitatea, compoziția și structura datelor, dar și măsurătorile utilizate. Această inconsecvență este o consecință a faptului că seturile de date mari sunt colectate adesea din mai multe surse incompatibile, într-un singur depozit. Prin urmare, este necesar ca partițiile de date să fie reformatate (sau, mai precis, formatele de date individuale să fie oarecum unificate). Această reformatare constă într-o secvență de sarcini simple executate pe date (cum ar fi conversia măsurătorilor, dacă este necesar), dar și de sarcini mai complexe, cum ar fi de exemplu identificarea valorilor aberante și a valorilor lipsă. În cazul valorilor lipsă, pot fi luate o serie de măsuri pentru a calcula aceste valori lipsă; metodele de învățare automată și rețelele neuronale pot fi utilizate pentru a estima valorile sau clasifica datele.

O problemă strâns legată de eterogenitatea seturilor de date mari este **durata de viață a seturilor de date**. În acest caz, se măsoară o serie de mărimi fizice (datele) utilizând surse de date ambientale (precum rețelele de senzori). Aceste măsurători sunt efectuate de un număr de senzori suficient de mare, pentru intervale de timp suficient de mici, sunt create fluxuri de date care necesită procesare și pregătire pentru o analiză de către sistem. Prin urmare, acest sistem trebuie să fie capabil să proceseze și să analizeze seturi de date care se modifică în timp.

O altă problemă semnificativă legată de seturile de date mari este **analiza datelor**. Analiza trebuie să fie susținută de computere de performanță înaltă, sau în cloud, descompunerea corectă a problemei și învățarea automată, modele de calcul pentru rețele fuzzy și neuronale. La analiza seturilor mari de date pot fi utilizate metode de analiză statistică, vizualizare a datelor precum și alte metode de analiză exploratorie sau analiză predictivă și estimativă a datelor, utilizând învățarea automată, sistemul de inferență fuzzy al abordărilor rețelelor neuronale (vezi Secțiunea 2).

1.2 PROBLEME COMUNE ÎN SETURILE DE DATE

Există o serie de probleme comune în ceea ce privește datele și care nu au fost descrise mai sus - în special legate de Big data.

După cum a fost menționat mai sus, cantitatea de date este într-o creștere continuă, ceea ce înseamnă că analiza datelor stocate și procesate durează mai mult. Cu toate acestea, analiza datelor este esența stocării acestora și, prin urmare, este imposibil de evitat. Acest lucru aduce cu sine nevoia de metode și proceduri care să permită obținerea de cunoaștere și sprijin în luarea deciziilor când se lucrează cu un set mare de date.

Seturile de date destinate unor sarcini specifice sunt create adesea prin combinarea datelor provenite din mai multe surse. Aceste surse pot fi caracterizate prin diversitate în formate și prin modul de alcătuire a unităților de date individuale. Prin urmare, este necesară o modalitate de colectare și unificare a unor astfel de date diferite pentru nevoile analizei ulterioare. Există încă o problemă asociată - deoarece datele provin din surse diferite, poate apărea situația în care înregistrările individuale se contrazic reciproc (sau nu sunt consecvente unele cu altele).

Securitatea datelor reprezintă o altă problemă serioasă, chiar dacă nu prezintă importanță în contextul acestui manual. Nu orice sursă de date este sigură și poate să nu fie în conformitate cu politica organizației care dorește să o utilizeze. În general, este necesar să se acorde atenție creării de autorizări și autentificari, monitorizarea utilizatorilor care lucrează cu date, asigurarea securității datelor brute și a celor achiziționate, protecția comunicării, adică transferul de date.

Nu în cele din urmă, două probleme semnificative în contextul creării de predicții sau estimări sunt reprezentate de **datele lipsă** și **valorile aberante** din date. Ambele probleme sunt naturale. În cazul datelor lipsă, lipsește una dintre valorile măsurate necesare din setul de date. În cazul valorilor aberante, unele valori măsurate se află mult în afara corpului setului de date. Aceste două probleme sunt conținutul principal al secțiunii 2 a acestui manual.

CAPITOLUL 2

PROCESAREA ȘI ANALIZA DATELOR

Această parte a manualului a fost scrisă de Adam Dudáš, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

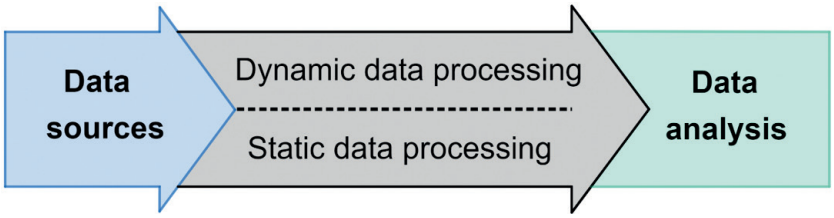


Figura 3. Procesarea și analiza datelor

În contextul acestui manual, când se lucrează cu date, pot fi identificate două activități principale: prelucrarea și analiza datelor. Obiectivul principal al acestei secțiuni este de a prezenta o serie de informații urmate de exemple de analiză a datelor, care nu pot fi efectuate în mod eficient fără date potrivite de intrare în acest proces. Pregătirea datelor într-o formă potrivită pentru o analiză lipsită de probleme se numește procesare de date. Necesitatea procesării și analizei datelor rezultă din mai multe caracteristici ale datelor moderne:

- **Surse de date** - în zilele noastre este ceva obișnuit să se lucreze cu seturi de date create prin combinarea unor seturi de date mai mici, colectate din surse diferite. Seturile de date create în acest fel pot proveni din baze de date diferite, senzori diferiți ai aceleiași rețele, dar și din combinația acestor două abordări. Seturile de date compuse din părți mai mici aduc cu ele probleme asociate cu acest tip de structură de date precum:
  - omogenizarea structurii datelor,
  - lucrul cu valori lipsă,
  - lucrul cu valori aberante.
- **Prelucrarea statică a datelor** – în sistemele moderne, există două tipuri de date care pot apărea sau care ar putea fi procesate într-un sistem dat – primul tip de date este reprezentat de datele statice. Datele statice sunt date care nu se modifică în timp. O abordare comună a procesării unor astfel de

date este așa-numita procesare în lot. În mod implicit, aceste loturi de lucrări includ încărcarea unui fișier, procesarea fișierului și scrierea rezultatului într-un fișier nou, fără ca utilizatorul să intervină manual.

- **Procesarea dinamică a datelor** – dacă sistemul utilizează surse de date ambientale (ca de exemplu un senzor activ în mod constant sau un set de senzori), acesta trebuie să fie capabil să capteze și să stocheze datele într-un timp apropiat de cel real. Nu contează ce tip de stocare este folosit în acest scop – este necesar să se accepte scalarea datorită volumului mare de date. Seturile de date care se schimbă dinamic se numesc fluxuri de date. Este posibil ca aceste seturi de date să trebuiască prelucrate, filtrate, agregate și pregătite pentru analiză. Spunem că datele prelucrate în acest mod sunt analizate. Problema procesării dinamice a datelor depășește ceea ce își propune acest manual, dar reprezintă o parte importantă a sistemelor moderne de date.
- **Analiza datelor** – este activitatea prin care se dobândesc cunoștințe necesare luării unor decizii mai bune, în contextul unui domeniu problematic selectat, și de a avea posibilitatea de predicție a unor valori pe baza datelor colectate sau de estimare a datelor nemăsurate. În a doua parte a acestei secțiuni, vor fi descrise tipurile de analiză a datelor și principalele probleme ale analizei datelor.

Această secțiune a manualului este axată pe prelucrarea datelor și pe problema selectată relativ la această acțiune. A doua parte a secțiunii reprezintă o introducere în analiza datelor care, ulterior, va fi discutată în detaliu în secțiunile următoare ale manualului.

2.1 PROCESAREA DATELOR

Nu există întotdeauna posibilitatea să se lucreze cu un set de date pregătit pentru o analiză directă a datelor. Prin urmare, este necesară **curățarea** și **formatarea** datelor înainte ca acestea să fie analizate.

O observație despre acest proces - prelucrarea datelor și toți pașii descriși în această secțiune a textului este recomandat să fie efectuate întotdeauna pe o copie a setului de date original, și nu pe setul de date în sine. De asemenea, în mod ideal ar trebui folosite metode care sunt sistematice și repetabile. La urma urmei, nu vrem să pierdem datele obținute cu greutate.

Consistența internă a setului de date

După cum s-a menționat la începutul acestei secțiuni a manualului, faptul că seturile de date moderne sunt obținute prin combinarea mai multor seturi de date mai mici, creează probleme legate de consistența internă a seturilor de date în sine. Această inconsecvență poate fi înțeleasă pe două niveluri - inconsecvența datelor în sine și inconsistența structurii setului de date.

În mod implicit, sunt câteva probleme tipice care pot cauza **inconsecvența datelor**:

- **Conversii de unități**. Atunci când sunt combinate două seturi de date care folosesc unități diferite pentru a măsura valorile atributelor (de exemplu, centimetri și milimetri), este necesară unificarea unității de măsură. De asemenea, este necesară unificarea seturilor de date măsurate pe continente care nu folosesc aceleași unități de măsură - de exemplu, pentru a măsura aceeași cantitate, se vor folosi centimetri în Europa și țoli (inches) în SUA.

- **Conversii numerice.** Valorile numerice înregistrate verbal trebuie convertite în numere. Această zonă de conversii include și probleme tipice legate de specificarea unităților în cadrul unei valori a atributului.
- **Conversii de nume.** În cazul înregistrării numelor persoanelor fizice este necesară unificarea modului de înregistrare a numelor și prenumelor. Cea mai mare problemă în cazul seturilor de date care utilizează atribute de nume de pe diferite continente sau din țări diferite sunt caracterele accentuate sau diacriticele (ca de exemplu š, cˇ, ä, ă, î, â, ș, ț).
- **Conversii de dată și oră**. În cazul analizelor care conțin informații despre oră, este necesar să se unifice formatul de înregistrare a orei și a datei în setul de date considerat.
- **Conversii financiare și valutare.** Valorile atributelor enumerate în diferite valute trebuie unificate cu una dintre monedele deja prezente în setul de date.

Al doilea caz este reprezentat de **inconsistența structurii setului de date**. Metoda ideală de stocare a datelor pentru analize ulterioare este metoda descrisă în secțiunea 1 a acestui manual - stocarea datelor sub formă de tabel. Cu toate acestea, acest lucru nu este întotdeauna posibil de realizat în mod simplu - în principal, problema fiind reprezentată de valorile lipsă.

Un set de date care conține valori lipsă este problematic de analizat atât cu ajutorul instrumentelor standard, cât și folosind orice instrumente software. Celulele dintr-un tabel imaginar din care lipsește o valoare, sunt completate cu valori NULL, care nu pot fi evaluate statistic și, în același timp, nu pot fi luate ca valori în sine (deoarece $0 \neq NULL$). Prin urmare, este necesară abordarea acestui tip de problemă într-un mod special.

Datele lipsă sau deteriorate

În acest manual, datele sunt privite ca măsurători ale unor mărimi fizice din lumea reală. Aceste măsurători sunt influențate de doi factori - instrumentul de colectare a datelor și metoda de prelucrare a datelor. În cazul ambilor factori, poate apărea o problemă, urmată de **pierderea sau deteriorarea datelor**. Dacă există o problemă cu instrumentul de colectare a datelor (parte distrusă a senzorului, înregistrări pierdute după o întrerupere a serverului și așa mai departe) există o pierdere de date care nu poate fi reconstruită. Opusul este pierderea sau deteriorarea datelor în timpul procesării acestora. Dacă există date brute disponibile, nu este o problemă corectarea erorii. Acest tip de pierdere sau deteriorare a datelor se numește **artefact**.

Dacă setul de date avut la dispoziție nu este complet, este necesară identificarea valorilor lipsă și apoi compensarea lor corespunzătoare. Problema este că unele dintre valorile lipsă este posibil să nu existe. Un exemplu în acest sens, poate fi valoarea unui atribut care conține ora sosirii într-o locație specificată, în situația în care nu s-a ajuns încă la acea locație.

Modalitățile de lucru cu valorile lipsă, atunci când datele brute nu sunt disponibile, pot fi împărțite în următoarele modalități de compensare:

- **Înlocuirea valorii lipsă cu o altă valoare** (0 / -1 / nonsens) – într-o astfel de abordare, se poate înlocui fiecare valoare lipsă (NULL) cu o valoare specială selectată. Această abordare nu este recomandată - valorile surogat pot fi adesea presupuse a fi corecte și vor fi interpretate incorect în analiza setului de date. De exemplu, dacă nu avem valoarea specificată a salariului unui angajat, nu se înlocuiește cu valoarea 0 sau -1, deoarece angajatul nu lucrează gratuit sau nu își plătește salariu pentru că vine la muncă.

- **Eliminarea interogărilor incomplete** – o situație mai bună comparativ cu cea precedentă este abordarea prin care se elimină fiecare înregistrare incompletă din setul de date. Această abordare este bună atunci când sunt date suficiente, dar poate conduce totuși la rezultate subiective.
- **Calculul valorilor lipsă (imputare)** – în cazul în care este necesară folosirea unor înregistrări care conțin valori lipsă, pot fi calculate aceste valori folosind una dintre metodele de mai jos. De asemenea, numim această abordare imputarea valorii.
  - *Imputarea folosind abordarea euristică* – în cazul în care există suficientă informație despre setul de date și relațiile din acesta, ar trebui să poată fi estimată valoarea unor atribute.
  - *Imputarea prin valoarea medie a atributului* – această metodă înlocuiește valorile lipsă cu valoarea medie pentru atributul dat. Utilizarea unei astfel de valori este avantajoasă din mai multe motive, cel mai important fiind că valorile medii ale atributelor nu sunt puternice în nicio direcție și, prin urmare, au un impact redus asupra potențialului predictiv al setului de date. Cu toate acestea, nu este întotdeauna adecvată înlocuirea valorilor lipsă cu valoarea medie a atributului dat. Pentru un salariu mediu, această abordare ar fi bună, însă o dată medie de sosire într-o anumită locație nu are sens.
  - *Imputarea prin valoare aleatorie a atributului* – pentru valoarea lipsă, se alege o valoare aleatorie a atributului dat care a fost înregistrată în setul de date.
  - *Imputarea folosind metode de învățare automată* – cea mai sofisticată abordare a calculării datelor lipsă este utilizarea metodelor de învățare automată. Cu toate acestea, aceste metode nu pot fi utilizate cu niciun set de date - sau, mai precis, nu este posibil să fie utilizate eficient pentru niciun set de date. Metodele de învățare automată funcționează pe baza corelațiilor dintre valorile individuale din setul de date, iar dacă aceste corelații sunt slabe sau inexistente, estimările valorilor setului de date vor fi inexacte. Această abordare este descrisă mai detaliat începând cu secțiunea 4 a acestui manual.

Aberațiile

Valorile aberante sunt valori care se află în afara corpului setului de date. Într-un set de date distribuit în mod normal, probabilitatea apariției unei valori în setul de date scade pe măsură ce crește distanța față de valoarea medie a setului de date dat. În orice caz, problema apare la seturile de date cu o distribuție anormală.

Valorile aberante apar în mai multe moduri:

- erori de măsurare,
- greșeli de tipărire în timpul procesării datelor,
- informații care nu sunt de încredere și care ridică semne de întrebare și despre alte înregistrări.

Valoarea aberantă este adesea o valoare reală care se abate de la situațiile standard (de exemplu, perioadele de poluare a aerului), așa că este necesară analizarea înregistrării în ansamblu.

Problema cu valorile aberante apare atunci când se încearcă **generalizări** bazate pe date care conțin valori aberante. În figura de mai jos, se poate observa o încercare de a descrie setul de date dat folosind o linie dreaptă. În partea stângă, este un set de date care conține 162 de înregistrări, dintre care 12 sunt situate semnificativ în afara corpului setului de date. În acest caz, se poate vedea că linia albastră care ar trebui să treacă prin centrul setului de date îl ratează complet, cu excepția unui punct. În partea dreaptă putem vedea același set de date după eliminarea celor douăsprezece valori aberante. Rezultatul generalizării este mult mai satisfăcător în acest caz.

Dacă se dorește o generalizare a setului de date, valorile aberante vor acționa ca un **element perturbator** și, prin urmare, este recomandat să nu se ia în considerare astfel de valori ale atributului (și înregistrările care le conțin), chiar dacă sunt corecte. În figura următoare – se dorește descrierea setului de date folosind o linie (funcție liniară de fapt). În cazul imaginii din stânga, linia a deviat din cauza prezenței unor valori aberante (colțul din dreapta jos al spațiului considerat). După eliminarea acestor valori aberante, se poate observa o creștere drastică a preciziei acestei generalizări (subfigura din dreapta).

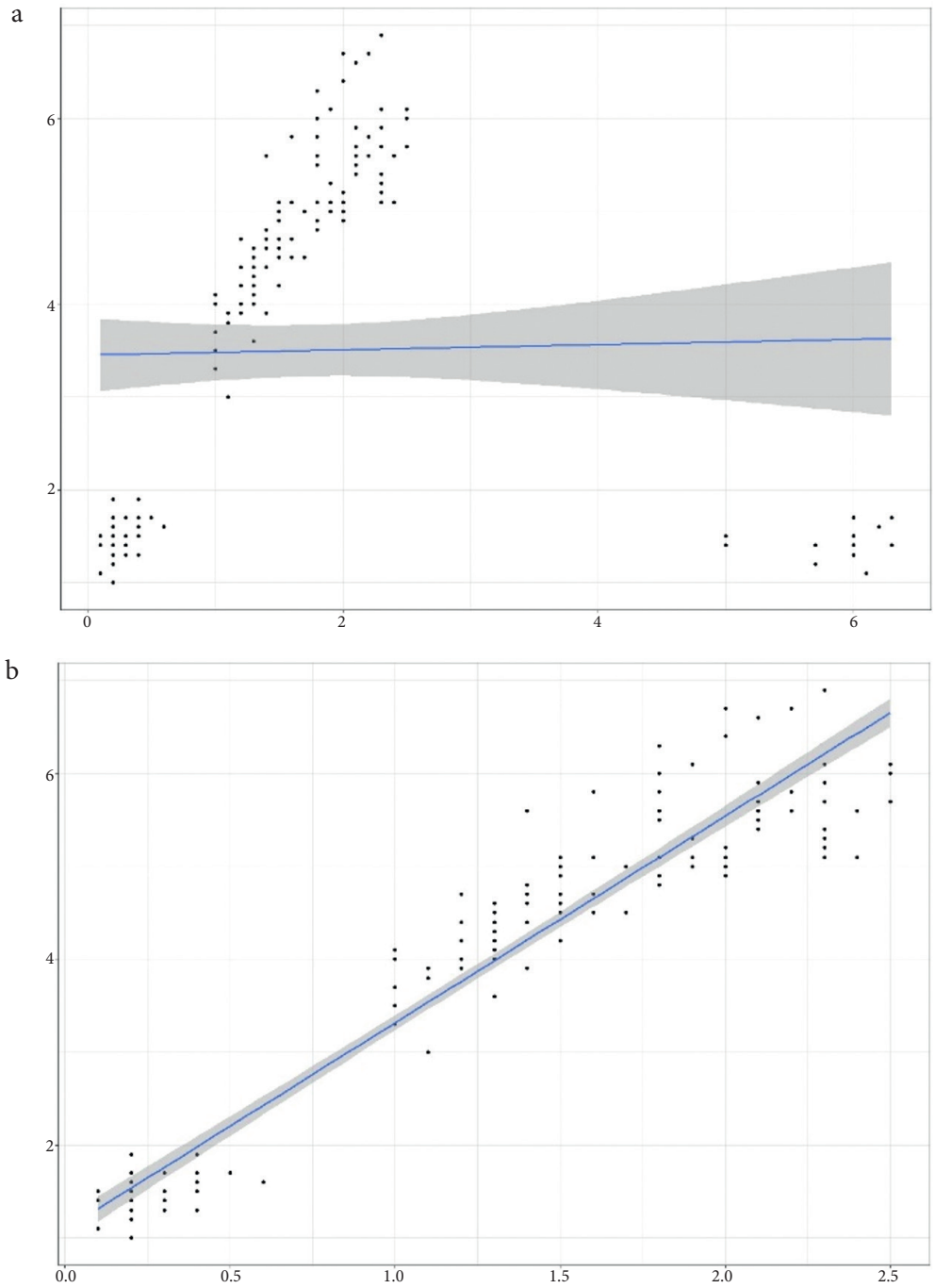


Figura 4. Descrierea setului de date cu ajutorul unei funcții liniare

2.2 ANALIZA DATELOR

Analiza datelor are ca scop **obținerea de cunoștințe (informații) utile din date** care să ajute la luarea unor decizii în cunoștință de cauză relativ la o problemă, predicția evenimentelor și a comportamentului obiectelor selectate, pe baza datelor prelucrate. Sunt cunoscute mai multe tipuri de analiză a datelor, în cadrul acestui manual fiind de interes doar trei tipuri, **cele mai frecvent utilizate**:

- **Analiza descriptivă și de diagnosticare a datelor** – cea mai simplă (și în același timp cea mai frecvent utilizată) metodă de analiză a setului de date. Analiza descriptivă urmărește tragerea unor concluzii (sau obținerea unor informații) din setul de date. Este folosită cel mai adesea în contextul descrierii și al măsurării proprietăților de bază ale setului de date - de exemplu, îndeplinirea planurilor în organizație. Analiza de diagnostic are ca scop clarificarea cauzelor care au condus la evenimentele identificate în analiza descriptivă. Analiza de diagnosticare este folosită adesea deoarece creează conexiuni între date și este utilă la identificarea unor tipare recurente în comportamentul obiectelor date. Acest tip de analiză a datelor se bazează pe crearea unor informații detaliate care pot fi utilizate ulterior, în mod repetat, la rezolvarea unor probleme similare.
- **Analiza exploratorie a datelor** – este cel mai natural tip de analiză pentru oameni (a căror majoritate este predominant vizuală). Este axată pe analiza datelor prin mijloace de explorare, cu ajutorul vizualizării datelor. Această analiză este eficientă în contextul identificării tiparelor și dependențelor în seturile de date, dar este importantă și din punct de vedere al prezentării rezultatelor altor analize. Pe lângă latura vizuală a analizei exploratorii a datelor, pot fi incluse aici și acțiuni asociate cu simplificarea setului de date sau reprezentarea setului de date - de exemplu reducerea dimensiunii, o operație în care se proiectează un set de date n-dimensional pe un set de date $m$dimensional în timp ce $m < n$.
- **Analiza predictivă a datelor** – este o extensie a tipurilor de analiză menționate mai sus. Obiectivul acesteia este de a utiliza datele colectate pentru a face predicții logice despre rezultatelor evenimentelor sau pentru a prezice și estima valori pe care nu le-am măsurat. În acest tip de analiză a datelor sunt utilizate metode de modelare bazate pe statistici, care implică necesitatea utilizării tehnologiilor de calcul pentru crearea unor modele de predicție. A se reține că predicțiile care sunt rezultatul modelelor create în timpul analizei predictive reprezintă doar estimări pentru setul de date dat și, prin urmare, acuratețea lor depinde în mod direct de calitatea datelor date.

Toate aceste tipuri de analiză a datelor funcționează în mod obișnuit cu doar două tipuri de probleme de rezolvat - problema de regresie și problema de clasificare. Următoarea parte a acestei secțiuni se concentrează pe descrierea acestor două probleme.

Problema regresiei

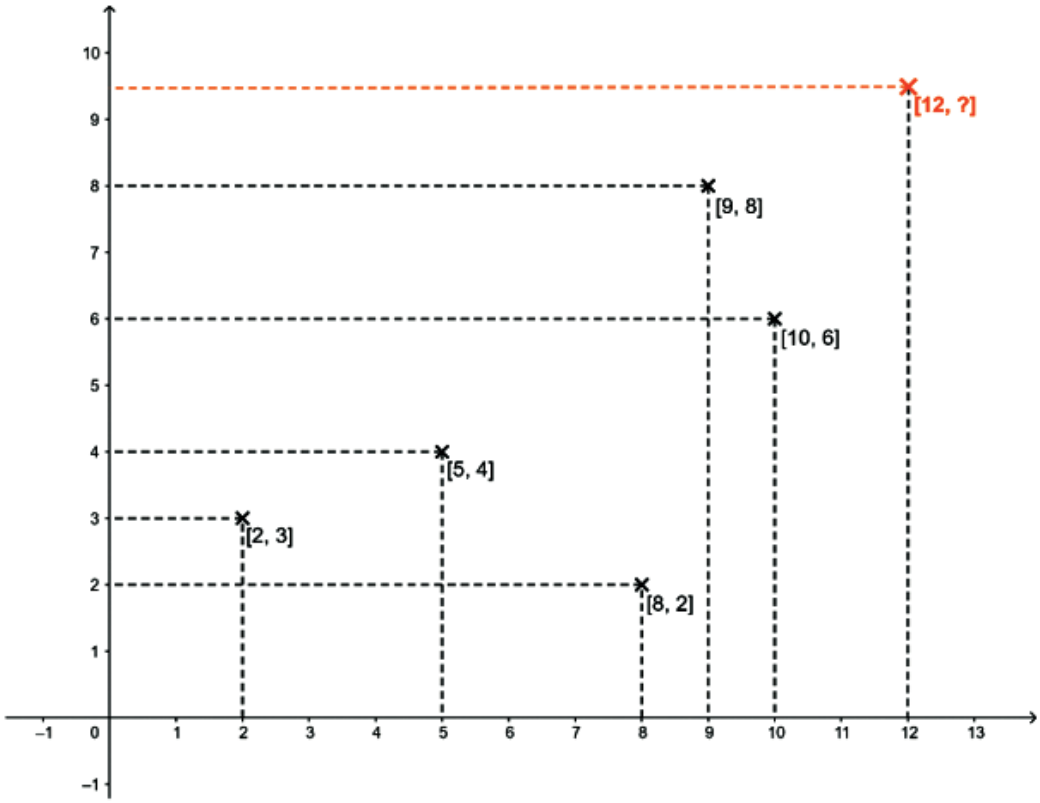


Figura 5. Exemplu de set de date pentru problema regresiei

În figura de mai sus se poate vedea un set de date care conține cinci puncte definite de valorile a două atribute - aceste valori sunt măsurate pe axele x și y, așa că ne vom referi la ele ca fiind valorile atributelor x și y. Acest set de date constă din punctele [2, 3], [5, 4], [8, 2], [9, 8] și [10, 6].

Problema regresiei, în acest caz, ar fi estimarea valorii reale a atributului $y \in R$ dacă se cunoaște valoarea lui x și modelul celor cinci puncte anterioare. Prin urmare, există o entitate care conține o valoare pentru atributul $x = 12$ și valoarea necunoscută pentru atributul y, pe care trebuie să o calculăm.

În general, se poate defini acest tip de problemă ca estimarea sau predicția valorii numerice a variabilei y pe baza valorii variabilei x, unde $x, y \in R$.

Problema clasificării

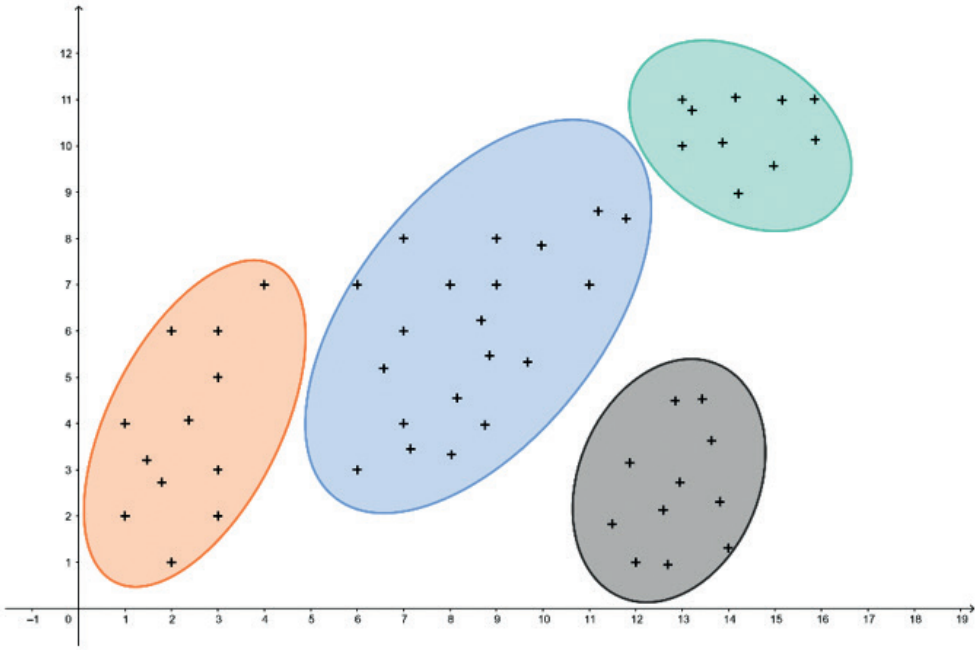


Figura 6. Exemplu de set de date pentru problema clasificării

O descriere generală a acestei probleme arată astfel: Având modelul x și spațiul X, să se estimeze ce valoare a atributului asociat $y \in \{1,..., n\}$ va fi dobândită de modelul x.

- **clasificarea ierarhică**, în care clasele sunt ele însele clasificate pe grupe, procesul fiind repetat la diferite niveluri pentru a forma un arbore,
- **partiționarea**, în care clasele se exclud reciproc, formând astfel o partiție a setului de entități,
- **aglomerarea**, în care clasele sau aglomerările se pot suprapune, iar o aglomerare și complementul său sunt tratate ca tipuri diferite de clasă.

CAPITOLUL 3

METODE DE EȘANTIONARE A DATELOR

Această parte a manualului a fost scrisă de Adam Dudáš, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

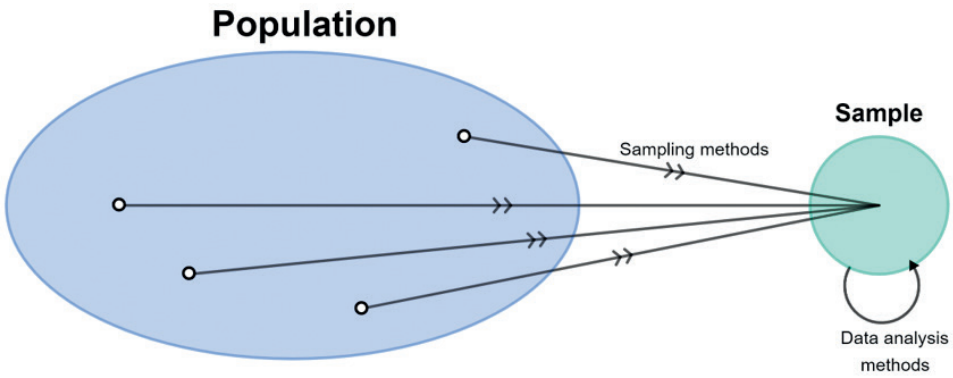


Figura 7. Ilustrarea grafică a eșantionării unei părți a populației

Eșantionarea se definește ca selectarea unei părți a populației (set de date) care este cea mai reprezentativă a acesteia, astfel încât să poată fi utilizată la analiza datelor și obținerea unor informații utile despre populație. Tehnica utilizată în eșantionarea populației este cunoscută ca metoda eșantionării.

Astfel, un eșantion este o parte sau o **fracțiune dintr-o populație** care a fost selectată astfel încât aceasta să permită realizarea unor deduceri despre populație. Populația este totalitatea tuturor subiecților sau subiecților studiați.

Sunt cunoscute mai multe metode de eșantionare. Cel mai adesea, acestea sunt împărțite în **două grupuri** - metode de eșantionare bazate pe probabilitate și non-probabilitate. Este important de subliniat faptul că tipul care va fi utilizat la selectarea unui eșantion depinde în întregime de problema care trebuie rezolvată. În general, însă, se poate spune că există:

- **metodele non-probabilistice** depind de persoana care alcătuiește eșantionul - deci este foarte ușor să se obțină rezultate la care o persoană s-ar putea aștepta (chiar dacă acestea pot să nu fie adevărate pentru întreaga populație).
- **metodele probabilistice** care evită mai mult sau mai puțin această problemă.

3.1 METODE DE EȘANTIONARE NON-PROBABILISTICE

Selectarea unui eșantion din populație depinde în principal de raționamentul persoanei care alcătuiește eșantionul. Prin urmare, aceste metode pot duce la distorsiunea unor valori raportat la populație. Unele metode de eșantionare non-probabilistică depind doar de comoditatea persoanei care compilează eșantionul - de exemplu, metoda de eșantionare numită **eșantionare convenabilă**, în care membrii populației sunt selectați în funcție de comoditatea compilatorului. În mod similar, în metoda numită eșantionare judecatorească, eșantionul este compilat pe baza cunoștințelor non-date ale persoanei care compilează eșantionul.

Unele metode de eșantionare non-probabilistică sunt directe, fiind doar o eșantionare de „bun simț". Prin urmare, în acest manual este oferită descrierea detaliată doar a uneia dintre metodele de eșantionare non-probabilistice.

Metoda de eșantionare intenționată

În această metodă de eșantionare, persoana care extrage eșantionul selectează membrii populației cu un scop specific. Deoarece membrii populației nu au aceleași șanse să fie incluși în eșantion, vorbim de o metodă de eșantionare neprobabilistică.

Un exemplu de astfel de eșantionare este reprezentat de necesitatea realizării unei analize a studenților din anul III de la domeniul de studiu informatică, adică de a crea un eșantion de studenți de la anul III informatică din populația tuturor studenților, din toți anii, din toate domeniile de studiu. Este evident că nu se dorește includerea în eșantion a studenților din primul, al doilea, al patrulea sau al cincilea an. De asemenea, nu sunt incluși în eșantion studenții care studiază în domenii precum matematica aplicată, biologia sau chimia criminalistică.

Pentru a descrie rezultatele metodelor individuale de eșantionare în acest punct al capitolului, vor fi folosite mostre din setul de date Iris, care este descris în Anexa A a acestui manual. Sarcina pentru metoda de eșantionare intenționată poate fi următoarea: *Să se analizeze valorile lungimii și lățimii sepalei pentru un anumit tip de floare - Iris Setosa.*

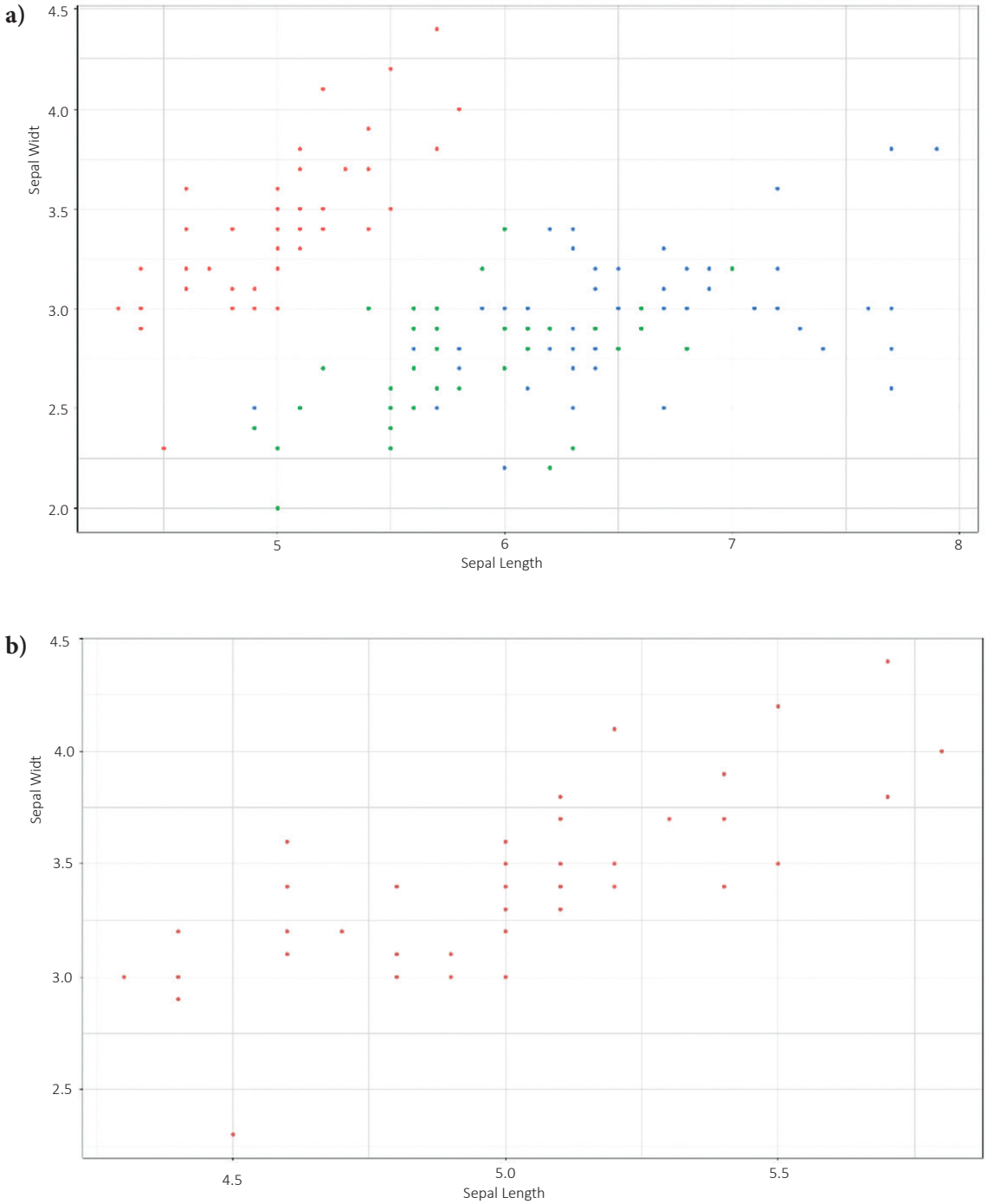


Figura 8. Comparație între valorile lungimii și lățimii sepalei în a) setul de date complet al irisului din subfigura din stânga (fiecare clasă a florii de iris este marcată cu propria sa culoare) și b) eșantionul setului de date constând dintr-o clasă - Iris setosa.

3.2 METODE DE EȘANTIONARE PROBABILISTICE

În aceste metode, toți membrii populației luate în considerare au șanse egale să fie selectați ca parte a eșantionului. Aceste metode previn (sau reduc) subiectivismul celui care face eșantionul atunci când adaugă obiecte la eșantion (au fost menționate în secțiunea metode non-probabilistice). Există tipuri diferite de metode de eșantionare probabilistică, utilizate în situații diverse la selectarea eșantioanelor din diferite populații.

Aceste metode impun utilizatorului să cunoască populația luată în considerare, metoda de eșantionare adecvată și modul de utilizare în fiecare situație întâlnită.

Există câteva metode de eșantionare probabilistică - eșantionare în mai multe etape, eșantionare în cluster, eșantionare sistematică și așa mai departe. În acest manual, accentul va rămâne pe patru metode probabilistice de eșantionare, simple și aplicabile într-o gamă largă de probleme rezolvate.

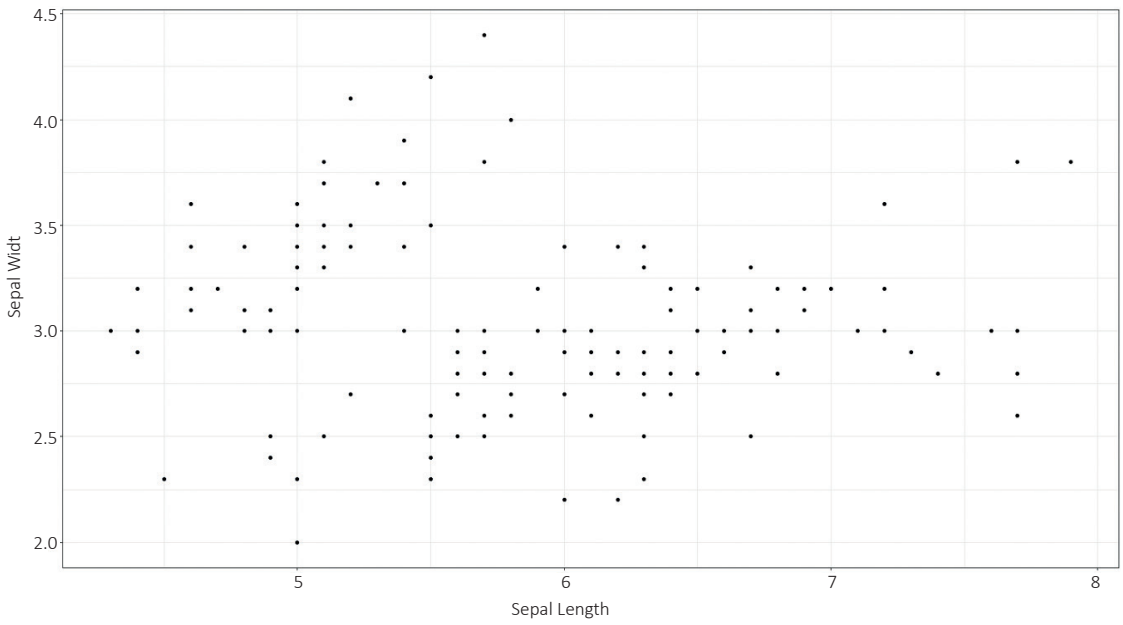
Metoda de eșantionare aleatorie simplă

Această metodă se bazează pe selecția aleatorie a indivizilor dintr-o populație. Cu alte cuvinte, un anumit eșantion este selectat din orice populație, fără niciun model matematic sau decizii logice. Deoarece fiecare individ (înregistrare) are aceeași șansă de a deveni parte a eșantionului, această metodă este considerată cea mai reprezentativă dintre metodele de eșantionare probabilistică.

Metoda simplă de eșantionare aleatorie are un singur parametru de intrare - dimensiunea dorită a eșantionului.

Exemplu: Populația noastră (stânga) conține 150 de indivizi (înregistrări) și selectăm aleatoriu 25 de reprezentanți din ea (dreapta) - acest set reprezintă un eșantion aleator simplu pentru noi.

a)



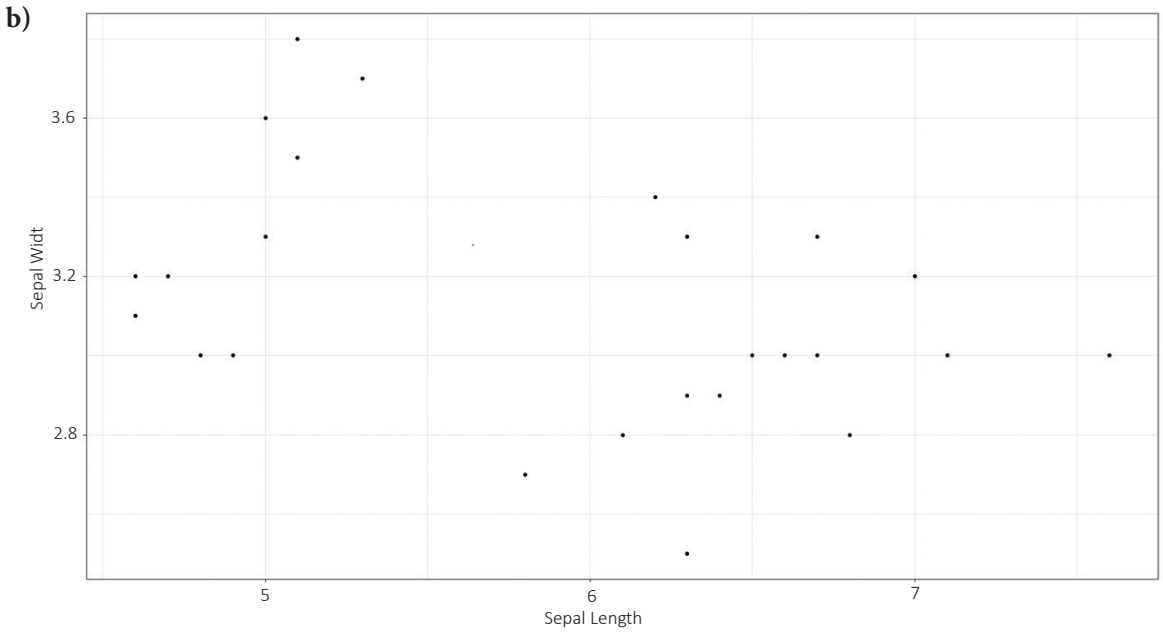


Figura 9. Populația considerată (stânga) conține 150 de indivizi (înregistrări) și din aceasta sunt selectați aleatoriu 25 de reprezentanți (dreapta) - acest set reprezintă un eșantion aleator simplu pentru cazul considerat.

Metoda de eşantionare cluster

O metodă în care un set întreg de date este împărțit în secțiuni sau grupuri (clustere). Clusterele sunt identificate și incluse în eșantion pe baza unui anumit atribut, cel mai adesea categoric, precum culoarea părului, sexul, etc. Metoda este aplicabilă la crearea unor eșantioane adecvate analizei unor subseturi deja existente de date.

Metoda de eșantionare în cluster are un singur parametru de intrare - atribut care trebuie utilizat pentru gruparea datelor.

Exemplu: Se împarte setul de date (rândul de sus) în funcție de atributul de clasă, care are trei valori - iris setosa, iris versicolor și iris virginica. Folosind metoda de eșantionare în cluster, se pot crea trei eșantioane (rândul de jos de cifre) care pot fi utilizate pentru a analiza caracteristicile indivizilor din clasele considerate. Este evident că eșantionul 2 (jos, mijloc) nu este potrivit pentru a trage concluzii despre întreaga populație, ci doar despre submulțimea populației al cărei atribut de clasă are aceeași valoare ca eșantionul 2. O utilizare adecvată a acestei metode este, de exemplu, pregătirea unei analize statistice care să descrie grupuri individuale, care va permite compararea caracteristicilor claselor de flori de iris.

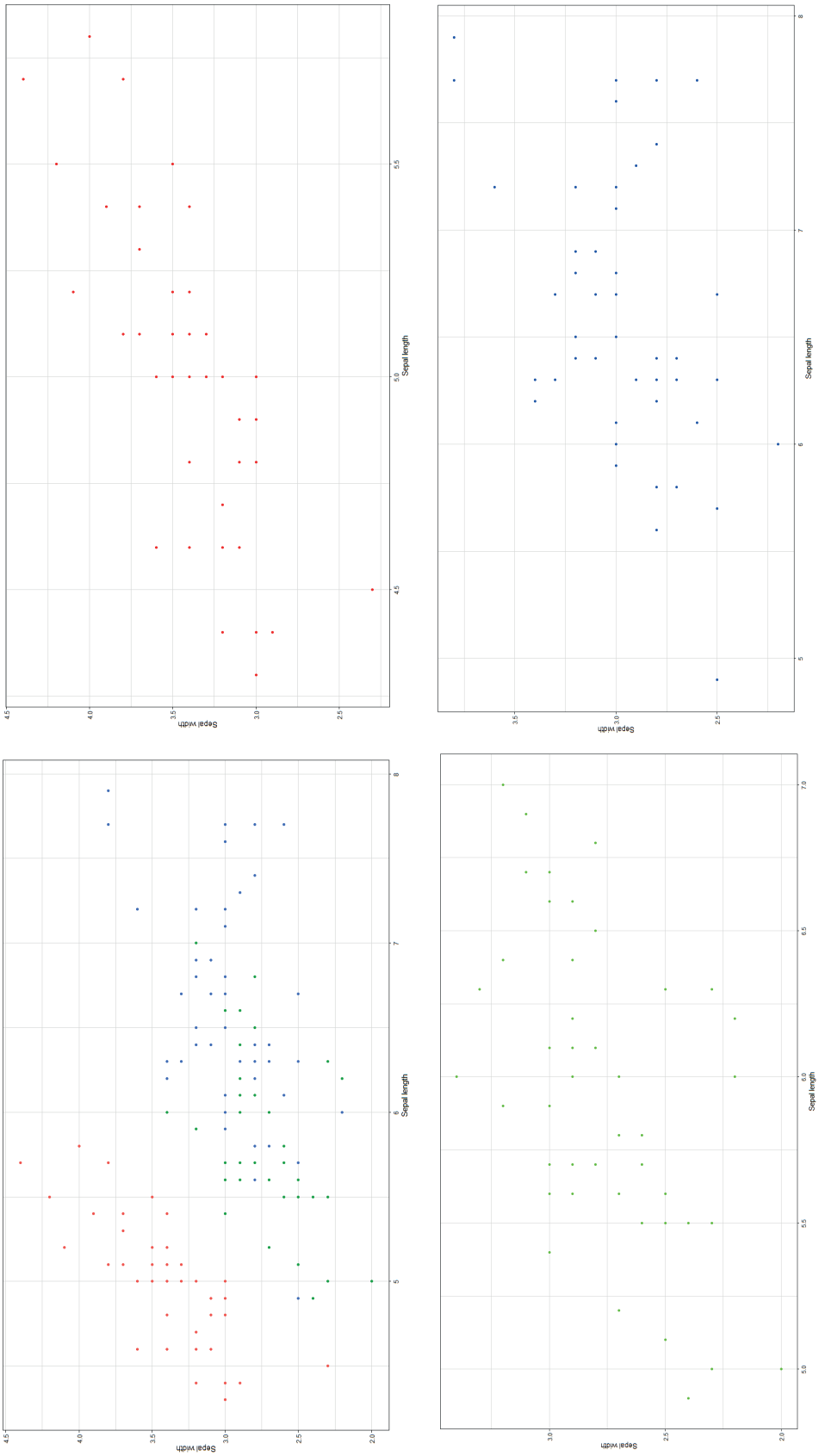


Figura 10. Ilustrarea exemplului prezentat anterior

Metoda de eșantionare sistematică

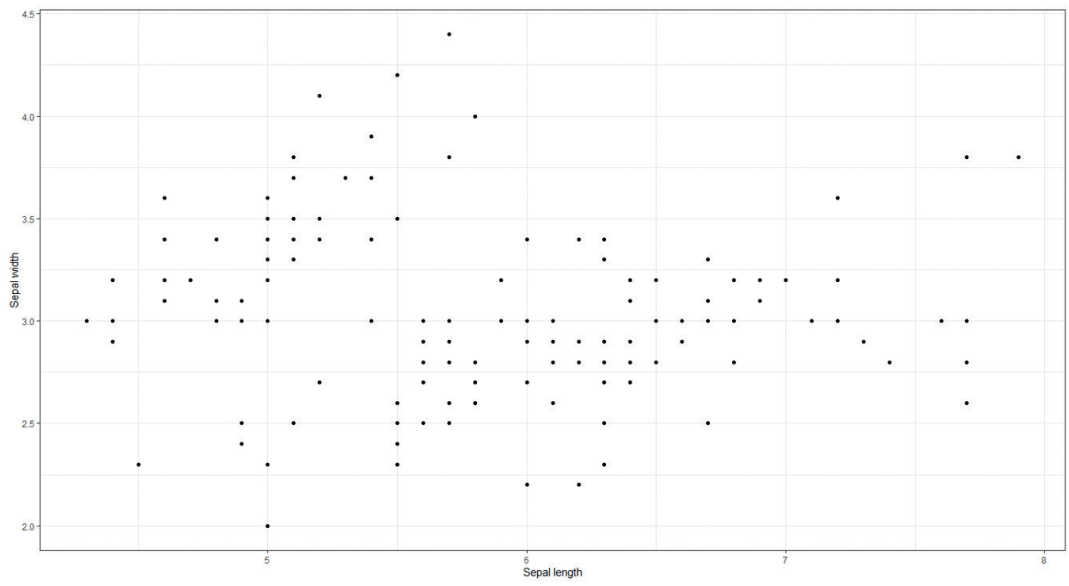
Această metodă este utilizată pentru a selecta la intervale regulate membrii eșantionului din populație. Acest tip de metodă de eșantionare are un domeniu de aplicare predefinit și, prin urmare, este tehnica de eșantionare care necesită cel mai puțin timp.

Metoda de eșantionare sistematică are doi sau trei parametri de intrare:

- punctul de plecare selectat pentru crearea eșantionului (primul individ care aparține eșantionului),
- intervalul în care indivizii sunt adăugați la eșantion și care implicit dau dimensiunea eșantionului,
- sau intervalul în care indivizii sunt adăugați la eșantion și crește dimensiunea eșantionului creat.

Exemplu: Deoarece prin metoda de eșantionare aleatorie simplă au fost selectați 25 de indivizi care reprezentau populația folosită, se dorește folosirea în continuare a metodei de eșantionare sistematică pentru crearea unui eșantion de 25 de indivizi. Setul inițial este format din 150 de reprezentanți și din moment ce 150/25 = 6, va fi selectat fiecare al șaselea individ (în cazul în care se începe de la prima înregistrare din setul de date). În figura de mai jos, se poate vedea întreaga populație (stânga) și eșantionul (dreapta) format din 25 de indivizi care au fost selectați prin procedura descrisă mai sus.

a)



b)

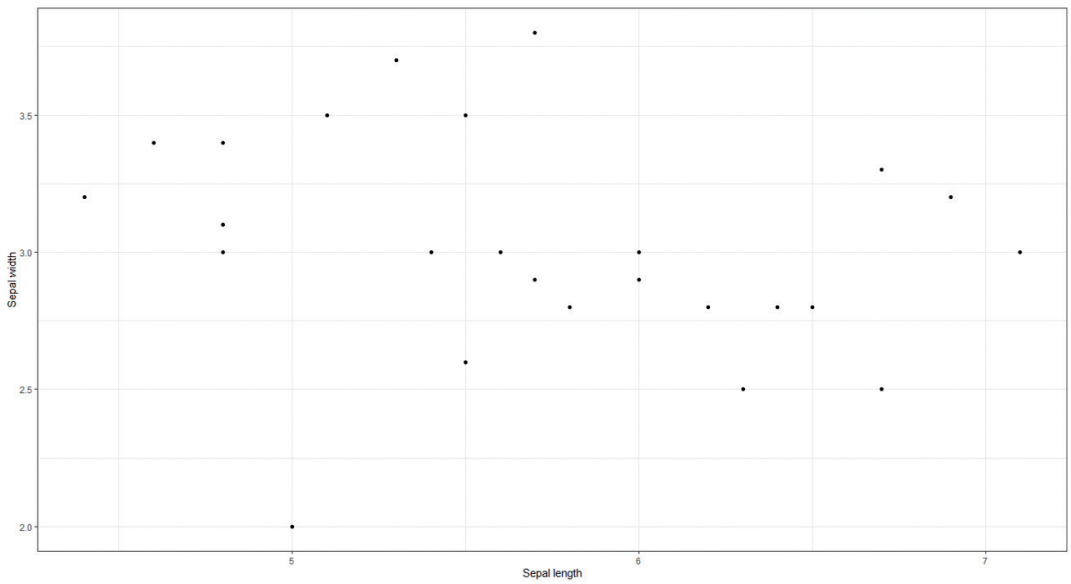


Figura 11. Ilustrarea exemplului de eșantionare sistematică

Metoda de eșantionare stratificată

Cu ajutorul metodei eșantionării stratificate, întregul set de date este împărțit în grupuri mai mici, disjuncte care reprezintă întreaga populație. În comparație cu metoda de eșantionare în cluster, această metodă creează grupuri de date, folosind o limită nou definită, pe unul dintre atributele prezente în setul de date original. Metoda de eșantionare în cluster nu creează aceste limite, ci utilizează unul dintre atribute (categorice) pentru a identifica grupurile din date.

Exemplu: Eșantioanele create folosind metoda de eșantionare stratificată din figura de mai jos pot fi definite ca intervale definite pe atributul lungimea Sepalei. Fiecare probă este diferită, dar în fiecare probă există reprezentanți a căror valoare a lungimii sepalei este similară din punctul de vedere al metodei selectate. În cazul considerat, sunt împărțite mostrele cu 1 cm de la cel mai mic la cel mai mare:

$$lungimea\_sepalei \in (4, 5]$$
$$lungimea\_sepalei \in (5, 6]$$
$$lungimea\_sepalei \in (6, 7]$$
$$lungimea\_sepalei \in (7, 8]$$

Astfel, figura conține patru eșantioane diferite prin intermediul culorii – eșantionul 1 marcat cu roșu, eșantionul 2 marcat cu verde și așa mai departe.

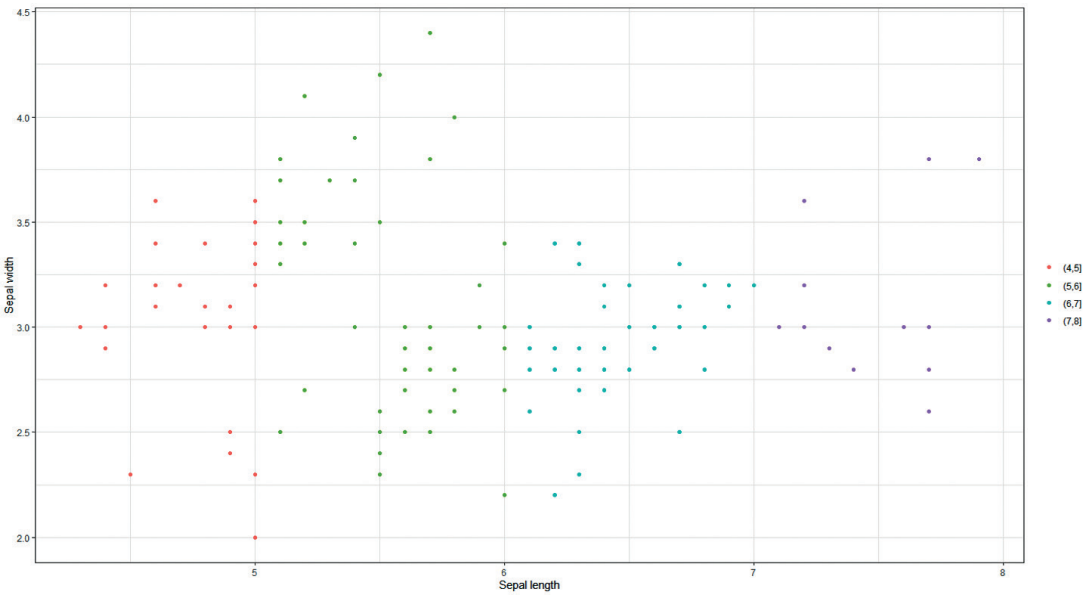


Figura 12. Ilustrarea celor patru eșantioane diferențiate prin intermediul culorii

3.3 CÂTEVA CUVÎNTE DESPRE CALİTATEA EȘANTİONULUİ

Eșantionarea corectă este una dintre tehnicile necesare atunci când se lucrează cu date mari (a se vedea secțiunea 1.1 ca referință). Metodele menționate mai sus creează eșantioane a căror calitate poate fi evaluată din mai multe puncte de vedere. În cadrul acestui manual, sunt prezentate doar două criterii pentru descrierea calitatății eșantionului. Doar unul dintre acestea este cu adevărat critic pentru utilizatorii obișnuiți (în principal utilizatori din afara domeniului informaticii):

- **Viteza de colectare a mostrelor** - în zilele noastre se utilizează programe moderne care conțin adesea funcții și pachete optimizate. Astfel de funcții includ metode de eșantionare, a căror implementare în instrumentul selectat a trecut prin optimizări și metode care au crescut eficacitatea funcției date (acest lucru este aproape garantat). Dacă se dorește crearea unui eșantion dintr-un set mare de date standard (nu de tip Big data), utilizatorul nu va intra în contact cu problema performanței insuficiente a sistemului, ceea ce ar duce la extinderea eșantionului (sau, un caz nedorit, la incapacitatea de a crea un eșantion). Cu toate acestea, atunci când se lucrează cu seturi de date mari adevărate, sistemul standard încetează să mai fie suficient de eficient. Un exemplu în acest sens este cel de creare a unui eșantion pe un set de date (populație) cu dimensiunea de o sută de milioane de înregistrări, în timp ce fiecare înregistrare conține șaisprezece atribute (a se reține faptul că acesta nu este un set de date atât de mare). Când s-a utilizat funcția metodei de eșantionare sistematică cu parametrul de intrare 4 (pentru a crea o dimensiune a eșantionului de 25% din populație) în limbajul R pe un computer standard, de utilizator, nu s-a putut crea un eșantion. Această problemă poate fi rezolvată în mai multe moduri, cea mai comună fiind cea de utilizare a metodelor de calcul de înaltă performanță și în cloud.
- **Reprezentativitatea unui eșantion.** O problemă mai importantă pentru evaluarea calității eșantionului decât viteza de colectare a eșantionului, este calitatea capacității acestuia de a descrie populația din care a fost creat. Ca și în cazul precedent, este evident că o astfel de măsurătoare nu va fi univer-

sală - așa cum s-a afirmat în descrierea metodei de eșantionare a grupurilor de mai sus, un eșantion dintr-un grup (un subset specific de date) nefiind potrivit pentru a trage o concluzie despre întreaga populație. În cazul în care este potrivită compararea caracteristicilor eșantionului și ale populației, se poate proceda în mai multe moduri, conform obiectivelor stabilite:

- **Descrierea statistică a eșantionului** - în cazul în care se dorește descrierea datelor cu un număr mic de valori, putem calcula metrici statistice critice. Din aceste valori numerice, se pot obține cunoștințe adecvate pentru a lucra în continuare cu date.
- **Vizualizarea eșantionului** - Big data sunt renumite pentru complexitatea vizualizării lor, de aceea este o idee bună să fie creat un eșantion reprezentativ care să conțină mai puțini indivizi și astfel să fie mai ușor de vizualizat.
- **Analiza potențialului predictiv al eșantionului** – dacă obiectivul este construcția de modele de predicție sau estimare bazate pe învățarea automată, este oportună analiza potențialului predictiv al atributelor individuale folosind metode precum analiza corelației sau folosind modele de tip arbore de decizie.

Toate aceste abordări sunt descrise detaliat în secțiunea 4 a acestui manual.

3.4 CÂTEVA CUVINTE DESPRE MĂRIMEA EȘANTIONULUI

În cazul în care este necesară compilarea unui eșantion dintr-o anumită populație, ar putea reprezenta o problemă **dimensiunea acestui eșantion pentru a obține rezultatele dorite** - pentru a putea obține cu precizie cunoștințele/informațiile de care avem nevoie. Soluția aceastei probleme depinde de metoda utilizată și de obiectivele stabilite:

- În cazul metodelor de eșantionare precum eșantionarea în cluster sau cea stratificată, soluția **este dată de metoda utilizată**. Aceste metode creează eșantioane a căror dimensiune este definită de apariția unei anumite valori în date și, astfel, în acest caz, nu este standard să se ia în considerare o dimensiune a eșantionului diferită de dimensiunea clusterului identificat prin metodă.
- În alte cazuri, cum este cel al eșantionării aleatorii, este necesară folosirea unui **model care identifică dimensiunea eșantionului potrivită nevoilor utilizatorului**. Acest model este definit implicit pentru două tipuri de populații - populații cu un număr limitat de indivizi, respectiv o populație fără limită a numărului de indivizi. În funcție de nevoile urmărite, se va lua în considerare cea mai naturală dintre aceste versiuni - o populație limitată:

$$\bar{n} = \frac{\frac{z^2 \bar{p}(1-\bar{p})}{\varepsilon^2}}{1 + \frac{z^2 \bar{p}(1-\bar{p})}{\varepsilon^2 N}}$$

unde

- este dimensiunea eșantionului,
- z este așa-numitul scor-z care arată nivelul de încredere, cel mai frecvent fiind setat la 90%, 95%, sau 99%, cu coeficienții de scor-z prezentați în tabelul următor:

Nivel de încredere	Scor-z
90%	1.65
95%	1.96
99%	2.58

Tabel 3. Tabel de scor-z care conțin valori de scor-z precalculate și pot fi căutate online pentru alte valori ale nivelului de încredere.

- este proporția populației - un procent (sau o fracțiune) din populație asociată cu problema cercetată (valoarea standard pentru populația necunoscută este setată la $p = 0,5$).
- ε este marja de eroare stabilită de utilizator.
- N este dimensiunea populației utilizate.

Cea mai ușoară modalitate de a calcula dimensiunea eșantionului este cea de utilizare a calculatoarelor online disponibile gratuit pentru dimensiunea eșantionului, care funcționează pe principiile enumerate mai sus.

CAPITOLUL 4

BAZELE ANALIZEI EXPLORATORII A DATELOR

Această parte a manualului a fost scrisă de Adam Dudáš, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

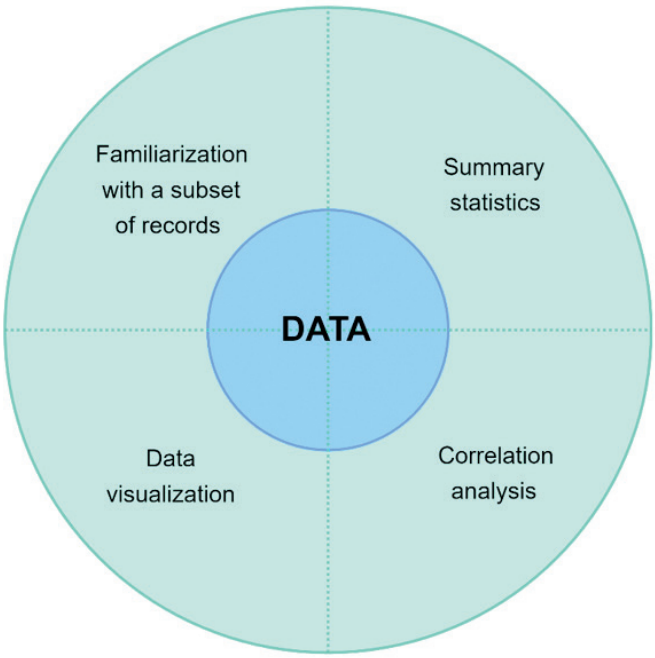


Figura 13. Analiza exploratorie a datelor

Așa cum a fost menționat în secțiunile anterioare ale acestui manual, în procesul de analiză a datelor se poate lucra cu întregul set de date sau cu un eșantion creat, folosind metodele menționate în secțiunea 3 a manualului. Pentru o analiză de bază a datelor descriptive, sunt utilizate metode de statistică descriptivă care oferă instrumentele de captare a caracteristicilor unui set sau eșantion de date.

Metodele de statistică descriptivă se bazează pe metode de agregare de reprezentare a subseturilor de date, de exemplu, valoarea medie a atributului, minimul, frecvența sau suma valorilor. Astfel de metode pot fi numite agregate (metode de reducere a datelor).

Atunci când sunt analizate datele prin utilizarea metodelor statisticii descriptive, sunt folosite în principal următoarele trei concepte:

- **Măsuri de tendință centrală**, cu ajutorul cărora sunt căutate centrele de date în jurul cărora datele sunt grupate sau distribuite.
- **Măsuri de variabilitate** sunt măsuri care descriu distribuția datelor în spațiul considerat, adică cât de departe sunt măsurătorile individuale față de cele centrale, identificate cu ajutorul măsurilor de tendință centrală.
- **Analiza corelației** care se bazează pe calculul coeficienților care descriu potențialul de predicție între atributele individuale din setul de date. Acești coeficienți sunt importanți în crearea modelelor de învățare automată, dar și în vizualizarea datelor, care reprezintă o parte esențială a acestei secțiuni a manualului.

În această secțiune a manualului, ne vom confrunta cu prima versiune a analizei datelor, ceea ce este foarte firesc din punct de vedere uman - **Analiza Exploratorie a Datelor** (EDA, din denumirea în engleză). După cum sugerează și numele, este o analiză a datelor folosind explorarea datelor cu scopul de a găsi modele și tendințe într-o anumită populație sau eșantion. În forma sa de bază, acest tip de analiză se realizează prin explorare vizuală și, prin urmare, metodele de vizualizare a datelor vor fi o parte importantă a unei astfel de analize.

Pentru a afla care părți ale setului de date considerat sunt adecvate pentru vizualizare și care nu, sunt folosite cunoștințe de bază obținute cu ajutorul metodelor de statistică descriptivă.

4.1 METODE STATISTICE DE BAZĂ

Din punct de vedere al metodelor statistice de bază, putem identifica metode cu care este măsurată centralitatea în date – sunt căutate centrele de date în jurul cărora datele sunt grupate sau dens distribuite. Cele mai comune astfel de metode sunt:

- **Media** este valoarea medie a elementelor matricei. Media este potrivită pentru caracterizarea datelor distribuite simetric, fără valori aberante (de exemplu, înălțimea, greutatea). Datele distribuite simetric sunt acelea în care numărul de elemente ale setului de date este similar atât sub, cât și peste limita mediei, în mod ideal același. Relația de calcul pentru valoarea medie este următoarea:

$$\mu_A = \frac{\sum_{i=1}^{n} Ai}{n},$$

unde μ_A este valoarea medie a atributului A, n este numărul de entități care conțin atributul A, și A_i este valoarea i a acestui atribut.

- **Mediana** este valoarea de mijloc a matricii sortate. Mediana este conectată cu ideea de simetrie, ceea ce înseamnă că numărul elementelor care sunt situate sub mediană este același ca și numărul de elemente de deasupra acesteia. O excepție de la această regulă este reprezentată de seturile de date care conțin un număr par de elemente (se alege una dintre cele două valori medii ca mediană - într-un

set de date rezonabil, acestea ar trebui să fie destul de aproape una de alta). Spre deosebire de medie, mediana este o valoare reală a unui atribut, deci este mai potrivită dacă datele conțin valori aberante sau sunt distribuite asimetric (de exemplu, salariile angajaților dintr-o anumită zonă). Mediana este calculată cu ajutorul următoarei relații:

$$median_A = \frac{(n+1)}{2} \text{ lea element al setului sortat } A,$$

unde n este numărul de entități conținute de atributul A.

- **Modul** este elementul care apare cel mai frecvent în atribut. Cu toate acestea, această măsură este dificil de utilizat și nu este precisă în majoritatea sarcinilor analitice. Un exemplu în acest sens ar putea fi modulul salariilor menționate, care ar fi în mare parte egal cu 0, deoarece exact 0 este câștigat de majoritatea oamenilor - șomeri, copii, pensionari.

În figura următoare, este oferită o vizualizare a măsurilor de centralitate pentru un set simplu de date. Poate fi observat comportamentul tipic al valorilor medii și mediane în datele care sunt distribuite normal în spațiul considerat - adică aceste valori sunt destul de apropiate una față de alta. Valoarea modului este greu de estimat, deoarece este valoarea elementului atributului care apare cel mai frecvent (prin urmare poate fi mare, poate fi scăzută, poate fi undeva la mijloc).

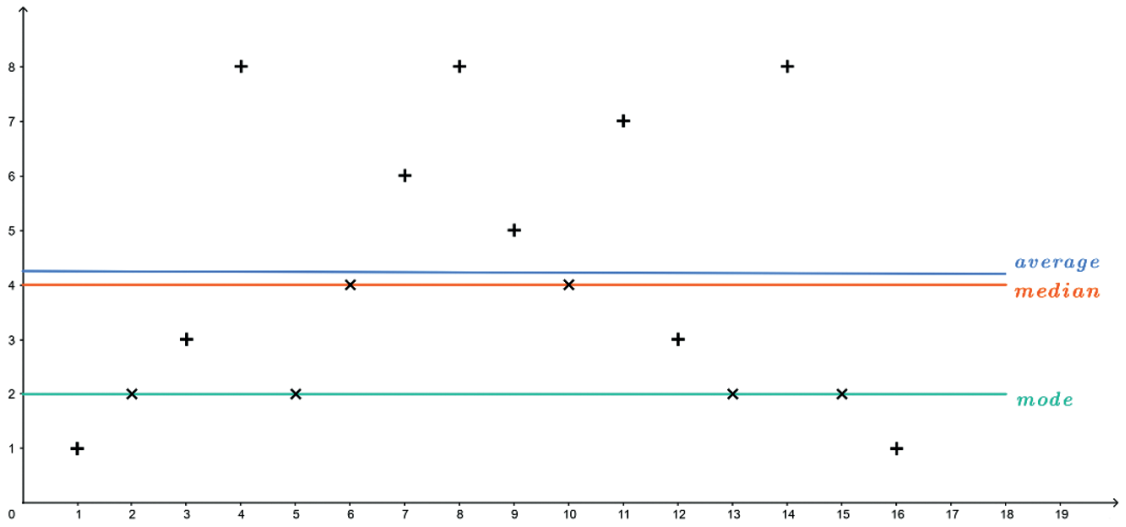


Figura 14. Vizualizarea măsurilor de centralitate pentru un set simplu de date

Pe lângă aceste măsuri standard, **frecvența valorilor** dintr-un atribut este de mare importanță. Prin titlul **distribuția frecvenței** înțelegem o listă, un tabel sau un grafic care arată frecvența de apariție a diferitelor rezultate dintr-un eșantion (set de date). Fiecare intrare din tabel conține frecvența (sau numărul) apariției valorii într-un anumit grup sau interval. Un exemplu de astfel de distribuție de frecvență pentru setul de date simplu utilizat mai sus arată după cum urmează:

Tabel 4. Frecvența valorilor unui atribut	
valoarea	frecvența
1	2
2	4
3	2
4	2
5	1
6	1
7	1
8	3

Acest tabel poate fi reprezentat cu ajutorul **graficului** de mai jos. O astfel de vizualizare a graficelor de frecvență este esențială, mai ales din punct de vedere al cunoașterii datelor și al detectării posibile a valorii aberante.

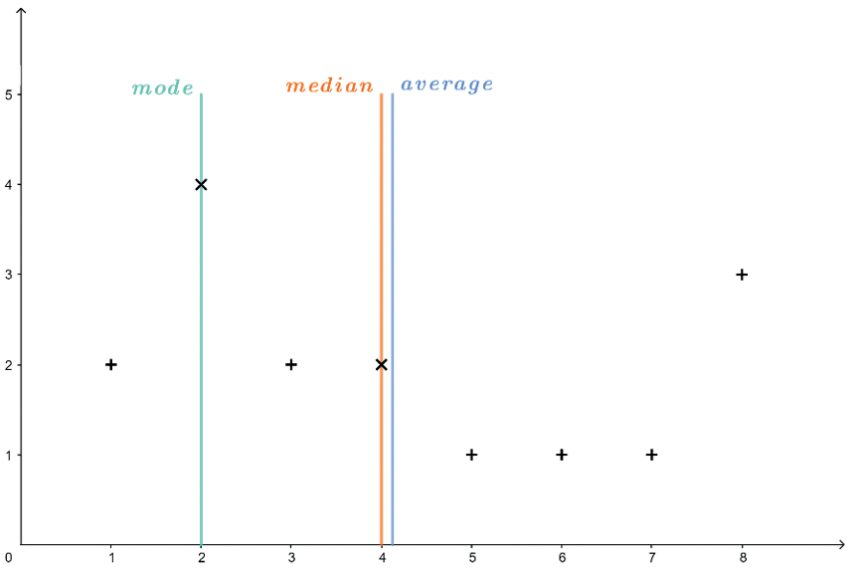


Figura 15. Vizualizarea graficelor de frecvență

Cealaltă față a monedei reprezentate de măsurile statistice standard este variabilitatea datelor în spațiul considerat. Cea mai comună măsură a variabilității este așa-numita **deviație standard** (σ), definită ca suma pătratelor diferenței dintre elementele individuale ale atributului și valoarea medie:

$$\sigma_A = \frac{\sqrt{\sum_{i=1}^{n}(A_i - \mu_A)^2}}{n-1},$$

unde μ_A este valoarea medie a atributului A, n este numărul de entități care conțin atributul A, și A_i este valoarea i a acestui atribut.

O mărime similară abaterii standard este **varianța** calculată astfel:

Exemplu: Se consideră următorul set simplu de date format dintr-un atribut cu cinci măsurători A = [20, 60, 40, 70, 50]. Să se calculeze media, mediana și abaterea standard pentru acest set de date.

$$\mu_A = \frac{\sum_{i=1}^{n} A_i}{n} = \frac{20+60+40+70+50}{5} = \frac{240}{5} = 48$$

$$\text{median}_A = \frac{(n+1)}{2} = \frac{6}{2} = 3rd \quad \textit{lea element al setului sortat} \rightarrow [20, 40, 50, 60, 70] = 50$$

$$\sigma_A = \frac{\sqrt{\sum_{i=1}^{n} (A_i - \mu_A)^2}}{n-1}$$

$$= \frac{\sqrt{(20-48)^2 + (60-48)^2 + (40-48)^2 + (70-48)^2 + (50-48)^2}}{4} = \frac{\sqrt{1480}}{4}$$

$$\sigma_A = \frac{\sqrt{\sum_{i=1}^{n} (A_i - \mu_A)^2}}{n-1}$$

$$\sqrt{370} \approx 19.235$$

$$= \frac{\sqrt{(20-48)^2 + (60-48)^2 + (40-48)^2 + (70-48)^2 + (50-48)^2}}{4} = \frac{\sqrt{1480}}{4}$$

$$\sqrt{370} \approx 19.235$$

Această abatere standard este relativ mare, ceea ce este natural, deoarece setul de date în sine este foarte împrăștiat. Vizualizarea acestor măsurători este prezentată în figura de mai jos.

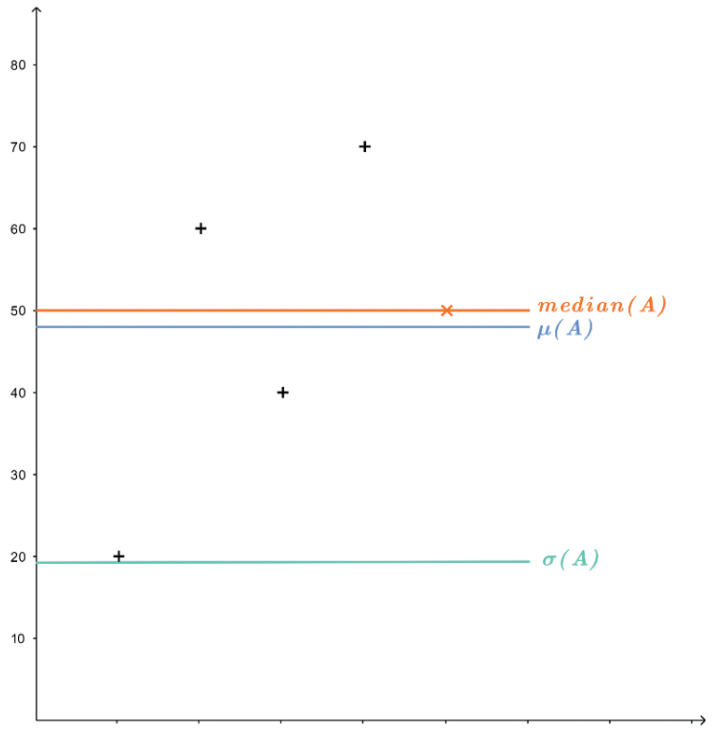


Figura 16. Vizualizarea măsurătorilor pentru exemplu prezentat

Metodele de calcul pentru centralitate și variabilitate sunt utile pentru descrierea setului de date cu ajutorul unui număr mic de valori. O astfel de abordare a descrierii setului de date este numită **descriere prin agregare**.

Example: Se consideră atributul A = [1, 2, 3, 8, 2, 4, 6, 8, 5, 4, 7, 3, 2, 8, 2, 1] (prezentat la începutul secțiunii 4.1). Acest set de date poate fi descris prin utilzarea a trei valori agregate, ca de exemplu (min(A), μ(A), max(A)), prin urmare A = (1, 4.125, 8).

Ultimul indicator din metricile statistice simple este **distribuția datelor setului de date în spațiu**. Utilizarea valorii medii a atributului și a abaterii standard caracterizează metrica menționată. Într-un set de date normal distrubuite, cel puțin $(1 - 1/k^2)$ dintre puncte se află la o distanță $k\sigma$ sau mai puțin față de medie. Un astfel de set de date nu conține valori aberante și este un candidat excelent pentru metodele de învățare automată și de analiză a datelor de inteligență artificială.

Exemplu: Pentru atributul nostru familiar A = [20, 60, 40, 70, 50], se poate calcula distribuția după cum urmează:

$\mu A = 48$

$\sigma A \approx 19.235$

$2\sigma = 2 * 19.235 \approx 38.47$

Se poate vedea că cel puțin 3 valori ar trebui să fie la cel mult 38,47 unități depărtare de valoarea medie (48). Acest lucru este valabil pentru toate măsurătorile atributului A.

4.2 ANALIZA CORELAȚIEI

Valorile statistice de bază descrise în secțiunea anterioară sunt indicatori importanți cu ajutorul cărora pot fi descrise datele. Din punct de vedere al obiectivelor analizei datelor însă, așa-numita analiză a corelației este considerată o metrică mult mai puternică.

În cazul în care setul nostru de date conține mai mult de un atribut numeric, putem măsura corelația dintre subseturile de două elemente ale acestui set de date. Fie două atribute ale setului de date A – A_1, A_2. Aceste atribute se corelează unul cu celălat când atributul A1 are **potențial** de predicție pentru atributul A_2. Un astfel de potențial de predicție arată **prezența unor tendințe și modele** în setul de date și posibilitatea de a construi modele analitice care funcționează cu datele.

Se măsoară corelația a două variabile utilizând **coeficientul de corelație** $r(A_1, A_2)$, care arată în ce măsură atributul A_1 este funcție de atributul A_2 și vice versa. Acest coeficient de corelație poate lua valori din intervalul [−1, 1], astfel:

- **1** indică **corelația completă** a două atribute. Cu alte cuvinte, când valoarea atributului A_1 crește, valoarea atributului A_2 crește de asemenea. Dacă există o corelație completă între valorile a două va-

riabile, vorbim despre un potențial predictiv puternic și, astfel, aceste atribute sunt potrivite pentru predicția reciprocă.

- **0** indică cea mai proastă situație din punct de vedere al corelării a două valori, pe care o numim **necorelare**. Când coeficientul de corelație dintre două atribute este apropiat sau egal cu 0, acestea sunt independente și sunt inutilizabile din punct de vedere al construirii modelelor analitice.
- **-1** este opusul corelației complete, pe care o numim **anticorelație**. În acest caz, putem identifica o tendință în care valoarea atributului A_1 crește, valoarea atributului A_2 scade sau invers. Ca și în cazul corelării complete, aceasta este o condiție satisfăcătoare pentru construirea modelelor analitice.

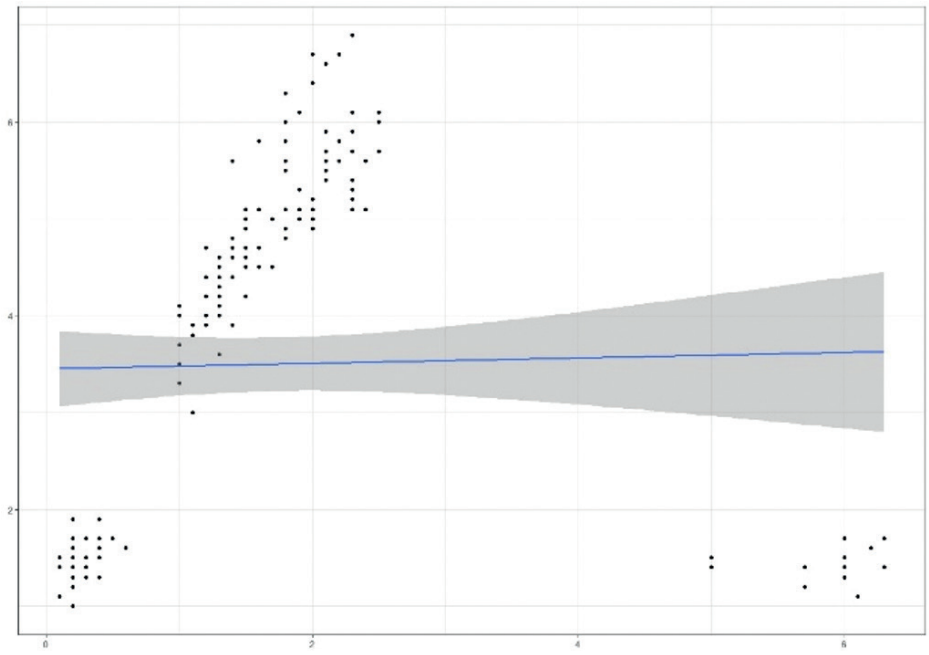
Se folosesc două metode standard pentru a analiza corelațiile și pentru a măsura coeficienții de corelație - desigur, există mult mai multe astfel de metode. În cele ce urmează, ne vom concentra pe coeficientul de corelație Pearson și pe coeficientul de corelație al rangului Spearman.

Coeficientul de corelație Pearson

Primul și cel mai puternic coeficient utilizat pentru a măsura corelația dintre două atribute ale setului de date este coeficientul de corelație Pearson. Acest coeficient este axat pe **predicția liniară a valorilor** și descrie relația dintre atributele A și B:

$$r = \frac{\sum_{i=1}^{n}(A_i - \mu(A))(B_i - \mu(B))}{\sqrt{\sum_{i=1}^{n}(A_i - \mu(A))^2}\sqrt{\sum_{i=0}^{n}(B_i - \mu(B))^2}},$$

unde $\mu(A)$ este valoarea medie a atributului A, similar, $\mu(B)$ este valoarea medie a atributului B, și n este numărul de măsurători (dimensiunea verticală a setului de date). Această dependență aparentă de valoarea medie aduce cel mai mare dezavantaj al coeficientului de corelație Pearson - sensibilitatea la valori aberante (după cum se arată în figura de mai jos).



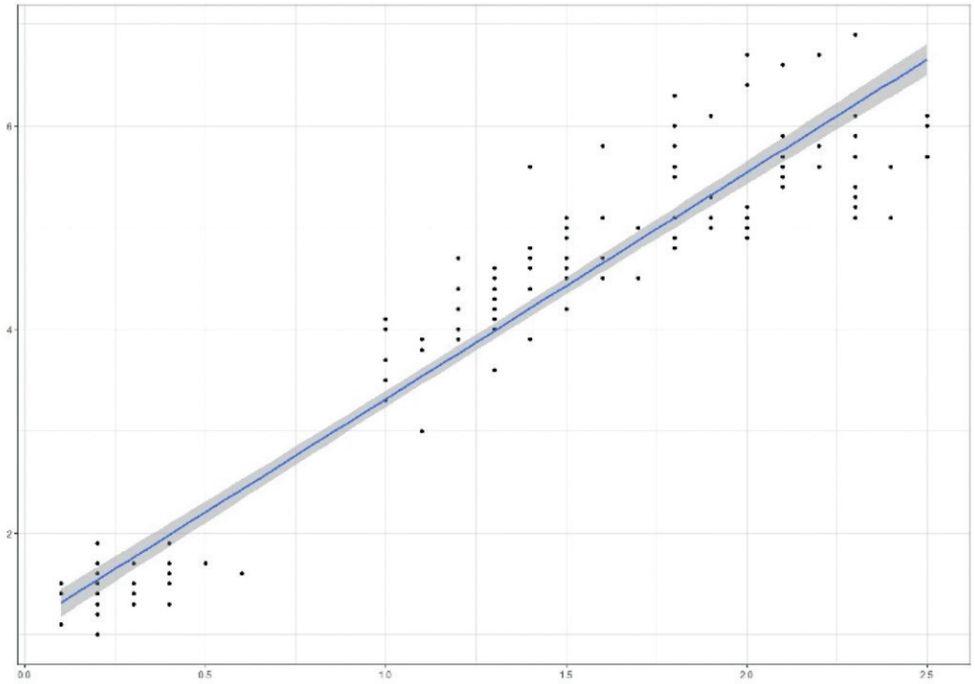


Figura 17. Sensibilitatea la valori aberante

Cu ajutorul coeficientul de corelație Pearson, se caută o dreaptă care descrie valorile atributelor datelor. În imaginea din stânga, se poate vedea o vizualizare a comparației valorilor a două atribute dintr-un set de date, în care există valori aberante (jos, dreapta). Se poate observa, de asemenea, că în dreptul dreptei alese, datele lipsesc complet și semnificativ (cu excepția unui punct) - de aici se poate concluziona că acest coeficient de corelație Pearson nu este potrivit pentru măsurarea potențialului de predicție pentru acest set de date. În partea dreaptă, sunt prezentate aceleași atribute ale setului de date după eliminarea valorilor aberante. În acest caz se poate observa că dreapta descrie tendințele prezente în date.

Prin urmare, **coeficientul de corelație Pearson poate fi utilizat când** atributele *A* și *B* conțin:

- relații liniare,
- distribuție normală (gaussiană),
- fără valori aberante.

Coeficientul de corelație a rangului (Spearman)

Pentru a lucra cu **seturi de date care conțin relații neliniare** sau cu valori aberante, se folosește un alt tip de coeficient de corelație - coeficientul de corelație a rangului (coeficientul Spearman). Această metodă de măsurare a corelației dintre atribute creează o **ierarhie** (clasare) a valorilor atributelor individuale în ceea ce privește funcționalitatea sa.

Exemplu: Se consideră atributul A = [a0 = 4, a1 = 8, a2 = 2, a3 = 6]. Ierarhia sau clasamentul menționat mai sus arată după cum urmează:

Când a1 > a3 > a0 > a2 rank(a1) = 1, rank(a2) = 4, și așa mai departe.

În acest mod, se măsoară **monotonia valorilor** din cadrul atributului și, prin urmare, se poate afirma că acest coeficient de corelație a rangului Spearman este cel mai potrivit pentru seturile de date cu relații monotone între atribute - când valoarea unuia dintre atribute crește, valoarea celuilalt nu scade niciodată, sau vice versa. Pe de altă parte, acest tip de coeficient de corelație nu se recomandă să fie utilizat atunci când există valori repetate (adică același rang) în setul de date. Acest efect este atenuat odată cu creșterea dimensiunii setului de date. Coeficientul de corelație a rangului Spearman se calculează cu formula următoare:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)},$$

unde $d = rang(a_i) - rang(b_i)$ și n este numărul de entități din atributele considerate.

Exemplu: Atenție - exemplul următor este popular în rândul studenților, ei s-ar putea să-și amintească acest exemplu mai mult decât de ideile de corelare în sine. Există două atribute - prețul kebabului și distanța locului kebab față de universitate.

Tabel 5. Exemplu de evidențiere a unor atribute

Indice	*Distanța în metri*	*Prețul în euro*
1	*10*	*4*
2	*70*	*3,50*
3	*85*	*3,30*
4	*100*	*3,20*
5	*130*	*3,80*
6	*195*	*2,90*
7	*215*	*3,10*
8	*300*	*3,90*
9	*420*	*3,15*
10	*505*	*3*

În primul rând, se calculează valoarea coeficientului de corelație a rangului Spearman. Această metodă necesită crearea unui clasament pentru ambele atribute și calculul valorilor lui d și d^2 după cum urmează:

Tabel 6. Clasamentul ambele atribute și calculul valorilor lui d și d^2						
Indice	*Distanța în m*	*Rang (distanța)*	*Prețul în euro*	*Rang(preț)*	*d*	*d^2*
1	*10*	*10*	*4*	*1*	*9*	*81*
2	*70*	*9*	*3.50*	*4*	*5*	*25*
3	*85*	*8*	*3.30*	*5*	*3*	*9*
4	*100*	*7*	*3.20*	*6*	*1*	*1*
5	*130*	*6*	*3.80*	*3*	*3*	*9*
6	*195*	*5*	*2.90*	*10*	*-5*	*25*
7	*215*	*4*	*3.10*	*8*	*-4*	*16*
8	*300*	*3*	*3.90*	*2*	*1*	*1*
9	*420*	*2*	*3.15*	*7*	*-5*	*25*
10	*505*	*1*	*3*	*9*	*-8*	*64*

Prin urmare, valorile din acest tabel pot fi înlocuite în relația de calcul al coeficientului de corelație a rangului Spearman:

$$\sum d^2 = 256$$

$$n = 10$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} = 1 - \frac{6 * 256}{10(100-1)} = 1 - \frac{1536}{990} = 1 - 1.55 = -0.55$$

Astfel, a fost găsită o corelație de -0,55 între aceste două atribute, care ar putea fi considerată ca o anticorelație moderat puternică.

Se calculează mai departe valoarea coeficientului de corelație Pearson. Pentru a calcula acest tip de coeficient, trebuie determinată valoarea medie a distanței μ (distanța) și valoarea medie a prețului kebabului μ (prețul):

- *μ(distanța) = 203, denumită în continuare μ(d)*
- *μ(prețul) = 3,39, denumită în continuare μ(p)*

Tabelul de mai jos conține mai multe coloane, dar acestea sunt doar precalculări ale părților necesare în relația finală a coeficientului de corelație Pearson (Datorită lipsei de spațiu în tabelul prezentat, folosim d pentru distanță și p pentru prețul kebabului)

Tabel 7. Coeficientul de corelație Pearson							
indice	*d*	*p*	*$d - \mu(d)$*	*$p - \mu(p)$*	*$(d - \mu(d))^2$*	*$(p - \mu(p))^2$*	*$(d - \mu(d))\ (p - \mu(p))$*
1	*10*	*4*	*-193*	*0.61*	*37 249*	*0,3721*	*-117,73*
2	*70*	*3,50*	*-133*	*0,11*	*17 689*	*0,0121*	*-14,63*
3	*85*	*3,30*	*-118*	*-0,09*	*13 924*	*0,0081*	*10,62*
4	*100*	*3,20*	*-103*	*-0,19*	*10 609*	*0,0361*	*19,57*
5	*130*	*3,80*	*-73*	*0,41*	*5 329*	*0,1681*	*-29,93*
6	*195*	*2,90*	*-8*	*-0,49*	*64*	*0,2401*	*3,92*
7	*215*	*3,10*	*12*	*-0,29*	*144*	*0,0841*	*-3,48*
8	*300*	*3,90*	*97*	*0,51*	*9 409*	*0,2601*	*49,47*
9	*420*	*3,15*	*217*	*-0,24*	*47 089*	*0,0576*	*-52,08*
10	*505*	*3*	*302*	*-0,39*	*91 204*	*0,1521*	*-117,78*

Deci poate fi calculat coeficientul de corelație Pearson după cum urmează:

$$\sum((\mathrm{d} - \mu(\mathrm{d}))^2) = 232\,710$$

$$\sum((\mathrm{p} - \mu(\mathrm{p}))^2) = 1.3905$$

$$\sum((\mathrm{d} - \mu(\mathrm{d}))\,(\mathrm{p} - \mu(\mathrm{p}))) = -252.05$$

$$\mathrm{r} = \frac{\sum_{i=1}^{n}(A_i - \mu(A))(B_i - \mu(B))}{\sqrt{\sum_{i=1}^{n}(A_i - \mu(A))^2}\sqrt{\sum_{i=0}^{n}(B_i - \mu(B))^2}} = \frac{-252.05}{\sqrt{232\,710}\sqrt{1.3905}} \approx \frac{-252.05}{569.232} \approx -0.44$$

S-a obținut o corelație Pearson de -0,44 între cele două atribute, care ar putea fi considerată o anticorelație moderat puternică. Pentru mai multe informații despre interpretarea rezultatelor coeficientului de corelație, se recomandă consultarea finalului acestei secțiuni a manualului.

Matricea de corelație și harta termică de corelație

Seturile de date conțin rareori doar două atribute, ceea ce conduce la necesitatea măsurării coeficienților de corelație între toate atributele sale. În acest scop, se folosește o matrice de corelație - un tabel care conține valorile coeficientului de corelație măsurat între toate perechile posibile de atribute din setul de date. În tabelul de mai jos, se poate vedea coeficientul de corelație măsurat între atributele A_1, A_2, A_3 (matricea de corelație), $r(A_1, A_2) = 0,238$, $r(A_1,A_3) = -0,834$, și așa mai departe.

Tabelul 8. Coeficientul de corelație măsurat între atributele A1, A2, A3

	A_1	A_2	A_3
A_1	*1*	*0,238*	*-0,834*
A_2	*0,238*	*1*	*-0, 362*
A_3	*-0,834*	*-0,362*	*1*

Această matrice are două **proprietăți naturale** - este simetrică față de diagonală și această diagonală conține întotdeauna valorile coeficientului de corelație egale cu 1. Corelația atributului A1 cu el însuși este întotdeauna $r(A1, A1) = 1$, indiferent de metoda utilizată, care este, de asemenea, naturală, deoarece valoarea atributului *A1* este complet dependentă de valoarea atributului *A1*.

O astfel de metodă de analiză a corelației este potrivită numai pentru un anumit număr de atribute din setul de date studiat. Evident, pentru un set de date care conține zeci de atribute, o astfel de matrice ar fi confuză și greu de citit. Prin urmare, o așa-numită hartă termică a corelației sau diagramă de corelație o înlocuiește adesea. Pentru matricea de corelație prezentă mai sus, harta termică poate fi construită după cum urmează:

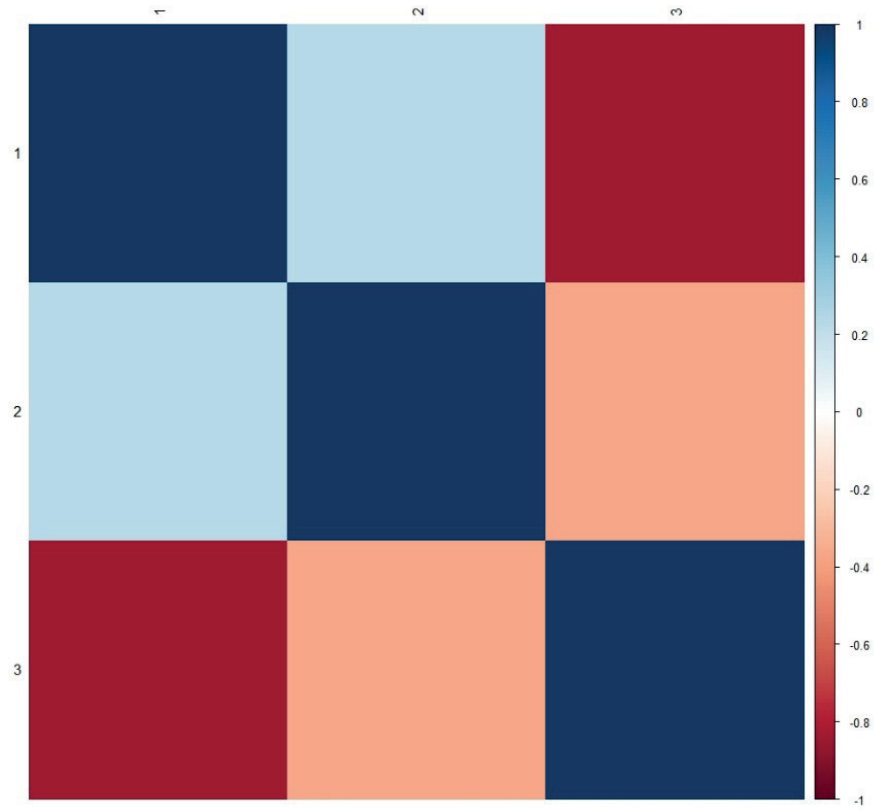


Figura 18. Harta termică a corelației

O astfel de hartă termică a corelației reprezintă doar o proiecție simplă a matricei de corelație într-o grilă de culoare, în care culoarea câmpului este definită de valoarea coeficientului de corelație pentru perechea dată de atribute. Pentru o citire mai ușoară, scala (din dreapta) conține intervalul de valori posibile pentru coeficientul de corelație și este indicată în dreapta hărții termice a corelației. În loc să se caute numere apropiate de extremele intervalului [1,–1] în matricea de corelație, se caută în harta de corelație câmpuri de grilă roșu închis sau albastru închis care indică fiecare aceeași proprietate și sunt mai ușor de identificat pentru majoritatea oamenilor.

Exemplu: Se dă setul de date Iris descris în Anexa A a acestui manual. Acest set de date conține cinci atribute măsurate pe 150 de entități, dintre care patru sunt atribute numerice, iar al cincilea conține o valoare lingvistică care indică clasa pentru entitatea dată. Ca parte a analizei corelației, este imposibil să se lucreze cu atribute lingvistice, așa că vor fi luate în considerare doar setul de date de dimensiunea 150×4. Valorile matricei de corelație Pearson a setului de date Iris (oarecum decupat) sunt următoarele:

Tabelul 9. Valorile matricei de corelație Pearson a setului de date Iris				
	lungimea sepalei	lățimea sepalei	lungimea petalei	lățimea petalei
lungimea sepalei	1	-0.1093692	0.8717542	0.8179536
lățimea sepalei	-0.1093692	1	-0.4205161	-0.3565441
lungimea petalei	0.8717542	-0.4205161	1	0.9627571
lățimea petalei	0.8179536	-0.3565441	0.9627571	1

Desigur, acest set de date nu conține un număr mare de atribute, astfel încât crearea unei hărți de corelație nu este necesară pentru analiza corelației. Cu toate acestea, va fi prezentată harta termică ca o demonstrație a proiecției valorilor coeficientului de corelație în scala de culori din harta termică a corelației.

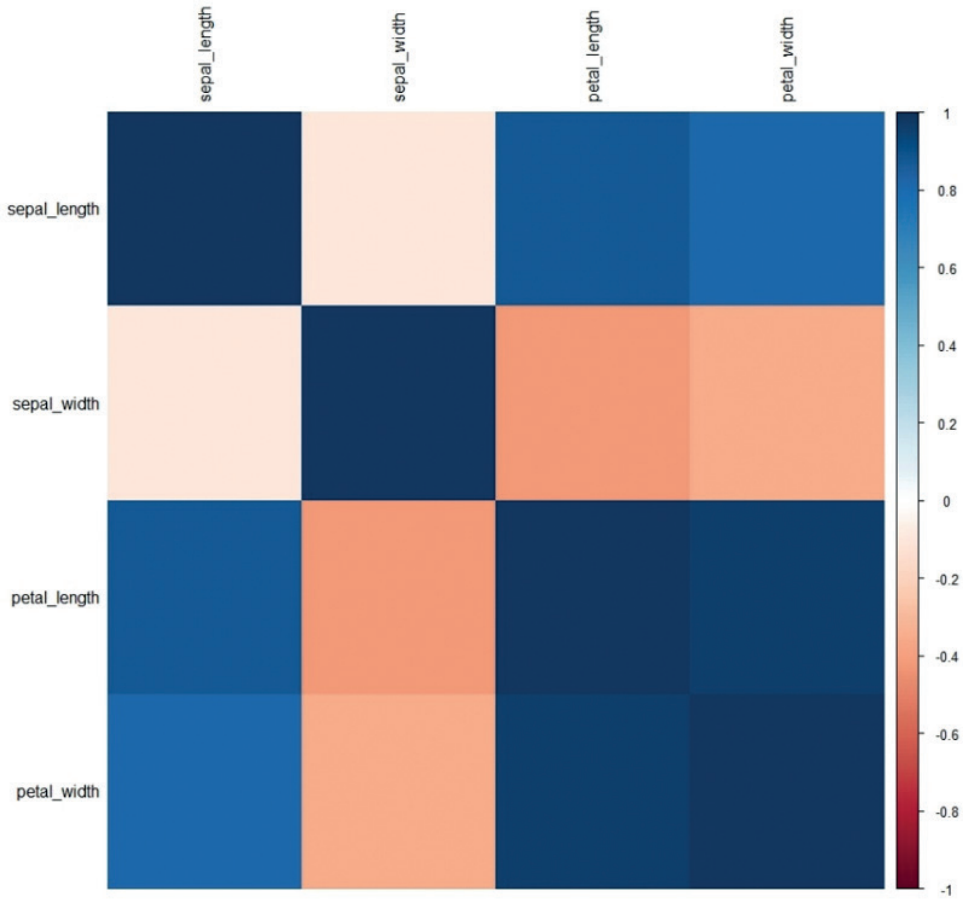


Figura 19. Harta termică pentru exemplul prezentat

Interpretarea rezultatelor coeficienților de corelație

Coeficientul de corelație arată cât de mult este posibilă estimarea valorilor atributului A_2 în eșantionul de date selectat pe baza atributului A_1. Cu cât valoarea acestui coeficient este mai apropiată de extremele intervalului considerat (adică de valoarea 1 sau -1), cu atât atributul dat A_1 este mai potrivit pentru a estima valorile atributului opțional A_2.

Evident, valoarea **0 nu spune nimic** relativ la corelația dintre două atribute. În acest caz, nu există nicio relație între valorile acestor atribute care să poată fi utilizată în construirea modelelor matematice pe setul de date utilizate.

Accepțiunea din literatura de specialitate diferă ușor în ceea ce privește nivelul de acceptabilitate a valorilor coeficienților de corelație - excepție fiind situația în care cu cât corelația este mai mare, cu atât este mai bine. În general, se spune că două atribute sunt puternic corelate când valoarea coeficientului de corelație măsurat între acestea atinge valori **mai mari de 0,8**. Se spune că între cele două atribute există o puternică anticorelație dacă coeficientul de corelație atinge o valoare **mai mică de -0,8**. Această limită de acceptabilitate a potențialului de predicție poate fi **relaxată când este mai aproape de valorile de 0,7 sau -0,7**, dar nu se recomandă să se depășească mai mult.

4.3 ANALIZA EXPLORATORIE A DATELOR ȘI VIZUALIZAREA DATELOR

Analiza exploratorie a datelor (denumită în mod obișnuit EDA din denumirea în engeză) este o metodă de analiză a datelor pe care le explorează pentru a găsi modele și tendințe într-o anumită populație sau eșantion. În forma sa de bază, acest tip de analiză se realizează prin explorarea vizuală a datelor. Înainte de vizualizarea în sine, este necesară parcurgerea mai multor pași care se dovedesc benefici în ceea ce privește căutarea ulterioară a cunoștințelor ascunse în date:

- **Familiarizarea cu setul de date** - este necesar să se răspundă la câteva întrebări despre setul de date dat înainte de a-l analiza:
  - **Cine a compilat setul de date, când și de ce?** Acest răspuns este esențial din punct de vedere al relevanței, actualității și utilității setului de date. Dacă setul de date ar fi compilat de experți în domeniu, ar fi mai relevant decât dacă ar fi compilat de un începător care a măsurat datele pe un senzor ieftin pentru consumator. Dacă setul de date a fost compilat cu 93 de ani în urmă, este posibil ca datele să nu fie la zi, măsurătorile ar putea fi mai puțin precise decât ar putea fi măsurate astăzi etc. Setul de date este construit cu un scop specific și, prin urmare, nu este universal (s-ar putea să nu fie potrivit pentru toate sarcinile).
  - **Cât de mare este setul de date?** Prin dimensiunea setului de date, ne referim la numărul de entități și atribute măsurate în setul de date. În cazul în care avem un set de date prea mare pentru a putea lucra confortabil (a se vedea secțiunea 1), trebuie selectat un eșantion din acesta pe baza principiilor menționate în secțiunea anterioară a acestui manual. Problema opusă - un set de date prea mic - este mult mai dificilă. Cu toate acestea, unii algoritmi pot funcționa cu seturi de date mici și abordări care generează noi entități bazate pe cele existente, numite supraeșantionare.
  - **Care este compoziția setului de date?** Acest punct este strâns legat de motivul compilării setului de date. Este important să fie parcurse toate atributele setului de date și să fie înțeles scopul acestora. De asemenea, este necesar să se analizeze dacă datele înregistrate în atributul dat sunt numerice sau categorice și în ce interval se deplasează valorile atributelor individuale.
- **Calculul statisticilor rezumative** - pentru fiecare atribut, se recomandă întocmirea statisticilor rezumative de bază. Valorile recomandate sunt extreme (min, max), mediană sau medie, abaterea standard și altele. Acesta este un pas foarte important și informativ în care obținem valorile medii ale atributelor date, cele mai mici și mai mari valori ale acestora și putem descrie în continuare atributele individuale prin agregare.
- **Efectuarea analizei corelațiilor** - pentru orice set de date, este recomandabil să compilați matricea

sau hărțile termice ale coeficienților de corelație. Această matrice măsoară corelațiile dintre valorile tuturor atributelor setului de date și astfel ne arată cât de dificil ne va fi să construim modele matematice pe setul de date dat. Analiza corelației ajută, de asemenea, la identificarea atributelor și a subseturilor de date potrivite pentru vizualizare.

Următorul pas al analizei exploratorii a datelor este reprezentat de vizualizarea efectivă a setului de date. Cu toate acestea, se vorbește despre vizualizarea eficientă a datelor, bazată pe câteva principii care vor fi prezentate în următoarea parte a manualului.

Vizualizarea eficientă a datelor

Spunem că vizualizarea datelor este eficientă când:

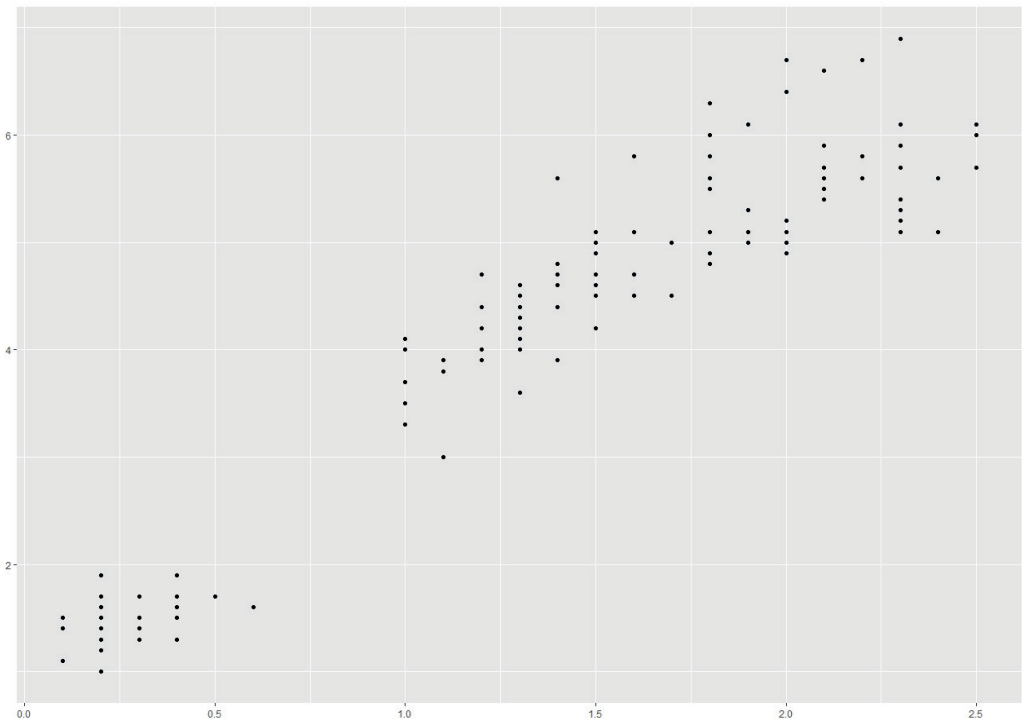
- putem să vizualizăm datele,
- le vizualizăm în mod corect,
- folosind tipul de grafic corect.

Aceste trei puncte sunt absolut naturale. Datele despre care se spune că sunt potrivite pentru vizualizare din punct de vedere al analizei datelor sunt cele care poartă unele informații, cel mai adesea cu potențial de predicție. După cum a fost prezentat în secțiunea anterioară a manualului, se poate vedea cu ușurință potențialul de predicție în seturile de date folosind o serie de metode de analiză a corelației. Prin urmare, sunt considerate potrivite pentru vizualizare acele părți ale setului de date în care sunt identificate corelații puternice sau anticorelații între valorile atributelor (a se vedea secțiunea Interpretarea rezultatelor coeficienților de corelație).

În cazul nostru, prin mod **corect de vizualizare a datelor**, se înțelege eliminarea a două probleme relativ comune prezente în vizualizarea seturilor de date:

- **Maximizarea raportului dintre culoarea utilizată și date** - deoarece se dorește vizualizarea datelor, în mod ideal, graficul ar trebui să conțină un minim de alte elemente grafice (de exemplu, o culoare de fundal, o grilă distinctivă și așa mai departe). Maximizarea raportului de culoare și date este esențială, mai ales când se vizualizează seturi mari de puncte care pot fuziona sau pot fi redate prin puncte foarte mici (sau alt tip de obiecte). Figura de mai jos conține un grafic de puncte standard realizat în limbajul R (în stânga) și o modificare a acestui grafic pentru a face datele mai vizibile (în dreapta).

a)



b)

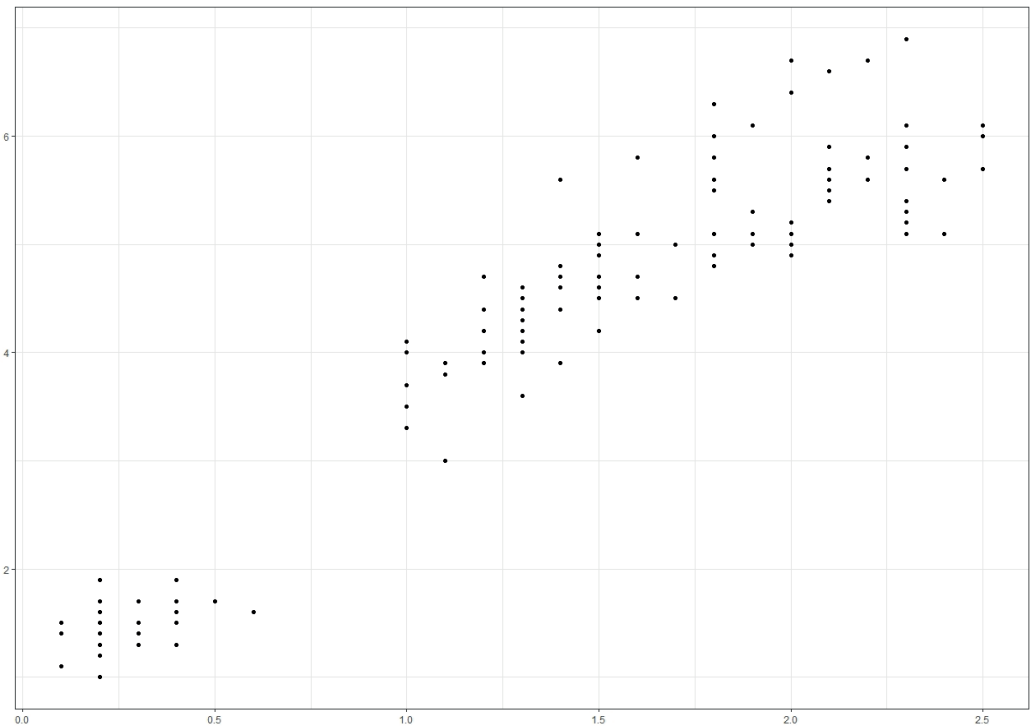


Figura 20. Grafic cu puncte realizat în limbajul R

- **Eliminarea distorsiunilor** - în prezentarea vizuală a datelor și interpretarea lor ulterioară, apar adesea distorsiuni din cauza setării incorecte a axelor sau din cauza distorsiunii fișierului imagine în sine. Graficele de mai jos ilustrează ambele probleme. Primul grafic cu bare compară temperatura a două persoane (A și B). A se reține că valorile din graficul din stânga par a fi semnificativ diferite, în timp ce valorile din dreapta sunt foarte asemănătoare, chiar dacă acestea sunt două măsurători identice prezentate cu setări diferite ale axei. În figura din stânga, este prezentată axa y de la valoarea de 36,5 la 37,3 grade. În graficul din dreapta, sunt prezentate valorile lui y de la 0 la 40 de grade. Acest exemplu este un factor tipic de confuzie în vizualizarea datelor.

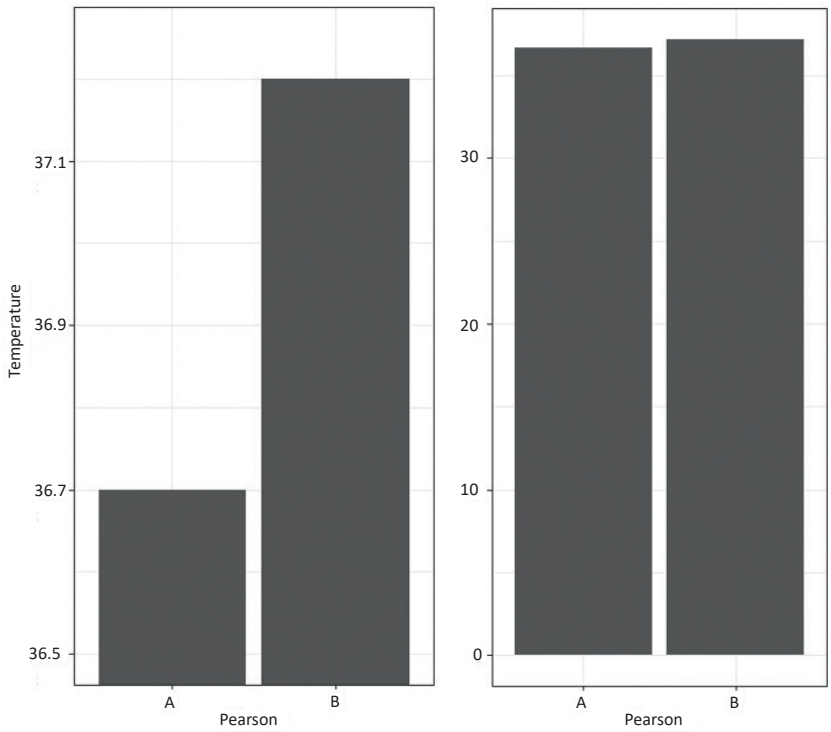


Figura 21. Distorsiune cauzată de setarea incorectă a axelor

A doua problemă de distorsiune este reprezentată de distorsiunea imaginii în sine. În imaginea următoare, poate fi văzut un grafic care este prezentat în trei rapoarte de aspect. Este clar că imaginea din partea de jos și cea din dreapta sunt nepotrivite din cauza distorsiunii formei reale a tendinței prezentate de graficul dat și, prin urmare, raportul ideal pentru aceste date va fi graficul de sus, din stânga - aproape de standardul de imagine al majorității proiectoarelor moderne - 16:9.

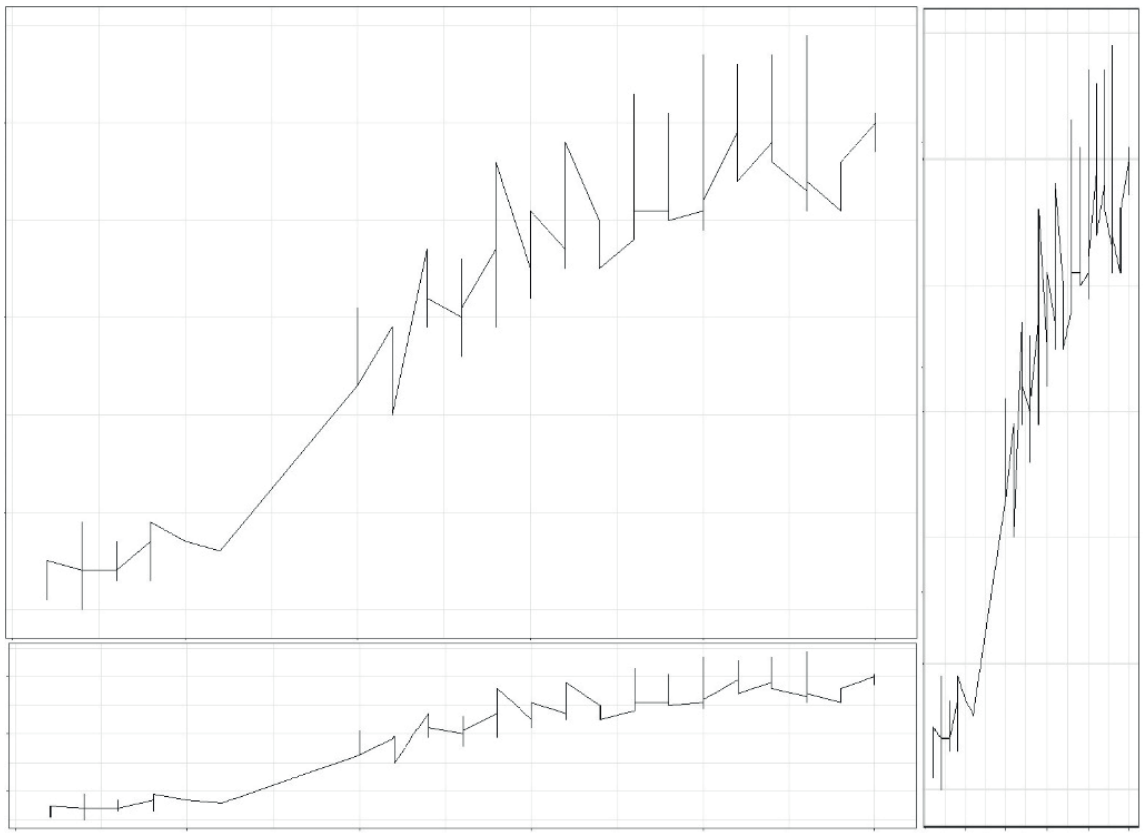


Figura 22. Distorsiunea imaginii în sine

Ultimul element foarte important al eficienței vizualizării este alegerea tipului corect de grafic. În general, sunt populare patru tipuri de grafice - grafice punct, linie, plăcintă și de bare. Fiecare tip de grafic este potrivit pentru scopuri diferite și are avantaje și dezavantaje. Această secțiune a manualului se va concentra doar pe cele mai comune două metode de vizualizare a datelor - grafice cu puncte și grafice cu linii.

Grafice de puncte

Graficele de puncte sunt folosite pentru a vizualiza relația dintre două (sau mai multe) atribute ale unui set de date folosind puncte. Modul standard de vizualizare este cel de comparare a valorilor a două atribute într-un plan:

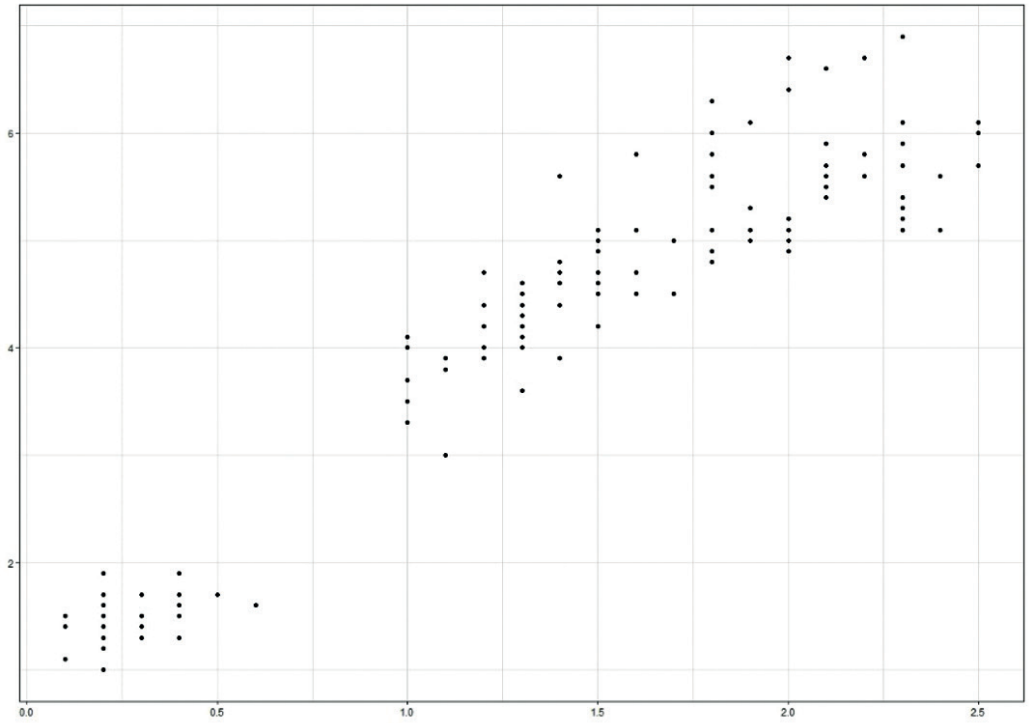


Figura 23. Graficele de puncte

Cu o astfel de abordare a vizualizării datelor, trebuie acordată o atenție specială dimensiunii punctului din grafic. Imaginea următoare arată suprasaturarea graficului cauzată de dimensiunea mare a punctului; o problemă similară apare când există un număr mare de puncte situate aproape unele de altele.

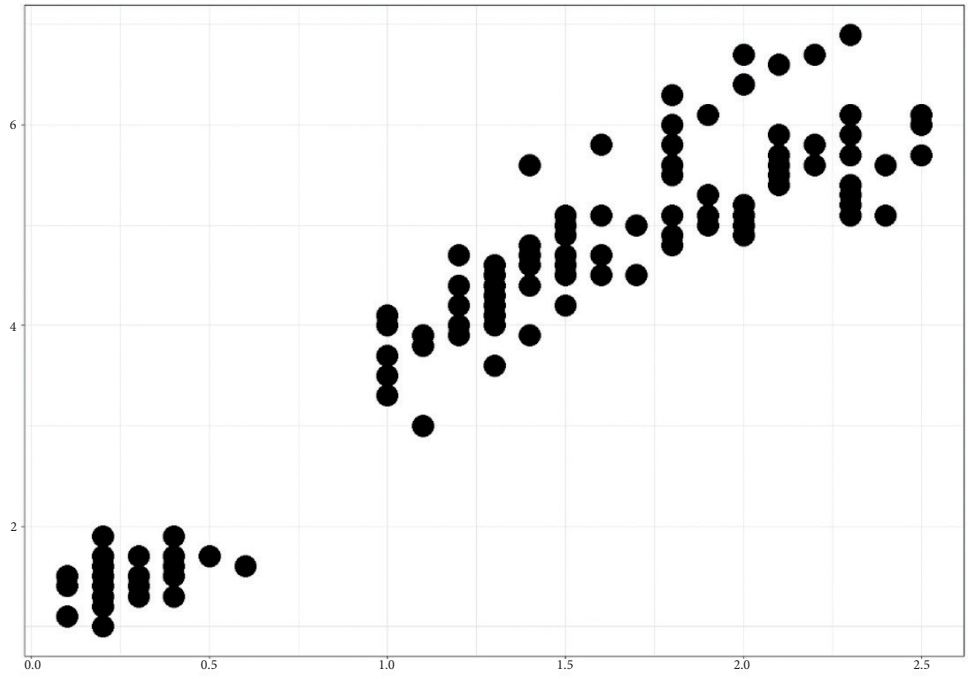


Figura 24. Suprasaturarea graficului cauzată de dimensiunea mare a punctului

Cu toate acestea, dimensiunea punctului poate fi utilizată în contextul vizualizării datelor pentru a transmite informații suplimentare. Dacă setăm dimensiunea punctului direct proporțională cu dimensiunea celui de-al treilea atribut din setul de date, putem vizualiza relația dintre cele trei atribute ale setului de date.

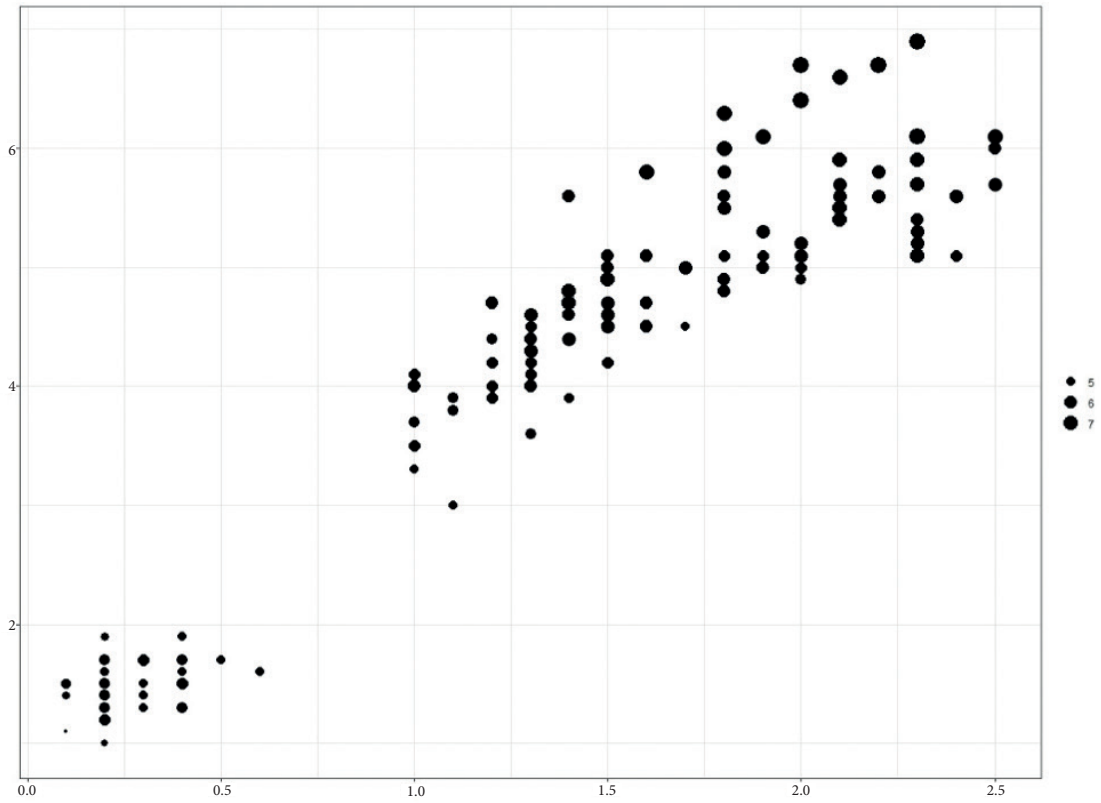


Figura 25. Setarea dimensiunii punctului direct proporțională cu dimensiunea celui de-al treilea atribut din setul de date

Putem extinde acest concept la alte proprietăți ale punctelor, de exemplu, culoarea lor (enumerate mai jos). Astfel, putem vizualiza relația dintre cele patru atribute ale setului de date. Cu toate acestea, această metodă de vizualizare este dificil de utilizat pentru un număr mare de puncte sau de atribute. În general, nu este recomandată vizualizarea a mai mult de trei sau patru atribute într-un plan.

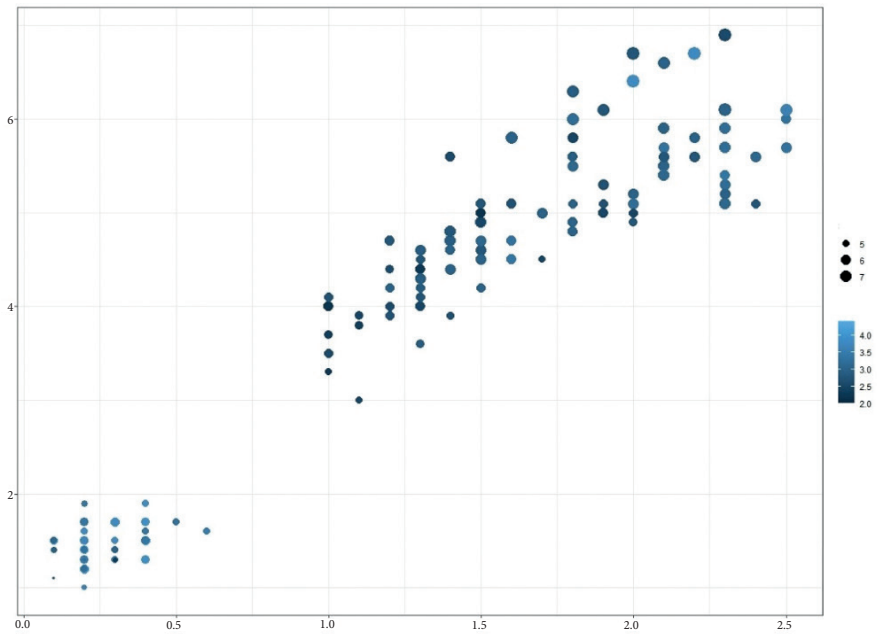


Figura 26. Setarea dimensiunii punctului și a culorii

Grafice cu linii

Graficele cu linii sunt folosite pentru a vizualiza evoluția valorii unui atribut în timp sau pentru a vizualiza fluctuația valorii unui atribut în funcție de alt atribut. A se reține că liniile prezente în graficele cu linii sunt o aproximare a punctelor - transformarea punctelor de date discrete în linii continue și, prin urmare, ar putea fi inexacte în unele locuri. În comparație cu graficele de puncte, graficele cu linii sunt greu de utilizat pentru date categorice (lingvistice).

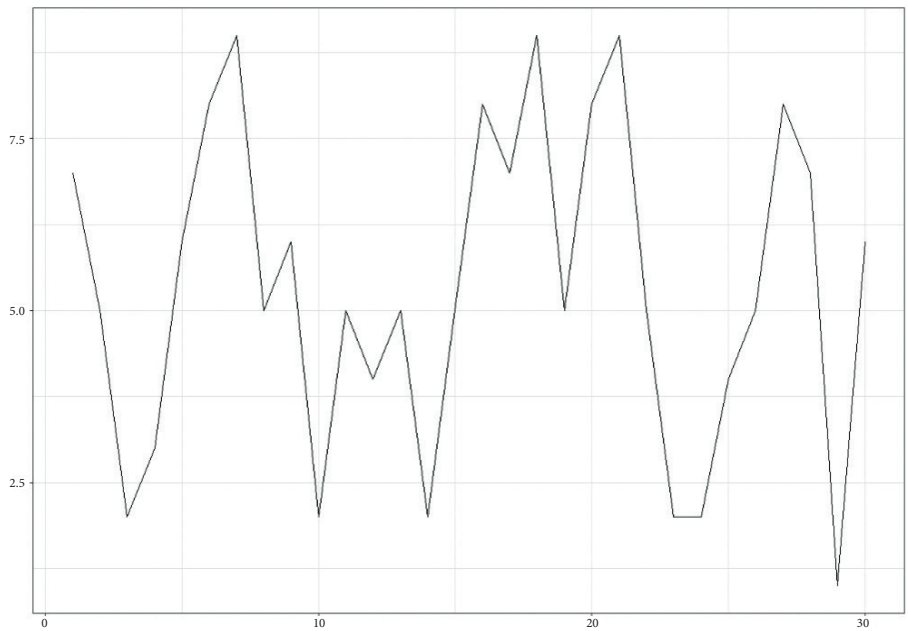


Figura 27. Graficele cu linii

Deoarece liniile sunt aproximări ale punctelor de date, se recomandă vizualizarea liniei și a punctelor pe care se bazează linia. Acest lucru scade ambiguitatea corectitudinii valorilor atributelor.

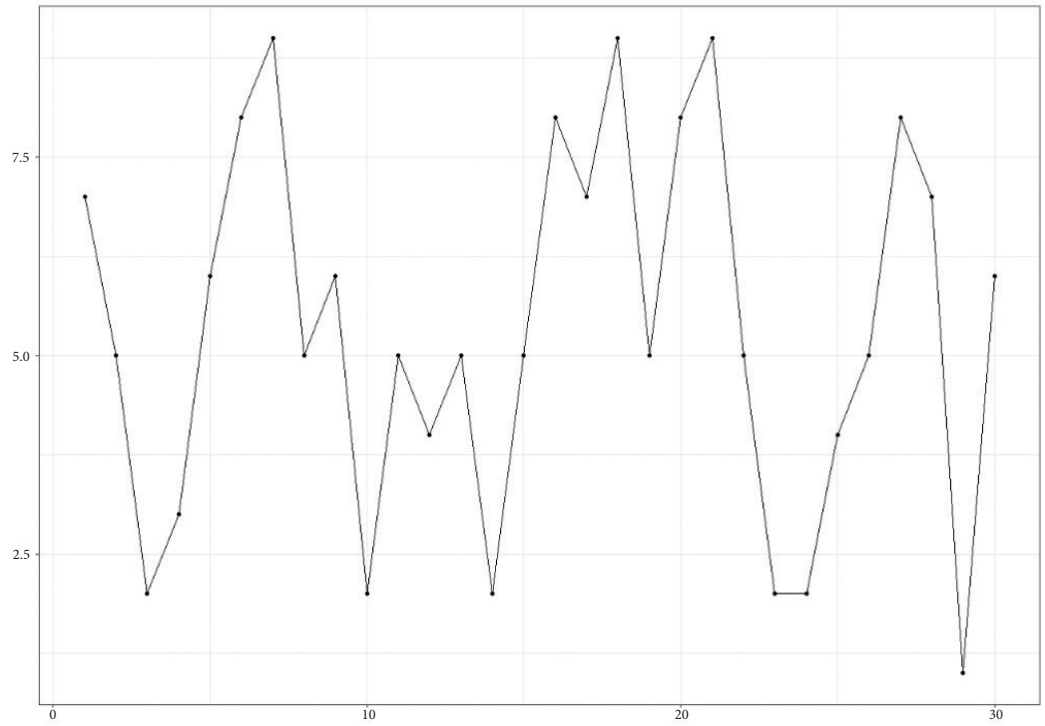


Figura 28. Graficele cu linii și puncte

4.4 ANALIZA EXPLORATORIE A DATELOR ÎN PRACTICĂ

Această parte a manualului se concentrează pe aplicarea în practică a metodelor exploratorii de analiză a datelor în limbajul R. Exemplele de mai jos au fost realizate în versiunea *R 4.3.0*, dar aceste concepte și comenzi sunt prezente în versiunile tuturor instrumentelor de programare adecvate pentru analiza datelor.

În primul rând, trebuie schimbat directorul de lucru pentru sesiunea care va urma în directorul în care sunt stocate datele structurate pe care le avem la dispoziție - acest lucru se poate face cu utilizarea barei de sus a programului, unde: *File → Change dir → set the working directory*. Pentru prezentarea metodei de analiză exploratorie a datelor, va fi folosit setul de date Iris, descris în *Anexa A*.

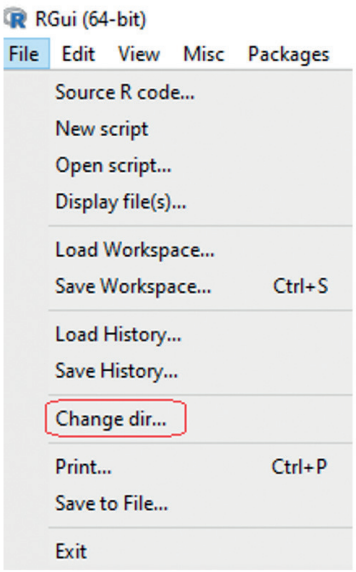


Figura 29. Meniu în limbajul R

După schimbarea directorului de lucru, se poate începe încărcarea setului de date în R. Această operație poate fi efectuată în mai multe moduri. În cele ce urmează, sunt prezentate cele mai simple comenzi *read.table* și *read.csv*, care funcționează similar pentru diferite tipuri de fișiere de intrare. Comanda în forma read.csv este utilizată pentru fișierele *.csv*, cea mai comună intrare structurată pentru instrumentele de analiză a datelor. Formularul *read.table* este utilizabil ca intrareîn formatul *.txt* sau *.data*.

```
read.table("title", header=T/F, sep="symbol")
read.csv("title", header=T/F, sep="symbol")
```

unde *title* este numele fișierului în care sunt stocate datele disponibile împreună cu extensia fișierului, partea *header* a comenzii indicând dacă fișierul de date de intrare are un antet (*T* pentru adevărat) sau nu (*F* pentru fals), iar *sep* reprezintă o parte a comenzii care așteaptă caracterul prin care valorile atributelor din fișier sunt separate.

Pentru a putea folosi mai departe setul de date încărcat în program, acesta trebuie salvat sub un titlu selectat, de exemplu, *title_of _data*:

```
title_of_data <- read.table("title", header=T/F, sep="symbol")
```

Un exemplu de încărcare a unui fișier de date stocat sub titlu *our_data.data* arată după cum urmează. În a doua comandă dată, se salvează setul de date sub titlul *data*:

```
read.table("our_data.data", header=T, sep=",")
data <- read.table("our_data.data", header=T, sep=",")
```

Analiza exploratorie a datelor - Pasul 1 - Familiarizarea cu un anumit set de date

Ca parte a cunoașterii setului de date, pot fi efectuate câteva operații foarte simple. Prima dintre ele este listarea întregului set de date în consola instrumentului R folosind *title_of _data*. Cu toate acestea, acest lucru nu este deloc practic pentru seturile mari de date, care conțin mii de entități. Prin urmare, este prezentată în cele ce urmează o a doua versiune a comenzii de listare a entităților setului de date – *head*. Această versiune listează pe consolă numărul de entități definit la începutul setului de date.

```
title_of_data

head(title_of_data, number_of_entities)
```

O exemplificare a conceptului menționat ar putea fi o listare a întregului set de date stocat sub numele de date sau o listare a primelor cinci entități ale acestui fișier.

```
data

head(data, 5)
```

Ieșirea acestei comenzi în limbajul R va fi un pseudo-tabel în formatul următor:

```
> data <- read.table("iris.data", header = T, sep = ",")
> head(data, 5)
  sepal_length sepal_width petal_length petal_width       class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
```

După familiarizarea cu setul de date, atributele acestuia și valorile din acesta, se trece la calculul măsurilor de centralitate și variabilitate. Toate aceste funcții sunt derivate din versiunea în limba engleză a numelui funcțiilor individuale (de exemplu, *sd* pentru abaterea standard), iar intrarea lor este doar unul dintre atributele setului de date scris sub forma

title_of _data$title_of_attribute.

Cea mai versatilă dintre aceste comenzi este funcția *summary*, care măsoară *minimul, prima cuartilă, mediana, media, a treia cuartilă* și *maximul* pentru toate atributele din setul de date.

```
mean(title_of_data$attribute_title)

median(title_of_data$attribute_title)

min/max/sum(title_of_data$attribute_title)

sd(title_of_data$attribute_title)

summary(title_of_data)
```

Un exemplu de analiză a proprietăților statistice prezente în date este utilizarea următoarelor comenzi:

```
summary(data)

sd(data$attribute_title)
```

Rezultatul acestor funcții executate pe setul de date Iris constă în următorul set de valori:

```
> summary(data)
  sepal_length    sepal_width     petal_length    petal_width          class
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> sd(data$sepal_length)
[1] 0.8280661
```

Analiza exploratorie a datelor - Pasul 2 – Analiza corelației

După cum s-a menționat în secțiunile anterioare ale acestui manual, analiza corelației este una dintre părțile esențiale ale analizei exploratorii a datelor. Forma de bază a comenzii pentru analiza corelației este funcția *cor* care face calculul coeficientului de corelație între două atribute ale setului de date. În sintaxa comenzii prezentată mai jos, se poate observa că tipul de coeficient de corelație care se dorește a fi calculat pentru date poate fi definit folosind metoda parametrilor = *type_of_corelation*. Coeficientul de corelație Pearson este implicit utilizat în această comandă.

```
cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2)

cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2,
method = “pearson”)

cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2,
method = “spearman”)
```

Cu toate acestea, ca parte a analizei setului de date, se dorește examinarea tuturor relațiilor dintre toate atributele setului de date și, prin urmare, va putea fi creată o matrice de corelație:

```
cor(title_of_data)

cor(title_of_data, method = “spearman”)
```

Analiza corelației setului de date Iris se realizează cu ajutorul următoarelor comenzi simple:

```
cor(data[, 1:4])

cor(data[, 1:4], method = “spearman”)
```

Observație: Deoarece setul de date Iris conține un atribut ale cărui valori sunt lingvistice (clasa de atribute) și matricea de corelație cuprinde numai valori numerice, funcția cor trebuie să aibă ca date de intrare doar primele patru atribute (numerice). Obținem acest lucru selectând coloanele 1:4 dintr-un set de date numit data: data[, 1:4].

```
> cor(data[,1:4])
             sepal_length sepal_width petal_length petal_width
sepal_length    1.0000000  -0.1093692    0.8717542   0.8179536
sepal_width    -0.1093692   1.0000000   -0.4205161  -0.3565441
petal_length    0.8717542  -0.4205161    1.0000000   0.9627571
petal_width     0.8179536  -0.3565441    0.9627571   1.0000000
> cor(data[,1:4], method = "spearman")
             sepal_length sepal_width petal_length petal_width
sepal_length    1.0000000  -0.1594565    0.8813864   0.8344207
sepal_width    -0.1594565   1.0000000   -0.3034206  -0.2775111
petal_length    0.8813864  -0.3034206    1.0000000   0.9360034
petal_width     0.8344207  -0.2775111    0.9360034   1.0000000
```

Așa cum s-a menționat în secțiunea 4.2, pentru seturile mari de date se recomandă utilizarea unei hărți de corelare. Trebuie instalat un pachet de funcții în limbajul R pentru a folosi această metodă de vizualizare. Metoda menționată conține o funcție de vizualizare a hărții de corelare, numită *corrplo*t. După instalarea acestui pachet, se încarcă folosind funcția *require* și apoi se creează o hartă termică de corelare:

```
install.packages(“corrplot”)

require(corrplot)

corrplot(cor(data), method = “color”)
```

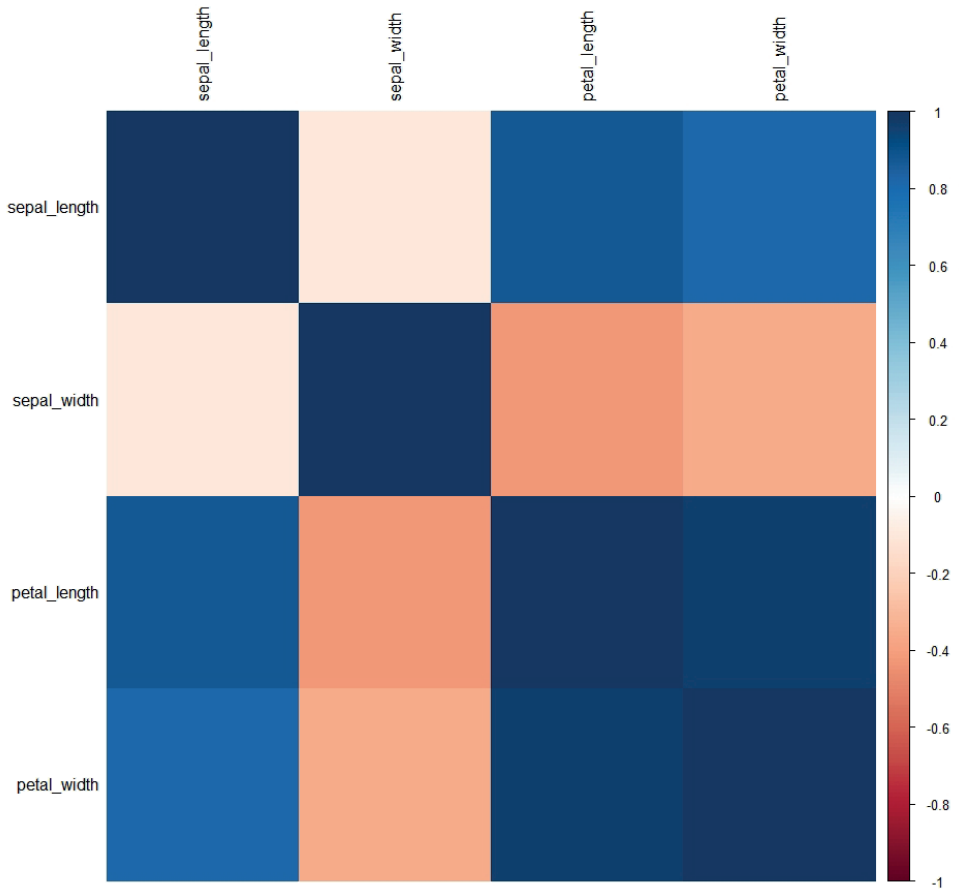


Figura 30. Harta termică de corelație pentru exemplul utilizat

Matricea de corelație și harta termică pentru setul de date Iris indică relațiile care pot fi utilizate în analiza ulterioară a datelor. Mai exact, arată toate relațiile dintre atribute în care orice tip de coeficient de corelație a fost mai mare de 0,8 sau mai mic de -0,8:

- ρ(sepal_length, petal_length) ≈ 0.87
- ρ(sepal_length, petal_width) ≈ 0.82
- ρ(petal_length, petal_length) ≈ 0.94

Aceste relații între atribute merită vizualizate.

Analiza exploratorie a datelor - Pasul 3 - Vizualizarea datelor

După analiza corelațiilor, urmează vizualizarea perechilor de atribute în care s-a observat o puternică corelație sau anticorelație. Cu toate acestea, înainte de vizualizarea în sine, este necesară instalarea unui pachet de vizualizare:

```
install.packages("ggplot2")
require(ggplot2)
```

Pachetul *ggplot2* este unul dintre cele mai populare pachete utilizate în vizualizarea datelor. Acesta conține funcții pentru trasarea graficelor cu ajutorul punctelor, graficelor cu linii, graficelor cu bare și multe altele. În această secțiune a manualului, vor fi alese câteva exemple simple de astfel de funcții.

Graficele de puncte

Cel mai puternic mod de vizualizare a datelor este reprezentat de graficul de puncte. În limbajul R și pachetul *ggplot2*, se folosește funcția ggplot pentru a construi orice tip de grafic, situație în care funcția așteaptă mai multe valori de intrare. Pentru cele mai simple grafice, aceste intrări sunt:

- Titlul setului de date cu care lucrăm (în cazul nostru, acest titlu este data).
- Secțiunea *aes* derivă din cuvântul englezesc estetică și se așteaptă informații despre cel puțin o axă. Aceste informații sunt prezentate sub forma *axis_title (x sau y) = title_of _attribute* reprezentat de axă.
- Tipul de grafic.

In general, sintaxa comenzilor pentru graficele de puncte conține alocarea a două atribute la axele graficului și secțiunea comenzii + *geom_point*(). Sintaxa generalizată pentru acest tip de comandă este prezentată mai jos:

```
ggplot(title_of_data, aes(x = title_of_attribute_1, y = title_of_
attribute_2)) + geom_point()
```

Se poate schimba culoarea punctelor folosind extensia secțiunii de comandă + *geom_point()*, adăugând o a doua secțiune *aes* valabilă numai pentru punctele în sine - și anume + *geom_point(aes(culoare = "nume culoare"))*. Ca alternativă la specificarea unei culori, se poate modifica culoarea punctelor dintr-un grafic prin specificarea numelui atributului din setul de date la care lucrăm în secțiunea comenzii + *geom_point()* în loc de numele culorii. În acest fel, se realizează vizualizarea în două dimensiuni (atribute) cu o dimensiune suplimentară marcată de culoarea punctelor, care se modifică în funcție de valorile atributului ales. Pentru a adera la principiile de vizualizare eficientă a datelor prezentate în subsecțiunea anterioară, se adăugă opțiunea + *theme_bw()* la sfârșitul comenzii, care asigură un fundal alb sub graficul în sine, maximizând astfel raportul dintre date și culoarea utilizată în grafic.

```
ggplot(title_of_data, aes(x = title_of_attribute_1,

y = title_of_attribute_2))) + geom_point(aes(color = "color"))

+ theme_bw()

ggplot(title_of_data, aes(x = title_of_attribute_1, y = title_of_
attribute_2))) + geom_point(aes(color = title_of_attribute_3)) + theme_bw()
```

O exemplificare simplă pentru această abordare este prezentată mai jos. De asemenea, sunt introduse două concepte suplimentare:

- • Graficul creat poate fi salvat sub un nume selectat, de exemplu, graph1, așa cum este arătat mai jos.
- • Pot fi adăugate alte părți comenzii la graficul salvat în acest fel. În exemplul de mai jos, folosim + *xlab()* și + *ylab()* pentru a adăuga etichetele axelor *x* și *y*. Graficul este apoi vizualizat prin apelarea numelui său în consolă.

```
graph1 <- ggplot(data, aes(x= atr_1, y = atr_2))
+ geom_point(aes(color = class) + theme_bw()
graph1 <- graph1 + xlab("X parameter") + ylab("Y parameter")
graph1
```

Astfel, următorul cod pentru setul de date Iris

```
> require(ggplot2)
> graph1 <- ggplot(data, aes(x = petal_length, y = petal_width)) + geom_point(aes(color = class)) + theme_bw()
> graph1 <- graph1 + xlab("Petal length measurements") + ylab("Petal width measurements")
> graph1
```

produce următoarea figură:

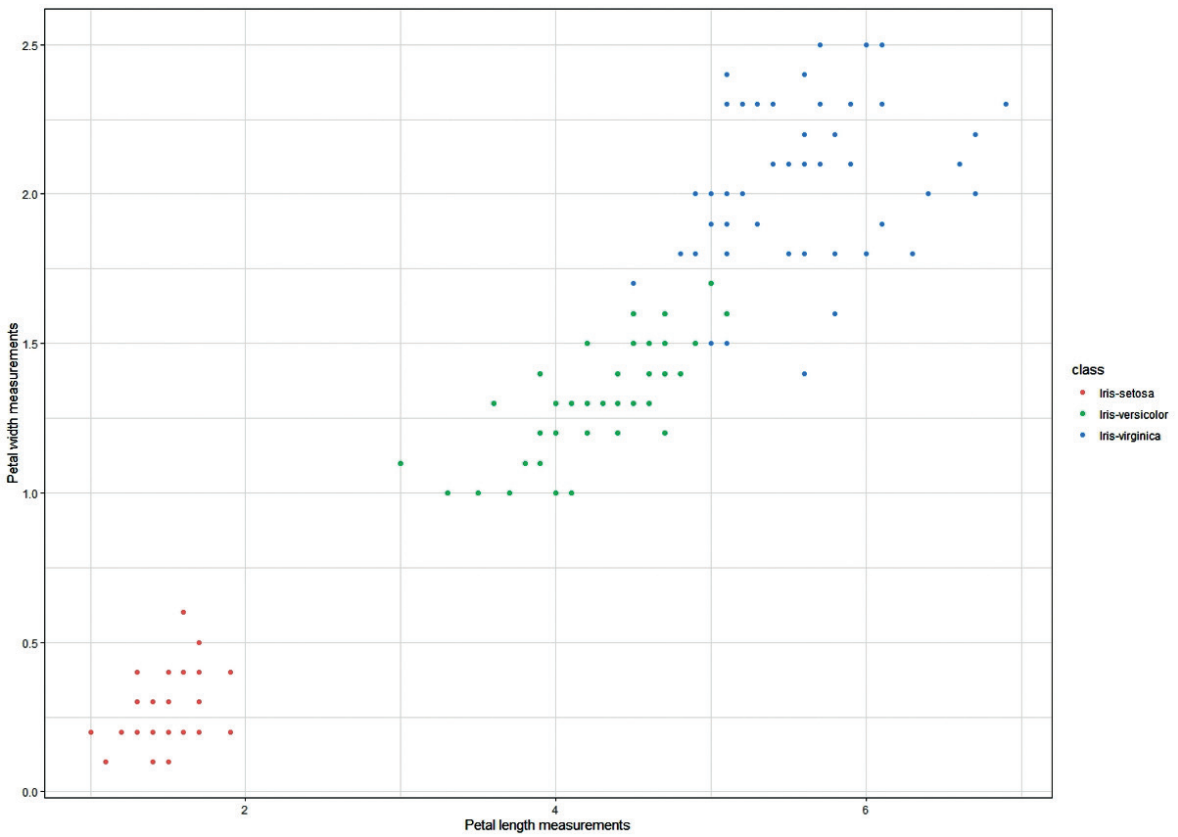


Figura 31. Grafice de puncte și culori pentru setul de date Iris

Graficele de linii

Acest tip de grafic este folosit pentru a vizualiza evoluția valorii unui atribut în timp sau pentru a vizualiza fluctuația valorii unui atribut în funcție de alt atribut. Sintaxa pentru comanda *line graph* din pachetul *ggplot2* nu este semnificativ diferită de exemplele anterioare prezentate în această secțiune a manualului. Singura diferență este dată de tipul de geometrie folosit la desenarea graficului - în cazul graficelor cu linii, acesta este + *geom_line*(). Folosind opțiunea *linetype* din secțiunea comenzii *geom_line()*, se poate schimba și tipul de linie folosită la trasarea graficului.

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) +
geom_line()
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “dashed”)
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “twodashed”)
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “dotted”)
```

Similar graficelor cu puncte, se poate schimba cu ușurință culoarea geometriei - în acest caz, linia - folosind opțiunea de culoare, care poate fi combinată cu toate tipurile de linie din grafic.

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line(color = “color”)

ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line(linetype = “type”, color = “color”)
```

Deoarece un grafic cu linii este întotdeauna o aproximare a punctelor de date, este adecvată vizualizarea punctelor de-a lungul graficului de linii. Acest lucru se poate face printro combinație de geometrii de linii și puncte, după cum urmează:

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line() + geom_point()
```

Se înțelege, este posibilă o combinație a tuturor acestor opțiuni:

```
ggplot(data,   aes(x=attribute_title_1,   y=attribute_title_2))   +   geom_
line(linetype = “dashed”, color = “blue”) + geom_point()
```

Astfel, următorul cod pentru setul de date Iris:

```
> ggplot(data, aes(x = petal_length, y = petal_width)) + geom_line(linetype = "dashed", color = "blue") +
+ theme_bw() + geom_point()
```

generează următoarea figură:

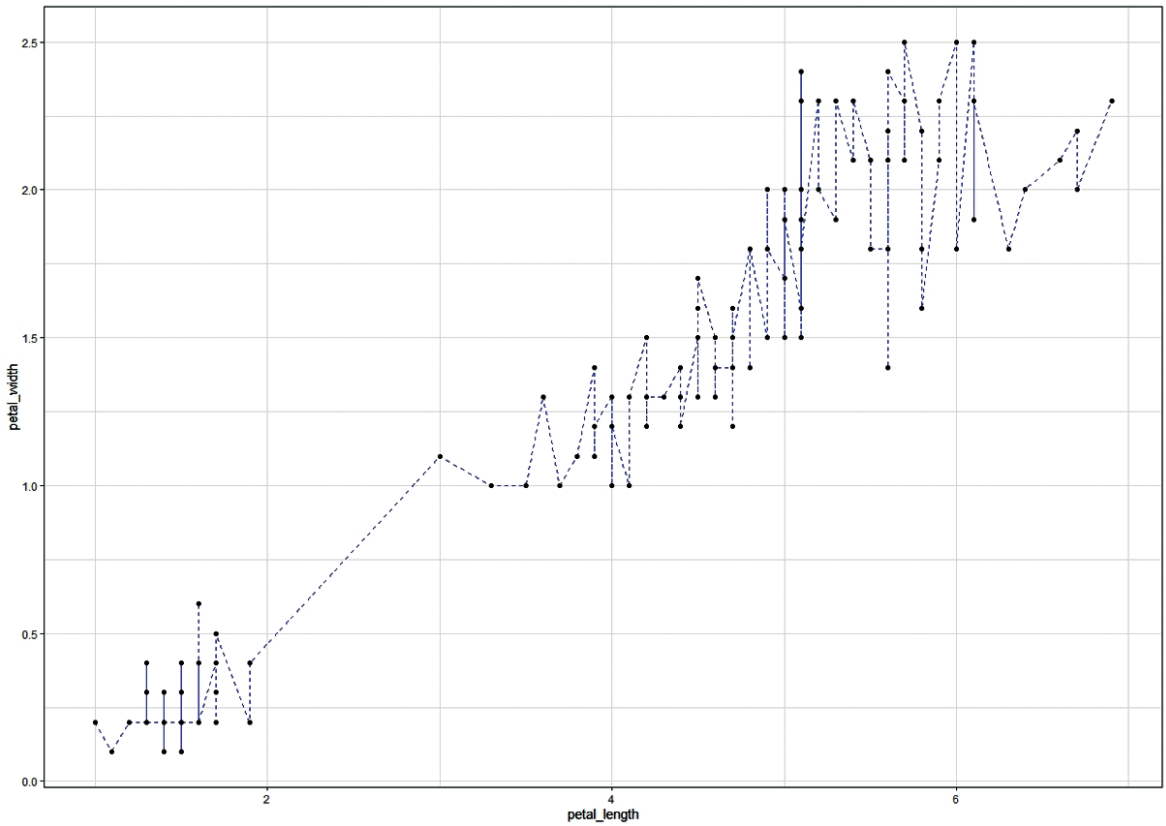


Figura 32. Grafice de linii

Adevărata putere a graficelor de linii constă în capacitatea lor de a vizualiza comparația dintre valorile unui atribut și un set de atribute - cu alte cuvinte, posibilitatea de a desena mai mult de o linie. În exemplul de sintaxă de mai jos, se poate vedea că funcția de bază *ggplot* conține un singur atribut (cel plasat pe axa *x*) și se folosește axa *y* pentru vizualizarea valorilor atributului 2 și atributului 3. Această vizualizare se face prin secțiuni separate + *geom_line()* de cod. Sunt prezentate și câteva combinații ale opțiunilor de mai sus utilizând această abordare.

```
ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2))
+ geom_line(aes(y= attribute_title_3))

ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2), color = “color”)
+ geom_line(aes(y= attribute_title_3), color = “color”)

ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2), linetype = “type”, color = “color”)
+ geom_line(aes(y= attribute_title_3), linetype = “type”, color = “color”)
```

Se consideră setul de date Iris în care au fost găsite corelații puternice între trei atribute - lungimea sepalei, lungimea petalei și lățimea petalei. Fluctuația lungimii și lățimii petalelor în funcție de lungimea sepalei poate fi vizualizată cu ajutorul a două linii, care vor fi separate după tipul sau culoarea lor. În cazul nostru:

- fluctuația valorilor lungimii petalelor pe baza lungimii sepalei este vizualizată cu ajutorul liniei roșii,
- fluctuația valorilor lățimii petalelor pe baza lungimii sepalei este vizualizată cu ajutorul liniei albastre.

```
ggplot(data, aes(x=attribute_title_1)) + geom_line(aes
(y= attribute_title_2), linetype = “dotted”, color = “red”)
+ geom_line(aes(y= attribute_title_3), color = “blue”)
So, the following code for the Iris dataset
```

Deci, următorul cod pentru setul de date Iris

```
> ggplot(data, aes(x = sepal_length)) + geom_line(aes(y = petal_length), color = "red") +
+ geom_line(aes(y = petal_width), color = "blue") + theme_bw() + xlab( "Sepal length measurements") +
+ ylab("Petal length (red) and petal width (blue)")
```

produce următoarea figură:

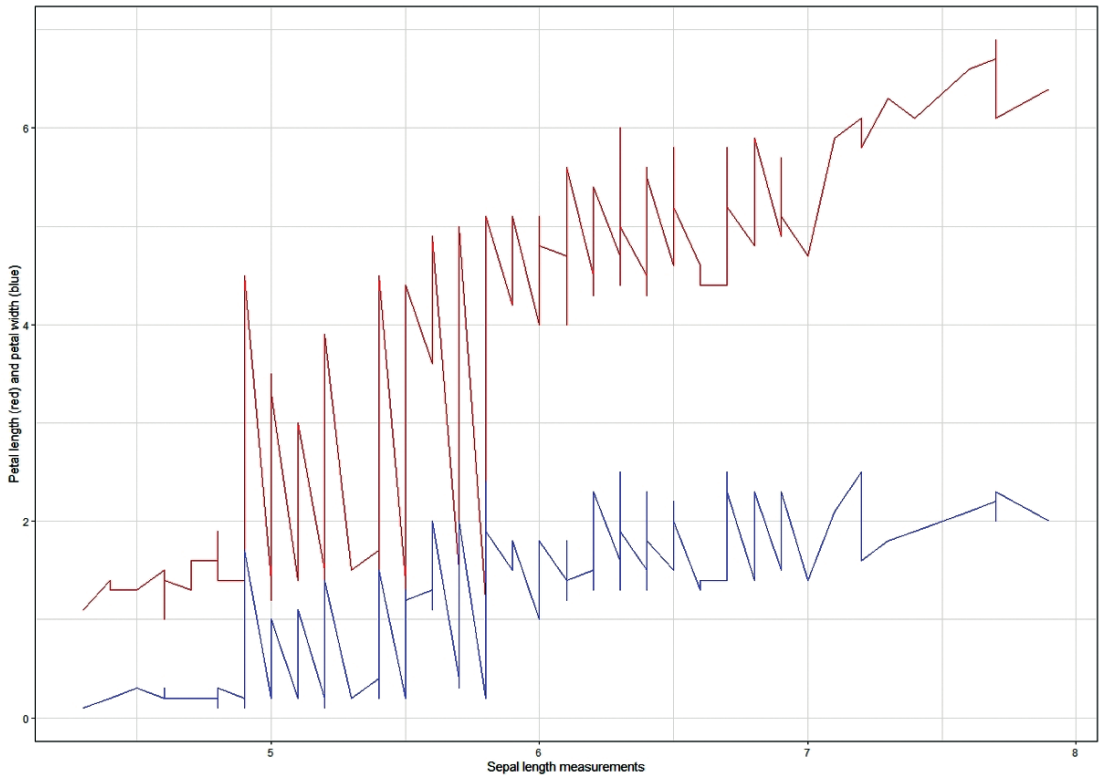


Figura 33. Grafice de linii, cu adăugare de culori

O astfel de vizualizare a datelor poate fi utilizată pentru a căuta tendințe și modele în date, care pot fi ulterior utilizate în abordări mai complexe ale analizei de date prezentată în alte secțiuni ale manualului. Analiza exploratorie a datelor are un dezavantaj - este greu de utilizat pentru seturi de date cu adevărat mari, care necesită reducerea dimensionalității și alte tehnici de analiză potrivite.

CAPITOLUL 5

SETURI DE DATE FUZZY

Această parte a manualului a fost scrisă de Alžbeta Michalíková, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

În viața de zi cu zi folosim deseori expresii vagi, cum ar fi: persoană tânără, puțină sare, ușor spre dreapta, vânt puternic, temperatură ridicată, preț scăzut... Aceste expresii nu au limite precise. Ele nu sunt clar definite. Le putem numi **vagi** sau **neclare**. Prima idee de modelare matematică prin concepte fuzzy ar putea fi găsită în lucrarea lui Lotfi A. Zadeh [1]:

ZADEH, L. A.: Fuzzy sets. Information and control. Volume 8, pp. 338-353, 1965.

Utilizarea mulțimilor fuzzy este denumită **operarea cu limbaj natural** (operează nu numai cu numere, ci și cu alți termeni ai vorbirii umane). În același timp, este de la sine înțeles că oameni diferiți percep concepte diferite, în mod diferit. Partea principală a următorului text se bazează pe manualul universitar [2] **Mulțimi fuzzy în informatică** (în slovacă), care este destinat studenților de la informatică aplicată la Departamentul de Informatică, Facultatea de Științe Aplicate, Universitatea Matej Bel din Banská Bystrica.

De ce se folosește logica fuzzy?

- ideea logicii fuzzy este ușor de înțeles,
- este un sistem flexibil care este tolerant la date inexacte,
- se poate lucra cu experiențele experților,
- se poate modela un sistem neliniar de orice complexitate,
- poate fi utilizat în echipamente tehnice standard.

Utilizări ale mulțimilor fuzzy

- sisteme expert,
- recunoașterea și clasificarea obiectelor,
- teoria controlului și reglementării,
- sisteme de baze de date,
- modelarea matematică;
- recent – rețele neuronale explicabile.

Domenii de aplicare a mulțimilor fuzzy

Mulțimile fuzzy pot fi utilizate ori de câte ori incertitudinea este inclusă în calcul. Acestea sunt adesea utilizate în dispozitive care nu reprezintă aparate scumpe din punct de vedere economic, de exemplu **aparate electronice de uz casnic** (mașini de spălat, cuptoare cu microunde, aspiratoare, aparate de ras, manometre etc.). Le putem găsi, de asemenea, în **dispozitive complicate și economice**, precum și în dispozitive de calcul **intensiv**, de exemplu:

- conducerea metroului în Japonia - (orașul Sendai – din 1988) [3],
- controlul unui furnal (controlul temperaturii se face mai eficient decât cu regulatoarele convenționale),
- gestionarea centralelor nucleare [4], ...

Exemplu: Avem un anunț de recrutare pentru un post care necesită ca vârsta candidaților să fie cuprinsă între 20-30 de ani. În cele ce urmează, descriem această mulțime de date.

Care este universul aplicației?

Se poate descrie această mulțime prin funcția sa caracteristică?

Poate o persoană care va împlini 31 de ani mâine să răspundă la această reclamă?

Ce este universul aplicației?

Acest set este de obicei notat cu litera X. Universul ar trebui să fie orice interval cu valorile realizabile. De exemplu, în cazul nostru ar putea fi .

Cum ar putea fi descris acest set prin funcția sa caracteristică?

Funcția caracteristică este o funcție care atribuie numărul **1** acelor elemente care aparțin setului considerat și, pe de altă parte, atribuie numărul **0** acelor elemente care nu aparțin setului considerat. Această funcție este notată de obicei cu litera χ. Pentru exemplul considerat, funcția are următoarea notație:

$$\chi_A : \mathbb{X} \to \{0, 1\} \qquad \chi_A(x) = \begin{cases} 1, & \text{if } 20 \le x \le 30, \\ 0, & \text{if } 0 \le x < 20 \text{ or } x > 30. \end{cases}$$

Poate o persoană care mâine va împlini 31 de ani să răspundă la acest anunț de recrutare?

NU! - pentru că în momentul în care cineva citește răspunsul, nu va mai îndeplini condiția cerută.

Exemplu: Se dă o situație similară: Într-o reclamă, există o cerință prin care compania caută tineri.

S-a schimbat situația față de exemplul anterior?

Care este universul de discurs al aplicației?

Cum poate fi descrisă această mulțime?

Poate o persoană care va împlini mâine 31 de ani să răspundă la această reclamă?

S-a schimbat situația față de exemplul anterior?

DA! – mulțimea de tineri reprezintă **setul fuzzy**. Nu există o limită clară pentru vârsta oamenilor care aparțin acestei mulțimi!

Care este universul?

Universul setului fuzzy ar trebui să fie orice interval cu valorile realizabile. Ar putea fi aceeași mulțimea ca și la cazul clasic, de exemplu .

Cum putem descrie acest set?

De exemplu, o persoană de 20 de ani este tânără cu siguranță, prin urmare i se atribuie un grad de tinerețe egal cu 1. În mod similar, o persoană de 30 de ani poate fi considerată cu siguranță tânără, prin urmare i se atribuie un grad de tinerețe egal cu 1, dar unui tânăr de 35 de ani i-am putea atribui un grad de tinerețe 0,8, ... Pentru a descrie seturile fuzzy, folosim așa-numitele **funcții de apartenență**. Acestea sunt notate cu μ. Prin utilizarea acestei funcții trebuie să atribuim o anumită valoare din intervalul unității (de exemplu, din interval la fiecare element al universului. În primul rând, să ne uităm la valorile vârstelor, pe care le-am considerat exact vârsta tinerilor. De exemplu, aceste valori ar putea fi din intervalul . Funcția de apartenență atribuie valoarea egală cu 1 acestor valori. Acum, să ne uităm la valorile vârstelor, pe care le-am considerat exact ca nefiind vârsta tinerilor. Un exemplu de astfel de valori ar putea fi valori mai mari de 55. Acestor valori funcția de apartenență atribuie valoarea egală cu 0. Pentru valorile din interval ne așteptăm ca valorile apartenenței la grupul de tineri să coboare secvențial de la 1 la 0 (a se vedea figura 34).

Notăm setul fuzzy de tineri cu B. Transcrierea unei funcții descrise este:

$$\mu_B: \mathbb{X} \to \langle 0,1\rangle \qquad \mu_B(x) = \begin{cases} 1, & \text{if } x \in \langle 0,30\rangle, \\ \frac{1}{25}(55-x), & \text{if } x \in (30,55\rangle, \\ 0, & \text{if } x > 55. \end{cases}$$

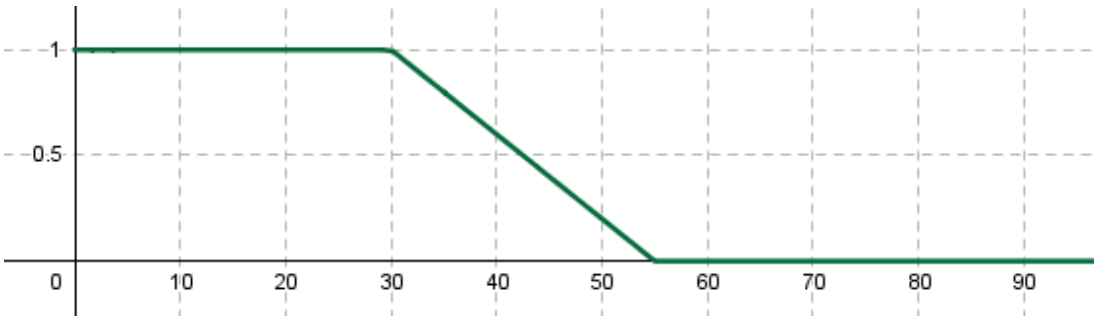


Figure 34. Funcția de apartenență a tinerilor din mulțimea fuzzy

Observații:

Termenii set fuzzy și funcție de apartenență sunt adesea considerați echivalenți.

Valoarea, care este atribuită valorii de intrare specifice, se numește **gradul de apartenență.**

Fie setul fuzzy determinat de formula (1) care descrie mulțimea de tineri. Putem atribui gradul de apartenență al persoanelor care au 20, 35 și 45 de ani? Folosind formula (1) obținem:

$\mu_B(20) = 1$, adică persoana de 20 de ani este tânără cu siguranță,

$\mu_B(35) = 0{,}8$, adică persoana de 35 de ani este tânără cu gradul 0,8,

$\mu_B(40) = 0{,}6$, adică persoana în vârstă de 40 de ani este tânără cu gradul 0,6.

Poate o persoană care va împlini 31 de ani mâine să răspundă la această reclamă?

DA! – deoarece gradul său de apartenență ce aparține mulțimii fuzzy μ_B este egal cu 0,96 (pentru că $\mu_B(31) = 0{,}96$). Această valoare reprezintă un grad ridicat de apartenență la mulțimea fuzzy de tineri.

Remarcă:

Există un număr de tipuri de funcții de apartenență ale mulțimilor fuzzy. Vor fi prezentate câteva dintre ele în exemplul următor.

Exemplu: Să modelăm mulțimea fuzzy C de numere reale care reprezintă termenul "aproximativ 7".
Care este universul de discurs al aplicației?
Cum putem descrie proprietățile acestei mulțimi?
Cu utilizarea acestui termen „aproximativ 7" ne putem imagina propoziții precum

Afară sunt aproximativ 7 grade Celsius.

sau

Am cheltuit aproximativ 7 € în magazin.

Ce este universul?

Ca univers, considerăm de obicei cea mai mare mulțime posibilă. În acest exemplu, ar putea fi întreg domeniul de numere reale, de exemplu .

Cum putem descrie proprietățile acestui set?

Notăm această mulțime cu C. Pentru funcția de apartenență al acestei mulțimi, μ_C ar putea avea două condiții:

$\mu_C(7) = 1$,

cu o diferență crescătoare $|x\text{-}7|$ valorile sale ar trebui să scadă la zero.

Unele dintre posibilități sunt prezentate în figura 35, figura 36 și figura 7.

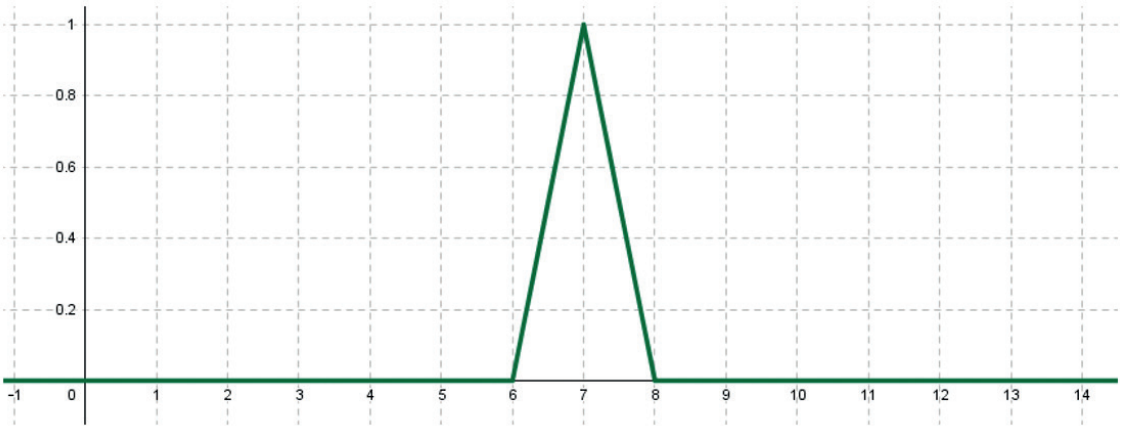


Figure 35. Funcția de apartenență triunghiulară reprezentând valoarea "aproximativ 7".

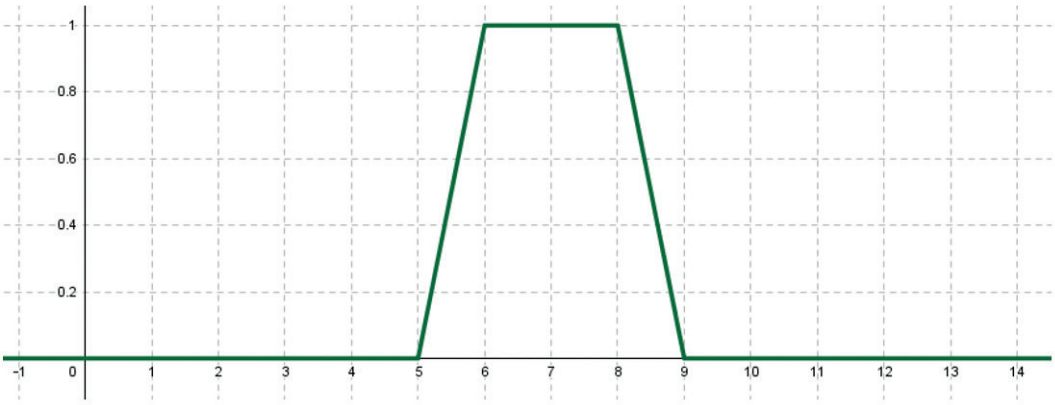


Figura 36. Funcția de apartenență trapezoidală reprezentând valoarea "aproximativ 7".

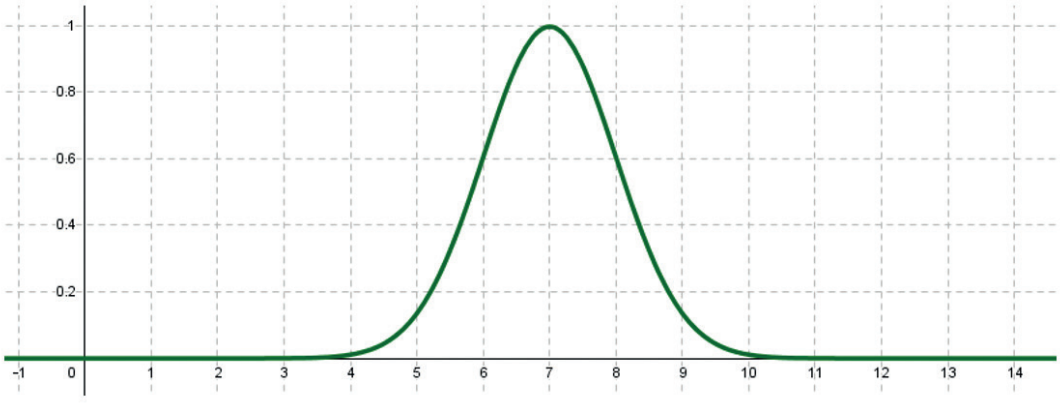


Figura 37. O altă funcție de apartenență reprezentând valoarea "aproximativ 7".

Tipurile de funcţii de apartenenţă

În figurile 2-4 am văzut că funcția de apartenență ar putea avea forme diferite. Vom arăta câteva dintre ele care sunt definite în MATLAB.

Funcţii de apartenenţă liniare

Funcţiile de apartenenţă liniare reprezintă cel mai simplu tip de funcţii de apartenenţă. Acestea sunt construite cu ajutorul unor părţi de linii drepte. Ele sunt împărţite în două grupe de bază:

- triunghiulare,
- trapezoidale.

Funcţia de apartenenţă triunghiulară

Funcţia de apartenenţă triunghiulară este formată din patru părţi (a se vedea figura 38). Prima parte atribuie valoarea de ieşire egală cu zero (interval (-∞ *a*) în figura 38) la valorile de intrare. A doua parte este liniar crescândă de la valoarea 0 la valoarea 1 (interval (*a; b*) în figura 38). A treia parte este liniar descrescătoare de la valoarea 1 la valoarea 0 (interval (*b, c*) în figura 38). Ultima parte atribuie din nou valoarea de ieşire egală cu 0 (interval (c, ∞) din figura 38) la valorile de intrare. În general, această funcţie de apartenenţă este descrisă de 3 parametri: *a, b, c*. În software-ul MATLAB este notat ca **trimf** şi pentru parametri folosim notaţia **[a b c]**. Observaţi că funcţia de apartenenţă triunghiulară atinge o valoare de ieşire egală cu 1 pentru o singură intrare (în special pentru valoarea de intrare *b*).

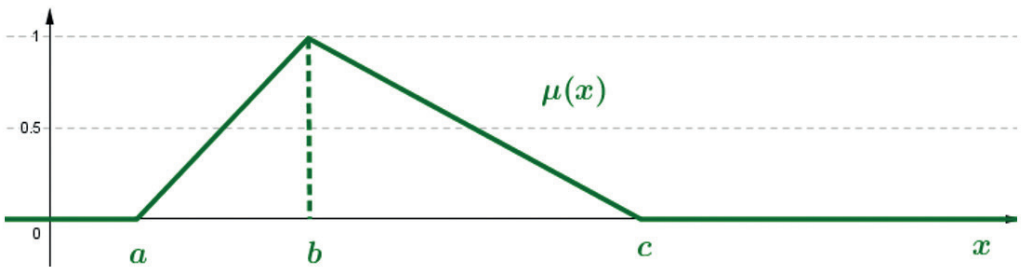


Figure 38. Funcţia generală de apartenenţă triunghiulară

Funcţia de apartenenţă trapezoidală

Funcţia de apartenenţă trapezoidală constă din cinci părţi (a se vedea figura 39). Spre deosebire de funcţia de apartenenţă triunghiulară, această funcţie constă în intervalul valorilor de intrare, care ating valoarea de ieşire egală cu 1. În general, această funcţie de apartenenţă este descrisă de 4 parametri *a, b, c, d*. În software-ul MATLAB este notat ca **trapmf** şi pentru parametri folosim notaţia **[a b c d].**

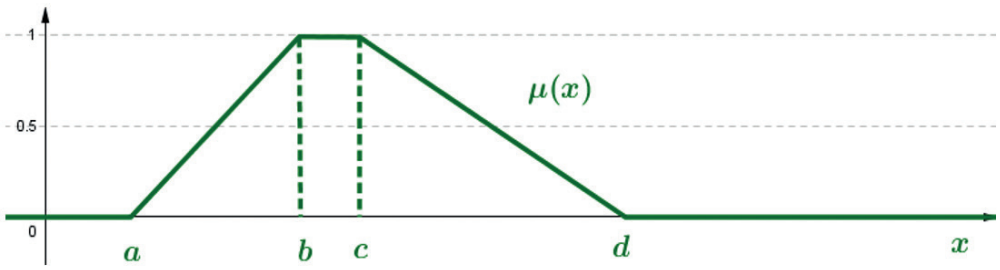


Figura 39. Funcţia generală de apartenenţă trapezoidală

Exemplu: Scrieți notația MATLAB a mulțimilor fuzzy A,B care sunt prezentate în figura 40:

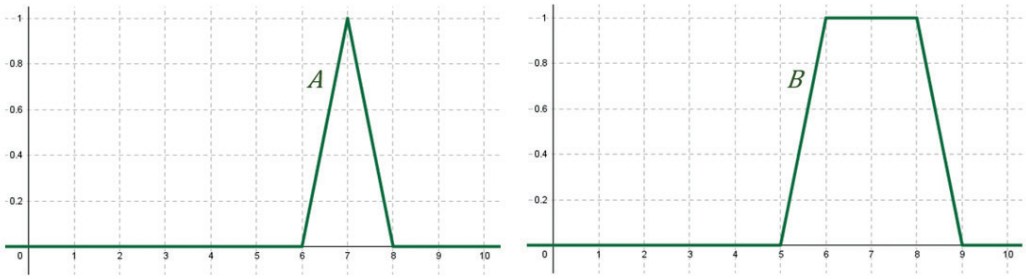


Figura 40. Mulțimile și din exemplu

Mulțimea fuzzy este reprezentată prin utilizarea funcției de apartenență triunghiulare. Notația sa în programul MATLAB este *AA [6 7 8]*. Mulțimea fuzzy este reprezentată prin utilizarea funcției de apartenență trapezoidale. Notația sa în programul MATLAB este *B [5 6 8 9]*.

Exemplu: În aplicațiile din viața reală pentru unele fenomene observate, folosim adesea termenii fenomen scăzut, mediu, înalt. Pentru temperatura apei am putea vorbi despre temperatură scăzută, temperatură medie și temperatură ridicată (vezi figura 41). În timp ce termenul temperatură medie (funcție) ar putea fi descrisă prin funcția de apartenență trapezoidală, așa cum am menționat în textul anterior, valorile temperatură scăzută (funcție) și temperatură ridicată (funcție) sunt specifice în acest sens, că trebuie descrise prin utilizarea funcțiilor de apartenență asimetrice. Cum putem scrie prescrierea acestor funcții în software-ul MATLAB?

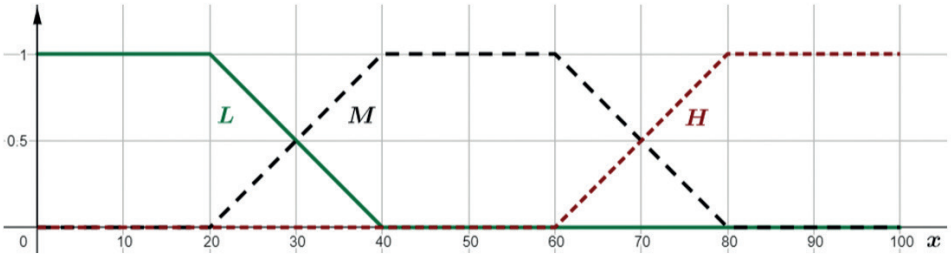


Figura 41. Mulțimile fuzzy și din exemplu

Pentru a descrie o mulțime fuzzy în software-ul MATLAB pot fi folosiți și parametri care nu aparțin universului variabilei studiate. Deci, în primul pas trebuie să determinăm universul termenului „temperatura apei". Este $\mathbb{X} = \langle 0, 100 \rangle$. Apoi am putea scrie

$$L\ [-20\ -\ 10\ 20\ 40], \quad M\ [20\ 40\ 60\ 80], \quad H\ [60\ 80\ 110\ 120].$$

Funcţii de apartenenţă bazate pe polinoame

Acest tip de funcţii se construieşte utilizând funcţiile polinomiale (pătratice). Acestea sunt împărţite în trei grupe de bază:

- curba Pi,
- curba S,
- curba Z.

Funcţia de apartenenţă de tip Pi

Funcţia de apartenenţă de tip Pi este definită de 6 parametri (a se vedea figura 42). Există patru părţi ale acesteia, care sunt definite de funcţii pătratice (intervale $\langle a,b\rangle$; $\langle b,c\rangle$; $\langle d,e\rangle$; $\langle e,f\rangle$), două părţi unde valoarea 0 (intervale $(-\infty,a\rangle$; (f,∞)) este atribuită fiecărei valori de intrare şi unei părţi în care valoarea 1 (interval $\langle c,d\rangle$) este atribuită fiecărei valori de intrare. În software-ul MATLAB este notat ca **pimf**.

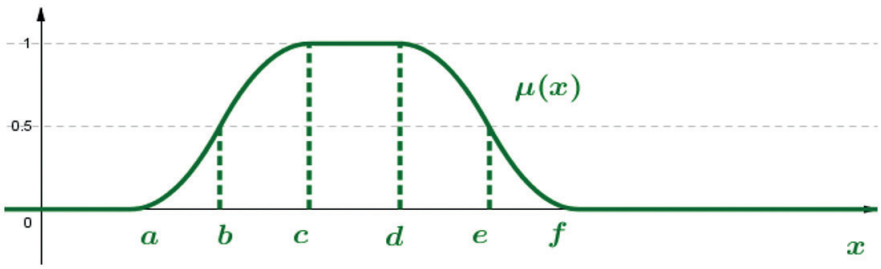


Figura 42. Funcţia de apartenenţă de tip Pi.

Funcţia de apartenenţă de tip S

Funcţia de apartenenţă de tip S este definită de 3 parametri (vezi Figura 43). Există două părţi ale acesteia care sunt definite de funcţii pătratice (intervale $\langle a,b\rangle$; $\langle b,c\rangle$), o parte în care valoarea 0 (intervale $(-\infty,a\rangle$) este atribuită fiecărei valori de intrare şi o parte în care valoarea 1 (interval $\langle c,\infty)$) este atribuită fiecărei valori de intrare. În software-ul MATLAB este notat ca **smf**.

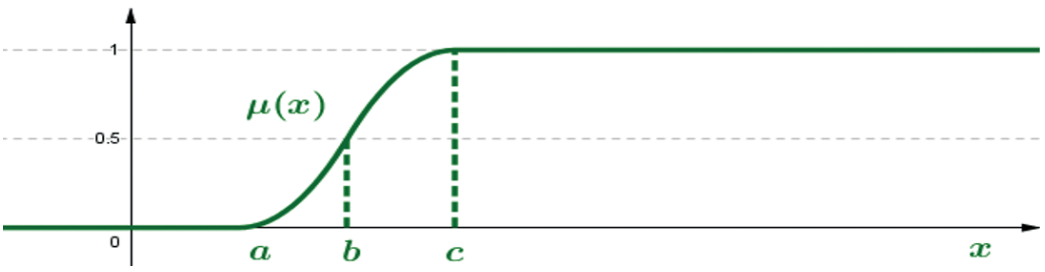


Figura 43. Funcţia de apartenenţă de tip S.

Funcţia de apartenenţă de tip Z

Funcţia de apartenenţă de tip S este definită de 3 parametri *a,b,c,* (a se vedea figura 44). Există două părţi ale acesteia, care sunt definite de funcţii pătratice (intervale $\langle a,b\rangle$; $\langle b,c\rangle$), o parte în care valoarea 1 (interval $(-\infty,a\rangle$) is atribuit fiecărei valori de intrare şi o parte în care valoarea 0 (interval $\langle c,\infty)$) este atribuită fiecărei valori de intrare. În software-ul MATLAB este notat ca **zmf**.

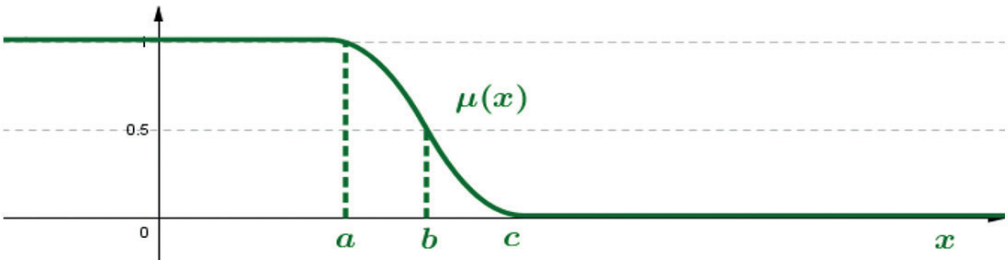


Figure 44: Funcția de apartenență de tip Z.

Remarcăm:

Funcțiile de apartenență de tip S și Z reprezintă funcții de apartenență asimetrice. Acestea ar putea fi utilizate pentru modelarea valorilor scăzute și ridicate ale variabilelor.

Funcție bazată pe o bază statistică

Dacă avem un set mare de date, îl putem procesa folosind o abordare statistică. **Funcțiile de apartenență gaussiane (gaussmf)** sunt derivate din curba clasică de distribuție gaussiană, care are doi parametri *c, σ* (a se vedea figura 45), unde *c,* reprezintă media *σ* și reprezintă abaterea standard a datelor.

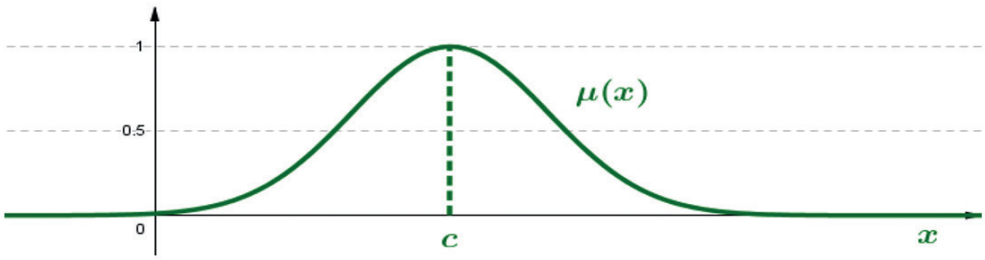


Figura 45. Funcțiile de apartenență gaussiane.

Funcții de apartenență sigmoide

Funcțiile de apartenență gaussiane nu pot specifica **funcțiile de apartenență asimetrice**. Din acest motiv, **funcțiile de apartenență sigmoide (sigmf)** au doi parametri *a, c* (a se vedea figura 46 și figura 47). Apoi parametrii *a, c* sunt obținuți din nou prin utilizarea abordării statistice.

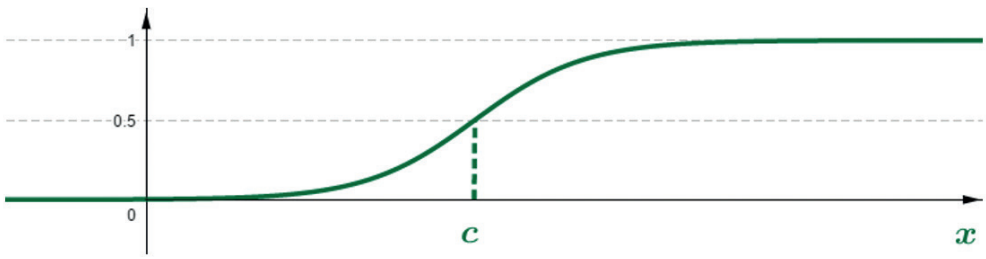


Figura 46. Funcții de apartenență sigmoide, unde.

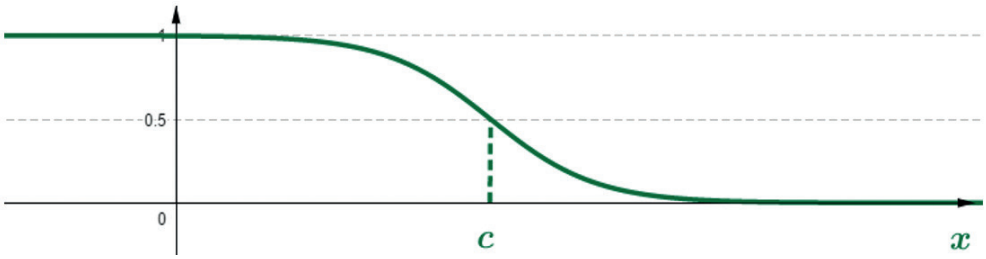


Figura 47. Funcții de apartenență sigmoide, unde .

Se remarcă următoarele:

Vom folosi funcțiile de **apartenență triunghiulare** și **trapezoidale**. În aplicațiile din viața reală sunt adesea utilizate funcții de apartenență gaussiene și sigmoidale, iar parametrii lor sunt aleși cu ajutorul analizei statistice a datelor.

În viața reală, de obicei**, cerem mai întâi expertului** să descrie problema prin funcții adecvate. Apoi, **în al doilea pas**, specificăm de obicei **parametrii funcțiilor** cu utilizarea tratamentului (statistic) al grupului mare de date.

Pentru a lucra cu mulțimi fuzzy, trebuie să definim operațiile de bază pe mulțimile fuzzy – **intersecție**, **reuniune** și **complement**. În mod similar, deoarece există multe tipuri de funcții de apartenență, **sunt definite și mai multe tipuri de operațiuni pe seturi fuzzy**. Vom menționa doar așa-numitele **operații standard pe seturi fuzzy,** care au fost propuse de profesorul Zadeh.

Definiție (intersecție standard)

Fie universul de discuție al aplicației și mulțimile fuzzy. **Intersecția standard a două mulțimi fuzzy** este mulțimea fuzzy cu funcția de apartenență

$$\mu_{A\cap B}(x) = \min\big(\mu_A(x), \mu_B(x)\big).$$

Definiție (reuniune standard)

Fie $\mathbb{X}$ universul de discuție al aplicației și A, B mulțimile fuzzy. **Îmbinarea standard a două mulțimi fuzzy** A,B este mulțimea fuzzy $A \cap B$ cu funcția de apartenență

$$\mu_{A\cup B}(x) = \max\big(\mu_A(x), \mu_B(x)\big).$$

Definiție (complementul standard)

Fie $\mathbb{X}$ universul de discuție al aplicației, iar A mulțimea fuzzy. **Complementul standard al mulțimii fuzzy** A este mulțimea fuzzy $\overline{A}$ cu funcția de apartenență

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

Exemplu: Există două mulțimi fuzzy A,B prezentate în figura 48. Prin utilizarea definițiilor anterioare, determinați grafic intersecția și unirea mulțimilor fuzzy A,B și, de asemenea, completați mulțimea fuzzy A.

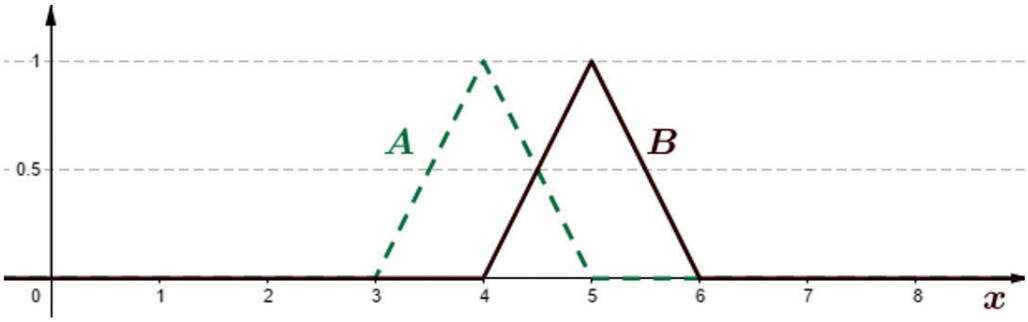


Figura 48. Mulțimile fuzzy din exemplu

Intersecția standard a două mulțimi fuzzy A,B este mulțimea fuzzy $A \cap B$ cu funcția de apartenență $\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$. Soluția este afișată în figura 49.

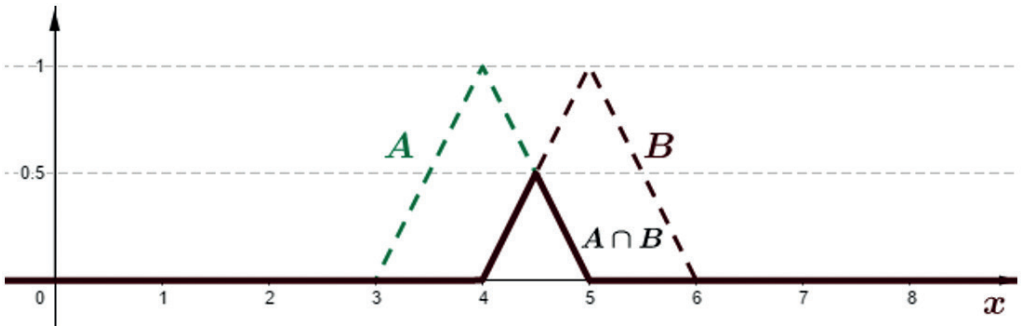


Figura 49. Intersecția standard a mulțimilor fuzzy din exemplu.

Reuniunea standard a două mulțimi fuzzy A,B este mulțimea fuzzy $A \cup B$ cu funcția de apartenență $\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$. Soluția este afișată în figura 50.

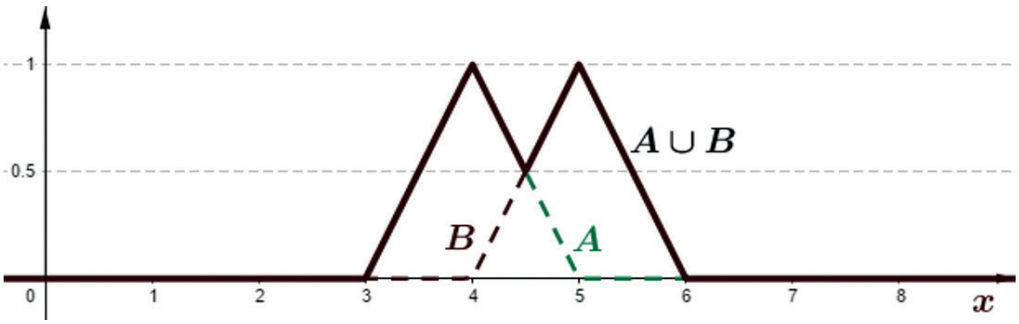


Figura 50. Reuniunea standard a mulțimilor fuzzy din exemplu.

CAPITOLUL 6

RAȚIONAMENTUL FUZZY

Această parte a manualului a fost scrisă de Alžbeta Michalíková de la Departamentul de Informatică, Facultatea de Științe Naturale, Universitatea Matej Bel din Banská Bystrica, Slovacia.

Raționamentul fuzzy este un proces **în care deducem consecințele pe baza informațiilor care constau în termeni vagi**. De exemplu, în viața reală sunt folosite adesea reguli precum

"Dacă este frig afară, mă voi îmbrăca în haine călduroase."

Aceste reguli se obțin prin observarea, învățarea, raționamentul și așa mai departe. În raționamentul fuzzy, sunt folosite așa-numitele **reguli IF - THEN fuzzy**, care au următoarea formă:

$\underbrace{IF\ \langle \dots \rangle}$	$\underbrace{THEN\ \langle \dots \rangle}$
= *antecedent*	= *în consecință*
(premisă)	*(concluzie)*

Pentru nevoile noastre se va modifica regula

"Dacă este frig afară, mă voi îmbrăca în hainele călduroase."

sub forma:

"IF Temperatura exterioară este scăzută, THEN rochia este călduroasă."

Cuvintele **"temperatură"** și **"rochie"** se numesc **variabile lingvistice**, de aceea sunt scrise cu majuscule. Valorile **"scăzut"** și **"cald"** se numesc **valorile variabilelor lingvistice**. Variabilele lingvistice care se află în partea anterioară se numesc **variabile lingvistice de intrare**. Variabilele lingvistice care sunt în consecință se numesc **variabile lingvistice de ieșire**. Folosind operațiile de conjuncție (AND), disjuncție (OR) și negație (NOT) putem crea reguli mai complexe, de exemplu:

"IF Temperatura exterioară este scăzută AND este foarte înnorat THEN rochia este călduroasă."

Dacă sunt definite toate regulile necesare, se obține setul de reguli, care se numește **baza regulilor**. Există diferite abordări pentru lucrul cu regulile bazei de reguli. În cele ce urmează va fi discutată și folosită una dintre ele – **Metoda Sugeno**.

Metoda Sugeno

Autorii acestei metode sunt **T. Takagi**, **M. Sugeno** și **G. Kang**. Au sugerat-o în anul 1985. Această metodă a fost concepută pentru modelarea problemelor în care este posibil să se descrie dependența dintre variabilele de intrare și ieșire ale funcției, care este neliniară, dar există unele părți care sunt liniare.

Metoda Sugeno a fost utilizată pentru prima dată în problema modelării parcărilor. Astăzi este utilizată pentru **aproximarea datelor** prin funcții neliniare, în **clasificare,** reglementare **și** control**, luarea deciziilor fuzzy**, sisteme expert etc.

În metoda Sugeno, valorile variabilelor **de intrare** sunt descrise de **funcțiile de apartenență.** Acestea sunt proiectate de expert. Variabilele **de ieșire** sunt **descrise de funcții** care ar putea fi fie funcții constante**, fie funcții liniare, fie funcții polinomiale de orice grad**.

Sugeno guvernează cu funcțiile de ieșire constante - variabila de ieșire a fiecărei reguli este descrisă de funcția care este constantă. În general, regula are forma

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ X_2\ is\ A_{2j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj}, THEN\ Y\ is\ b_j.$$

Sugeno guvernează cu funcțiile de ieșire liniare - variabila de ieșire a fiecărei reguli este descrisă de funcția liniară. În general, regula are forma

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj},\qquad THEN\ Y\ is\ a_{1j}x_1 + \cdots + a_{nj}x_n + b_j,$$

unde $a_{1j}, \dots, a_{nj}, b_j$ sunt numere reale.

Reguli Sugeno cu funcțiile de ieșire polinomiale - variabila de ieșire a fiecărei reguli este descrisă de funcția polinomială de orice grad.

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj}, THEN\ Y\ is\ a_{1j}{x_1}^{m_1} + \cdots + a_{nj}{x_n}^{m_n} + b_j,$$

unde $a_{1j}, \dots, a_{nj}, b_j$ sunt numere reale și $m_1, \dots, m_n$ sunt numere naturale.

Exemplu: regula Sugeno cu funcția de ieșire constantă

Vor fi evaluați studenții folosind metoda Sugeno. Acest lucru ar putea fi descris prin regulile de următorul tip

R: IF Prezentarea este foarte bună AND valoarea punctajului de la Test este mare,

THEN Evaluarea este egală cu 1 (=A).

Exemplu: Regulile Sugeno cu funcțiile de ieșire liniară

Se știe că pentru unele valori (ca de exemplu, pentru valori mici) ale poziției mașinii, mașina va merge de-a lungul unui traseu în linie dreaptă care poate fi ușor determinată. Regula are următoarea formă:

R: IF valoarea Poziției este mică,THEN traseul Liniar este 3,25x+2,5 .

Avem **baza de reguli cu elemente**.Fie n valori de intrare, de exemplu $\mathbb{x} = (x_1, x_2, \dots, x_n)$. Se consideră că fiecare regulă are funcția de ieșire de forma $y_j = a_{1j}{x_1}^{m_1} + a_{2j}{x_2}^{m_2} + \cdots + a_{nj}{x_n}^{m_n} + b_j$.

Apoi rezultatul final y_x se calculează după formula

$$y_{\mathbb{x}} = \frac{\sum_{j=1}^{k} w_j y_j}{\sum_{j=1}^{k} w_j}$$

unde w_j este ponderea regulii j (a se vedea figura 51).

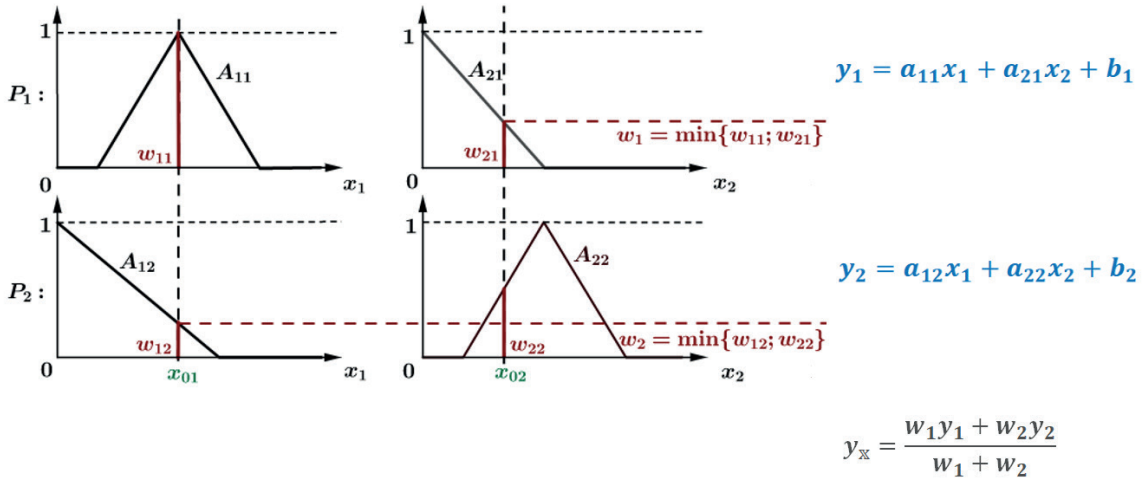


Figura 51. Rezultatul final în metoda Sugeno cu două variabile de intrare și două reguli.

Observație privind modul de obținere a ponderilor:

Presupunem că există regula R_j cu n variabile de intrare $(x_1, x_2, \dots, x_n)$. În primul rând, vor fi calculate ponderile w_{ij} ca intersecție între valoarea intrării măsurate x_i și funcția de apartenență respectiv A_{1j}. În al doilea rând, va fi calculată ponderea w_j utilizând următoarea formulă:

$$w_j = \min_i w_{ij} .$$

Cum se pot proiecta în mod corespunzător regulile fuzzy?

- Se poate cere unui expert să-și descrie cunoștințele folosind funcțiile corespunzătoare.
- Se pot determina valorile parametrilor funcțiilor prin prelucrarea unei cantități mari de date cunoscute.

Această metodă are mai multe nume, de exemplu **metoda Sugeno, sistemul de inferență fuzzy** Takagi-Sugeno, **regulatorul Takagi-Sugeno** etc. Aceste nume reprezintă aceeași metodă. Denumirea utilizată este legată de zona în care este folosită metoda.

În cele ce urmează, se va demonstra utilizarea metodei Sugeno în două domenii diferite:

- în clasificarea datelor,
- în aproximarea datelor.

În ambele cazuri, autorii sunt considerați experții care vor proiecta regulile sistemului dat [2], [6], [7]. Pentru a crea valorile variabilelor **lingvistice de intrare**, vor fi folosite cele mai simple tipuri de funcții de apartenență - **funcții de apartenență liniare**. Pentru a crea valorile variabilelor **lingvistice de ieșire**, vor fi folosite **funcții constante pentru clasificare** și **funcții liniare pentru aproximare**.

CAPITOLUL 7

UTILIZAREA METODEI SUGENO PENTRU CLASIFICAREA DATELOR

Această parte a manualului a fost scrisă de Alžbeta Michalíková de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

În această parte a manualului se va lucra cu setul de date **Iris** (a se vedea Anexa A). Setul de date **Iris** constă din 150 de mostre de flori de Iris. Pentru fiecare floare, sunt patru atribute de bază - lungimea și lățimea sepalelor, precum și lungimea și lățimea petalelor, toate măsurate în centimetri (sau milimetri). Pe de altă parte, florile pot fi clasificate în trei clase corespunzătoare a trei specii de Iris (**Iris Setosa**, **Iris Virginica** și **Iris Versicolor**).

În această parte, se va combina procesarea datelor în două software – în Excel și MATLAB.

*Exemplu: Clasificați datele din setul de date Iris într-un număr adecvat de clase folosind metoda **Sugeno**. (Soluția acestui exemplu poate fi găsită în Anexa B.)*

În primul rând, se încearcă să se răspundă la următoarele întrebări:

1. *Câte **variabile de intrare** există în setul de date Iris?*
2. *Ce vom folosi pentru **a descrie variabilele de intrare**?*
3. *Ce tip de **funcții de apartenență fuzzy** vom folosi?*
4. *Care va fi **rezultatul**?*
5. *Ce vom folosi pentru **a descrie variabilele de ieșire**?*
6. ***Ce tip de reguli** vom folosi?*
7. ***Scrie un exemplu** de o singură regulă!*

În al doilea rând, se descarcă setul de date Iris de pe o pagină web și se copiază în fișierul Excel. **Se marchează** primele 50 de entități cu **culoarea roșie**, următoarele 50 de entități cu **culoarea albastră**, iar restul cu **culoarea verde** Pentru simplitatea utilizării software-ului care conține denumiri standard, vor fi folosiți termeni și denumiri din limba engleză (**red**, **blue**, **green**). În Excel, se creează patru foi independente și se copiază tabelul colorat în fiecare dintre ele. În prima foaie, se sortează valorile (de la cel mai mic la cel mai mare) în funcție de prima coloană. În mod similar, sunt sortate valorile din fiecare

foaie în funcție de una dintre coloane. Pentru a modela variabilele de intrare, vor fi folosite **funcții trapezoidale**. Se determină valorile parametrilor variabilei de intrare din aceste date și se completează în tabelele următoare.

INPUT 1:

Name	Parameters
Universe	
Red	
Blue	
Green	

INPUT 2:

Name	Parameters
Universe	
Red	
Blue	
Green	

INPUT 3:

Name	Parameters
Universe	
Red	
Blue	
Green	

INPUT 4:

Name	Parameters
Universe	
Red	
Blue	
Green	

Tabelul 10. Parametrii variabilelor de intrare

În al treilea rând, se pot determin valorile parametrilor de ieșire. În acest sens, se completează următorul tabel cu valorile corecte dacă, pentru **variabila lingvistică de ieșire**, se folosesc **funcții constante**.

OUTPUT:

Name	Parameters
Universe	
Red	
Blue	
Green	

Tabelul 11. Parametrii variabilei de ieșire

În al patrulea rând, se sugerează un număr de reguli și se scriu corect.

Reguli:

În continuare, valorile obținute sunt procesate cu ajutorul software-ului MATLAB. Se deschide software-ul MATLAB și se scrie comanda **fuzzy** în fereastra de comandă (Command Window). Această comandă deschide Fuzzy Logic Designer. Se utilizează metoda Sugeno (Sugeno fuzzy inference system = Sugeno FIS). Prin urmare, trebuie deschis acest tip de FIS (a se vedea Figura 52a). Putem redenumi și salva acest FIS, de exemplu, ca fișier IRIS_Sugeno a se vedea 52b). Mai departe, sunt necesare patru

variabile lingvistice de intrare – se adaugă trei variabile de intrare noi (a se vedea Figura 53a), apoi se editează parametrii funcțiilor de apartenență (a se vedea Figura 53b). Pentru fiecare variabilă de intrare, se modifică intervalul variabilei pas cu pas, se adaugă numele funcțiilor de apartenență, se schimbă tipul de funcții de apartenență și se adaugă parametri pentru fiecare funcție (se va utiliza Tabelul 10). Acești pași sunt prezentați în Figura 53.

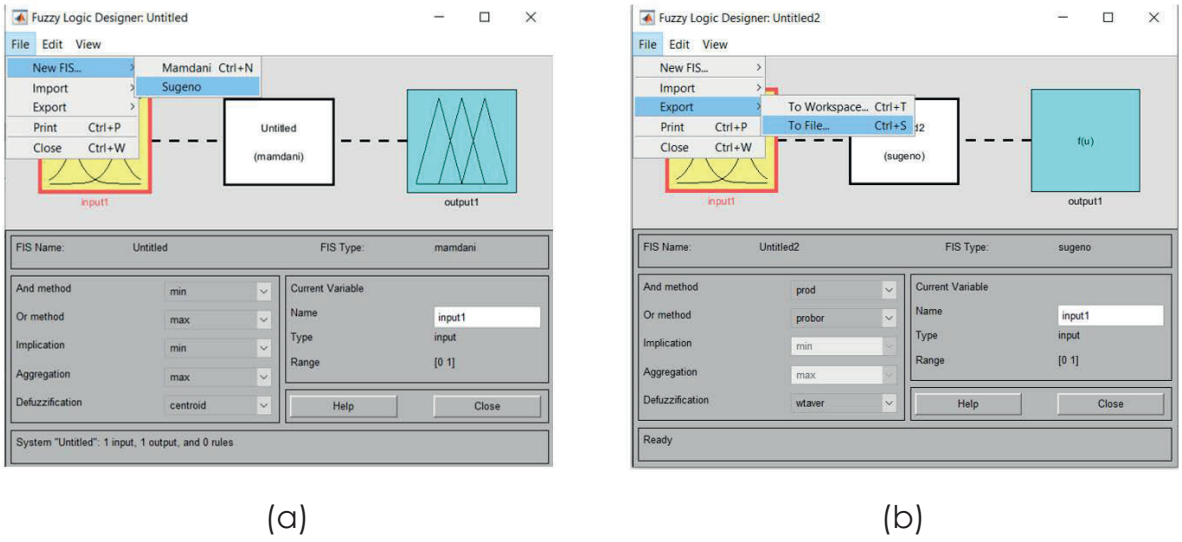


Figure 52. Se deschide un nou Sugeno FIS (a) și se redenumește/salvează FIS (b) în MATLAB

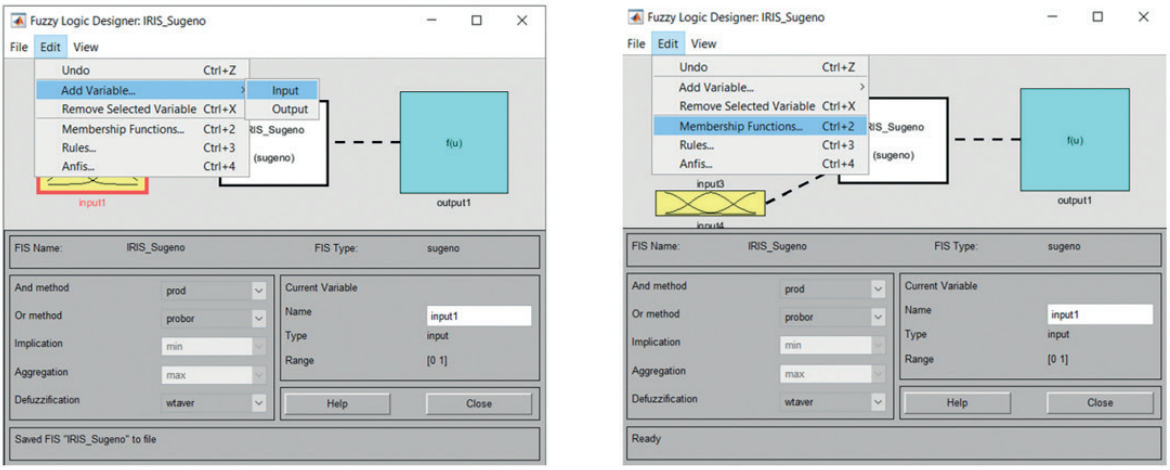


Figure 53: Se folosește meniul Add Variables (a) apoi se editează funcția de apartenență (b) în MATLAB

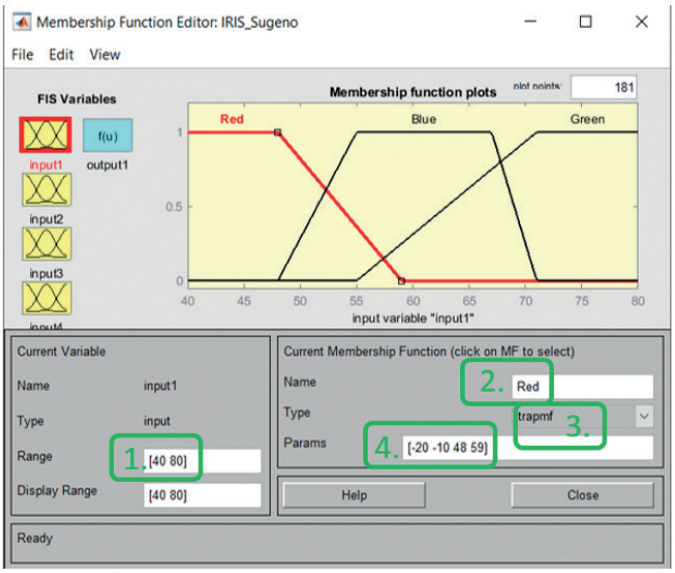


Figure 21: Changing the input membership function parameters in MATLAB software

În continuare sunt schimbate valorile variabilei de ieșire. Pentru a edita variabila de ieșire, se face dublu clic pe dreptunghiul albastru numit output1. Se obține noul meniu pentru variabila de ieșire, așa cum este prezentat în Figura 55. Se completează valorile din Tabelul 11 în acest meniu.

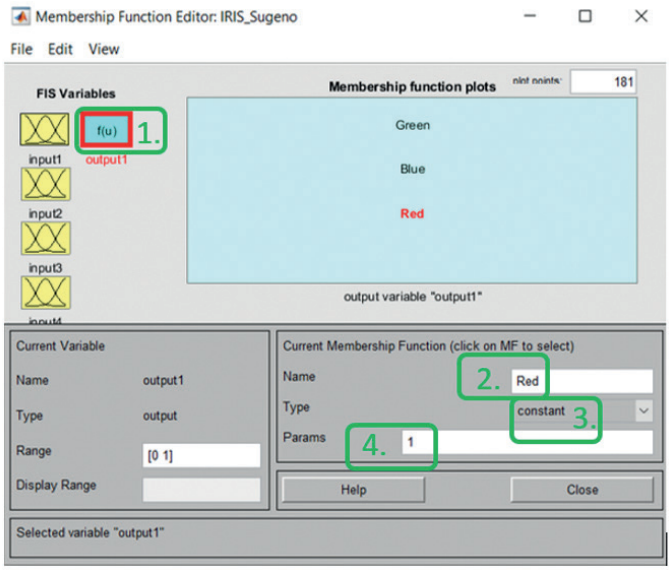
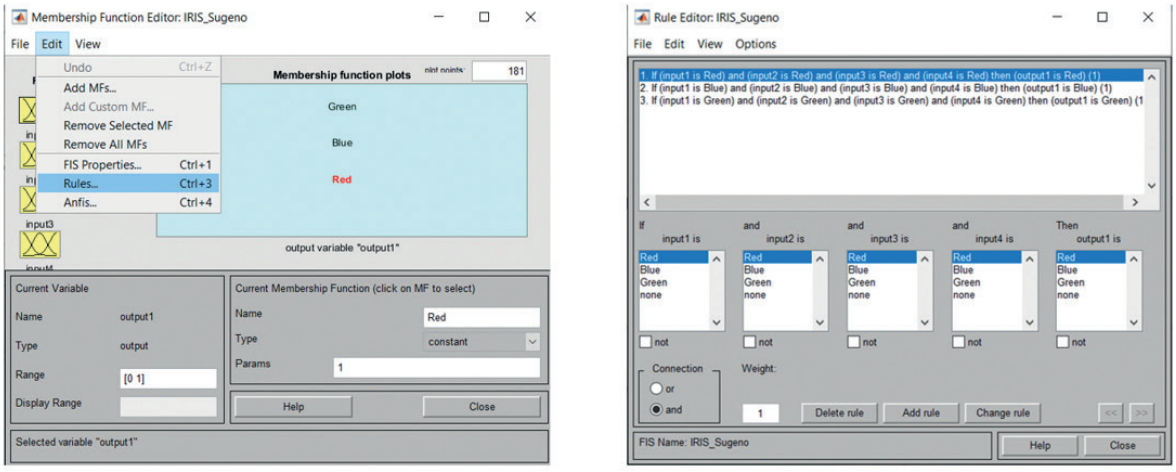


Figure 55. Modificarea parametrilor de ieșire în MATLAB

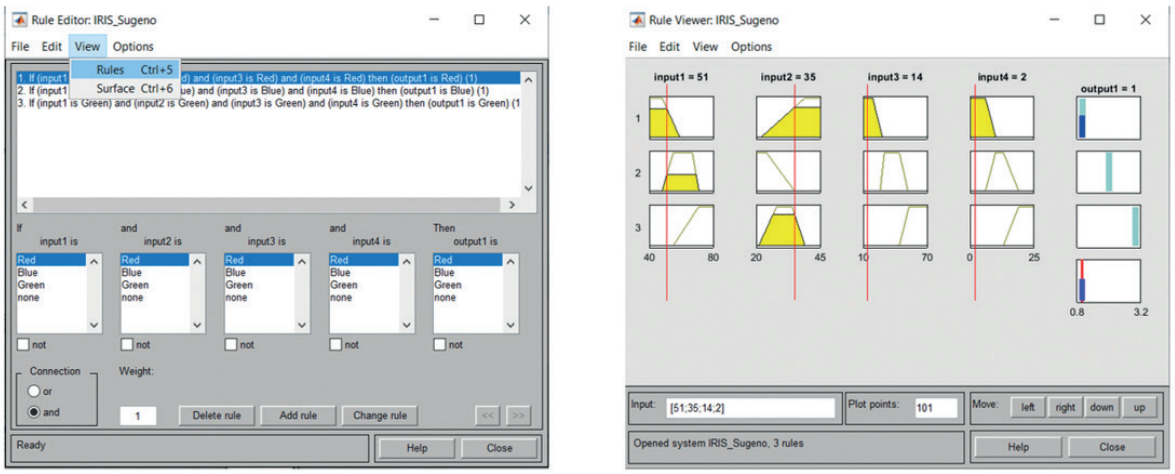
Ultimul pas este cel de creare a regulilor sistemului nostru. Se deschide meniul de reguli (a se vedea Figura 56a) și se folosiesc trei reguli simple (vezi Figura 56b).



(a) (b)

Figure 56. Deschiderea meniului de reguli (a) și adăugarea regulilor (b) în MATLAB

Sistemul este pregătit în acest moment. Pot fi evaluate, pentru început, rezultatele oferite de sistem pentru intrări cunoscute. Se deschide vizualizatorul de reguli (vezi Figura 57a) și se adaugă o valoare specifică fiecărei variabile de intrare (vezi Figura 57b). Aceste valori pot fi adăugate prin mutarea liniilor roșii în partea de sus a meniului sau prin modificarea valorilor parametrilor din grupul situat în partea de jos a meniului.



(a) (b)

Figure 57. Deschiderea vizualizatorului de reguli (a) și adăugarea unor valori specifice la parametrii de intrare (b) în MATLAB

Observații:

Valorile de intrare afișate în Figura 57b aparțin primului rând al tabelului de date Iris. După cum se poate observa, sistemul a clasificat obiectul cu aceste atribute de intrare în clasa 1 (output1 = 1). Este valoarea așteptată ca rezultat!

Anterior s-a arătat cum se clasifică un obiect din setul de date Iris. Această abordare poate fi utilizată pentru fiecare rând al tabelului. De asemenea, pot fi clasificate toate rândurile de tabel într-un singur pas folosind o secvență de comenzi din linia de comandă. Astfel, se calculează rata de succes a clasificării folosind FIS-ul dat. Folosind parametrii menționați în Anexa B, s-a ajuns la o rată de succes de 94,6667%, adică 142 de flori din cele 150 au fost clasificate corect.

Rata de succes a clasificării poate fi îmbunătățită prin utilizarea mai multor abordări diferite. De exemplu, pot fi utilizate mai multe valori ale variabilelor lingvistice de intrare și apoi pot fi create mai multe reguli. Sunt folosite trei valori pentru fiecare variabilă de intrare (**roșu – albastru – verde**). De asemenea, pot fi utilizate cinci valori ale fiecărei variabile de intrare, care reprezintă valorile: **very_small_value – small_value – middle_value – high_value – very_high_value**. Apoi pot fi create mai multe reguli, combinând valorile acestor variabile de intrare. Pe de altă parte, pot fi folosite și alte metode, care au fost concepute pentru a optimiza parametrii valorilor variabilelor de intrare și de ieșire. Unul dintre ele este **ANFIS** (Adaptive Neuro-Fuzzy Inference System), care optimizează parametrii FIS folosind o rețea neuronală. Cunoștințele de bază referitoare la rețelele neuronale sunt prezentate în următoarea parte a acestui manual.

CAPITOLUL 8

UTILIZAREA METODEI SUGENO PENTRU APROXIMAREA DATELOR

Această parte a manualului a fost scrisă de Alžbeta Michalíková de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

Există o mulțime de date care trebuie procesate. Adesea este utilă aproximarea acestor date printr-o funcție mai simplă care furnizează o aproximare a datelor de ieșire reale, pentru valori de intrare exacte. Acest proces se numește **aproximare**. Metoda Sugeno a fost concepută pentru aproximarea unor astfel de date pe unele părți ale domeniului liniar (în 2D ele pot fi aproximate cu o parte a liniei). Pe restul domeniului, acestea trebuiec aproximate printr-o funcție adecvată. În continuare, vor fi aproximate datele reprezentând traseul unei mașini.

Sunt combinate procesarea datelor din programele Excel și MATLAB.

Exemplu: Ne imaginăm dezvoltarea un vehicul autonom. Una dintre problemele care trebuie rezolvată este găsirea funcției care va descrie parcarea într-un loc de parcare. O soluție ar fi să îi cerem șoferului profesionist să parcheze într-un anumit loc de un anumit număr de ori și să captăm traseul mașinii cu ajutorul senzorilor (vezi Figura 58).

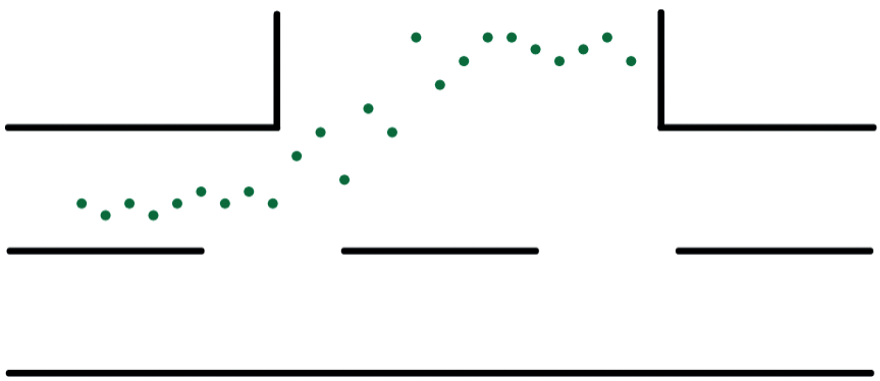


Figure 58. Poziția mașinii în timpul procesului de parcare

Soluție: Această mișcare a mașinii poate fi descrisă prin linii în cele două părți. Prima parte - deplasare pe drum drept înainte de parcare. A doua parte - mișcare pe locul de parcare. Mișcarea dintre aceste două părți va fi aproximată prin metoda Sugeno.

În primul pas, vor fi plasate datele existente în sistemul de coordonate carteziene (vezi Tabelul 12 și Figura 59). De asemenea, pot fi desenate liniile de mișcare dreaptă și vor fi notate y_1 și y_2. După cum putem vedea, unele puncte de date vor contribui la descrierea unei linii (puncte cu valoarea x din intervalele și) și, de asemenea, date care vor contribui la descrierea a două linii (puncte cu valoarea x din intervalele). Această informație este importantă atunci când proiectăm funcțiile de apartenență ale setului fuzzy utilizat.

Tabelul 12. Coordonatele datelor estimate (aproximate)

x	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
y	*8*	*7,5*	*8*	*7,5*	*8*	*8,5*	*8*	*8,5*	*8*	*10*	*11*	*9*
x	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*
y	*12*	*11*	*15*	*13*	*14*	*15*	*15*	*15*	*14*	*15*	*15*	*14*

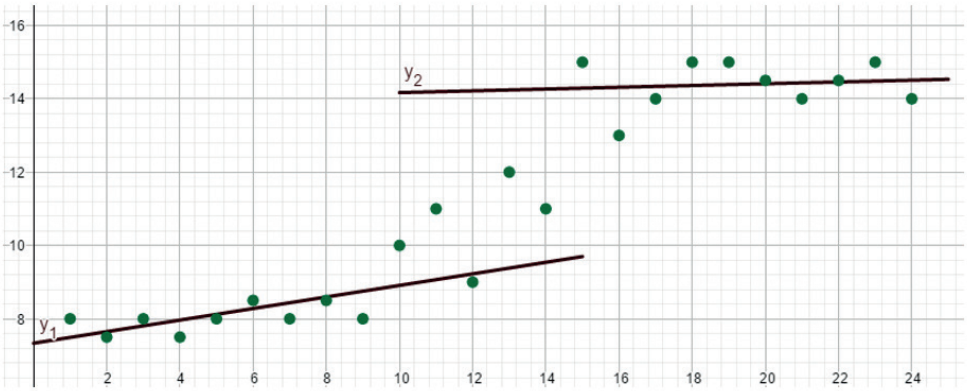
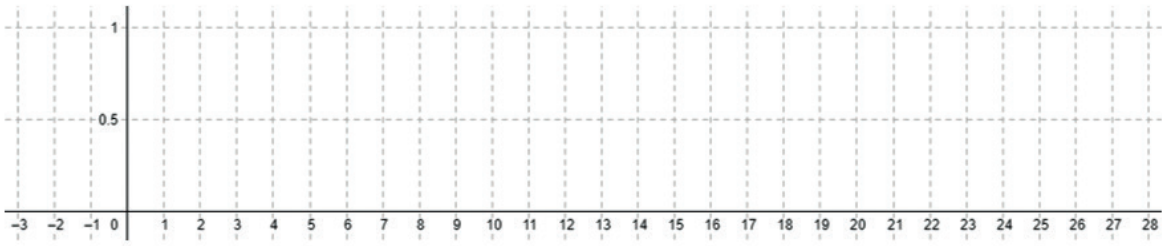


Figura 59. Plasarea datelor în sistemul de coordonate carteziene

Răspundeți la următoarele întrebări:

1. Câte **variabile de intrare** sunt? Să se denumească aceste variabile!
2. Câte **valori ale variabilelor de intrare** vor fi folosite? Să se numească aceste valori ale variabilelor!
3. Cum pot fi **descrise variabilele de intrare**?
4. **Ce tip de funcții de apartenență fuzzy** vor fi folosite?
5. Pot fi **desenate** aceste **funcții fuzzy**? Să se utilizeze următoarea grilă:



6. Pot fi **scrise universul** și **parametrii** acestor funcții? Se pot **scrie etapele** pentru aceste funcții pentru MATLAB?
7. Care va fi **rezultatul**?
8. Cum pot fi **descrise variabilele de ieșire**?
9. **Câte reguli** vor fi folosite?
10. **Să se scrie un exemplu** de regulă!

Răspuns:

Există doar o variabilă de intrare – numită **Poziția** (mașinii) **pe axa x**. Această variabilă de intrare are două valori – **o valoare scăzută a coordonatei *x*** și **o valoare ridicată a coordonatei *x***. Acestea pot fi descrise prin seturi fuzzy. Vor fi folosite **funcții de apartenență trapezoidală** care pot fi desenate ca în Figura 60.

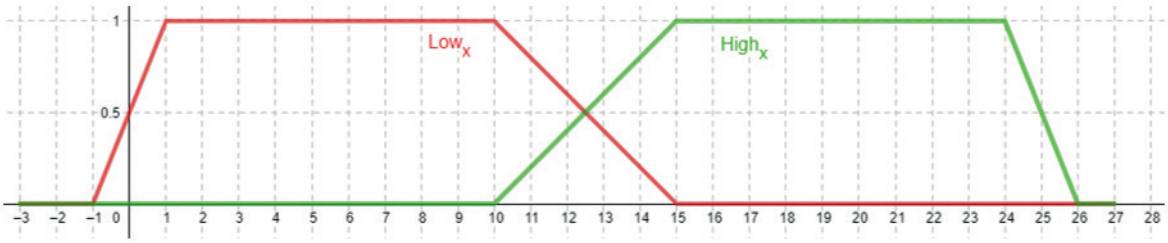


Figura 60. Funcții de apartenență trapezoidale pentru aproximarea datelor

Universul (domeniul) acestor funcții fuzzy este X=[1,24]. Pentru valoarea scăzută a coordonatei x avem parametrii *Low x*=[-1,1,10,15]. Pentru valoarea ridicată a coordonatei x avem parametrii *High x*=[10,15,24,26].

Variabila de ieșire reprezintă **Poziția mașinii pe axa y**. Ca rezultat, se va folosi o funcție liniară (linie). Vor fi utilizate două linii y_1 și y_2. Pentru a descrie parametrii acestor linii, **se va folosi software-ul Excel** (a se vedea mai jos). Vor fi două reguli **IF-THEN** (DACĂ-ATUNCI) care ar putea fi scrise după cum urmează:

R1: IF Poziția mașinii pe axa x este Low_x, THEN Poziția mașinii pe axa y este y_1.

R2: IF Poziția mașinii pe axa x este High_x, THEN Poziția mașinii pe axa y este y_2.

Este necesar să se calculeze parametrii funcțiilor liniare, care reprezintă rezultatul regulilor. Acești parametri vor fi calculați din acele valori de date care contribuie la descrierea exactă a unei linii, de exemplu, la descrierea liniei y_1 au contribuit primele 10 puncte de date. Să le copiem în Excel – fiecare punct întro singură linie (vezi Figura 61a). Apoi sunt marcate aceste puncte și se utilizează următorii pași: **Insert** → **Charts** → **Points**.

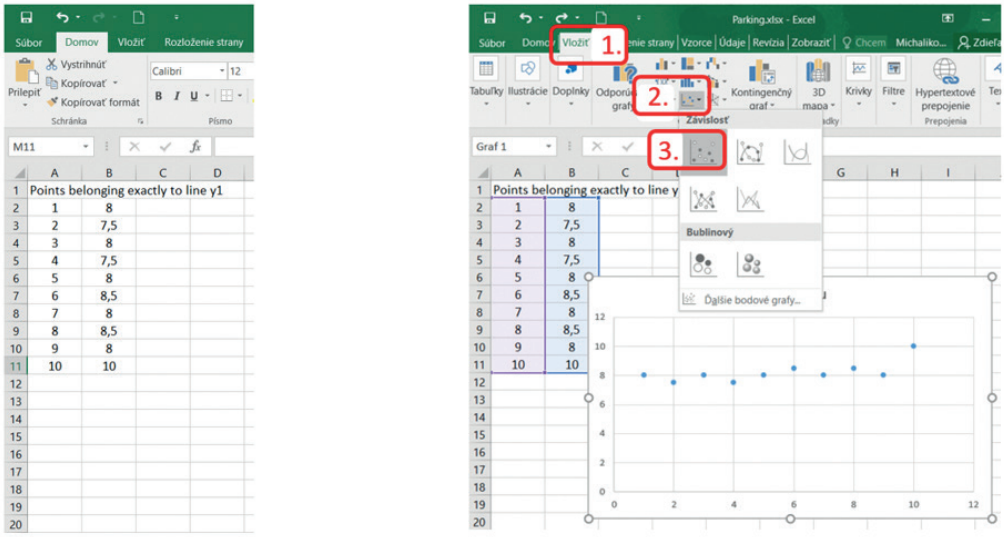


Figura 61. Prelucrarea datelor de intrare în Excel

Apoi se adaugă elementul graficului, după cum este indicat în Figura 62a și se afișează ecuația dreptei (a se vedea Figura 62b).

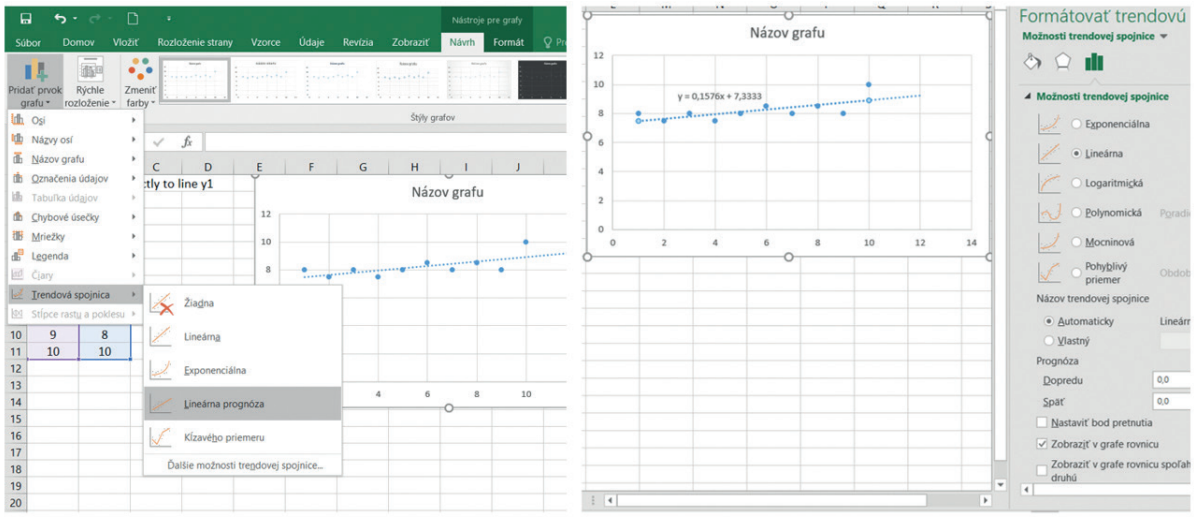


Figura 62. Afișarea ecuației liniei drepte în Excel

Se obțin parametrii liniei y_1. Folosind aceeași procedură, se obțin parametrii liniei y_2. Apoi y_1=0,1576x + 7,3333 și y_2= 0,0242x + 13,927. În MATLAB, se folosește y_1 [0,1576 7,3333] și y_2 [0,0242 13,927].

Tabelul 13. Sinteza parametrilor de intrare și de ieșire pentru aproximarea datelor

INPUT:

Name	Parameters
Universe	[1, 24]
Low x	[-1, 1, 10, 15]
High x	[10, 15, 24, 26]

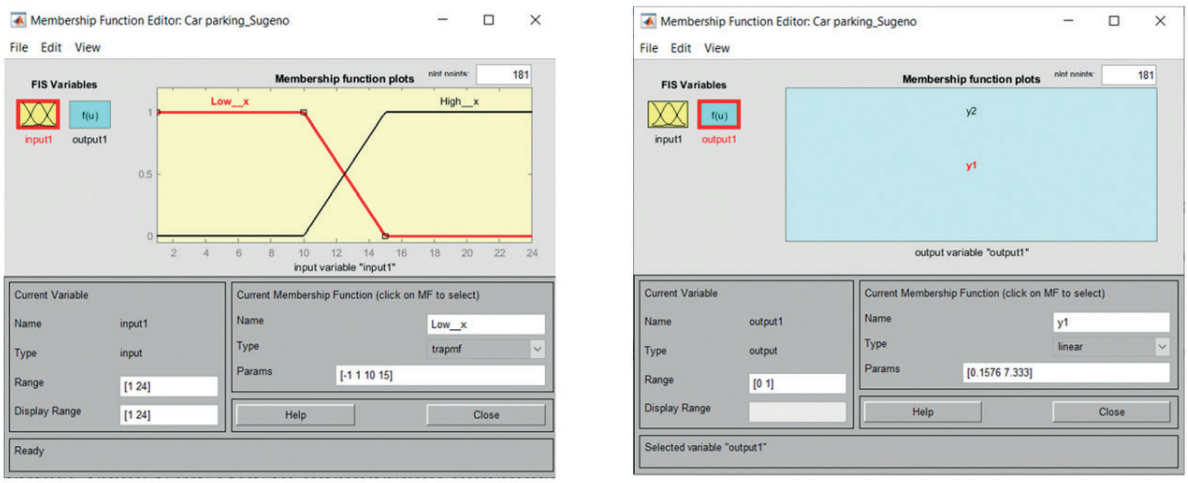
OUTPUT:

Name	Parameters
Universe	---
y_1	[0,1576 7,3333]
y_2	[0,0242 13,927]

Având toți parametrii, poate fi creat FIS de tip Sugeno în MATLAB. Primii pași sunt similari cu cei menționati în subsecțiunea 3 (clasificarea datelor). Se deschide software-ul MATLAB și, în fereastra de comandă se scrie comanda **fuzzy**. Această comandă deschide Fuzzy Logic Designer. Se va folosi metoda Sugeno (Sugeno Fuzzy Inference System = Sugeno FIS). Prin urmare, trebuie deschis acest tip de FIS (vezi Figura 52a). De exemplu, se poate redenumi și salva acest FIS (vezi Figura 52b) ca fișier **Parking_Sugeno**.

Există doar **o variabilă lingvistică de intrare**. Pentru această variabilă sunt doar **două funcții de apartenență**. Pentru a elimina una dintre ele, se face clic pe una dintre funcțiile din grafic și se utilizează *Delete* de pe tastatură. Apoi se editează parametrii funcțiilor de apartenență (se utilizează valorile din tabelul 13). Configurația finală pentru funcțiile de apartenență de intrare este afișată în Figura 63a.

În continuare vor fi schimbate valorile variabilei de ieșire. Pentru a edita variabila de ieșire, se folosește din nou dublu clic pe dreptunghiul albastru numit output1. Se obține meniul pentru variabila de ieșire. În mod similar, așa cum a fost la crearea FIS-ului anterior, sunt completate toate valorile (din Tabelul 13). În acest FIS, tipul funcției de ieșire este liniar (vezi Figura 63b).



(a) (b)

Figura 63. Configurarea variabilelor de intrare și ieșire în MATLAB

Ultimul pas este cel de creare a regulilor sistemului. Sunt folosite două reguli simple (a se vedea Figura 64a). Sistemul este gata. Pot fi evaluate la acest moment rezultatele pe care le oferă sistemul pentru intrări cunoscute. Se deschide vizualizatorul de reguli și se adaugă variabilei de intrare valoarea sa specifică (a se vedea Figura 64b).

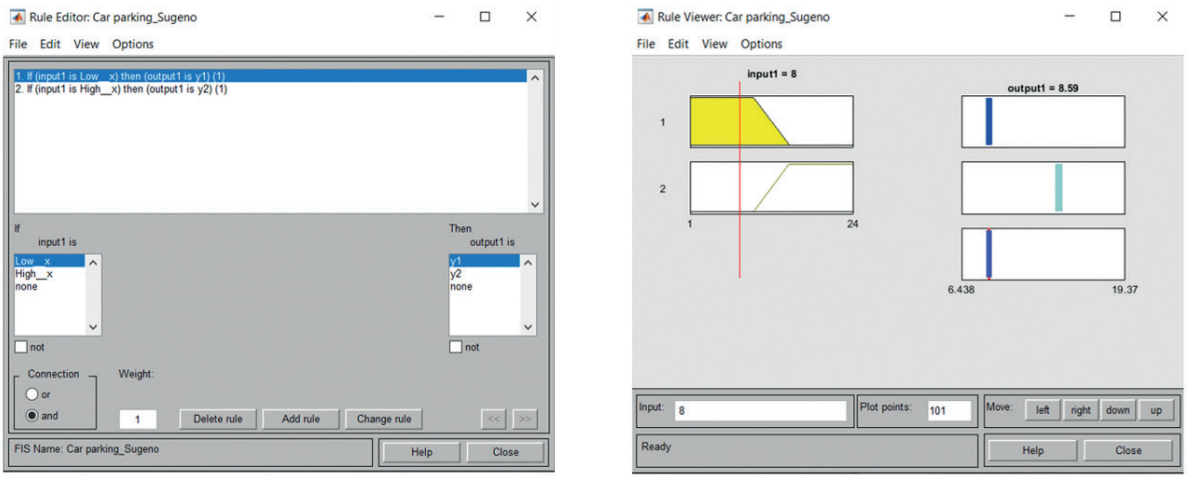


Figura 64. Configurarea regulilor și evaluarea rezultatelor în MATLAB

Există și posibilitatea de a arăta funcția pe care o creăm folosind acest FIS (a se vedea Figura 65a).

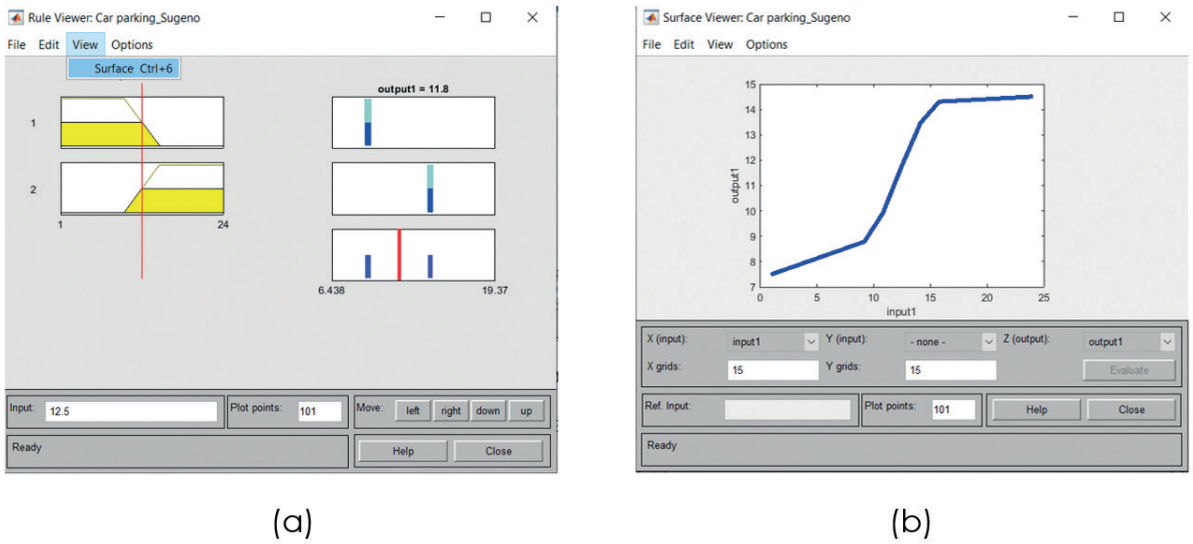


Figura 65. Deschiderea vizualizatorului (Surface Viewer) și funcția finală a FIS creat în MATLAB

Observații:

Valoarea de intrare afișată în Figura 65b este egală cu 8. După cum se poate vedea, sistemul a furnizat o ieșire de 8,59 acestei valori de intrare. Valoarea reală (a se vedea Tabelul 3) a fost 8,5. Prin urmare, valoarea obținută reprezintă o bună aproximare pentru acest punct.

Pot fi comparate datele originale (reale) cu funcția obținută. Există două abordări de bază pentru compararea (evaluarea) rezultatelor. Prima este o comparație grafică. A doua constă în evaluarea eroarii de calcul a sistemului creat. Figura 66 prezintă **o comparație grafică** a datelor reale și a funcției obținute.

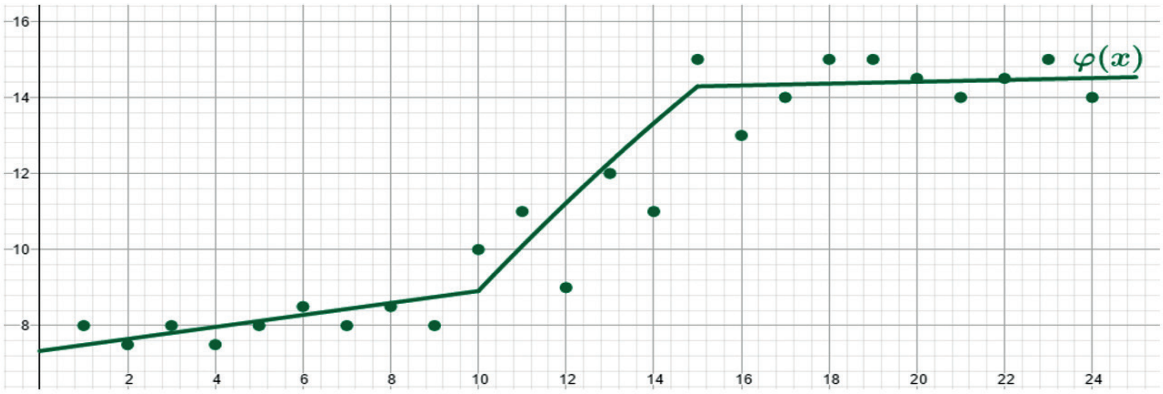


Figura 66. Compararea grafică a datelor reale și a funcției obținute

Până la acest punct s-a arătat cum se obține o valoare de ieșire aproximativă pentru o anumită valoare de intrare. Desigur, pot fi aproximate toate intrările din Tabelul 12 într-un singur pas, prin utilizarea unei secvențe de comenzi din linia de comandă. Se poate calcula și eroarea obținută. Ar putea fi calculate mai multe tipuri de erori. Eroarea pătratică medie (MSE) este cea mai răspândită eroare și se calculează cu formula

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - \varphi(x_i)]^2 ,$$

unde n reprezintă numărul de date de intrare, valorile lui $f(x_i)$ reprezintă ieșirile reale și $\varphi(x_i)$ reprezintă ieșirile calculate de FIS. Valoarea MSE este potrivită pentru utilizare dacă se dorește compararea a două sau mai multe abordări diferite. Atunci cea mai mică valoare reprezintă abordarea mai bună. Pentru sistemul considerat, s-a obținut valoarea MSE egală cu 0,7263

Calitatea aproximației poate fi îmbunătățită prin utilizarea mai multor abordări diferite. De exemplu, se utilizează mai multe funcții de apartenență și apoi pot fi create mai multe reguli. În exemplul prezentat, s-au folosit două funcții de apartenență (**Low_x – High_x**). Pot fi folosite trei valori pentru variabilele de intrare, reprezentând valorile **Low_x – Medium_x – High_x**. Apoi pot fi estimate 3 linii și crea 3 reguli combinând valorile de intrare și de ieșire. În situația când există un set mare de intrări, se poate folosi și un alt tip de funcție de apartenență (au fost menționate în subsecțiunea 2 a acestei secțiuni). Distribuția statistică a datelor poate determina ulterior parametrii funcțiilor. Pe de altă parte, se poate utiliza și o altă metodă, care a fost concepută pentru a optimiza parametrii valorilor de intrare și respectiv ale celor de ieșire.

CAPITOLUL 9

INTRODUCERE ÎN OPTIMIZARE

Această parte a manualului a fost scrisă de Fatih Kilic, de la Facultatea de Informatică și Calculatoare din cadrul Universității de Științe și Tehnologie Adana Alparslan Turkes Science din Adana, Turcia.

În multe domenii de studiu, este abordată o problemă de optimizare x pentru a găsi soluția optimă în spațiul de căutare (toate soluțiile fezabile) folosind metode matematice și euristice. Există diferite probleme de optimizare, precum cele de inginerie, financiare, medicale și de producție. Figura 67 prezintă principalii pași de rezolvare a problemelor de optimizare. În primul pas, factorii de decizie doresc să rezolve o problemă de optimizare pentru a îmbunătăți sistemele actuale sau pentru a sugera noi sisteme. Spre exemplu, dacă se dorește localizarea unui spital în cea mai bună poziție, ținând cont de cererea și potențialii pacienți. În al doilea rând, această problemă trebuie formulată matematic ca structură de soluție, obiectiv și constrângeri. Structura soluției constă în variabile de decizie. Variabilele de decizie sunt pozițiile posibile ale spitalelor candidate pentru această problemă. Funcția obiectivă măsoară calitatea unei soluții de evaluat în rândul soluțiilor candidate. Funcția obiectivă poate fi suma distanțelor dintre spitale și potențialii pacienți pentru problema din exemplu. Toate soluțiile ar putea fi fezabile sau nefezabile din cauza constrângerilor predefinite. Aceste constrângeri sunt definite de specialist. Pentru această problemă, cel puțin un spital ar putea fi într-o subzonă solicitată de părțile interesate. În trei pași, sunt implementate metode bine cunoscute pentru a găsi soluții bune. Aceste metode generează o soluție optimă sau soluții bune apropiate de soluția optimă. Părțile interesate interpretează soluțiile și fac orice revizuiri minore ale soluției, dacă este necesar. În cele din urmă, soluția este realizată.

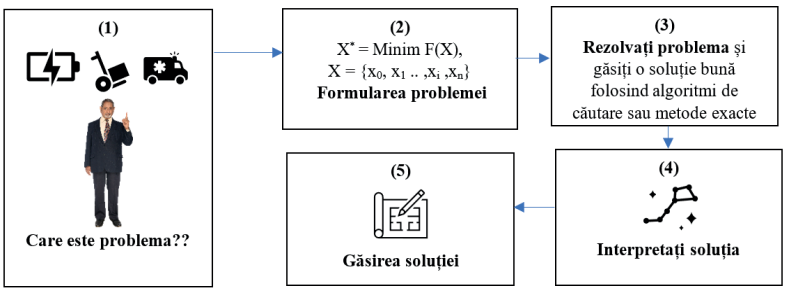


Figura 67. Etapele principale ale rezolvării problemelor de optimizare

Matematic, orice optimizare poate fi explicată după cum urmează:

$$\max_{x \in F \subseteq S} / \min f(x), \qquad \text{Eq. (1)}$$

unde x arată un set de variabile de decizie, F conține soluții fezabile, S reprezintă spațiul soluțiilor și f(x) demonstrează funcția obiectivă, iar max/min reprezintă scopul de a găsi valorile maxime și minime ale lui f(x).

Putem formula constrângeri și o gamă de date și variabile. Un exemplu este dat după cum urmează:

$$\sum_{j}^{n} x_j < \mathrm{b} \qquad \text{Eq. (2)}$$

$$x_j \in \{0,1\} for\ j = 1 \ldots . n \qquad \text{Eq. (3)}$$

unde poate fi 0 sau 1 pentru toate j, iar suma tuturor elementelor x este mai mică decât b.

Problemele care încearcă să găsească variabile continue sunt clasificate ca probleme de optimizare continuă. Tabelul 1 prezintă probleme bine cunoscute de optimizare continuă. Soluția (X) constă în valori reale D-dimensionale. Fiecare dimensiune se află între numerele minime și maxime predefinite. Dimensiunea este numărul de variabile de decizie. Acestea sunt utilizate pentru a-și demonstra performanța atunci când sunt introduși algoritmi de optimizare.

Tabelul 14. Funcții unimodale.

Dimensiune	Interval	Ecuație
5	[-100, 100]	$F_1(x) = \sum_{i=1}^{n} x_i^2$
	[-10, 10]	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
	[-100, 100]	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$
	[-100, 100]	$F_4(x) = \max_i \{ \lvert x_i \rvert,\ 1 \le i \le n \}$
	[-30, 30]	$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$
	[-100, 100]	$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$
	[-1.28, 1.28]	$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}\,[0, 1]$

9.1 ALGORİTMİ DE CĂUTARE LOCALĂ

Algoritmii de căutare locală (Local search algorithms - LSAs) sunt folosiți pentru a rezolva probleme de optimizare în informatică și științe computaționale conexe. Acești algoritmi sunt definiți ca algoritmi euristici de căutare generalizată și pot fi implementați pentru diferite probleme de optimizare după formularea problemelor.

În mod obișnuit, algoritmii LSA caută o singură soluție pentru a identifica cea mai bună variantă în orice moment. Călirea simulată, căutarea Tabu, urcarea dealurilor și căutarea variabilelor învecinate sunt algoritmi de căutare locali bine cunoscuți.

Algoritmul 1 arată pașii principali ai algoritmului de urcare a dealurilor (Hill Climbing - HC).

Algoritmul 1: Urcarea dealurilor

```
1  currentSolution = Generați soluția inițială
2  Evaluare currentSolution
3  iteration =0
4  while (!Stop condiții)
5     NeighbourSolution = Movement(currentSolution)
6     If NeighbourSolution este mai bun decât currentSolution
7        currentSolution = NeighbourSolution
8     end if
9     iteration = iteration+1
```

În primul pas, soluția inițială este generată aleatoriu și este atribuită soluției curente. De exemplu, X = [60.15, -50.07, 10.08, -80.01, 17.59] pentru funcția F1 din tabelul 1. Fiecare element al acestui vector are între -100 și +100 și este selectat aleatoriu. În al doilea rând, soluția vecină este generată folosind soluția curentă și funcția de modificare minoră în fiecare iterație. Numerele de index aleatorii sunt selectate, iar elementul selectat este modificat pentru vectorul X. Dacă soluția vecinului este mai bună decât soluția curentă, atunci soluția curentă este actualizată cu soluția vecinului. Iterațiile sunt efectuate până când soluția curentă este satisfăcătoare sau iterația este maximă.

9.2 CALCUL EVOLUTİV

Calculul evolutiv (Evolutionary computation - EC) este un algoritm popular de optimizare și bazat pe populație care imită evoluția biologică, cum ar fi reproducerea, recombinarea, mutația, selecția și supraviețuirea indivizilor. Diferite variante ale EC sunt introduse folosind procese de evoluție biologică. Algoritmul genetic este inventat de John Holland (1962), în timp ce strategiile evoluționiste sunt inventate de Ingo Rechenberg (1965).

Pașii unui EC tipic sunt descriși de algoritmul 2.

Algoritmul 2: Calcul evolutiv	
1	Population = Generați soluții inițiale aleatorii
2	iteration =0
3	while (!Stop Condiții)
4	Valorile condiției fizice sunt calculate pentru fiecare persoană în Population
5	Persoanele fizice sunt selectate ca părinți în Population pe baza valorilor condiției fizice
6	Crossover-ul și mutația sunt efectuate pentru a genera descendenți
7	Actualizează Population în funcție de noii descendenți și de valorile lor de fitness
8	iteration = iteration+1

În primele etape, populația este efectuată aleatoriu ca o dimensiune predefinită. O populație constă în soluții. Fiecare individ reprezintă o soluție. Al doilea pas este un set de procese iterative. În a doua etapă, valoarea condiției fizice este calculată pentru fiecare persoană care utilizează o funcție obiectivă. Părinții sunt selectați pe baza condiției lor fizice sau a diferitelor tehnici. Crossover și mutație sunt procese de reproducere pentru a genera noi soluții. Pentru generația următoare, procesul de selecție se realizează folosind valorile de fitness ale indivizilor. Acești pași se repetă până când sunt îndeplinite condițiile de oprire.

Operatorul Crossover

Operatorul crossover este utilizat în schimbul de informații între cromozomii a doi părinți selectați pentru a genera doi noi descendenți. Acest operator este un operator important de explorare în CE. Există diferite tehnici generale de încrucișare, cum ar fi crossover-uri cu un singur punct, multi-punct, uniforme și tehnici de încrucișare specifice problemelor (pentru probleme de optimizare combinatorie), cum ar fi recombinarea marginilor, crossover-ul parțial mapat multi-părinte și crossover-ul bazat pe ordine. Acest operator este efectuat în funcție de probabilitatea încrucișării.

Tabelul 15 reprezintă exemplul operatorului crossover cu un singur punct. Părinții 1 și 2 sunt persoane selectate, prezentând două soluții ca italic și subliniat, respectiv în primul și al doilea rând. Un punct de "tăiere" este selectat aleatoriu, iar părinții sunt împărțiți în două părți pentru fiecare individ. Copiii 1 și 2 sunt generați să schimbe a doua parte a părinților și să ia aceleași prime părți ale părinților.

Tabelul 15. Exemplul operatorului crossover într-un singur punct

	X_1	X_2	X_3	X_4	X_5
Părinte 1	60.15	-50.07	10.08	-80.01	17.59
Părinte 2	40.22	30.08	20.09	-20.05	60.85
Copil 1	60.15	-50.07	20.09	-20.05	60.85
Copil 2	40.22	30.08	10.08	-80.01	17.59

Operatorul mutație

Se efectuează un operator de mutație pentru a asigura diversitatea populației. Operatorul mutației modifică un părinte pentru a produce urmași. Se selectează o poziție aleatorie a soluției și se modifică genul sau bitul corespunzător pentru a efectua operatorul mutației. Există diferiți operatori de mutație. Unul dintre operatorii mutației este mutația la scară largă care actualizează simultan poziția multiplă a individului.

O probă de mutație este prezentată în tabelul 16. X_3 este selectat aleatoriu, iar metoda flip-flop este utilizată și o nouă valoare a X_3 trebuie să fie 0.

Tabelul 16. Exemplu de operator mutație

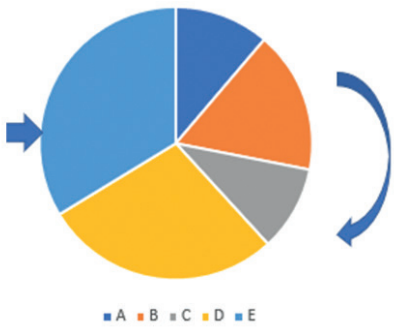
	X_1	X_2	X_3	X_4	X_5
Individ	*1*	*0*	***1***	*0*	*1*
Individ nou	*1*	*0*	***0***	*0*	*1*

Strategii de selecție

Strategiile de selecție sunt utilizate pentru a crește probabilitatea de supraviețuire a indivizilor și a descendenților cu o condiție fizică mai bună în generația următoare și pentru a selecta părinții. Selectarea roții ruletei și selectarea turneului sunt strategii populare de selecție.

Selecția roții ruletei: O roată circulară este formată din n sectoare, unde n reprezintă numărul de soluții din populație. Fiecare soluție primește o parte din întreg pe baza valorii sale de fitness. Este selectat un punct de pe circumferința roții și roata circulară este rotită.

Solutions	Fitness Values
A	10
B	15
C	9
D	25
E	30



Tabelul 17. Un eșantion de populație

Selecția turneului: În această abordare, în selecția „turneu, k", k - indivizii sunt selectați aleatoriu din populație și cei care au cea mai bună stare de sănătate participă la un turneu.

9.3 PROBLEMA RUCSACULUİ – REZOLVAREA PROBLEMEİ

În problema rucsacului, un pachet dintr-un set de articole cu greutăți și valori se dorește pus în rucsac cu valoarea totală maximă. Tabelul 18 prezintă setul de date de testare pentru problema rucsacului.

Tabelul 18. Set de date de testare pentru problema rucsacului

	Articol 1	Articol 2	Articol 3	Articol 4	Articol 5	Articol 6	Articol 7
Greutate	30	20	10	45	15	33	25
Valoare	10	5	30	16	50	13	13
Exemplu soluție	1	0	1	1	0	1	1

Folosim următoarele notații, parametri și variabile de decizie.

Notație:

j: Index articole, $j \in \{1...J\}$, J este numărul de articole

Parametri:

v_j: valoarea articolului j

w_j: greutatea articolului j

W: capacitatea maximă a rucsacului

Variabile de decizie:

$$x_j = \begin{cases} 1, & \textit{if item } j \textit{ is selected into the knapsack} \\ 0, & \textit{otherwise} \end{cases}$$

$$\text{Mazimize F} = \sum_{j=1}^{J} v_j x_j$$

$$\text{subject to} = \sum_{j=1}^{J} w_j x_j < W$$

Codul funcției de fitness:

Fitness Function Code:

```
function Fit = MyFitness(x)
   global wSet vSet maxCapacity;
   sumV = sum(x(1,:).* vSet);
   sumW = sum(x(1,:).* wSet);
   if sumW <= maxCapacity
     Fit= sumV;
   else
     Fit = 0;
   end
```

Genetic Algorithm Code:

```
clc;
```

```
clear;
close all;

global nItem wSet vSet maxCapacity;
wSet = [30, 20, 10, 35, 15, 33, 25, 25, 25, 15, 25,54]; % weights of each item
vSet = [10, 5, 30, 16, 50, 13, 13, 23, 14, 52, 10,50]; % value of each item
maxCapacity = 120;
nItem = size(wSet,2);
FitnessFunction = @(x) MyFitness(x);
WeighFunction = @(x) MyFitnessW(x);
popSize = 20;
maxIter = 50;

muProbility = 0.2;
individual.Solution = [];
individual.FitnessValue = [];
individual.Weight = [];
population = repmat(individual, popSize, 1);
round(rand(1,nItem));
for i = 1:popSize
   population(i).Solution = round(rand(1,nItem));
   population(i).FitnessValue = FitnessFunction(population(i).Solution);
   population(i).Weight = WeighFunction(population(i).Solution);
end

% Sort Population
FitnessValues = [population.FitnessValue];
[FitnessValues, SortOrder] = sort(FitnessValues,'descend');
population = population(SortOrder);

BestSol = population(1);
BestFitness = zeros(maxIter, 1);
TournamentSize=3;

for t = 1:maxIter
   % Crossover operator
   populationCrossover = repmat(individual, popSize/2, 2);
   for j = 1:popSize/2
      i1 = TournamentSelection(population, TournamentSize);
```

```
      i2 = TournamentSelection(population, TournamentSize);
      p1 = population(i1);
      p2 = population(i2);
      % Perform Crossover
      [populationCrossover(j, 1).Solution, populationCrossover(j, 2).Solution] =
      Crossover(p1.Solution, p2.Solution);

   % Evaluate Offsprings
   populationCrossover(j, 1).FitnessValue = FitnessFunction(populationCrossover
   (j, 1).Solution);
   populationCrossover(j, 2).FitnessValue = FitnessFunction(populationCrossover
   (j, 2).Solution);
   populationCrossover(j, 1).Weight = WeighFunction(populationCrossover
   (j, 1).Solution);
   populationCrossover(j, 2).Weight = WeighFunction(populationCrossover
   (j, 2).Solution);
end

   populationCrossover = populationCrossover(:);

   % Mutation operator
   mutPop =0;
   populationMutation = repmat(individual, 0,1);
   for j = 1:popSize
      p = population(i);
   if (rand < muProbility)
      mutPop=mutPop+1;
      k= randi(nItem);
      p.Solution(k) = 1- p.Solution(k);
      p.FitnessValue = FitnessFunction(p.Solution);
      p.Weight = WeighFunction(p.Solution);
      populationMutation(mutPop) = p;
   end
end

populationMutation = populationMutation(:);
   population = [population
   populationCrossover
   populationMutation];
```

```
FitnessValues = [population.FitnessValue];
[FitnessValues, SortOrder] = sort(FitnessValues,'descend');
population = population(SortOrder);
population = population(1:popSize);
FitnessValues = FitnessValues(1:popSize);

BestSol = population(1);

BestFitness(t) = BestSol.FitnessValue;
disp(['Generation : ' num2str(t) ': Best Fitness value = ' num2str(BestFitness(t))]);
end

plot(1:maxIter,BestFitness);
```

CAPITOLUL 10

REȚEAUA NEURONALĂ ÎNTR-UN SINGUR STRAT

Această parte a manualului a fost scrisă de Onder Tutsoy, de la Facultatea de Inginerie din cadrul Universității de Științe și Tehnologie Adana Alparslan Turkes Science din Adana, Turcia.

Rețelele neuronale (NN) stochează informații sub formă de greutăți învățate fie din perspectivele de învățare supravegheate (recunoașterea modelelor), fie din cele nesupravegheate (aproximarea funcțiilor). NN sunt, în esență, abordări de modelare non-parametrică utilizate pentru reprezentarea aproximativă a sistemelor reale. Prin urmare, analiza sa analitică (matematică, riguroasă) este o provocare. Pentru a antrena NN-urile, ponderile ar trebui actualizate pe baza informațiilor furnizate prin intermediul datelor de intrare. Abordarea sistematică utilizată pentru actualizarea greutăților este numită regula de învățare care utilizează informațiile de intrare furnizate. În esență, mapează informațiile de intrare la informațiile de ieșire. Deoarece învățarea este singura modalitate pentru NN de a stoca și de a-și aminti informațiile în mod sistematic, regula de învățare este o componentă vitală a procesului de învățare, discutată în continuare.

10.1 REGULA DELTA

Regula delta este o regulă reprezentativă de învățare a NN-urilor cu un singur strat. Procesul de instruire al unui NN cu un singur strat poate fi prezentat ca în figura de mai jos.

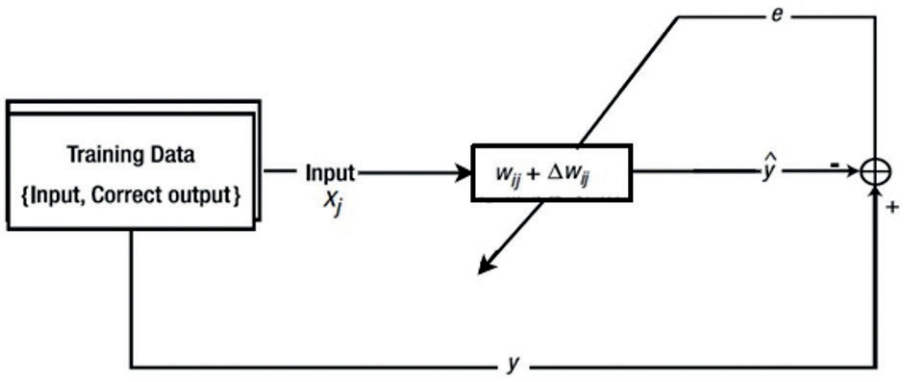


Figura 68. Schema bloc a unui proces de instruire NN cu un singur strat

Este important de reținut că NN cu un singur strat poate fi o singură intrare o singură ieșire (SISO), o singură intrare multiplă ieșire (SIMO), mai multe intrări o singură ieșire (MISO) sau intrări multiple și ieșiri multiple (MIMO). Numărul de intrări și ieșiri variază în funcție de caracterul problemei de învățare. De asemenea, rețineți că dinamica cuplată poate fi învățată numai de către NN care au mai multe intrări sau ieșiri. Cu toate acestea, dacă problema de învățare nu este cuplată, dar problema de învățare a NN este construită ca fiind cuplată, atunci eficiența NN se va reduce. Prin urmare, caracterul datelor de intrare ar trebui analizat inițial și, pe baza informațiilor obținute despre datele de intrare, ar trebui construite NN-urile.

Pseudo cod: Regula de învățare Delta pentru intrări și rezultate NNs

1. **Intrări:** Pregătirea intrărilor $x \in \mathbb{R}^{m \times l}$, unde l este lungimea fiecărui m numărul datelor de intrare, etichetate cu $y \in \mathbb{R}^{l \times n}$, unde n este numărul matricei/vectorului de parametri necunoscuți inițializați aleatoriu rezultatul estimat $w \in \mathbb{R}^{m \times n}$, saturate și estimate $\hat{y}_o \in \mathbb{R}^{n \times l}$, eroarea de antrenament $e \in \mathbb{R}^{l \times n}$, parametrul rata de învățare $0 < \eta \le 2 / x(:,1)^T x(:,1)$, numărul simulărilor multiple simMultiple, matrice de stocare w_s, eroare de stocare e_s, pragul de oprire a erorilor e_t, memorarea rezultatelor estimate $\hat{y}_s$.
2. **Rezultate:** Valoarea terminală a parametrilor instruiți memorarea erorilor de învățare , memorarea rezultatului
3. for *i* to simMultiple
4. for *j* to *l*
5. 1. Calculează rezultatul estimat curent $\hat{y}_o$
6. $\hat{y}_o(:, j) = w^T x(:, j)$
7. 2. Aplicarea pragului σ rezultatului (dacă este necesar)
8. $\hat{y}(:, j) = \sigma(\hat{y}_o(:, j))$
9. 3. Determinarea erorii
10. $e(:, j) = y - \hat{y}(:, j)$
11. 4. Actualizarea și memorarea parametrului necunoscut
12. $w \rightarrow w + \eta e(:, j) x(:, j)^T$
13. $w_s = [w_s; reshape(w_s, 1, [])]$
14. end *j*
15. Momorarea erorii și a rezultatului saturat
16. $e_s = [e_s; reshape(e, 1, [])]$
17. $\hat{y}_s = [\hat{y}_s; reshape(\hat{y}, 1, [])]$
18. If $e(:, j) < e_t$ then
19. break
20. end if
21. end *i*

Regula delta actualizează parametrii necunoscuți iterativ, în loc să rezolve totul dintr-o dată. Este un tip de metodă numerică iterativă care utilizează metoda gradientului (*iterative gradient descent*

method). Metoda gradientului începe de la valoarea inițială și continuă spre soluție. Numele său provine de la comportamentul său prin care caută soluția ca și cum o minge se rostogolește pe deal de-a lungul celei mai abrupte căi. În această analogie, poziția bilei este ieșirea ocazională din model, iar partea de jos este soluția. Este demn de remarcat faptul că metoda de coborâre a gradientului iterativ nu poate arunca mingea în partea de jos cu o singură aruncare. Întregul proces se repetă, deoarece recalificarea modelului cu aceleași date poate îmbunătăți modelul.

Exemplu: Regula Delta

Luați în considerare un NN care constă din trei noduri de intrare și un nod de ieșire, așa cum se arată în figura următoare.

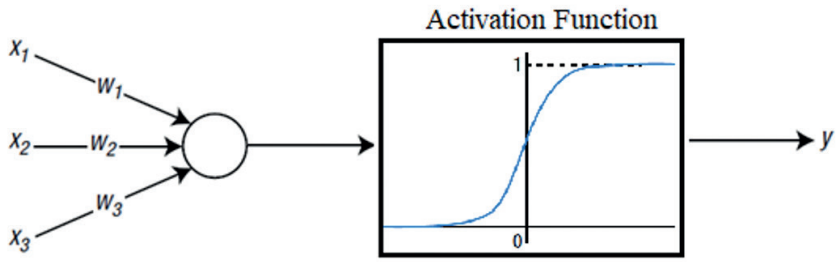


Figura 69. NN care constă din trei noduri de intrare și un nod de ieșire.

După cum se poate observa din figura 69, funcția sigmoid este utilizată pentru funcția de activare a nodului de ieșire. Avem patru puncte de date de antrenament, așa cum se arată în tabelul următor.

Tabelul 18. Poarta Săli de Operații (OR Gate) - puncte de date de instruire cu etichete

{0,0,1,**0**}
{0,1,1,**1**}
{1,0,1,**1**}
{1,1,1,**1**}

Deoarece este utilizat pentru învățarea supravegheată, fiecare punct de date constă dintr-o pereche de ieșire corectă. Ultimul număr al fiecărui set de date este rezultatul corect. Aceasta este o problemă OR Gate (având ultima valoare a intrării 1).

Deoarece este un singur strat și conține date simple de antrenament, codul nu este complicat. Odată ce urmați codul, veți înțelege în mod clar comportamentul de învățare al NN.

Codul corespunzător procedează după cum urmează:

Inițial, parametrii de antrenament sunt definiți ca în următoarea funcție "**trainPar**":

```
function trainPar = trainParameters()
% Training input data where the last value of 1 represents the bias
trainPar.x = [0 0 1; 0 1 1; 1 0 1; 1 1 1]';
% Labelled output data

trainPar.y = [0 1 1 1]';
% Randomly intialized unknown parameters

trainPar.w = rand(size(trainPar.x,1),size(trainPar.y,2));
% Intitialize the estimated output
trainPar.yo_hat = zeros(size(trainPar.x,2),size(trainPar.y,2));

% Intitialize the estimated output
trainPar.y_hat = zeros(size(trainPar.x,2),size(trainPar.y,2));

% Intitialize the error
trainPar.e = zeros(size(trainPar.x,2),size(trainPar.y,2));

% Intitialize the learning rate
trainPar.mu = zeros(size(trainPar.y));

% Learning rate upper scaling
trainPar.mur = 2;

% Stopping error threshold
trainPar.et = 0.001;

% The number of the multiple trainings
trainPar.simMultiple = 1000;

% The output saturation function upper limit (sigmoid)
trainPar.satUppper = 1;

end
```

După definirea parametrilor de antrenament aferenți, se utilizează următoarea funcție pentru procesul de învățare.

```
% This m-file trains a single layer NN for the OR problem
% Upload the training parameters

trainPar = trainParameters();
% Upload the allocated error

e = trainPar.e;
% Upload the allocated estimated output

yo_hat = trainPar.yo_hat;
 Upload the allocated ouput with threshold

y_hat = trainPar.y_hat;
 Upload the allocated unknown parameter
```

```
w = trainPar.w;
% Upload the allocated learning rate

mu = trainPar.mu;
% Introduce the store matrix for the unknown parameter

ws = [];
% Introduce the stora matrix for the error

es = [];
% Introduce the store matrix for the estimated output

ys_hat = [];
for i=1:trainPar.simMultiple

   for j=1:size(trainPar.x,2)
      % Calculate the estimated current output
      yo_hat(j,:) = w'*trainPar.x(:,j);

      % Apply a threshold for the estimated output
      y_hat(j,:) = satOutput(yo_hat(j,:),trainPar);

      % Determine the instant error
      e(j,:) = trainPar.y(j,:) - y_hat(j,:);

      % Update the learning rate
      mu(i,j) =trainPar.mur/(trainPar.x(:,j)'*trainPar.x(:,j));

      % Update the unknown parameter vector/matrix
      w = w + mu(i,j)*e(j,:)*trainPar.x(:,j);

      % Store the unknown parameter vector/matrix
      ws = [ws;reshape(w,1,[])];

end

% Store the error history
es = [es;reshape(e,1,[])];

% Store the estimated output
ys_hat = [ys_hat;reshape(y_hat,1,[])];

end
```

unde funcția „**satOutput**” este creată pentru funcția de activare sigmoid, așa cum este prezentat mai jos:

```
function y_sat = satOutput(y_unsat, trainPar)

y_sat = trainPar.satUppper / (1 + exp(-y_unsat));

end
```

*Apoi, rezultatele estimate **ys_hat** pentru fiecare set de intrări sunt reprezentate folosind următorul bloc de cod:*

```
figure(1),
plot(1:length(ys_hat),ys_hat(:,1),'r','LineWidth',2),
hold on,
plot(1:length(ys_hat),ys_hat(:,2),'b','LineWidth',2),
plot(1:length(ys_hat),ys_hat(:,3),'g','LineWidth',2),
plot(1:length(ys_hat),ys_hat(:,4),'y','LineWidth',2),
hold off
title('Estimated Outputs for the OR Gate')
```

Execuția acestui cod realizează următoarea diagramă:

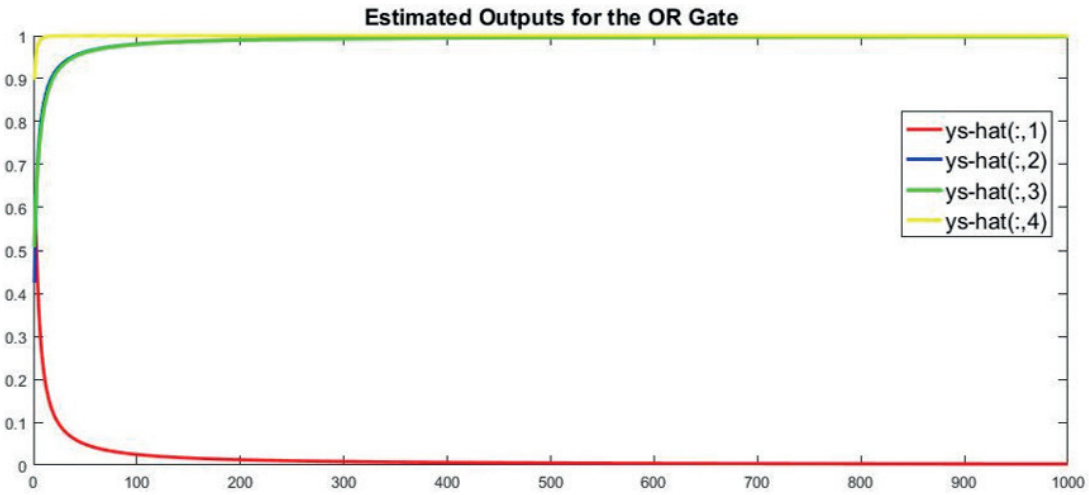


Figura 70. Rezultatele estimate ale realizărilor pentru poarta sala de operații (OR Gate)

Executarea acestui cod produce următoarele valori. Valorile de ieșire sunt foarte apropiate de rezultatele corecte din valoarea țintă y. Prin urmare, concluzionăm că NN a fost instruit corespunzător pentru a învăța OR Gate.

$$\begin{bmatrix} 0.0025 \\ 0.9980 \\ 0.9980 \\ 1.0000 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Pentru a reprezenta grafic eroarea de antrenament,

```
plot(1:length(es),es(:,1),'r','LineWidth',2),
hold on,
```

```
plot(1:length(es),es(:,2),'b','LineWidth',2),
plot(1:length(es),es(:,3),'g','LineWidth',2),
plot(1:length(es),es(:,4),'y','LineWidth',2),
hold off
title('Training Error for the OR Gate')
```

se utilizează blocul de cod și rezultatul determinat este reprezentat astfel:

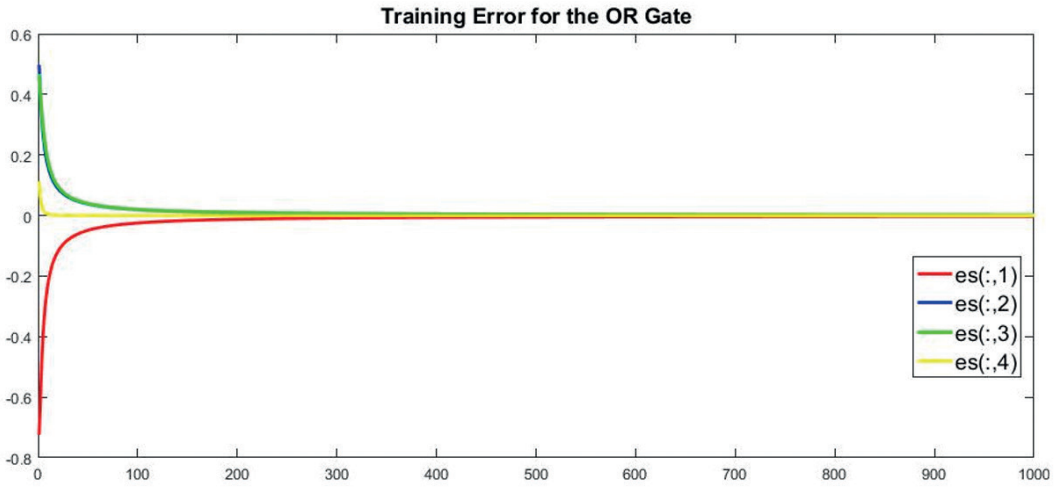


Figura 71. Rezultatele erorilor de instruire pentru poarta OR

După cum se poate observa din figura 4, eroarea converge la zero pentru punctele de date OR Gate corespunzătoare.

În cele din urmă, parametrii necunoscuți instruiți sunt reprezentați grafic și afișați folosind următorul bloc de cod:

```
figure(),
plot(1:length(ws),ws(:,1),'r','LineWidth',2),
hold on,
plot(1:length(ws),ws(:,2),'b','LineWidth',2),
plot(1:length(ws),ws(:,3),'g','LineWidth',2),
plot(1:length(ws),ws(:,4),'y','LineWidth',2),
hold off
title(,Parametri necunoscuți instruiți pentru poarta OR')
```

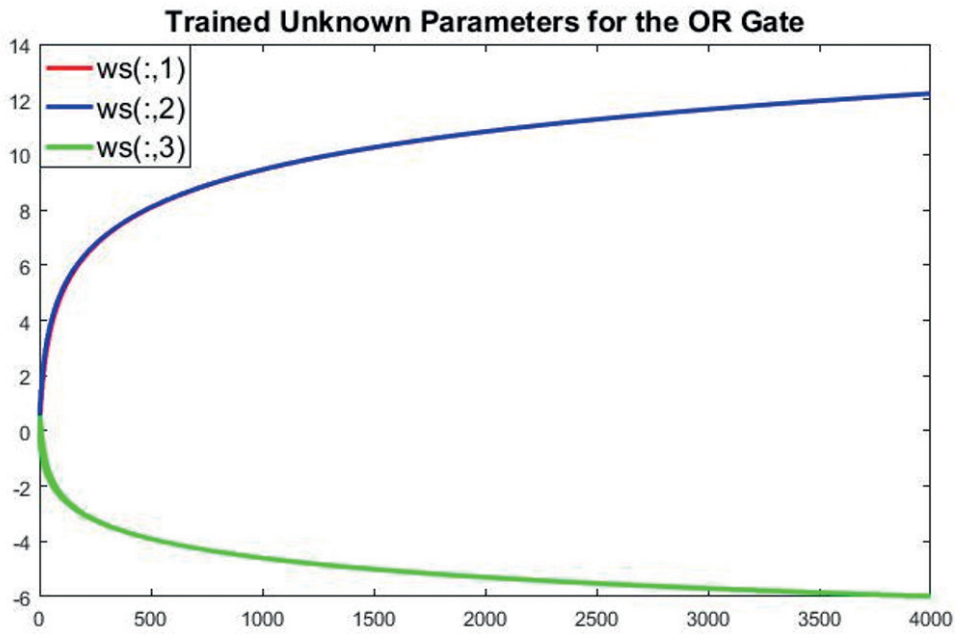


Figura 72. Parametri necunoscuți instruiți pentru poarta sala de operații (OR)

După cum se poate observa din 72, sunt reprezentați grafic doar 3 parametri necunoscuți instruiți. Acest lucru se întâmplă datorită înmulțirii matricelor din următorul bloc de cod.

```
trainPar.w = rand(size(trainPar.x,1),size(trainPar.y,2));
```

*unde **size(trainPar.x)**este **3x4** și **size(trainPar.y)**este **4x1**. Astfel, se determină un vector 3x1. De fapt, este destul de simplu desenarea tuturor parametrilor necunoscuți instruiți aici. După toate aceste explicații, se poate face cu ușurință actualizarea necesară în cod.*

10.2 LIMITĂRILE NNS CU UN SINGUR STRAT

Această secțiune prezintă motivul critic pentru care NN cu un singur strat a trebuit să evolueze într-un NN cu mai multe straturi. Vom încerca să arătăm acest lucru printr-un caz particular. Luați în considerare același NN care a fost discutat în secțiunea anterioară. Se compune din trei noduri de intrare și un nod de ieșire, iar funcția de activare a nodului de ieșire este o funcție sigmoidă. Să presupunem că avem patru puncte de date de antrenament, după cum se arată mai jos.

Tabelul 19. Puncte de date de antrenament ale porții XOR cu etichete

{0,0,1,**0**}
{0,1,1,**1**}
{1,0,1,**1**}
{1,1,1,**0**}

După cum se arată în tabelul 2, aceasta este o problemă XOR Gate având ultima valoare a intrării ca părtinire a lui 1. Este diferită de secțiunea „Regula Delta" prin faptul că a doua și a patra ieșire corectă sunt comutate, în timp ce intrările rămân aceleași. Ei bine, diferența este abia vizibilă.

Deoarece avem în vedere același NN, îl putem antrena folosind funcția „**trainPar**" din secțiunea „Exemplu: regula Delta", cu excepția faptului că are valori diferite pentru y, așa cum am menționat anterior. Înainte de a executa codul, blocul codului de date de ieșire etichetat în funcția „**trainPar**" este actualizat după cum urmează.

```
% Date de ieșire etichetate
trainPar.y = [0 1 1 0]';
```

Executând acest cod, vor apărea următoarele valori care constau în ieșirea din NN instruită corespunzătoare datelor de antrenament. Le putem compara cu rezultatele corecte date de „y" ca:

$$\begin{bmatrix} 0.5297 \\ 0.5000 \\ 0.4703 \\ 0.4409 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

După cum se poate observa din ecuația determinată, avem două seturi total diferite. Instruirea NN pentru o perioadă mai lungă nu face diferența.

Ce s-a întâmplat de fapt?

Ilustrarea datelor de antrenament poate ajuta la elucidarea acestei probleme. Să interpretăm cele trei valori ale datelor de intrare drept coordonatele X, Y și, respectiv, Z. Deoarece a treia valoare (coordonata Z) este fixată 1, datele de antrenament pot fi vizualizate pe un plan așa cum se arată în figura următoare.

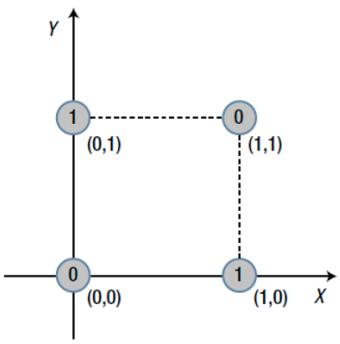


Figura 73. Interpretarea celor trei valori ale datelor de intrare sub forma coordonatelor X, Y și Z

Valorile 0 și 1 din cercuri sunt ieșirile corecte atribuite fiecărui punct. Un lucru de observat din această figură este că nu putem împărți regiunile 0 și 1 cu o linie dreaptă. Cu toate acestea, o putem împărți cu o curbă complicată, așa cum se arată în figura următoare. Spunem că acest tip de problemă este liniar inseparabilă.

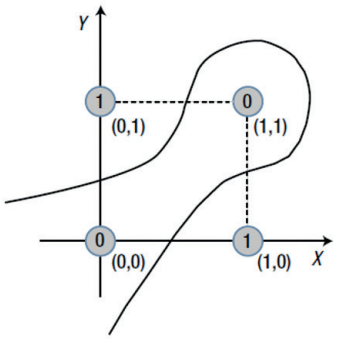


Figura 74. Separarea lui 0 și 1 printr-o curbă complicată (liniar inseparabilă)

În același proces, datele de antrenament din secțiunea «Exemplu: regula Delta» din planul X-Y apar ca:

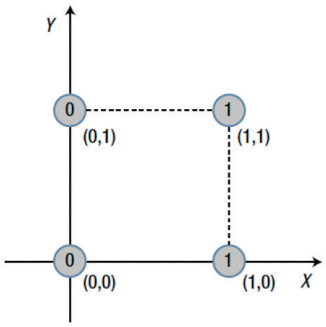


Figura 75. Datele de antrenament pentru regula delta

În acest caz, o linie dreaptă de frontieră care împarte regiunile 0 și 1 poate fi găsită cu ușurință. Aceasta este o problemă separabilă liniar, așa cum este arătat în figura următoare:

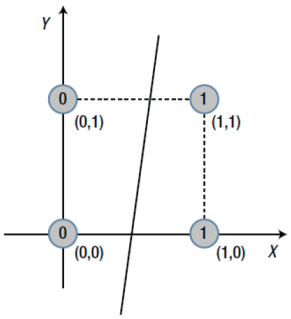


Figura 76. Problemă separabilă liniar

Mai simplu spus, NN cu un singur strat poate rezolva doar problemele separabile liniar. Acest lucru se datorează faptului că NN cu un singur strat este un model care împarte liniar spațiul de date de intrare. Pentru a depăși această limitare a NN cu un singur strat, avem nevoie de mai multe straturi în rețea. Această necesitate a dus la apariția NN multistrat, care poate realiza ceea ce NN cu un singur strat nu poate. A se reține că NN cu un singur strat este aplicabil pentru anumite tipuri de probleme. NN multistrat nu are astfel de limitări. Pentru mai multe detalii, pot fi consultate notele bibilografice.

CAPITOLUL 11

Implementarea Rețelei Neuronale

Această parte a manualului a fost scrisă de Jarmila Škrinárová, de la Departamentul de Informatică, Facultatea de Științe ale Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

Matlab este un limbaj de nivel înalt și un mediu interactiv pentru calcul numeric, vizualizare și programare pentru:

- analiza datelor,
- dezvoltare de algoritmi,
- creare de modele și aplicații.

Scopul secțiunii este de a-i ajuta pe cei interesați să învățe cum să lucreze cu Matlab și cum să creeze rețele neuronale simple. În cele ce urmează, va fi introdusă o metodologie și vor fi prezentate trei exemple de rețele neuronale în Matlab. Exemplele sunt orientate pe funcția de ajustare a creației și sarcinile de clasificare.

11.1 SCURTĂ ÎNTRODUCERE ÎN MATLAB – MATRİX LABORATORY

În primul rând, va fi prezentat mediul Matlab. Există o bandă de instrumente în partea superioară a ferestrei. Sub bara de instrumente, zona este împărțită în patru ferestre, care sunt destinate navigării (deplasarea prin structura de directoare), editării scripturilor executabile, afișării spațiului de lucru și o fereastră de comandă (a se vedea Figura 77).

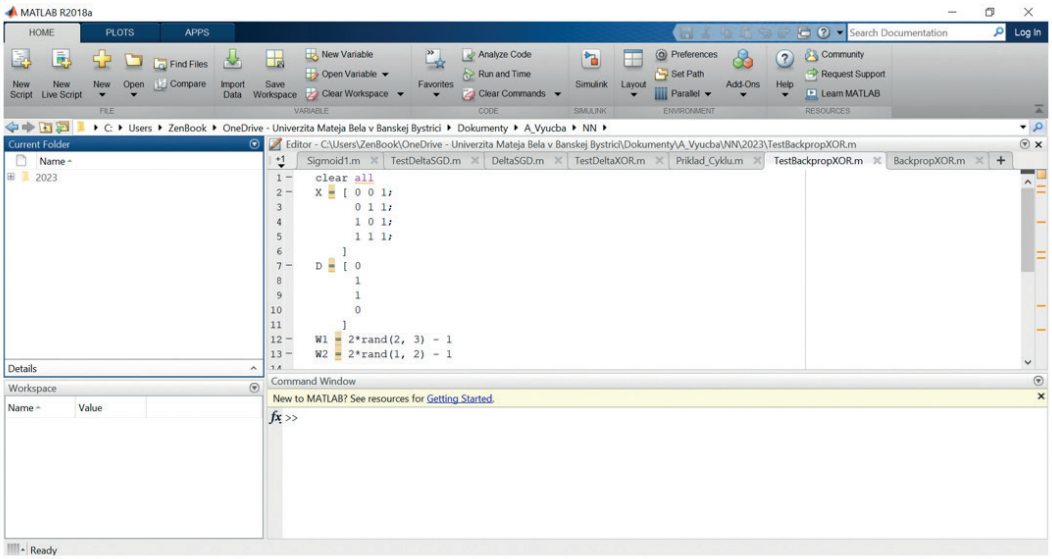


Figura 77. Editor, fereastra de comandă, spațiul de lucru, navigator

Primul pas, este de a învăța cum să se lucreze în fereastra de comandă, unde se scriu comenzi după semnul ">>" (a se vedea Figura 78).

Figura 78. Fereastra de comandă

Exemple de calcule simple, lucrul cu variabile, vectori și matrici

Calculul cu **variabile** (exemplele 1 – 4) poate fi exersat treptat direct în fereastra de comandă:

Exemplul 1	Exemplul 2	Exemplul 3	Exemplul 4
>> 12+34 ans = 46	**>> a =5, b = a^2** a = 5 b = 25	**>> (101+79) /(47 -17)** ans = 6	**>> 15*12** ans = 180

Un **vector** este o matrice unidimensională de elemente. De obicei sunt scrise elementele individuale ale vectorilor între paranteze pătrate și se separă cu o virgulă sau un spațiu. A se reține că se scrie nota după semnul %. În următoarele părți ale acestui capitol, se va lucra cu rețele neuronale și va fi nevoie de date de intrare și țintă pentru învățarea rețelelor neuronale. Dacă rețeaua are o singură intrare, datele de intrare sunt o matrice unidimensională de elemente (vector). Dacă rețeaua are o singură ieșire, datele țintă sunt, de asemenea, o matrice unidimensională de elemente [3].

Exemplul 5:	Exemplul 6:	Exemplul 7:	Exemplul 8:
>> u1=[1 2 3 4] %row vector u1 = 1 2 3 4	**>> u2=[1 2 1 2]** %row vector u2 = 1 2 1 2	**>> u1.*u2** %scalar product of two vectors ans = 1 4 3 8	**>> v=[-1; -7; -3]** %column vector v = -1 -7 -3

Exemplul 9:	Exemplul 10:	Exemplul 11:	Exemplul 12:
>> w=[1 7 -2]' %transposed vector w = 1 7 -2	**>> 6:2:12** % To generate a regular vector, we define the first and last elements of the vector and the step. ans = 6 8 10 12	**>> m=15:-3:0** m =15 12 9 6 3 0	**>> x=12** x = 12 **>> z=[x, 2*x, 3*x]** z = 12 24 36

Exemplul 13:	Exemplul 14:
>> W=2*rand(1,3)-1 W = 0.9298 -0.6848 0.9412	**>> x2=linspace(-1, 4, 8)** % -1 to 4 is interval and 8 is number of elements x2 =-1.0000 -0.2857 0.4286 1.1429 1.8571 2.5714 3.2857 4.0000

Pentru a folosi mai mult de o intrare sau o țintă în rețelele neuronale, trebuiesc pregătite date sub formă de câmpuri bidimensionale. În Matlab, tablourile bidimensionale sunt reprezentate prin matrici. Prin urmare, mai departe va fi exersat lucrul cu **matrici**:

Exemplul 15:	*Exemplul 16:*
>> A=[1 -1 2 -3; 3 0 4 5; 3.2, 5 -6 12] *%matrix* *A = 1.0000 -1.0000 2.0000 -3.0000* *3.0000 0 4.0000 5.0000* *3.2000 5.0000 -6.0000 12.0000*	**>> O=[]** *%empty matrix* *O = []*

Exemplul 17:	Exemplul 18:
>> **B=[A; u1]** %Matrix expansion by 1 row (vector u1). B = 1.0000 -1.0000 2.0000 -3.0000 3.0000 0 4.0000 5.0000 3.2000 5.0000 -6.0000 12.0000 1.0000 2.0000 3.0000 4.0000	**>> C=[A, v]** %Extending the matrix by 1 column (vector v). C = 1.0000 -1.0000 2.0000 -3.0000 -1.0000 3.0000 0 4.0000 5.0000 -7.0000 3.2000 5.0000 -6.0000 12.0000 -3.0000

Exemplul 19:	Exemplul 20:
>> Z=zeros(2,5) %Creating a null matrix of size 2 rows by 5 columns. Z = 0 0 0 0 0 0 0 0 0 0	**>> O1=ones(3,4)** %Creating a unit matrix with dimension 3 rows by 4 columns. O1 = 1 1 1 1 1 1 1 1 1 1 1 1

Exemplul 21:	Exemplul 22:
>> A=[1 -1 2 -3; 3 0 4 5; 3.2, 5 -6 12] A = 1.0000 -1.0000 2.0000 -3.0000 3.0000 0 4.0000 5.0000 3.2000 5.0000 -6.0000 12.0000 **>> A(2, :)** % Listing of the 2nd row of matrix A ans = 3 0 4 5 **>> A(:, 3)** % Listing of the 3rd column A matrix ans = 2 4 -6	**>> I=eye(5,8)** %Creating a diagonal matrix of size 5 rows by 8 columns. I = 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
Exemplul 23:	Exemplul 24:
>> R1=rand(3,5) %Create a random matrix of size 3 rows by 5 columns, with values in the range 0 to 1 R1 = 0.1419 0.7922 0.0357 0.6787 0.3922 0.4218 0.9595 0.8491 0.7577 0.6555 0.9157 0.6557 0.9340 0.7431 0.1712	**>> R2=randn(4)** %matrix with random elements - standard distribution R2 = 0.8884 -2.9443 1.3703 0.3192 -1.1471 1.4384 -1.7115 0.3129 -1.0689 0.3252 -0.1022 -0.8649 -0.8095 -0.7549 -0.2414 -0.0301

Exemplul 25:	Exemplul 26:
>> A=[1 5 0; -1 2 3; 1 2 1] *A = 1 5 0* *-1 2 3* *1 2 1* **>> c=2** *c = 2* *%matrix multiplication by vector* **>> D=A*c** *D = 2 10 0* *-2 4 6* *2 4 2*	**>> B=[1 2 3; 1 2 3; 1 2 3]** *B = 1 2 3* *1 2 3* *1 2 3* **>> E=B*D** *%matrix multiplication* *E = 4 30 18* *4 30 18* *4 30 18*

Exemplul 27:	Exemplul 28:
>> A =[2 3; 0 10] *B =[1 0; -3 5]* *%matrix multiplication* *A = 2 3* *0 10* *B = 1 0* *-3 5* **>> C=A*B** *C = -7 15* *-30 50*	*%matrix scalar multiplication, matrices A and B are from previous example* **>> C=A.*B** *C = 2 0* *0 50*

Sunt folosite adesea date care conțin multe elemente. Prin urmare, este practică scrierea aceste date într-un **fișier** pentru a le putea folosi mai târziu. Toate datele cu care s-a lucrat în Matlab sunt stocate în spațiul de lucru și pot fi văzute în fereastra din stânga jos. Pot fi afișate informații despre conținutul spațiului de lucru folosind exemplele 29 și 30.

Exemplul 29:

>> who % workspace listing only with names of variables, vectors and matrices
Your variables are:
A B C O ans u v w x z

Exemplul 30:

>> whos % workspace

Name	Size	Bytes	Class	Attributes
A	3x4	96	double	
B	4x4	128	double	
C	3x5	120	double	
O	0x0	0	double	
ans	1x1	8	double	
u1	1x4	32	double	
u2	1x4	32	double	
v	3x1	24	double	
w	3x1	24	double	
x	1x1	8	double	
z	1x3	24	double	

Pot fi **salvate** toate variabilele, vectorii și matricele **din spațiul de lucru** într-un fișier (a se vedea exemplul 31) sau numai variabilele, vectorii și matricele selectate (a se vedea exemplul 32)

Exemplul 31:	Exemplul 32:
>> save data % save all data to file data.dat	**>> save data1 u1 u2 v** % save variables u1, u2 and v into data1.mat

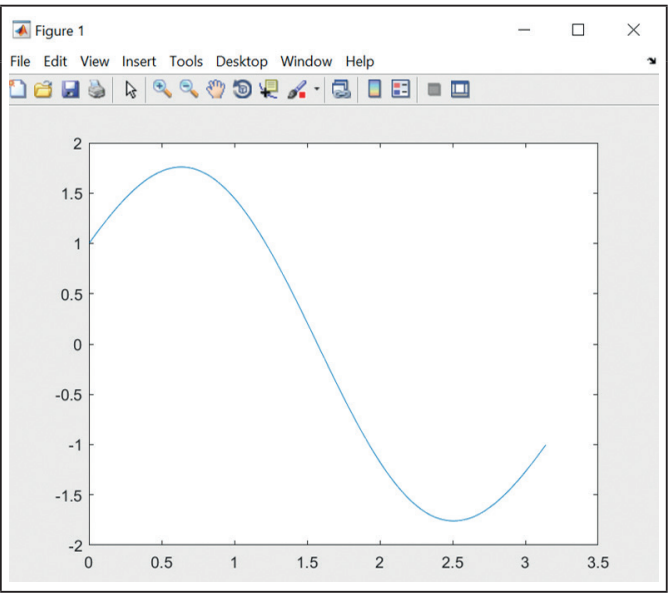
Datele stocate în fișiere pot fi încărcate oricând în spațiul de lucru al instrumentului Matlab pentru a lucra cu ele. A se vedea exemplele 33 și 34.

Exemplul 33:	Exemplul34:
>> load data1 % read all variables saved in data1.mat	**>> load data.dat -MAT** % read all variables saved in data.dat

La **reprezentarea grafică** a unei anumite funcții, numărul de elemente de pe axa x trebuie să fie același cu cel de pe axa y. Se poate vedea funcția desenată în imaginea din exemplul 35. Funcția este trasată într-o fereastră nouă care poate fi editată – se pot instroduce nume de axe, titluri etc.

Example 35:

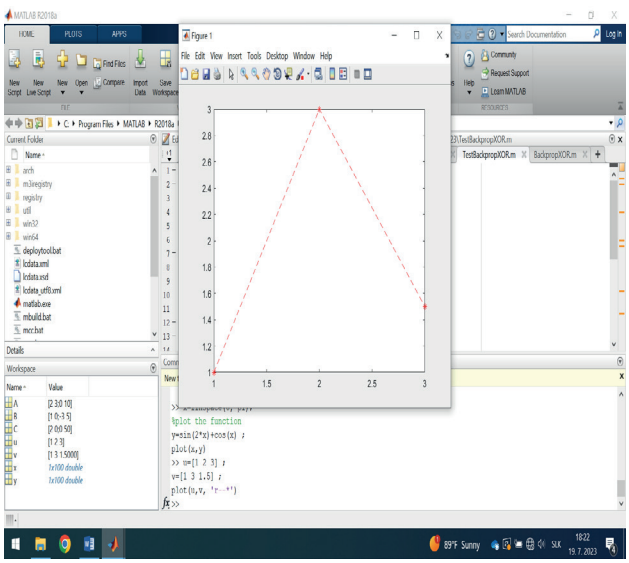
```
>> x=linspace(0, pi);
%plot the function
>> y=sin(2*x)+cos(x) ;
>> plot(x,y)
```



Atât **culoarea**, cât și **tipul de linie** pot fi modificate în comanda plot. A se vedea exemplul 36.

Example 36:

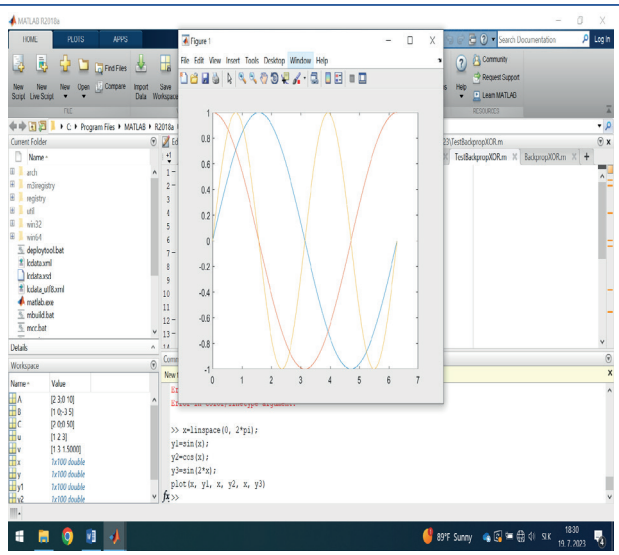
```
>> u=[1 2 3] ;
>> v=[1 3 1.5] ;
>> plot(u,v, 'r--*')
```



De asemenea, este posibilă **reprezentarea mai multor funcții** într-o singură imagine. A se vedea in acest sens exemplul 37.

Example 37:

```
>> x=linspace(0, 2*pi);
>> y1=sin(x);
>> y2=cos(x);
>> y3=sin(2*x);
>> plot(x, y1, x, y2, x, y3)
```



Pentru ilustrare, este prezentat un exemplu de program simplu (exemplul 38), în care rândurile individuale ale matricei X sunt scrise secvenţial într-un ciclu. Ieşirea din program este în coloana din dreapta.

Example 38:

```
>> clear all
X = [ 0 0 1;
      0 1 1;
      1 0 1;
      1 1 1;
   ] ;
N = 4; % print always one row of matrix X
for k = 1:N
x = X(k, :)
end
```

```
x = 0 0 1

x = 0 1 1

x = 1 0 1

x = 1 1 1
```

În secțiunile anterioare, s-a arătat cum se scriu comenzile în linia de comandă. Acest lucru nu este practic dacă trebuie scrise multe comenzi. Astfel de comenzi succesive ar putea fi scrise într-un editor de text şi ulterior salvate într-un fişier cu extensia „m”. Acesta este modul de creare a unui **script** care să deschidă şi ruleze în mediul Matlab. Comenzile scrise în script trebuie mai întâi testate în linia de comandă Matlab pentru a evita erorile.

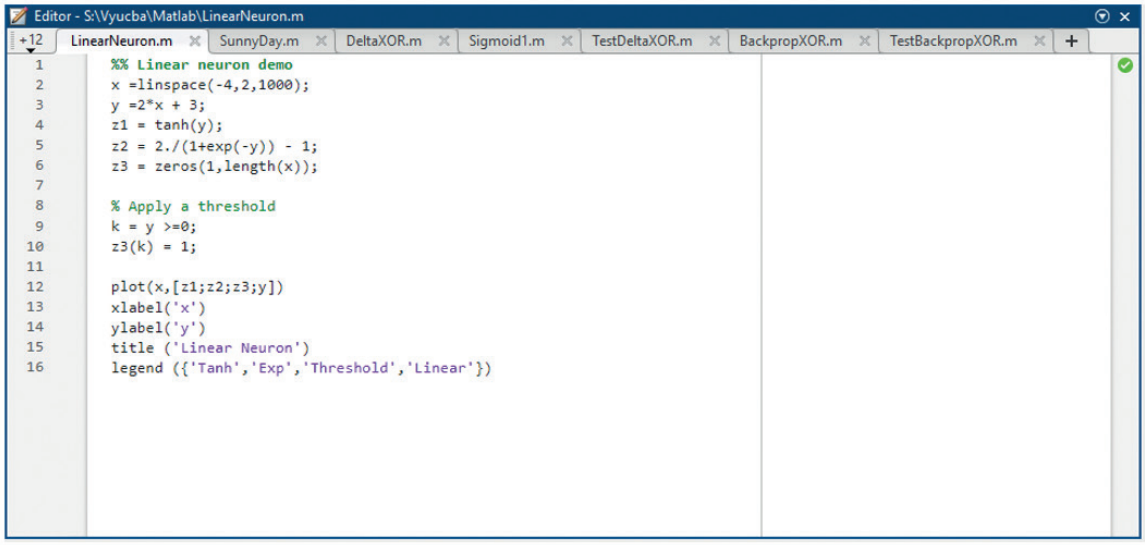


Figura 79. Exemplu de fișier- M-file în fereastra de editare în Matlab

11.2 IMPLEMENTAREA REȚELELOR NEURONALE ÎN MATLAB

În lumea reală, se întâmplă adesea să se poată realiza diverse măsurători, dar să nu se poată descrie comportamentul unui anumit sistem cu un model matematic simplu. Aceasta înseamnă că sunt necesare valorile măsurate ale intrărilor în sistem și ieșirile corespunzătoare, dar nu pot fi calculate ieșirile pe baza intrărilor. În acest scop, se folosesc rețele neuronale pentru a învăța relația dintre intrări (de la un anumit interval de valori) și ieșiri sau pentru a clasifica intrările în anumite grupuri. O rețea neuronală bine antrenată poate produce ieșiri corecte pentru diferite valori de intrare (din același interval).

Această subsecțiune își propune să prezinte **metodologia** de creare a rețelelor neuronale în mediul Matlab. Ulterior, vom rezolva **exemple simple** care să ajute la înțelegerea procesului de creare a rețelei neuronale.

Metodologia de creare a rețelelor neuronale în mediul grafic Matlab

Pașii individuali ai metodologiei sunt următorii:

Pasul 1: Pregătirea datelor. De obicei sunt necesare două seturi de date (setul de intrare și setul de ieșire corespunzător). În cazul eșantionului de date în care o intrare corespunde unei ieșiri așteptate (țintă), atunci ambele seturi de date au aceeași dimensiune, adică 1 rând x numărul de coloane (eșantioane).

Pasul 2: Se selectează o aplicație potrivită din opțiunea APPS din mediul Matlab. De exemplu, din categoria Machine Learning, se lectează aplicația Neutral Net Fitting (a se vedea Figura 80).

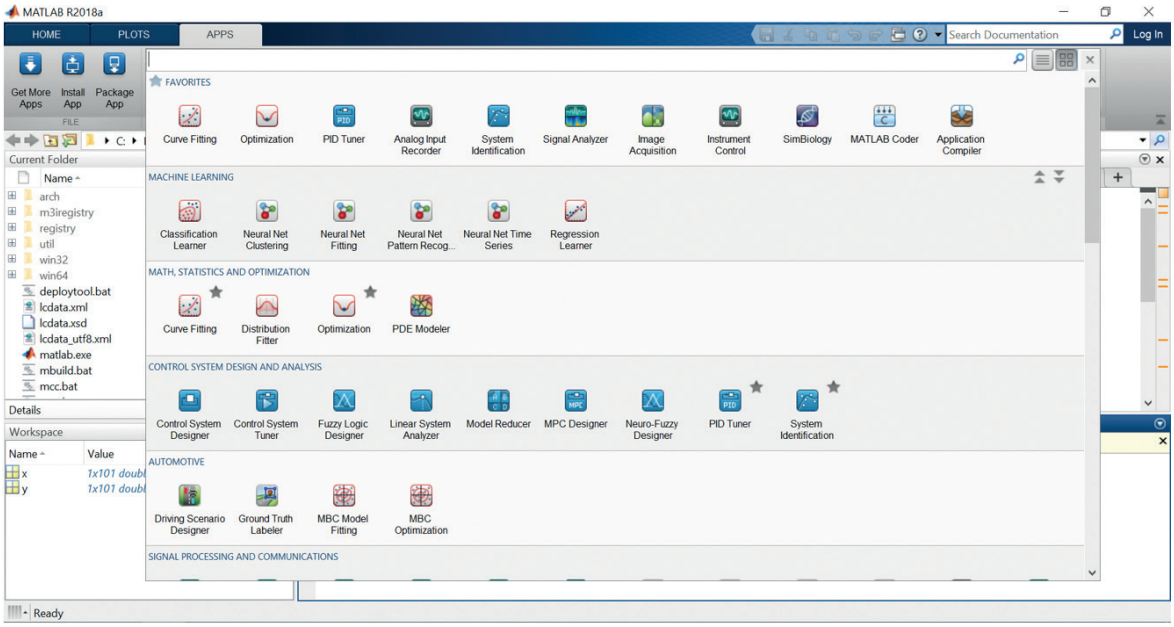


Figura 80. Aplicații care fac parte din mediul Matlab

Pasul 3: Se încarcă datele de intrare din setul de date și datele țintă din setul de date (cele pregătite la Pasul 1).

Pasul 4: Este introdus raportul în care datele trebuie împărțite în trei seturi de antrenament, validare și testare, ca de exemplu, 70%, 15% și 15%.

Pasul 5: Se proiectează arhitectura rețelei. Numărul de intrări și ieșiri din rețea este setat automat în funcție de datele de intrare și țintă. Este necesară setarea numărului de neuroni din straturile ascunse. De exemplu, în cazul în care se proiectează o rețea de perceptron multistrat care are 2 straturi ascunse, trebuie specificat numărul de neuroni pentru primul și al doilea strat ascuns.

Pasul 6: Este selectat algoritmul de învățare. Se alege unul dintre algoritmii de învățare pregătiți, de exemplu, Levenberg - Marquardt, Regularizare Bayesiană sau gradient conjugat la scară.

Pasul 7: Se începe procesul de învățare în rețea. Trebuie remarcat faptul că anumite valori sunt prestabilite în aplicație. De exemplu, numărul de epoci de învățare poate fi setat la 1000, iar acuratețea învățării se exprimă folosind MSE și R. Eroarea pătratică medie (MSE) este diferența pătratică medie dintre rezultatele la ieșirea din rețea și țintele înaintea procesului de instruire. Scopul este de a obține cele mai mici valori ale erorilor. O valoare zero înseamnă nicio eroare. Regresia (R) exprimă corelația măsurată între rezultate și ținte. O valoare R de 1 indică o corelație strânsă, iar 0 înseamnă că nu există corelație sau, cu alte cuvinte, există o relație aleatorie.

Pasul 8: Procesul de învățare în rețea se încheie dacă precizia de învățare obținută este suficientă pentru noi. În caz contrar, trebuie schimbată arhitectura rețelei (numărul de straturi ascunse și numărul de neuroni din ele), sau se modifică algoritmul de învățare, sau numărul de epoci de învățare ale rețelei (dacă este posibil). Aceasta înseamnă că se repetă procedura de la pasul 5. Trebuie avut în vedere că un număr mare de epoci de învățare poate duce la așa-numita reînvățare în rețea.

Exemplu de creare simplă a funcției de rețea neuronală

În acest exemplu, va fi arătat cum o rețea neuronală învață valoarea unei funcții. Va fi urmată metodologia prezentată în secțiunea 2.1 a acestui capitol.

Pasul 1: Pentru simplitate, nu vor fi utilizate datele măsurate, însă vor fi create datele de intrare și de ieșire în Matlab. Folosind comenzi din Matlab, vor fi create două seturi de date (a se vedea codul de mai jos). Primul set de date este denumit data1.mat și conține valori ale intrărilor, iar al doilea set de date este denumit data2.mat și conține valori ale țintelor, adică ieșirile așteptate după învățarea rețelei. Elementele de intrare și țintele sunt aranjate într-o astfel de ordine astfel încât elementele de intrare individuale corespund elementelor țintă respective [2].

```
>> x=0:0.1:10
>> y=2*x.^3
>> plot(x,y)
>> save data1 x
>> save data2 y
```

După rularea comenzilor din Lista 1, sunt scrise valorile vectorilor x și y, graficul funcției este trasat (vezi Figura 81) iar seturile de date sunt salvate în fișiere.

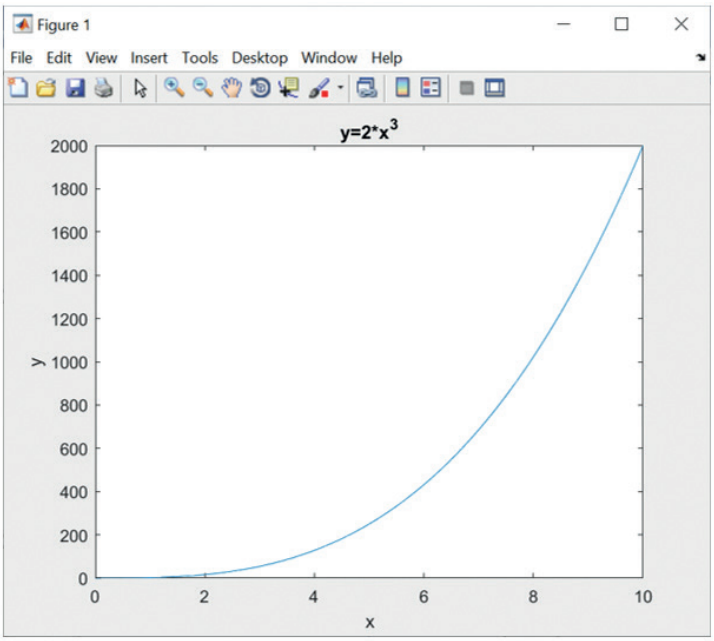


Figure 85. Graficul funcției $y= 2x^3$

Pasul 2: În mediul Matlab, alegem o aplicație potrivită din APPS. Din Machine Learning, se alege aplicația Neutral Net Fitting și practic se începe utilizarea aplicației Neutral Net Fitting (vezi Figura 82). Utilizând butonul următor, se navighează în aplicație.

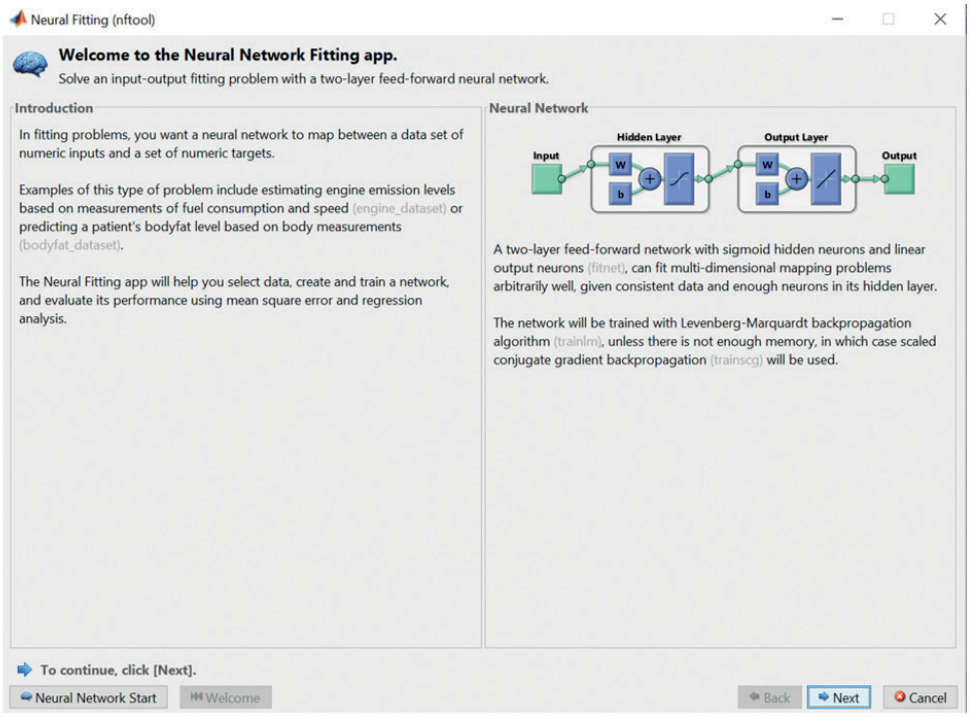


Figura 82. Neural Net Fitting în Matlab

Pasul 3: Se încarcă seturile de date de intrare și de date țintă din fișierele pregătite (a se vedea Figura 83, din stânga). Deoarece numărul de intrări și ținte este același, se verifică dacă fișierele au aceeași dimensiune (a se vedea Figura 83, dreapta).

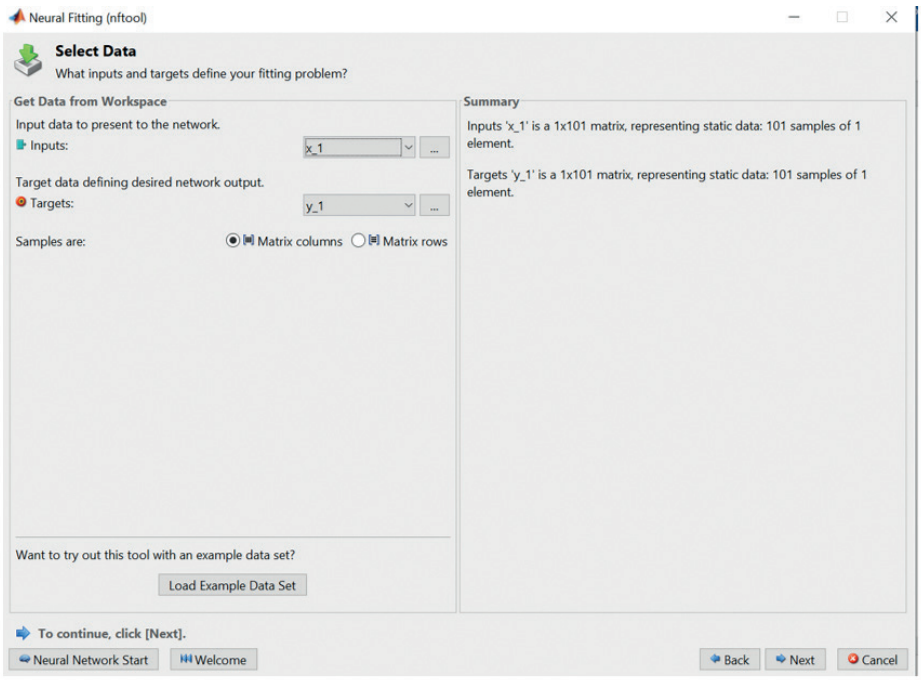


Figura 83. Metoda de încărcare a valorilor de intrare și țintă

Pasul 4: Se introduce raportul în care trebuie împărțite datele în cele trei seturi: pentru antrenament, validare și testare. În cazul nostru, s-a ales 70% din date pentru instruirea rețelei, 15% pentru validare în cadrul procesului de învățare al rețelei și 15% pentru testare (a se vedea Figura 84).

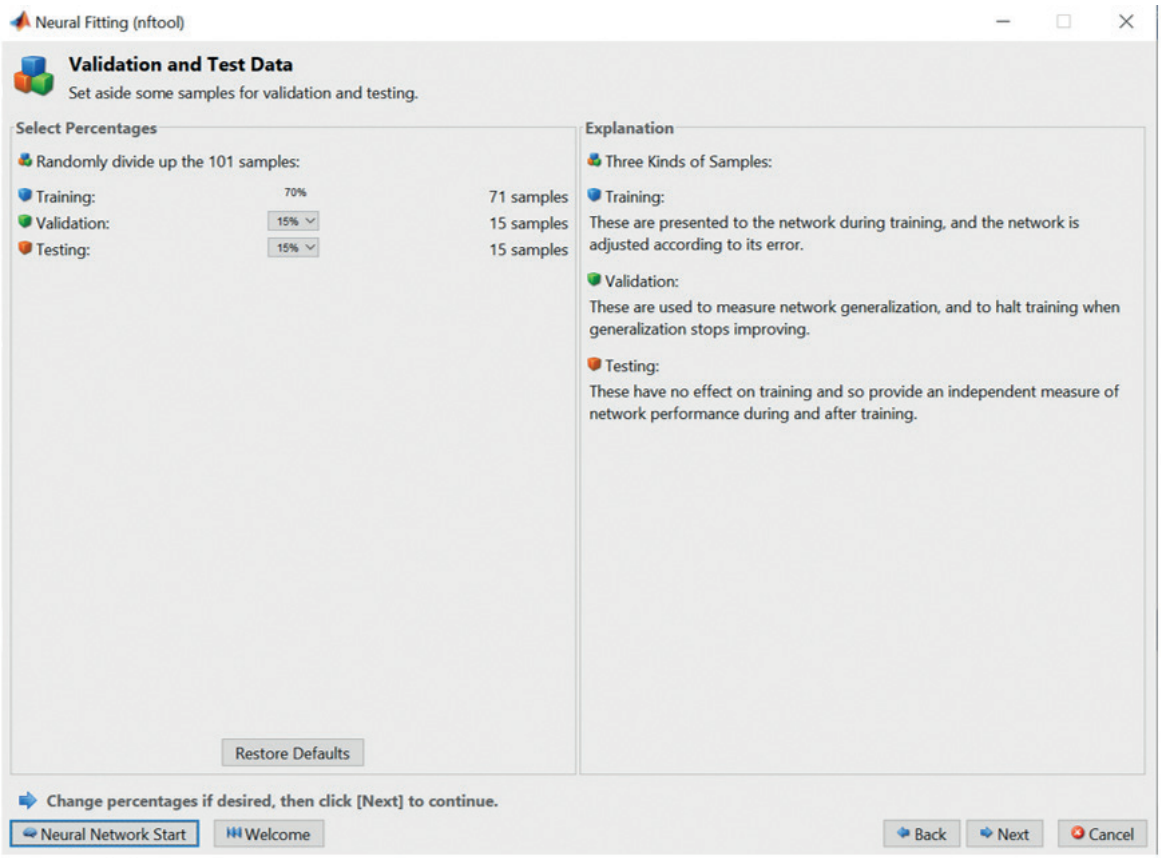


Figure 84. Metoda de distribuire a eșantioanelor de date în sarcina rezolvată

Pasul 5: În continuare se proiectează arhitectura rețelei. Numărul de intrări și ieșiri ale rețelei a fost setat automat în funcție de datele de intrare și de ieșire, astfel încât rețeaua aleasă de noi să aibă o intrare și o țintă (a se vedea Figura 85). Stratul de ieșire are un singur neuron, deoarece avem o ieșire din rețea. De asemenea, numărul de neuroni din stratul de ieșire este setat automat. În cazul descris, există un singur strat ascuns deoarece este utilizată aplicația Neutral Net Fitting. Se setează numărul de neuroni din stratul ascuns la 100. Dacă rețeaua nu învață relația dintre intrări și ținte suficient de precis, se poate reveni la setarea arhitecturii rețelei și schimba numărul de neuroni ascunși.

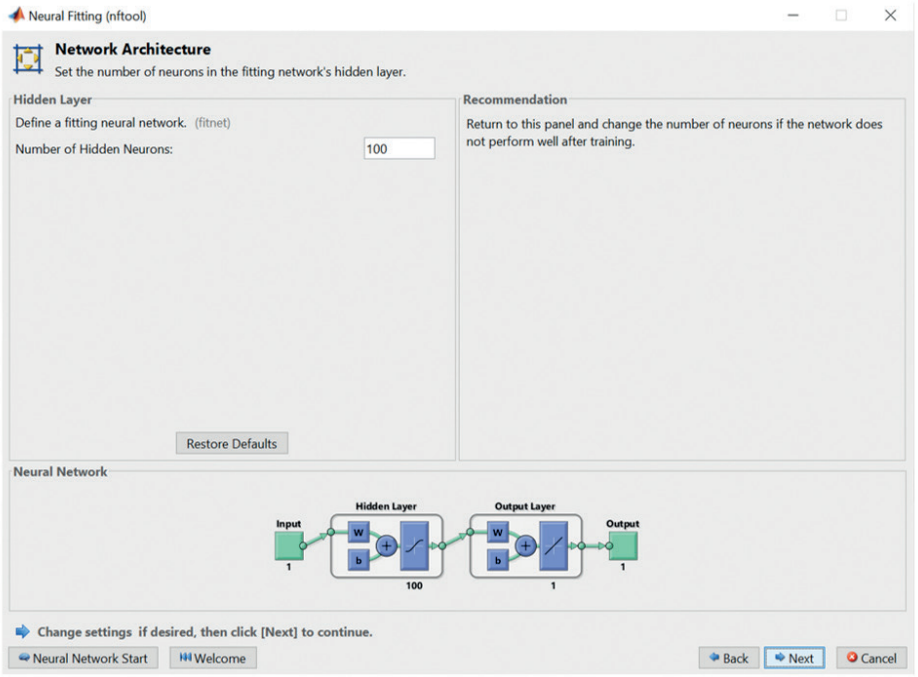


Figure 85. Proiectarea arhitecturii de rețea pentru sarcina aleasă de noi

Pasul 6: Selectarea algoritmului de învățare. Este ales unul dintre cei trei algoritmi de învățare: Levenberg – Marquardt, Regularizare Bayesiană sau Gradient conjugat la scară. În cazul de față este ales algoritmul de regularizare Bayesian (a se vedea Figura 86).

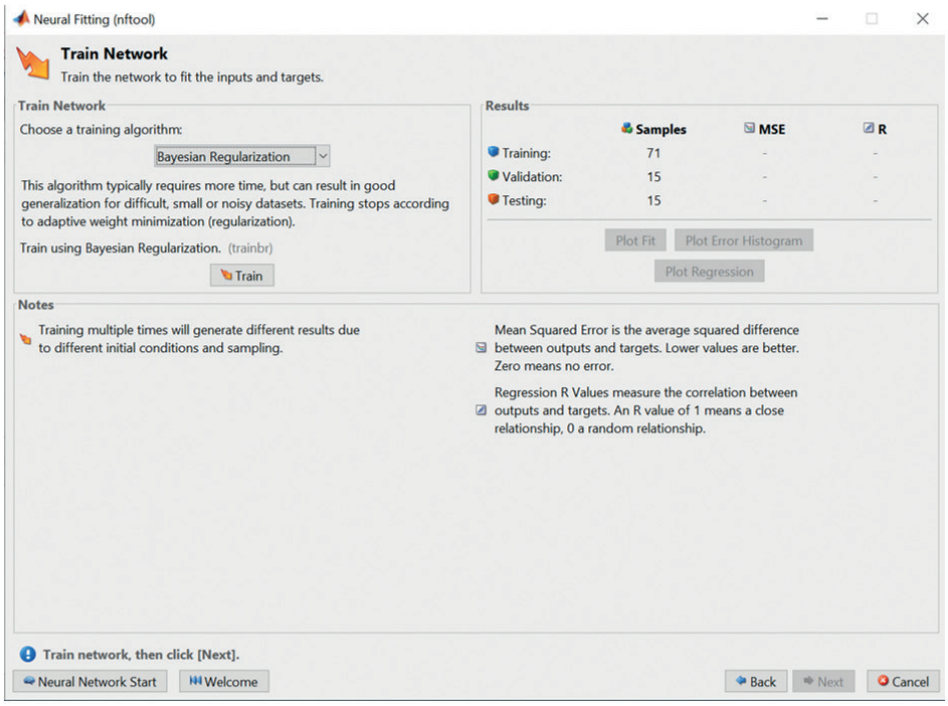


Figura 86. Selectarea algoritmului de învățare

Pasul 7: Procesul de învățare a rețelei începe prin apăsarea butonului Antrenează. Numărul de epoci de învățare se setează la 1000 și astfel se urmărește procesul de învățare după cum este aratat în Figura 87.

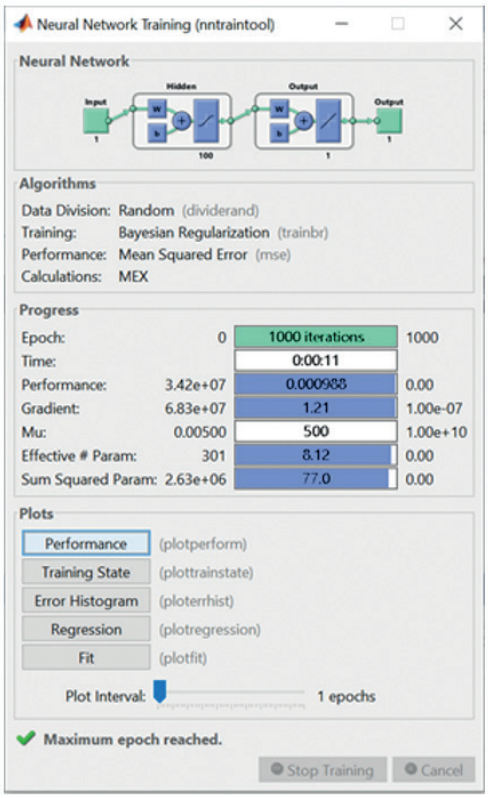


Figure 87. Progresul învățării în rețea

Acuratețea învățării este exprimată folosind MSE și R (a se vedea Figura 88). Eroarea medie pătratică (MSE) este diferența pătratelor mediilor ieșirilor din rețea și obiectivele înaintea procesului de instruire. Scopul este de a obține cele mai mici valori ale erorilor. O valoare zero înseamnă nicio eroare. Se poate observa că rețeaua a învățat caracteristica noastră cu eroarea MSE $8{,}95*10^{-7}$, care este o eroare nesemnificativă. Testarea rețelei a confirmat că rețeaua a învățat corect cu o eroare MSE scăzută de $3{,}14*10^{-5}$. Regresia (R) exprimă corelația măsurată între rezultate și ținte. O valoare R de 1 înseamnă o corelație strânsă, iar 0 înseamnă că nu există o corelație sau există o relație aleatorie. Calculul corelațiilor după antrenamentul rețelei și testarea rețelei a ajuns la valoarea 1, confirmând că rețeaua a învățat foarte bine relația dintre intrările și ieșirile din rețea.

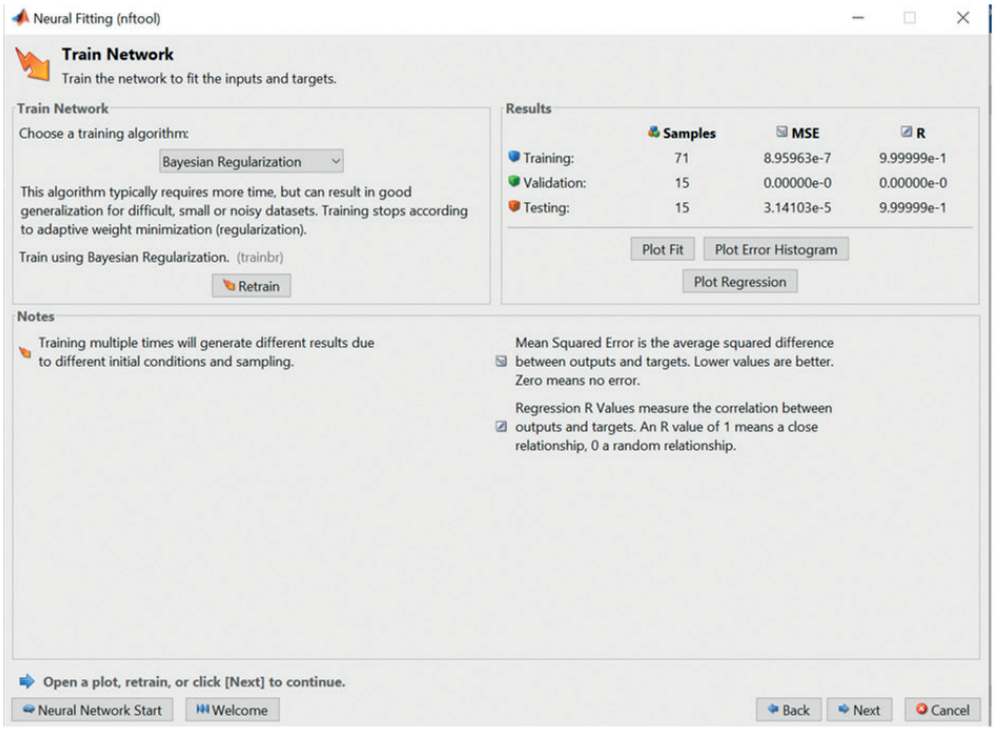


Figura 88. Erorile și corelațiile după procesul de învățare și testare a rețelei

Pasul 8: Se poate concluziona că acuratețea obținută în învățarea și testarea rețelei este suficientă, iar procesul de învățare al rețelei se încheie. Prin urmare, pot fi observate valorile și progresul funcției învățate în detaliu dacă se apasă succesiv butoanele Plot Fit, Plot Error Histogram și Plot Regression (a se vedea Figura 88). După apăsarea Plot Fit, se cunoaște evoluția valorilor țintelor și ieșirilor rețelei în funcție de intrările după învățarea și testarea rețelei (a se vedea Figura 89).

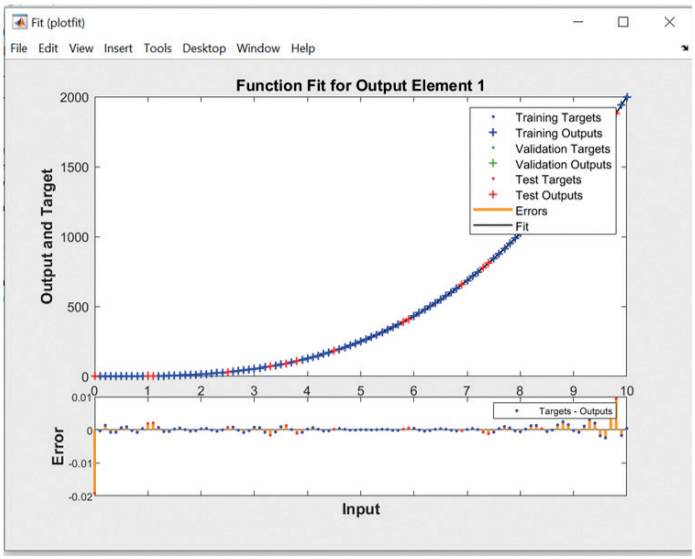


Figura 89. Evoluția obiectivelor și ieșirilor rețelei în funcție de intrările după procesul de învățare și respectiv de testare a rețelei

După apăsarea **Plot Error Histogram**, se observă valorile de eroare și frecvența acestora (vezi Figura 90). În acest caz, eroarea absolută este diferența dintre valoarea țintă și ieșirea rețelei despre o anumită intrare în rețea. Putem vedea că eroarea 0.000119 a apărut cel mai des.

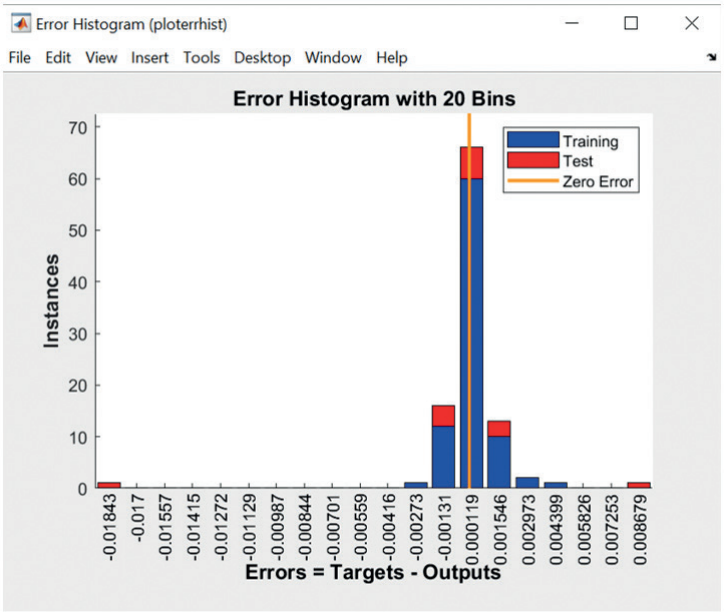


Figura 90. Valorile erorilor absolute și frecvența acestora

După apăsarea butonului **Plot Regression**, se văd valorile de corelație dintre valorile **țintă** și cele de ieșire în procesul de antrenament, procesul de testare și ambele procese (vezi Figura 91). Calculul corelațiilor după antrenamentul rețelei și testarea rețelei a ajuns la valoarea de 1, confirmând că rețeaua a învățat foarte bine relația dintre intrările și ieșirile din rețea.

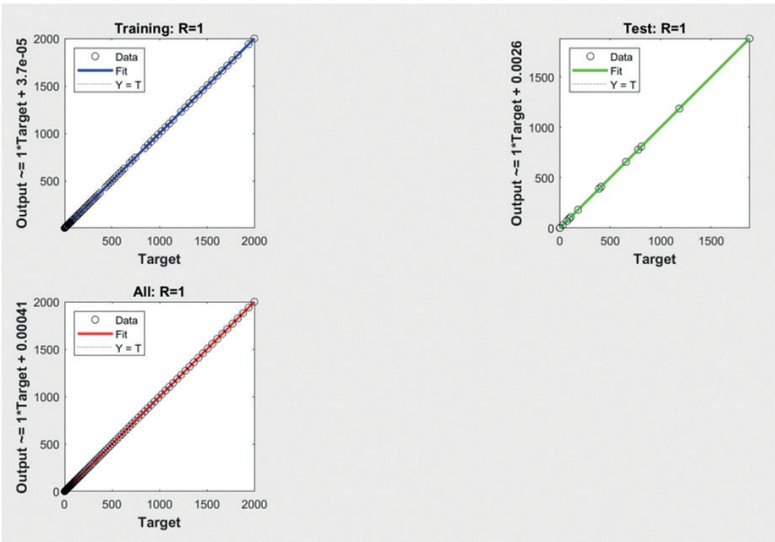


Figure 91. Corelațiile între valorile țintelor și valorile de ieșire în procesul de instruire, procesul de testare și ambele procese

Exemple de funcție de ajustare a rețelei neuronale prin crearea de valori măsurate

Acest exemplu își propune să învețe rețeaua neuronală valorile obținute prin măsurare. Va fi urmată metodologia prezentată în secțiunea 2.1 a acestui capitol.

Pasul 1: Pregătirea datelor. Sunt 500 de valori ale datelor măsurate care sunt stocate în fișierul data4 (a se vedea Figura 92). Pentru claritate, se trasează axa x de la valoarea 0,1 la valoarea 50, cu un pas de 0,1. A se vedea codul de mai jos:

```
x=0.1:0.1:50
save data3 x
load data4 y
plot(x,y)
```

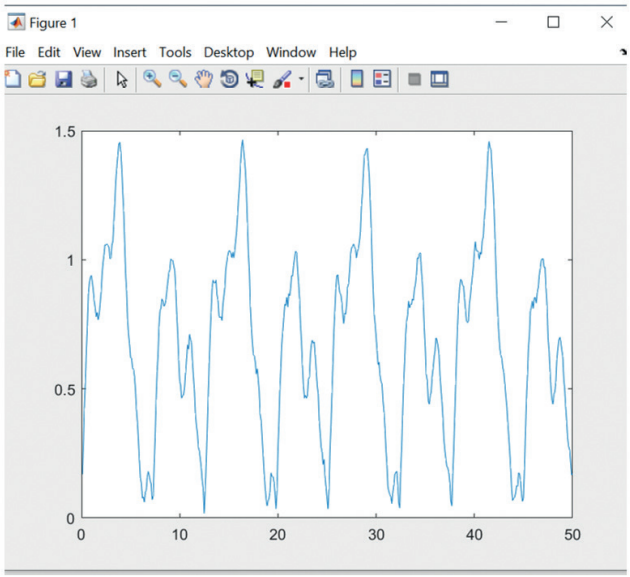


Figura 92. Valorile măsurate

Uneori, datele măsurate au deviații și, prin urmare, trebuie ajustate [1]. Se utilizează în acest scop un instrument mobil care netezește datele, astfel încât rețeaua neuronală să le poată învăța. Media mobilă trece treptat prin toate valorile netezite și înlocuiește valoarea curentă cu valoarea medie. Fereastra constă din valoarea curentă și un anumit număr de valori înainte și după valoarea curentă. Acum va fi arătat cum se netezesc datele măsurate stocate în fișierul data4.mat. Va fi extinsă lista 2 cu comenzi prin netezirea datelor măsurate folosind o medie mobilă cu o fereastră care are nouă valori lățime. Datele modificate vor fi salvate în vectorul m în fișierul data5.mat și folosite ulterior ca ținte atunci când este învățată rețeaua (a se vedea codul de mai jos) și Figura 93.

```
x=0.1:0.1:50
save data3 x
load data4 y
plot(x,y)
m = movmean(y,7)
plot(x,y,x,m)
save data5 m
```

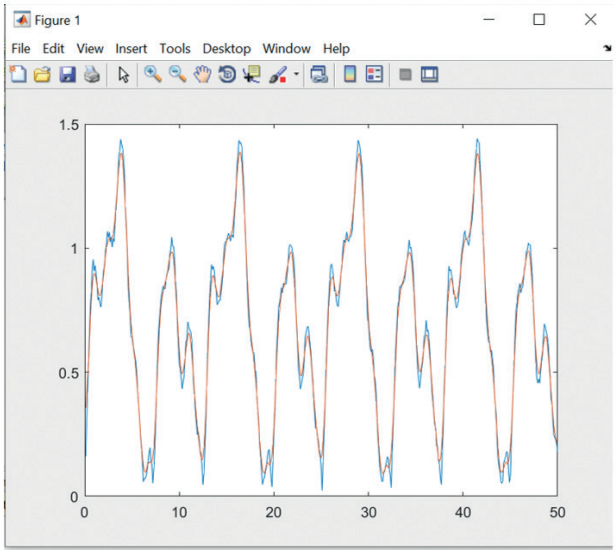


Figura 93. Valorile măsurate sunt indicate cu albastru şi datele netezite sunt colorate cu roşu

Pasul 2: Se selectează aplicația Neutral Net Fitting în mediul Matlab.

Pasul 3: Se face încărcarea datelor în aplicație. Fișierul data3.mat conține datele de intrare, iar fișierul data5.mat conține ținte.

Pasul 4: Se păstrează raportul ca în exemplul anterior.

Pasul 5: Se va proiecta arhitectura rețelei. Se aleg 50 de neuroni în stratul ascuns.

Pasul 6: Este ales algoritmul de învățare Bayesian Regularization (Regularizare Bayesiană).

Pasul 7: Se începe procesul de învățare (a se vedea Figura 94).

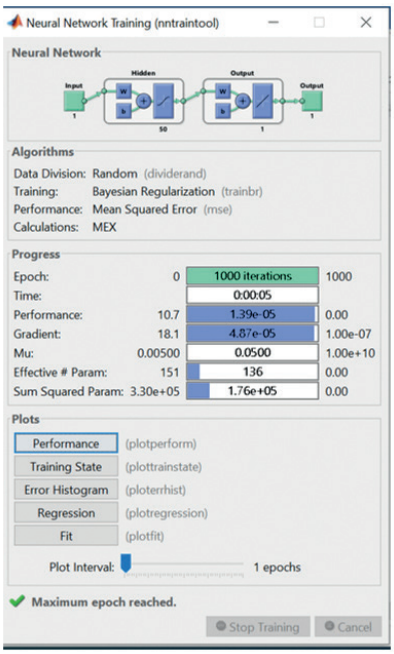


Figura 94. Progresul învățării rețelei

Pasul 8: Aruncând o privire mai atentă la rezultatele învățării rețelei (Figura 95), concluzionăm că rețeaua a învățat valorile corecte pe baza erorii de învățare a $1{,}39x10^{-5}$ și eroarea de testare a rețelei este de $1{,}39x10^{-5}$.

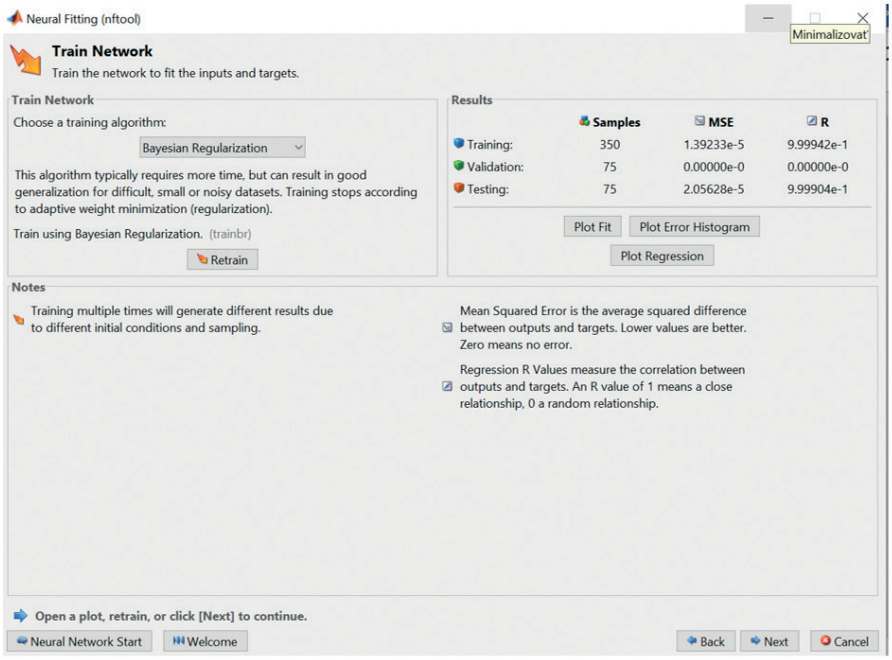


Figura 95. Afișarea erorilor de învățare

În Figura 96, se observă progresul valorilor țintelor și ieșirilor rețelei în funcție de intrările după învățarea și testarea rețelei. Valorile învățate se suprapun cu valorile țintă. În partea de jos a imaginii, se văd erorile afișate, care sunt minime.

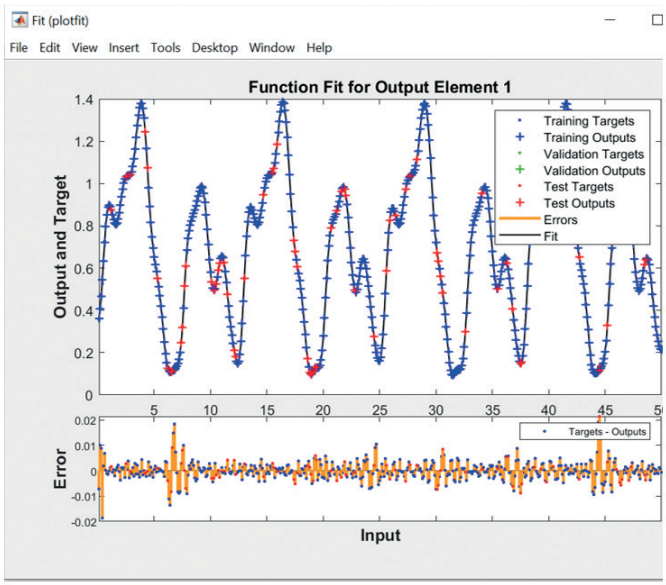


Figura 96. Rezultatele învățării în rețea – obiective (țințe) de progres și valori învățate.

Pe histogramă, în Figura 97, se poate observa dimensiunea și frecvența erorilor, ceea ce demonstrează din nou că erorile sunt minime.

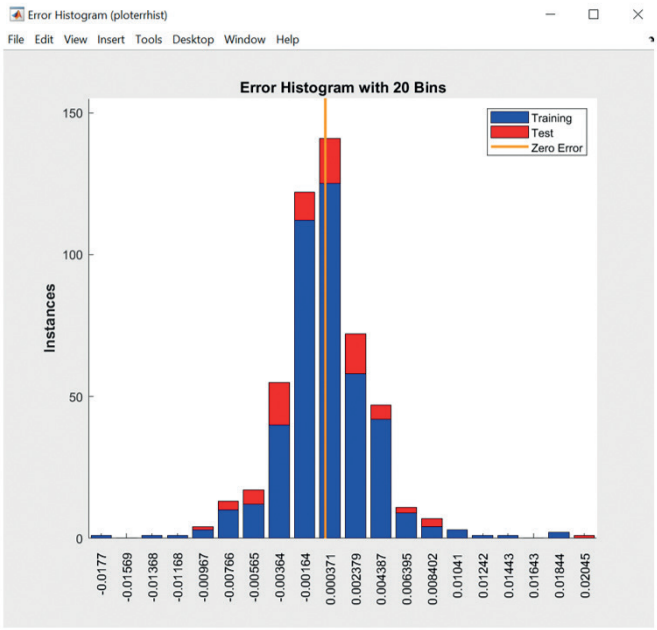


Figura 97. Rezultatele învățării în rețea - histograma - dimensiunea și frecvența erorilor

Regresiile prezentate (a se vedea Figura 98) care au atins valoarea de 1, din punct de vedere al antrenamentului, testării, și toate intrările indică faptul că avem o rețea foarte bine pregătită.

În general, concluzionăm că acuratețea obținută în învățarea și testarea rețelei este suficientă, iar procesul de învățare al rețelei se încheie.

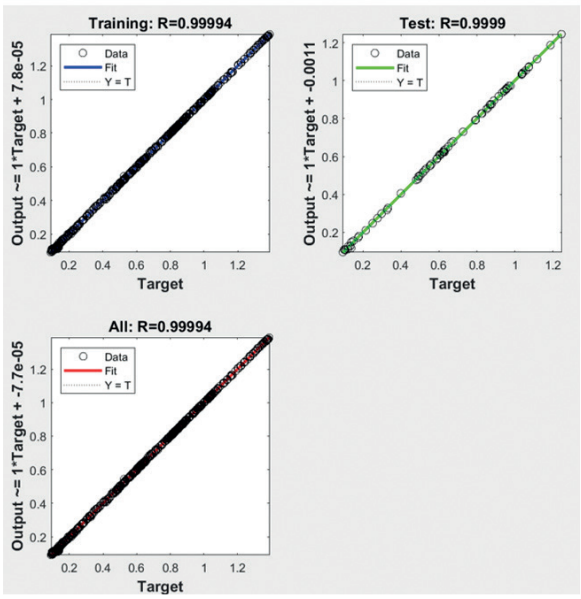


Figura 98. Rezultatele învățării în rețea - calculul regresiei

Exemplu de grupare a datelor prin intermediul rețelei neuronale cu hărți auto-organizate

În acest exemplu, este rezolvată binecunoscuta sarcină de clasificare a florilor de iris. Va fi utilizat setul de date IRIS și va fi prezentat un exemplu rezolvat, care face parte din mediul Matlab. Florile de iris sunt descrise cu ajutorul a 4 parametri, în timp ce valorile (lungimea sepalei, lățimea sepalei, lungimea petalei și lățimea petalei) din setul de date sunt date în centimetri. Prin urmare, fiecare floare este caracterizată cu ajutorul a 4 elemente. Sarcina noastră este de a crea o rețea neuronală cu hartă auto-organizată care să clasifice tipurile de flori de iris în clase, astfel încât tipurile similare să fie situate într-un grup apropiat unul de celălalt. Harta este creată pe baza asemănării mostrelor, iar rețeaua neuronală învățată poate clasifica chiar și eșantioane necunoscute [4].

Se procedează conform metodologiei de implementare a rețelei neuronale în mediul grafic Matlab.

Pasul 1: Se sare peste acest pas. Se utilizează un set de date disponibil în Matlab. Acest set de date va fi descris în detaliu la pasul 3.

Pasul 2: În mediul Matlab, se selectează aplicația corespunzătoare din fila APPS, din categoria Machine Learning, se selectează aplicația Neutral Net Clustering și apoi se lansează aplicația. Această aplicație ajută la crearea unei rețele neuronale cu hărți cu auto-organizare. A se vedea în acest sens Figura 99.

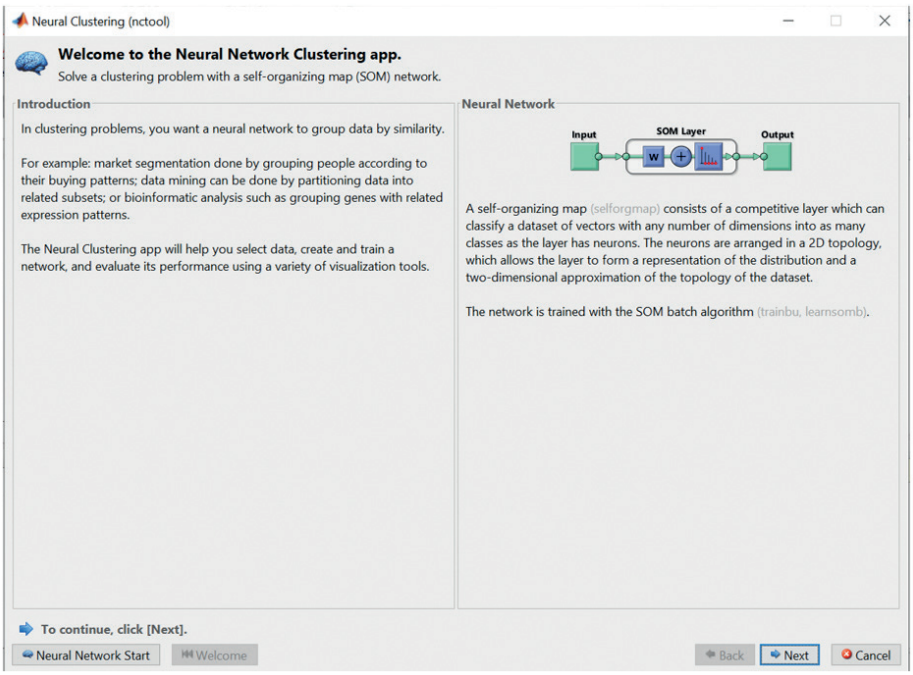


Figura 99. Aplicația pentru clusterizarea rețelelor neuronale

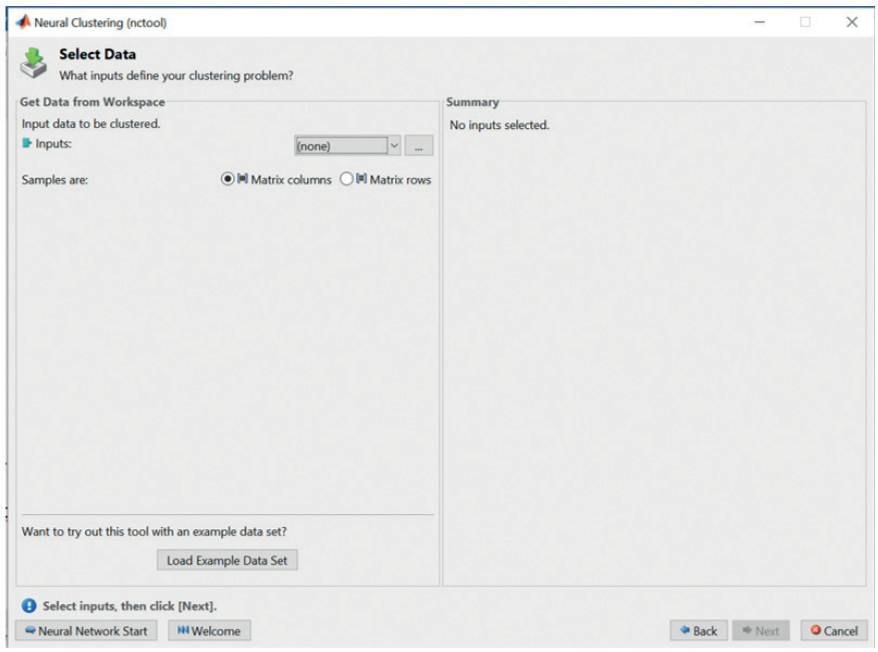


Figura 100. Încărcarea setului de date pregătit

Pasul 3: Se încarcă seturile de date. A se vedea Figura 100. Din moment ce se dorește folosirea setului de date pregătit IRIS (a se vedea Anexa A), se apasă Încărcare exemplu set de date (Load Example Data Set), se selectează Flori de iris (Iris Flowers) și se importă datele, așa cum se poate vedea în Figura 101.

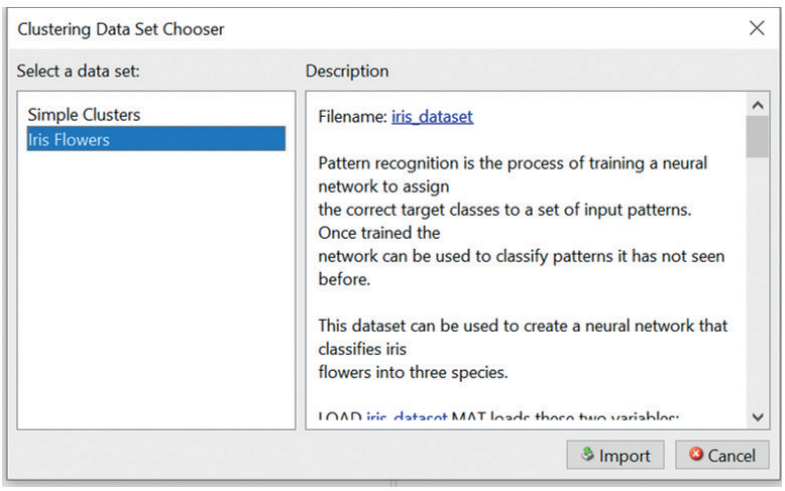


Figura 101. Importul setului de date al florilor de iris

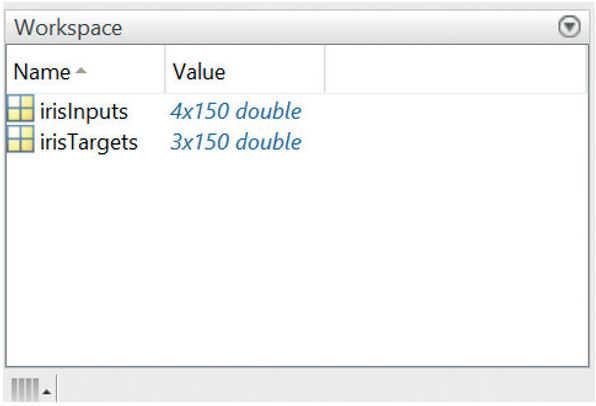


Figura 102. Spațiul de lucru Matlab după încărcarea setului de date Iris flowers.

După încărcarea setului de date, matricele irisInputs și irisTargets sunt afișate în fereastra spațiului de lucru Matlab, a se vedea Figura 102. Se observă că matricea irisInputs are dimensiunea de 4 linii x 150 coloane. Patru parametri descriu o floare, prin urmare o coloană reprezintă o probă de floare. Setul de date de intrare conține 150 de mostre de flori. Matricea irisTargets specifică clasificarea fiecărui eșantion de intrare într-una din cele 3 clase. Ambele matrici se scriu în fereastra de comandă în mod individual pentru a înțelege mai bine datele. Mostre de date pot fi văzute în codurile de mai jos:

```
>> irisInputs

irisInputs =

    Columns 1 through 11

        5.1000 4.9000 4.7000 4.6000 5.0000 5.4000 4.6000 5.0000 4.4000 4.9000 5.4000
        3.5000 3.0000 3.2000 3.1000 3.6000 3.9000 3.4000 3.4000 2.9000 3.1000 3.7000
        1.4000 1.4000 1.3000 1.5000 1.4000 1.7000 1.4000 1.5000 1.4000 1.5000 1.5000
        0.2000 0.2000 0.2000 0.2000 0.2000 0.4000 0.3000 0.2000 0.2000 0.1000 0.2000

>> irisInputs

irisTargets =

Columns 1 through 18

    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 19 through 36

    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Columns 37 through 54

1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Pasul 4: Împărțirea eșantioanelor în două categorii de instruire și respective testare este prestabilită și, prin urmare, nu este necesar să fie introduse în această aplicație.

Pasul 5: Proiectarea arhitecturii rețelei. În acest caz, trebuie introdus numărul de neuroni ai stratului SOM. Un strat reprezintă o matrice pătrată bidimensională. Prin urmare, atunci când este specificată o dimensiune a matricei de 12, practic este creată o matrice bidimensională de 12x12 elemente. A se vedea Figura 103. Apoi harta de la ieșirea rețelei va fi 12x12, adică 144 de elemente. Topologia hărții de ieșire predefinită este hexagonală.

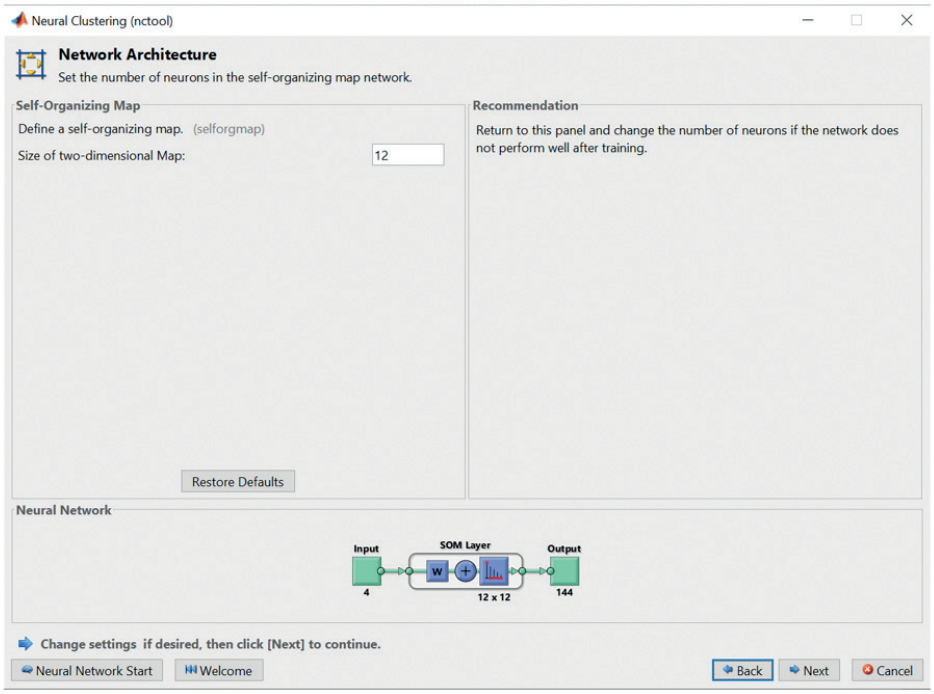


Figura 103. Proiectarea arhitecturii rețelei

Pasul 6: Selectarea algoritmului de învățare. Aplicația are un algoritm SOM batch predefinit, așa că va fi omis acest pas.

Pasul 7: Se începe procesul de învățare în rețea prin apăsarea butonului Antrenează. Numărul de epoci de învățare este setat la 200 și astfel se poate urmări procesul de învățare după cum este arătat în Figura 104.

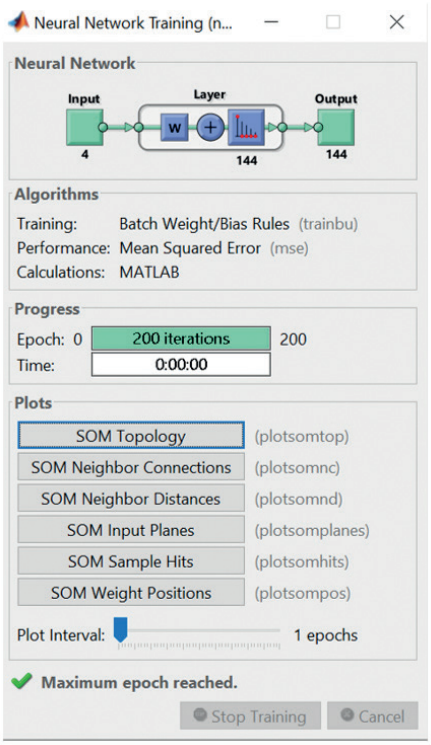


Figura 104. Progresul învățării în rețea

Pasul 8: Afișarea rezultatelor învățării rețelei se realizează folosind patru butoane (a se vedea Figura 105). Primul rezultat reprezintă clasificarea claselor pentru fiecare floare, iar Plot som hits eșantion (a se vedea Figura 106) arată numărul de flori din fiecare clasă. Zonele neuronilor cu valori mai mari reprezintă clase de flori reprezentate frecvent în mod similar. Dimpotrivă, zonele cu valori mici indică flori mai puțin abundente.

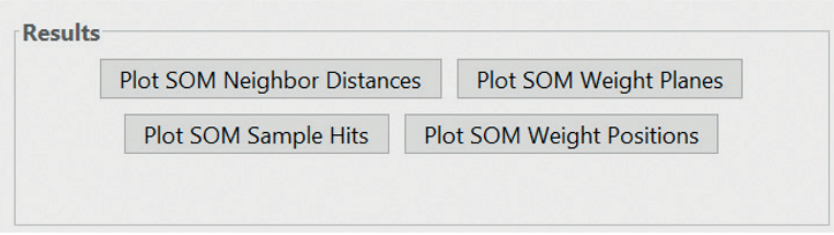


Figura 105. Rezultatele învățării în rețea

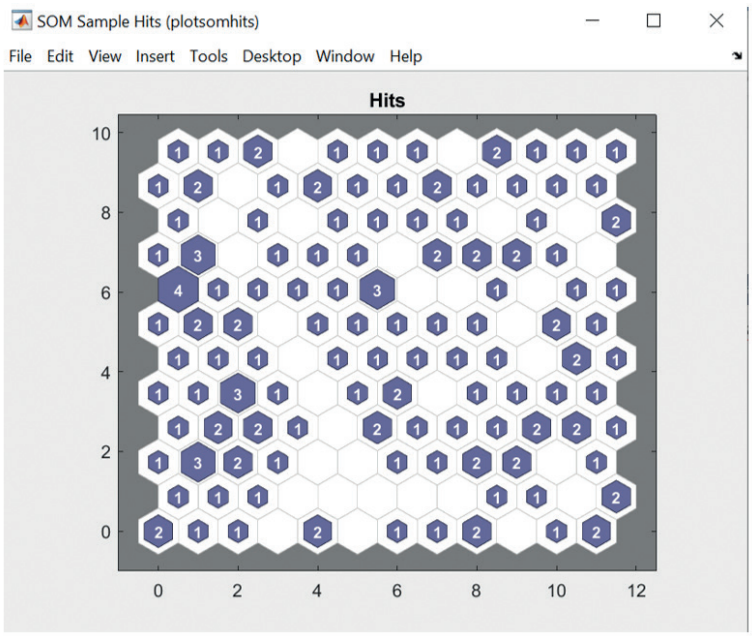


Figura 106. Rezultatele învățării în rețea - numărul de flori din fiecare clasă

Rezultatul afișat folosind Plot SOM Neighbor Distances exprimă distanța euclidiană a unei clase de neuroni față de vecinii săi. Grupurile de neuroni care formează conexiuni luminoase înseamnă o asemănare mare a florilor în setul de intrare. În schimb, conexiunile întuneric-lumină reprezintă zone cultivate cu mai puține flori sau zone fără flori. A se vedea Figura 107. Marginile întunecate (articulațiile) separă zone mari din spațiul de intrare și indică faptul că florile din zone separate au caracteristici diferite.

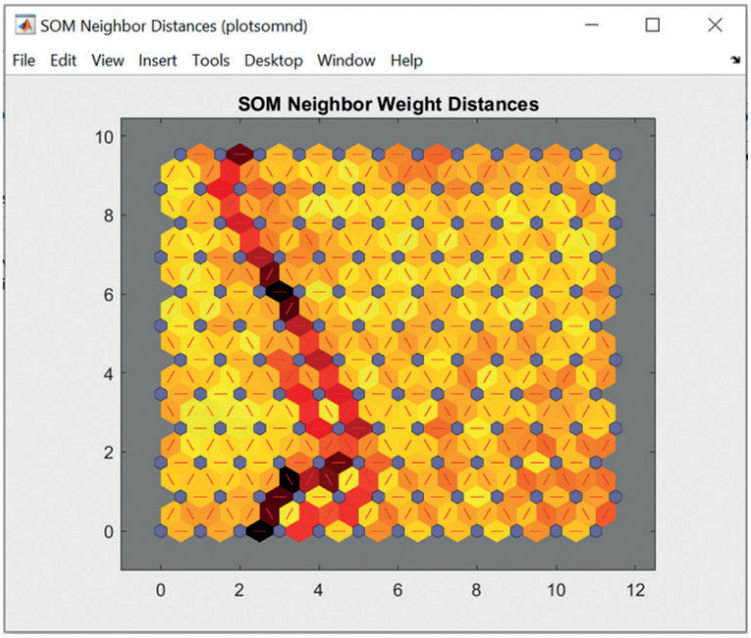


Figura 107. Rezultatele învățării în rețea - abundența și clasele de flori în spațiul de intrare

Greutățile rezultate ale rețelei din punct de vedere al celor patru caracteristici de intrare ale florilor vor fi afișate folosind planuri de greutate Plot SOM ca în Figura 108.

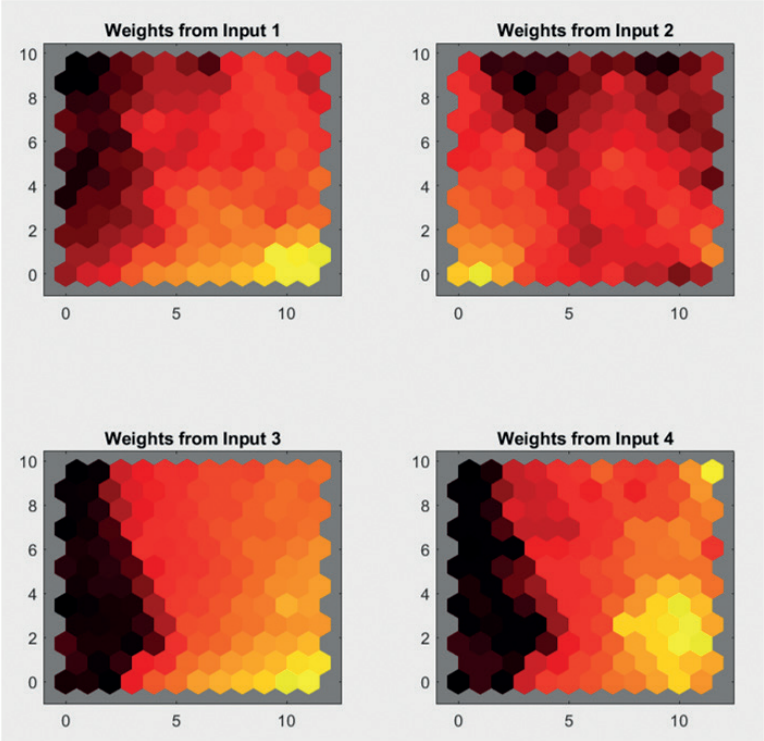


Figura 108. Rezultatele învățării în rețea - hărți de greutate pentru intrări individuale în rețea

Greutățile conectează fiecare intrare la fiecare dintre cei 144 de neuroni de ieșire ai rețelei. Culorile închise reprezintă greutăți mai mari. Intrările care au aceeași culoare pe hartă sunt strâns corelate.

CAPITOLUL 12

ANEXE

Această secțiune conține anexe la conținutul prezentat în corpul principal al manualului. Cinci anexe conțin diverse date și exerciții legate de materialul din acest manual, și anume:

- **Anexa A** descrie setul de date Iris folosit în exemplele din secțiunile 3 – 8.
- **Anexa B** conține exemple de soluții la problemele prezentate în secțiunea 7.
- **Anexa C** se axează pe prezentarea seturilor de date legate de poluarea aerului și schimbările climatice, folosite ca surse pentru analiza datelor.
- **Anexa D** descrie impactul poluării aerului asupra sănătății umane.
- **Anexa E** conține curriculumul propus pentru cursul la care poate fi folosit prezentul manual.

A-SCURTĂ DESCRIERE A SETULUI DE DATE IRIS

Această parte a manualului a fost scrisă de Alžbeta Michalíková și Adam Dudáš de la Departamentul de Informatică, Facultatea de Științele Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

Setul de date Iris este unul dintre cele mai folosite seturi de date în analiza datelor, lucrând cu modele de predicție și reglând algoritmi pentru procesarea datelor.

Edger Andersen a creat acest set de date, prezentat pentru prima dată în contextul analizei datelor în publicația Fisher, R.A. *"Folosirea unor măsurători multiple în problemele taxonomice", Annual Eugenics, 1936.* Setul de date are **cinci atribute** măsurate de la peste **150 unități** ale florii de Iris (setul de date are dimensiunea 150×5). Floarea are șase elemente (petale și sepale) structurate în două cicluri, unde:

- sepalele formează ciclul interior al florii,
- petalele formează ciclul exterior al florii.

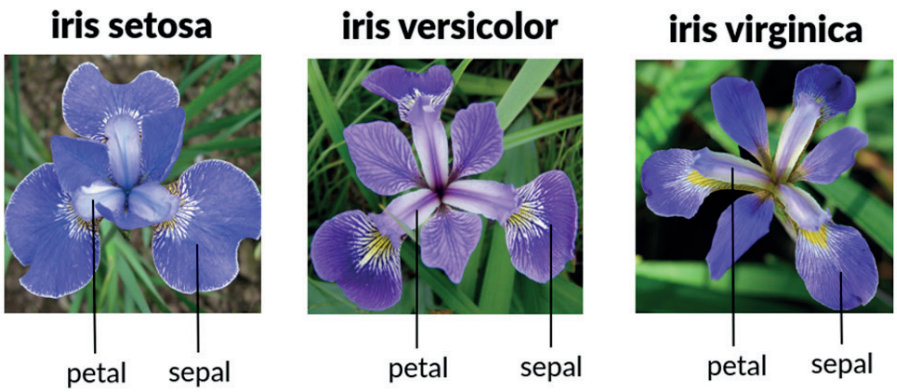


Figura A1. Setul de date Iris

Setul de date Iris se compune prin măsurarea a două valori pentru fiecare tip de petală sau sepală (lățime și lungime), ceea ce creează **patru atribute numerice**:

- lungimea și lățimea sepalei măsurate în centimetri sau milimetri,
- lungimea și lățimea petalei măsurate în centimetri sau milimetri.

Al cincilea atribut al setului de date este valoarea categorică a **clasei** sau uneori **speciei**, care împarte entitățile setului de date în trei clase:

- iris setosa,
- iris versicolor,
- iris virginica.

Fiecare dintre aceste clase este reprezentată în setul de date în mod egal - de **50 de entități**. Mai jos, prezentăm un exemplu al unei entități eșantion din fiecare clasă a setului de date Iris:

Tabelul A.1 Exemplu entități eșantion

Entitatea	Lungimea sepalei	Lățimea sepalei	Lungimea petalei	Lățimea petalei	Clasa
1	5.1	3.5	1.4	0.2	setosa
2	7.0	3.2	4.7	1.7	versicolor
3	6.3	3.3	6.0	2.5	virginica

Lucrul cu setul de date Iris

Setul de date Iris este atât de standardizat încât majoritatea instrumentelor de procesare și analiza datelor au o comandă internă care poate fi utilizată pentru încărcarea acestui set de date.

De exemplu, în limbajul R, în locul numelui fișierului de date, folosim doar *iris*.

Exemplu: Tastând numele setului de date încărcat în limbajul R, obținem ieșirea pe consolă cu toate atributele și entitățile setului de date. În cazul setului de date iris, putem tasta iris (fără a încărca setul de date).

```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
```

În cazul când se lucrează cu un instrument care nu are un set de date iris disponibil în acest mod, există posibilitatea descărcării gratuite a setului de date, de exemplu de pe:

https: //archive.ics.uci.edu/ml/datasets/iris

B-SOLUȚII LA ÎNTREBĂRILE CLASIFICĂRII FUZZY

Această parte a manualului a fost scrisă de Alžbeta Michalíková de la Departamentul de Informatică, Facultatea de Științele Naturii, Universitatea Matej Bel din Banská Bystrica, Slovacia.

Folosind metoda Sugeno, se clasifică datele din setul de date Iris într-un număr rezonabil de clase.

Soluția:

Trebuie să se răspundă la următoarele întrebări:

1. Câte **variabile de intrare** sunt în setul de date Iris?
Sunt patru variabile de intrare în setul de date IRIS.

2. Ce funcții pot fi utilizate pentru **a descrie variabilele de intrare**?
Pentru a descrie variabilele de intrare, vor fi folosite funcțiile apartenenței fuzzy.

3. Ce tip de **funcții de apartenență** fuzzy vor fi folosite?
Vor fi folosite funcțiile de apartenență trapezoidală.

4. Care va fi **rezultatul**?
Rezultatul va fi clasa specifică din care fac parte florile de Iris individuale (rânduri ale tabelului / obiecte).

5. Ce funcții pot fi folosite **pentru a descrie variabilele de ieșire**?
Pentru a descrie variabilele de ieșire, vor fi folosite funcțiile constante (constantele).

6. **Ce tip de reguli** vor fi folosite?
Vor fi folosite regulile Sugeno IF-THEN (DACĂ-ATUNCI).

7. **Să se scrie un exemplu** de astfel de regulă!
Dacă valorile pentru Intrarea 1 și Intrarea 2 sunt mici, Intrarea 3 este mijlocie, iar Intrarea 4 este ridicată, atunci Ieșirea este clasa 1 (sau Iris_Setosa).

Să se determine parametri variabilelor de intrare din aceste date și să se completeze în tabelele următoare.

Tabelul B.1: Parametri variabilelor de intrare

INPUT 1:

Name	Parameter
Universe	[40 80]
Red	[-20 -10 48 59]
Blue	[48 55 67 71]
Green	[55 71 81 90]

INPUT 2:

Name	Parameter
Universe	[20 45]
Red	[22 39 46 50]
Blue	[0 10 24 35]
Green	[21 28 34 39]

INPUT 3:

Name	Parameter
Universe	[10 70]
Red	[0 5 19 28]
Blue	[26 30 44 52]
Green	[44 53 75 80]

INPUT 4:

Name	Parameter
Universe	[0 25]
Red	[-10 -5 6 10]
Blue	[6 10 13 19]
Green	[13 19 30 35]

Să se determine valorile parametrilor de ieșire. Să se completeze tabelul următor cu valorile corecte, dacă pentru **variabila lingvistică de ieșire**, sunt folosite **funcțiile constante**.

Ieșirea:

Tabelul B.2: Parametrii variabilelor de ieșire

Nume	**Parametru**
Univers	**[1 3]**
Roșu	1
Albastru	2
Verde	3

Să se desemneze numărul de reguli și să se scrie în forma corectă.

Reguli:

1. Dacă Intrarea 1 este Roșu, Intrarea 2 este Roșu, Intrarea 3 este Roșu și Intrarea 4 este Roșu, atunci Ieșirea este Roșu.
2. Dacă Intrarea 1 este Albastru, Intrarea 2 este Albastru, Intrarea 3 este Albastru și Intrarea 4 este Albastru, atunci Ieșirea este Albastru.
3. Dacă Intrarea 1 este Verde, Intrarea 2 este Verde, Intrarea 3 este Verde și Intrarea 4 este Verde, atunci Ieșirea este Verde.

C-SCURTĂ DESCRIERE A SETURILOR DE DATE LEGATE DE SCHIMBĂRILE CLIMATICE

Această parte a manualului a fost scrisă de Mihaela Tinca Udriștioiu de la Departamentul de Fizică, Facultatea de Științe și Silvia Puiu, de la Departamentul de Management, Marketing și Administrarea Afacerilor, Facultatea de Economie și Administrarea Afacerilor, Universitatea din Craiova, România.

Cercetarea progresează mai ușor și mai rapid când oamenii au acces deschis și liber la informații. Cu posibilitatea de a accesa Internetul la un clic distanță, trebuie să se cunoască unde pot fi găsite surse de informare precise, actualizate și de încredere. De aceea, rolul bazelor de date este atât de important. Informația este structurată și, de obicei, într-un mod care poate fi ușor transformat și prelucrat în funcție de nevoile cercetătorului sau utilizatorului.

Comisia Europeană a creat baze de date open-source (cu acces liber) legate de schimbările climatice. Este ușoară descărcarea seturilor de date; mai dificile sunt procesarea și analiza (analiza exploratorie și predictivă) seturilor de date precum și folosirea diferiților algoritmi de creare a unor modele matematice. Copernicus, European Climate Assessment and Dataset, Climate Explorer și Indecis sunt doar câteva baze de date open-source. Datele sunt necesare pentru a monitoriza schimbările climatice și a face diverse prognoze de vreme, pentru a observa sensibilitatea climatică în funcție de diferiți parametri, pentru a crea diverse scenarii și a observa evoluția anumitor procese, atât pe termen scurt cât și pe termen lung. În cele ce urmează, autorii vor prezenta pe scurt câteva baze de date.

***Centrul European pentru Prognoza Meteo pe Termen Mediu** (European Center for Medium-range Weather Forecast - ECMWF) procesează date de la aproximativ 90 de sateliți, în date operaționale zilnice, de asimilare și monitorizare a activităților. Sunt disponibile zilnic aproximativ 60 de milioane de observații coontrolate calitativ în Sistemul Integrat de Prognoză, dintre care majoritatea sunt măsurători satelitare. ECMWF beneficiază, de asemenea, de observații din surse nesatelitare, inclusiv rapoarte de suprafață și provenite de la aeronave.*

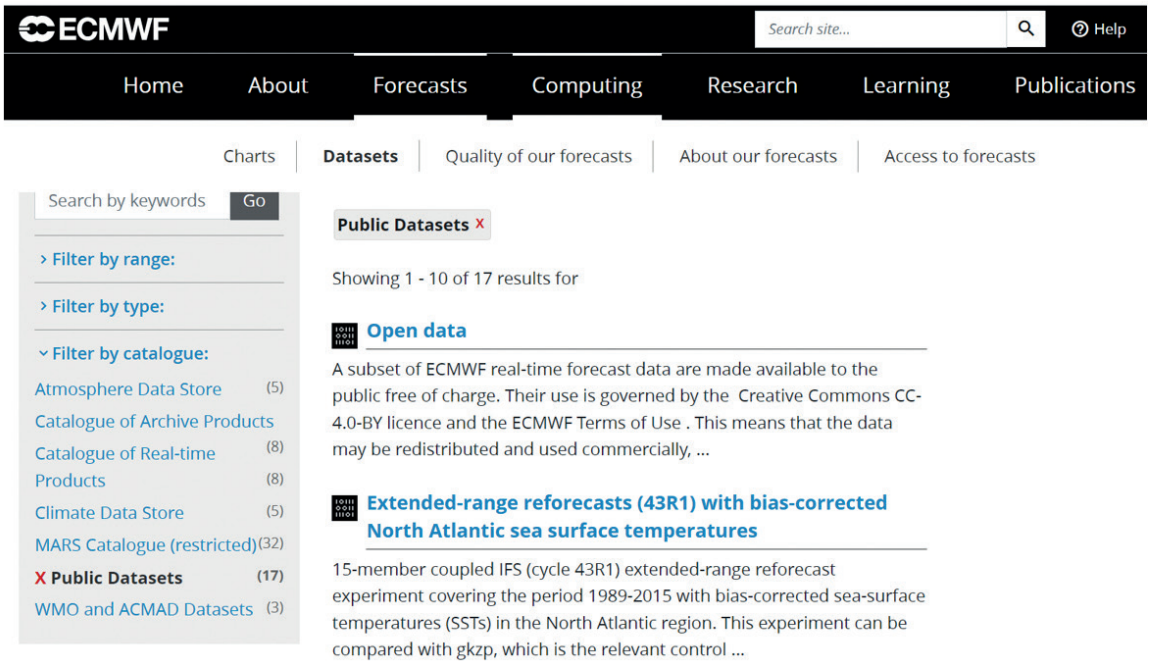


Figura C.1: Captură de ecran a secțiunii de Seturi de date publice de pe site-ul web ECMWF (sursa: https://www.ecmwf.int/en/forecasts/datasets/search)

Copernicus este componenta de observare a Pământului din cadrul Programului Spațial al UE. Comisia Europeană (CE) este cea care îl gestionează. CE implementează Copernicus în parteneriat cu statele membre ale UE, Agenția Spațială Europeană - European Space Agency (ESA), Organizația Europeană pentru Exploatarea Sateliților Meteorologici - European Organization for the Exploitation of Meteorological Satellites (EUMETSAT), Centrul European pentru Prognoza Meteo pe Termen Mediu - European Centre for Medium-Range Weather Forecasts (ECMWF), Centrul Comun de Cercetare - Joint Research Centre (JRC), Agenția Europeană de Mediu - European Environment Agency (EEA), Agenția Europeană de Siguranță Maritimă - European Maritime Safety Agency (EMSA), Frontex, SatCen și Mercator Océan. Copernicus conține seturi de date climatice din diferite surse (reanalize, produse satelitare, proiecții climatice). Baza de date Copernicus este una dintre cele mai frecvent utilizate pentru seturi de date legate de schimbările climatice, lucrând cu modele de predicție și reglarea algoritmilor pentru procesarea datelor. Are sateliți (SENTINELS 1-6) cu misiuni importante.

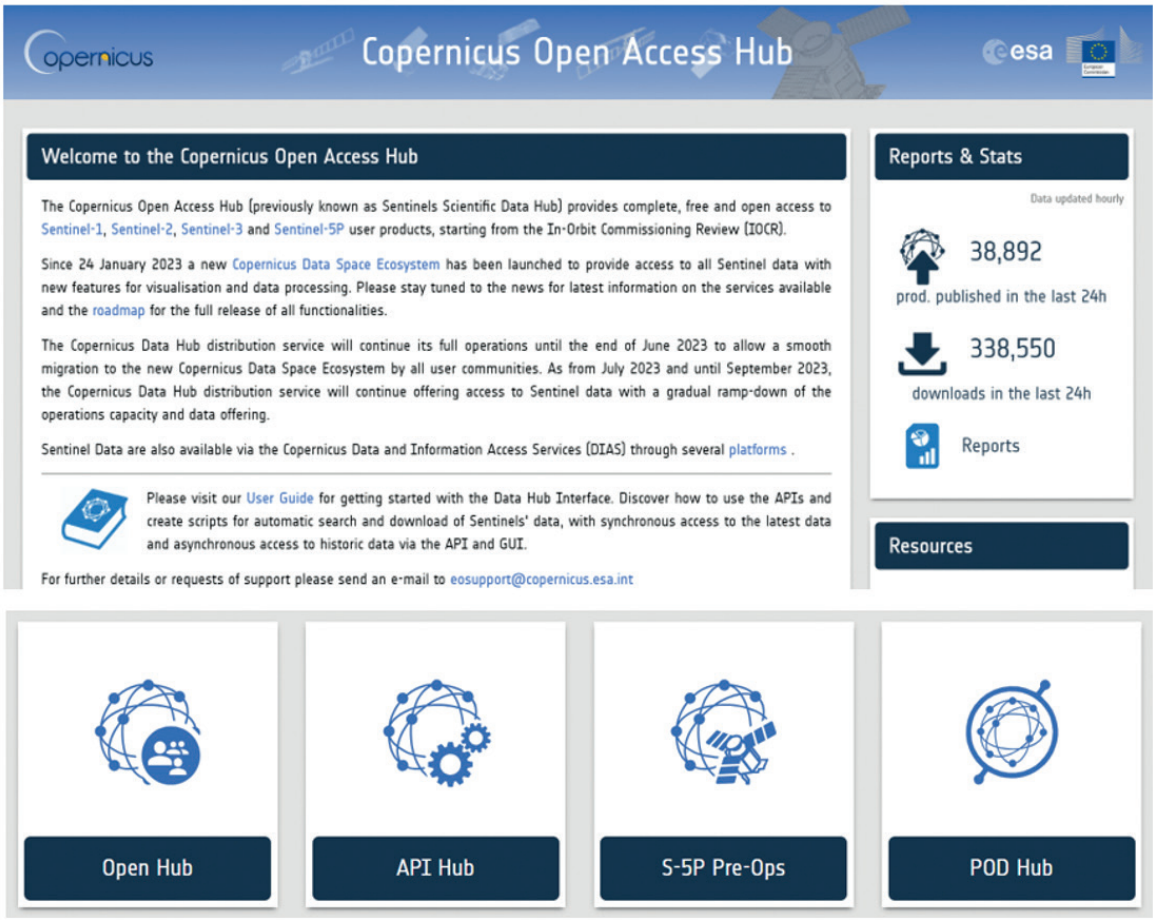


Figura C.2: Imagine a interfeței Copernicus Open Access Hub (sursa: https://scihub.copernicus.eu/)

Spre exemplu, SENTINEL-1 are doi sateliți care se deplasează pe orbite polare, care operează 24 de ore din 24, și 7 zile din 7, fără oprire. SENTINEL 1 folosește imagistica radar pentru a capta imaginile, indiferent de vreme.

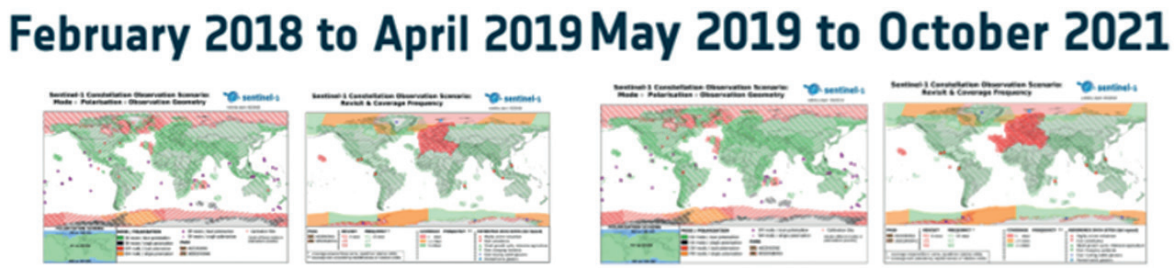


Figura C.3: Imagine oferită de SENTINEL 1 pentru două intervale de timp
(sursa: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/observation-scenario/archive)

SENTINEL-2 are doi sateliți cu orbită polară pe aceeași orbită sincronă cu soarele, la 180° distanță. Acești sateliți țin evidența schimbărilor apărute, în condițiile de la suprafața pământului. Raza sa mare de acțiune (290 km) și timpul lung de revenire (10 zile la ecuator cu un satelit și cinci zile cu doi sateliți în circumstanțe fără nori, rezultând în 2-3 zile la latitudini mijlocii) ajută la monitorizarea schimbărilor de la suprafața Pământului. Produsele SENTINEL-2 sunt disponibile utilizatorilor. Există produse care sunt doar pentru experți (radiațiile de vârf ale atmosferei în geometria senzorilor), și produse care sunt pentru toți utilizatorii (reflectanțele de vârf ale atmosferei în geometria cartografică și reflectanțele de suprafață corectate atmosferic în aceeași geometrie). Produsele pilot sunt generate doar la cerere. Există două categorii de produse pilot: SENTINEL-2 armonizat + Landsat-8/9 cu reflectanțe de suprafață în geometria cartografică.

SENTINEL-3 realizează măsurători ale topografiei la suprafața mării, ale temperaturii de la suprafața mării și a uscatului, și ale culorii suprafeței oceanelor și pământului. Ideea este de a sprijini sistemele de prognoză oceanică și de monitorizare ale mediului și climei.

SENTINEL-4 monitorizează urmele de gaze și aerosoli cheie pentru calitatea aerului în Europa, sprijinind Serviciul Copernicus de Monitorizare a Atmosferei - Copernicus Atmosphere Monitoring Service (CAMS) cu un timp de revenire rapid. Radianța Pământului care este calibrată spectral și radiometric și geolocalizată, iradierea solară calibrată spectral și radiometric sunt disponibile ca și parametri pentru toți utilizatorii, în timp ce parametrii de procesare a datelor, calibrarea și datele de diagnostic instrumentar sunt doar pentru utilizatorii experimentați.

SENTINEL-5 este un sistem spectometric de rezoluție înaltă operând într-un interval de la ultraviolete până la infraroșu, având șapte benzi spectrale diferite: UV-1 (270-300nm), UV-2 (300-370nm), VIS (370-500nm), NIR-1 (685-710nm), NIR-2 (745-773nm), SWIR-1 (1590-1675nm) și SWIR-3 (2305-2385nm). Sentinel-5 oferă informații despre calitatea aerului și interacțiunea compoziție atmosferă-climă (O_3, NO_2, SO_2, HCHO, CHOCHO și aerosoli). Sentinel-5 furnizează parametri de calitate pentru CO, CH_4 și O_3 stratosferic cu acoperire globală zilnică pentru climă, calitatea aerului și aplicații pentru UV de suprafață/ozon.

SENTINEL-5P realizează măsurători atmosferice cu o rezoluție spațio-temporală ridicată pentru calitatea aerului, ozon și radiațiile UV, precum și prognoza și monitorizarea climei.

Copernicus SENTINEL-6 Michael Freilich se axează pe măsurarea creșterii nivelului apei mării din cauza schimbărilor climatice și este următoarea misiune de referință pentru altimetria radar de extindere a măsurătorilor de înălțime de la suprafața mării, cel puțin până în anul 2030.

O altă bază de date importantă este **ECA&D** care conține observații de la stațiile meteorologice și seturi de date provenite de la acestea, la nivel european; cercetătorii consideră aceste date ca date de referință. Acest site conține informații referitoare la schimbările vremii și ale extremelor climatice precum și setul de date zilnic necesar pentru monitorizarea și analiza acestor extreme.

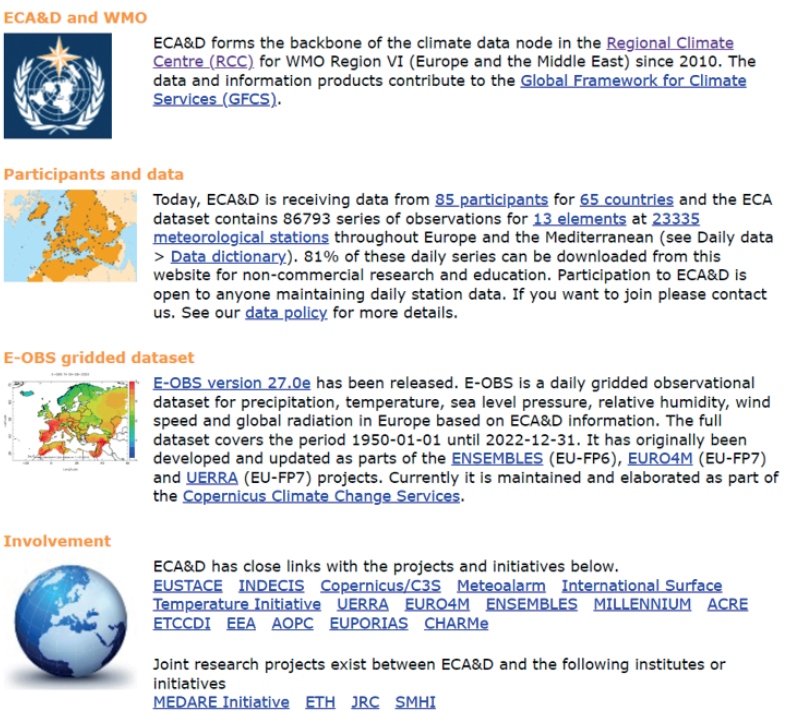


Figura C.4: Interfața ECA&D și WMO (sursa: https://www.ecad.eu)

KNMI Climate Explorer este o altă bază de date care conține o clasă de date climatice (serii de timp sau câmp) din reanalize și modele climatice, inclusiv proiecții climatice; are avantajul unei interfețe mai prietenoase (incluzând reprezentările geografice). Din acest motiv, reprezintă un instrument educațional bun. Utilizatorii pot descărca serii temporale despre datele zilnice și lunare ale stațiilor, precum și indici climatici. La nivel anual, doar indicii climatici anuali sunt disponibili. Cercetătorii pot descărca această informație prin selectarea unui câmp precum câmpuri zilnice, observații lunare, câmpuri de reanaliză lunare, reconstrucți istorice lunare și sezoniere, retrospective sezoniere lunare, rulări de scenarii lunare CMIP3+, rulări de scenarii lunare CMIP5, extreme CMIP5 anuale, rulări de scenarii lunare CMIP6, rulări de scenarii lunare CORDEX, runde de atribuire.

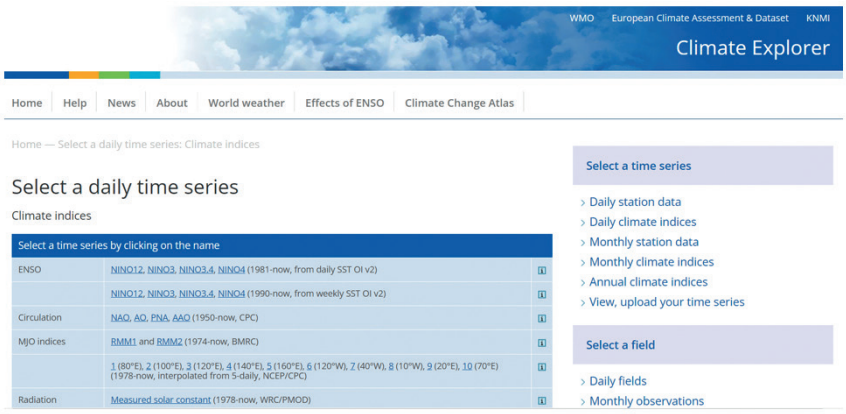


Figura C.5: Figura C.5 – Captură de ecran a KNMI Climate Explorer (sursa: https://climexp.knmi.nl/selectdailyindex.cgi?id=someone@somewhere)

Met Office Hadley Centre oferă seturi de date legate de variabilele meteorologice. Cercetătorii folosesc această informație în monitorizarea și cercetarea climatică. Clasele sunt următoarele: indicatori climatici cheie, seturi de date marine, date zilnice și orare/ indici extremi, date de suprafață pe uscat, date combinate uscat/presiune marină, date din stratul de aer superior, date unice care însoțesc articolele din jurnale și seturi de date mai vechi.

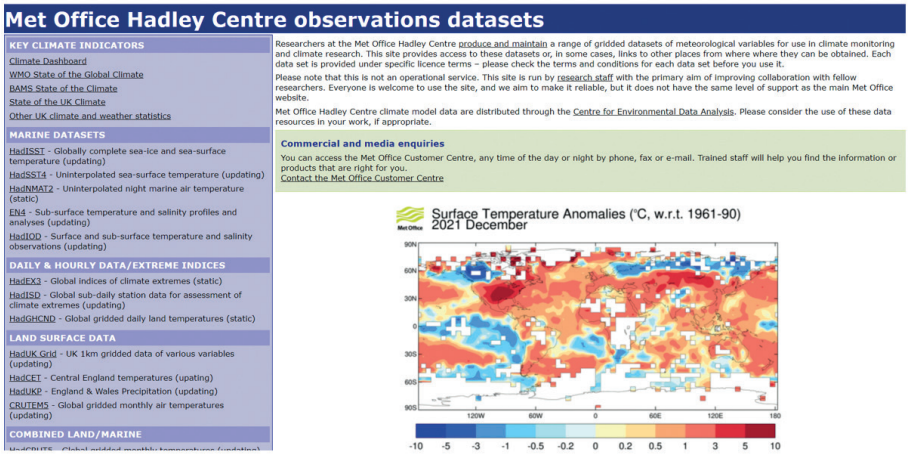


Figura C.6: Interfața Met Office Hadley Centre (sursa: https://www.metoffice.gov.uk/hadobs/index.html)

Indecis conține date climatice pentru agricultură, reducerea riscului de dezastre, energie, sănătate, apă și turism (*http://indecis.eu/indices.php*). Aici pot fi găsiți doar indici climatici – mulți cu diverse aplicații; platforma are definiții pentru indicii climatici, cu reprezentare grafică în stil hartă și serii de date în punct, plus posibilitatea de descărcare. Această bază de date este, de asemenea, un instrument educațional bun. Conține date zilnice ale stațiilor, date controlate calitativ ale stațiior, date omogenizate ale stațiilor, date recuperate de la stații și versiuni grilă ale indicilor.

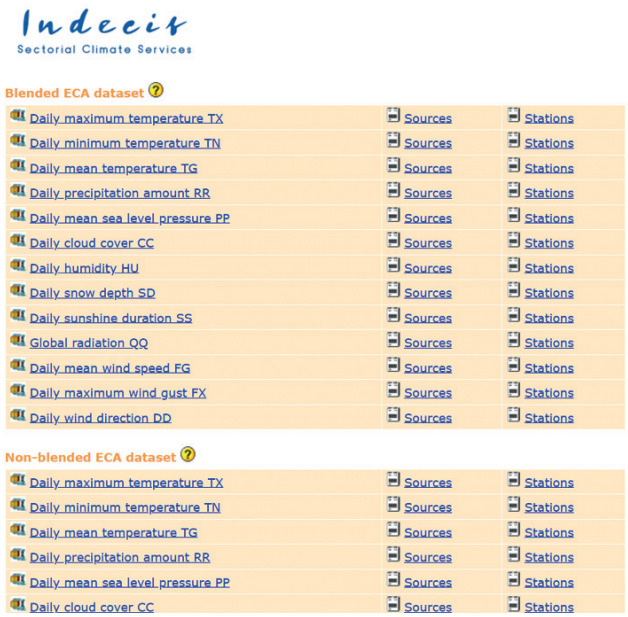


Figura C.7 : Clase de date care pot fi descărcate de pe Indecis (sursa: https://www.ecad.eu/dailydata/predefinedseries.php)

Există baze de date open-source (cu acces liber) oferite de Agenția Europeană de Mediu (European Environment Agency) pentru calitatea aerului.

Figura C.8: Seturi de date oferite de Agenția Europeană de Mediu (sursa: https://www.eea.europa.eu/themes/air/explore-air-pollution-data)

World's Air Pollution conține informații de la senzorii Agențiilor de Mediu Naționale și oferă informații despre indicii de calitatea aerului în timp real.

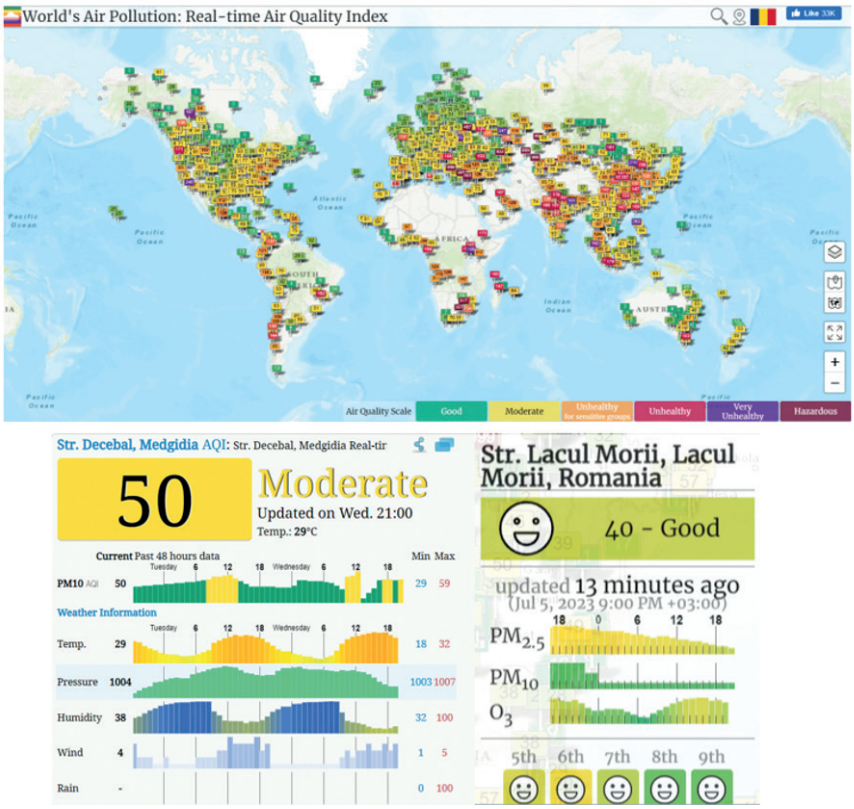


Figura C.9 : Harta senzorilor pentru calitatea aerului la nivel global și informații adiționale la accesarea unui senzor (sursa: https://waqi.info/)

OECD oferă informații despre indicatori precum aerul și emisiile gazelor cu efect de seră, expunerea la poluarea aerului și efectele poluării aerului, sub formă de grafice, hărți, sau tabele pentru a fi cât mai simplu pentru toată lumea de vizualizat evoluția datelor în timp. Un simplu clic deschide seturi de date organizate în diverse tipuri de grafice, hărți și tabele.

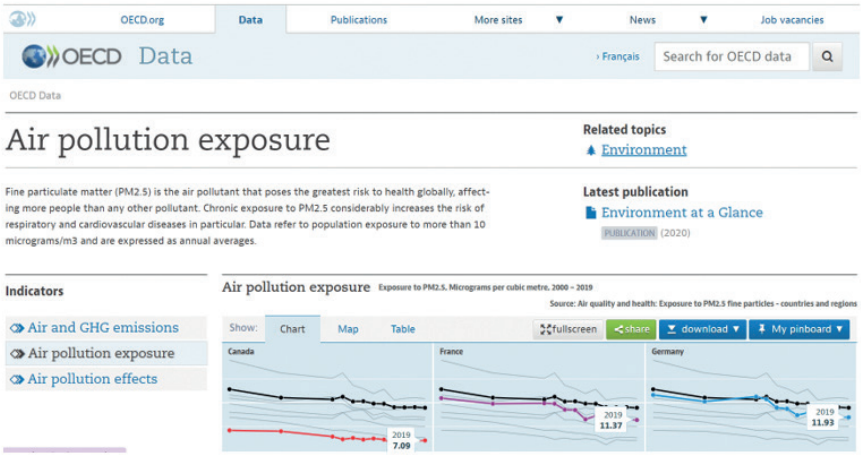


Figura C.10 : Interfața OECD (sursa: https://data.oecd.org/air/air-pollution-exposure.htm)

Există, de asemenea, inițiative în rândul cetățenilor și rețele de senzori create de comunități locale. Aceste rețele conțin senzori cu un cost scăzut care monitorizează calitatea aerului de către cetățeni în beneficiul comunităților lor și au o acoperire excelentă a unei mari suprafețe a Europei. O parte dintre aceste rețele au fost construite de voluntari în proiecte cu scop educațional. uRADMonitor® este un astfel de exemplu în România. Rețeaua asigură acces liber la date în timp real. Administratorii pot asigura date istorice la cerere. Inițiativele cetățenești promovează transparența și responsabilitatea în monitorizarea mediului. Alte exemple sunt următoarele: Community Air Sensor Network (CAIRSENSE), rețeaua Smart Citizen®, Public Laboratory for Open Technology and Science sau rețeaua Public Lab, inițiativa Eye on Earth, Global Learning and Observations to Benefit the Environment (GLOBE), HabitatMap©, Imperial County Community Air Monitoring Project și Citizen Weather Observer Program (CWOP).

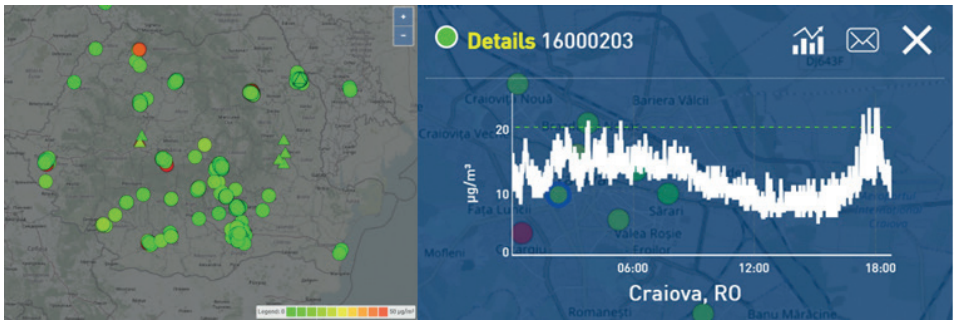


Figura C.11: Captură de ecran a rețelei uRADMonitor® (sursa: https://www.uradmonitor.com/)

Bazele de date statistice ajută la avansarea cercetării, asigurând o cantitate importantă de date, pentru variabile diferite și pentru perioade lungi de timp. Acesta este un aspect fundamental pentru a trage concluzii, a crea, prezice sau atenua diverse scenarii. De exemplu, dacă avem nevoie de date legate de

schimbările climatice, cum ar fi nivelul de poluare, avem multiple baze de date open-source pe care le putem accesa și utiliza pentru obiectivele noastre.

Deciziile se bazează pe intrările de date; dacă nu avem informația corectă, deciziile noastre nu vor fi nici ele bune. Astfel, este important să alegem surse de informare de încredere, cum ar fi cele de la organizațiile naționale și internaționale bine cunoscute.

Una dintre aceste surse este reprezentată de baza de date Eurostat care asigură multe date statistice în legătură cu multe aspecte de interes pentru țările europene. Unul dinte motivele pentru care putem fi siguri de această bază de date este lunga sa istorie (70 de ani) și faptul că este sub umbrela Uniunii Europene.

Dăm un exemplu de cum pot fi accesate datele legate de schimbările climatice folosind baza de date Eurostat. Se poate accesa site-ul web în mod direct și se caută *climate change* (schimbări climatice) sau se face acest lucru folosind un motor de căutare. Dacă se accesează linkul *https://ec.europa.eu/eurostat/web/climate-change/database*, pot fi găsite diverse date care pot fi folosite pentru un anumit scop, așa cum se observă din Fig. C.12.

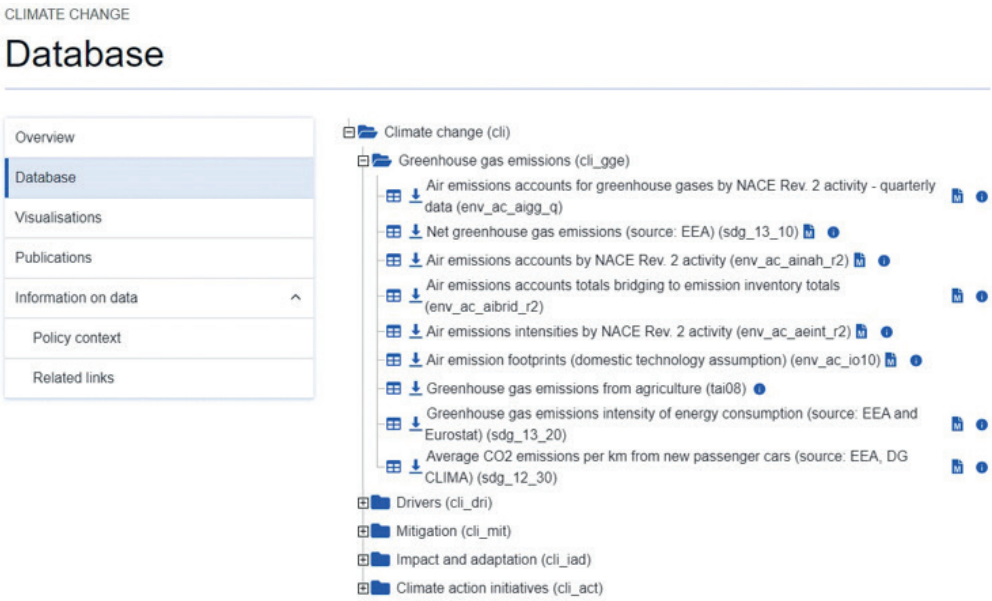


Figura C.12: Captură de ecran de pe site-ul web Eurostat referitor la informațiile despre schimbările climatice (sursa: https://ec.europa.eu/eurostat/web/climate-change/database)

Baza de date de schimbări climatice este structurată pe mai multe domenii: emisiile gazelor cu efect de seră; factori care produc schimbările climatice; atenuarea schimbărilor climatice; impact și adaptare; inițiative de acțiune climatică. Fiecare dintre acestea conține date care pot fi descărcate de utilizator. Informația este gratuită și liberă tuturor, în diferite formate.

În directorul despre emisiile gazelor cu efect de seră, se poate observa că sunt mai multe tipuri de date. Dacă mergem la primul tip - Emisiile atmosferice generate de gazele cu efect de seră (date trimestriale) - Air emissions accounts for greenhouse gases, pot fi găsite mai multe informații dacă dăm clic pe butonul aferent. Fereastra care se deschide este cea din Figura C.13. Astfel, se observă că informația este disponibilă pentru 13 ani, din 2010 până în 2022 și că această informație a fost actualizată în mai 2023.

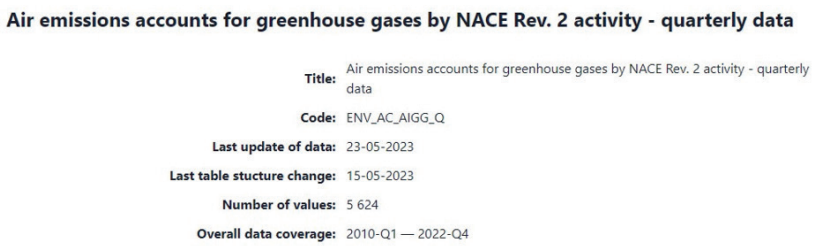


Figura C.13: PCaptură de ecran după accesarea butonului de informații suplimentare (sursa: https://ec.europa.eu/eurostat/web/climate-change/database)

Dacă suntem interesați să aflăm mai multe despre factorii care duc la schimbările climatice, se selectează cel de-al doilea director și se observă în Figura C.14 că sunt disponibile date pentru fiecare dintre aceștia, precum energia; transportul; procesele industriale; deșeurile; agricultura; cultivarea pământului, schimbările legate de utilizarea pământului și silvicultura. Pentru Energie, pot fi descărcate date legate de consumul final de energie, consumul final de energie pe cap de locuitor, consumul final de energie pe sectoare, etc.

Este importantă selectarea datelor necesare utilizatorului pentru atingerea obiectivelor sale. Datele sunt în stare brută și neprocesate, astfel că utilizatorul trebuie să folosească diverse instrumente pentru prelucrarea datelor, observarea unor tendințe, anticiparea anumitor scenarii și prezentarea rezultatelor într-o manieră mai prietenoasă. Aceste rezultate vor fi, mai departe, utilizate de afaceri, indivizi, guverne și alte persoane cu responsabilitate pentru a preveni anumite aspecte sau a schimba lucrurile în bine.

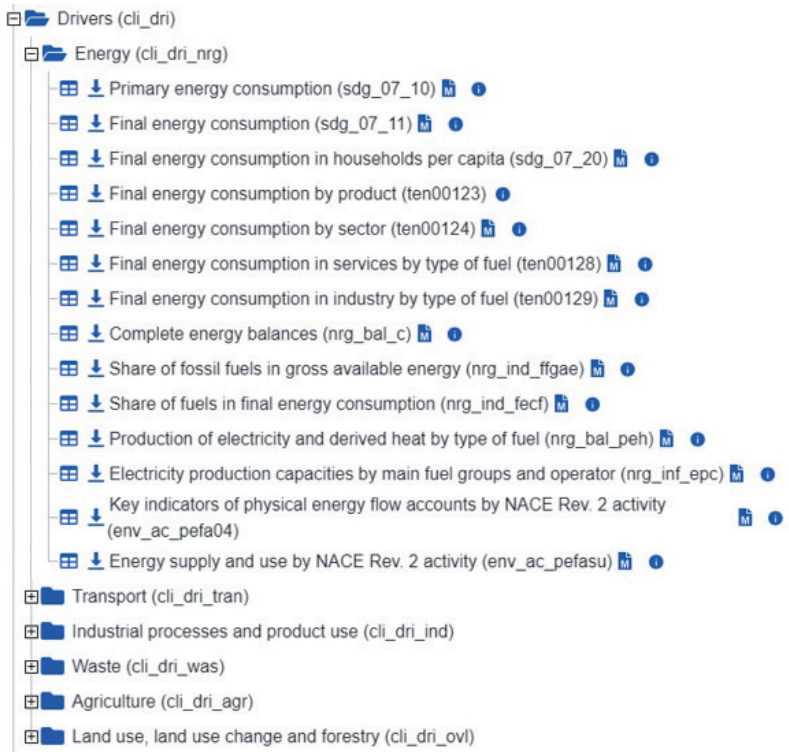


Figura C.14: Captură de ecran cu informațiile disponibile pe Eurostat referitor la factorii cauzatori ai schimbărilor climatice (sursa: https://ec.europa.eu/eurostat/web/climate-change/database)

În continuare, să vedem cum arată informația dacă dorim să verificăm consumul final de energie în gospodării, pe cap de locuitor. În Fig. C.12, se vede că există un cod între paranteze lângă acest indicator: SDG 7. Acesta este, de fapt, o referință la al șaptelea obiectiv de dezvoltare sustenabilă din Agenda 2030 a Organizației Națiunilor Unite. Practic, acesta se referă la *Energie accesibilă și curată*.

Dacă se accesează prima iconiță care arată ca un tabel, pot fi citite explicațiile referitoare la indicator, dar există și posibilitatea de a alege formatul datelor (tabel, liniar, cu bare, hartă) și variabilele de care avem nevoie (țări și ani) – Fig. C.15 și C.16.

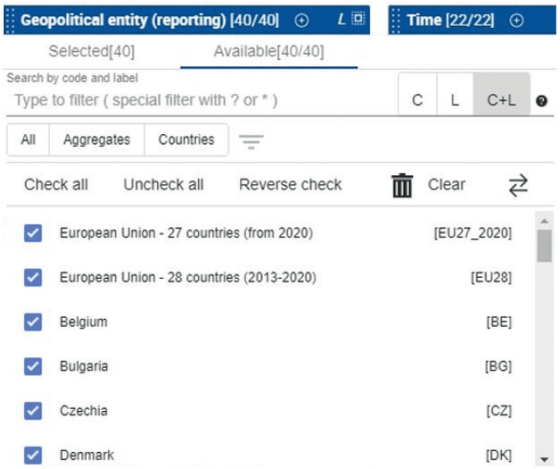


Figura C.15: Captură de ecran asupra filtrelor ce pot fi utilizate pentru datele de pe Eurostat (sursa: https://ec.europa.eu/eurostat/web/climate-change/database)

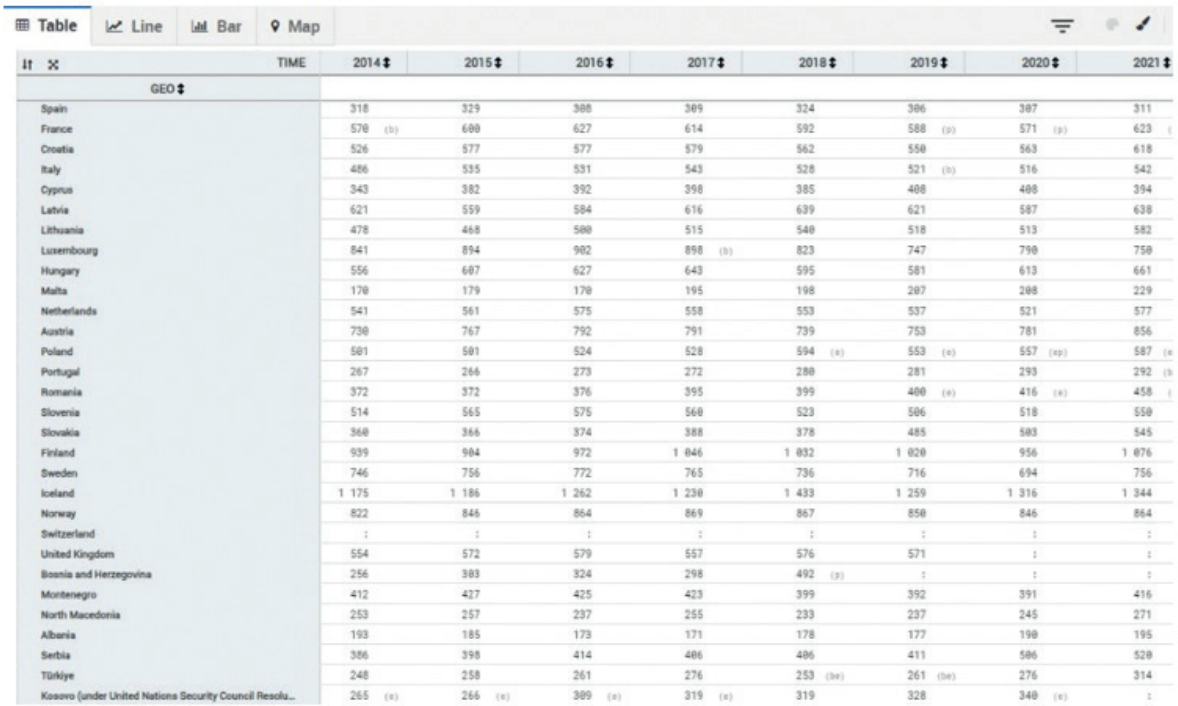


Figura **C.16 :** Captură de ecran asupra informațiilor afișate dacă alegem formatul tabelar (sursa: https://ec.europa.eu/eurostat/web/climate-change/database)

Pe lângă Eurostat, sunt și alte baze de date open-source care pot fi utilizate de cercetători. Poate fi menționată Our World in Data care are ca principal obiectiv afișat pe site-ul web (*https://ourworldindata.org/*) publicarea de cercetări și date pentru a progresa în soluționarea celor mai mari probleme ale lumii, ceea ce se referă la dinamica populației, energie și mediu, sănătate, hrană, sărăcie, educație, condiții de trai, drepturile omului, schimbări tehnologice și violență și război. Funcționează sub umbrela unei organizații non-profit, dar este intens citat în literatura de specialitate și în media.

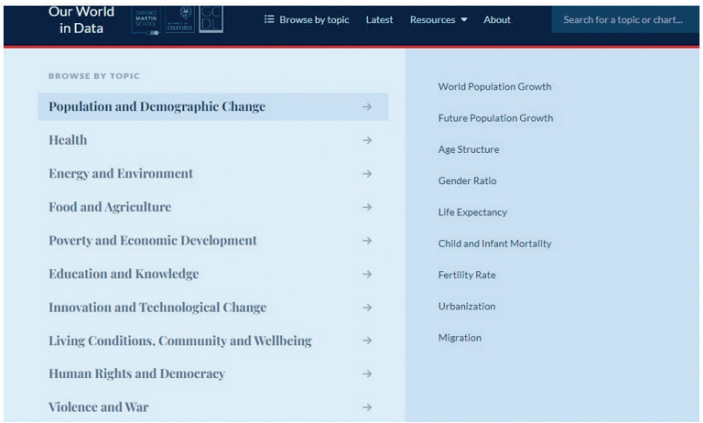


Figura C.17: Captură de ecran pe site-ul web Our World in Data referitor la subiectele adresate de această publicație (sursa: https://ourworldindata.org/)

În cee ace privește poluarea aerului, se poate selecta *Energie și Mediu* și există posibilitatea de a alege între poluarea aerului din interior sau exterior. Acest site web oferă atât articole, cât și date statistice în stare brută, pe care le poți folosi în cercetarea sau activitatea ta. Astfel, se poate afla câte decese pot fi atribuite poluării aerului la nivel global (Fig. C.18). Se poate afla spre exemplu care este rata de deces cauzat de poluarea aerului, pe vârste și pot fi descărcate datele sub formă de tabel sau grafic (Fig. C.19).

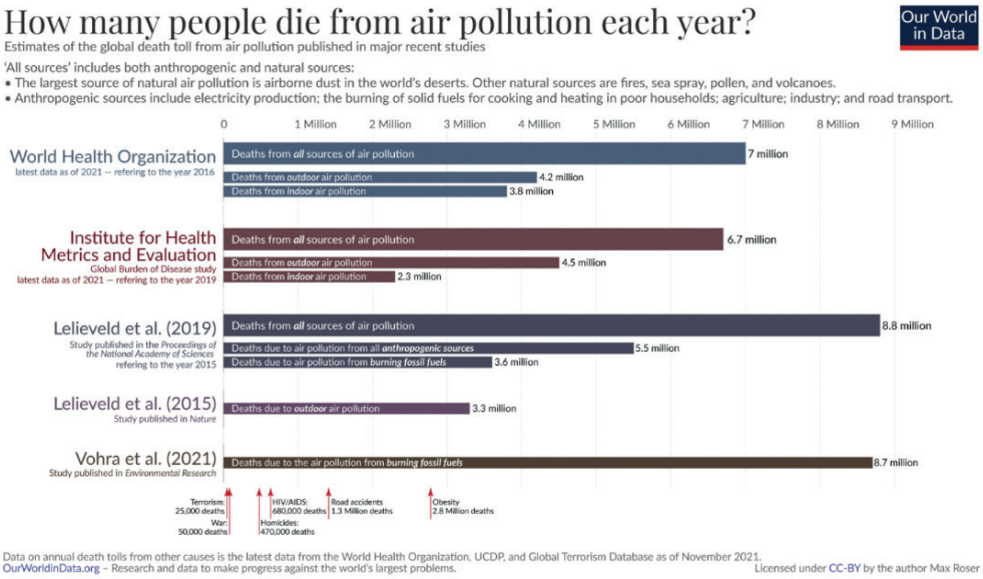


Figura C.18: Decese globale generate de poluarea aerului (sursa: https://ourworldindata.org/data-review-air-pollution-deaths)

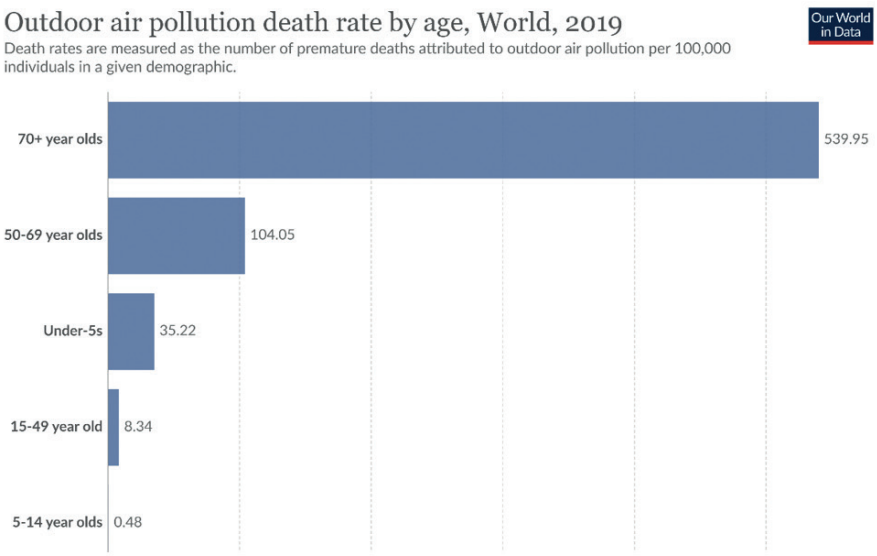


Figura C.19: Rata de deces cauzat de poluarea aerului, pe vârste
(sursa: https://ourworldindata.org/outdoor-air-pollution#outdoor-air-pollution-deaths-by-age)

D. IMPACTUL POLUĂRII AERULUI ASUPRA SĂNĂTĂȚII UMANE

Această parte a manualului a fost scrisă de Slaveya Petrova de la Departamentul de Ecologie și Conservarea Mediului, Facultatea de Biologie, Universitatea Plovdiv "Paisii Hilendarski," Bulgaria.

Poluarea aerului reprezintă contaminarea mediului intern și extern cu agenți biologici, chimici sau fizici care schimbă caracteristicile naturale ale atmosferei.

Dispozitivele de ardere din gospodării, vehiculele cu motor și facilitățile industriale sunt surse comune de poluare a aerului. Poluanții cu risc major pentru sănătatea publică includ particule fine (PM), monoxid de carbon (CO), ozon (O3), dioxid de azot (NO2) și dioxidul de sulf (SO2).

Potrivit Agenției Europene de Mediu - European Environment Agency (EEA), fiecare poluant al aerului poate fi asociat cu diferite surse:

- Consumul de energie rezidențial, comercial și instituțional a fost principala sursă de particule fine în 2020. Industria extractivă și producătoare, precum și agricultura au fost, de asemenea, surse semnificative de PM10. O tendință în scădere s-a observat între 2005 și 2020 în legătură cu emisiile de particule fine (PM10 și PM2.5) – au scăzut cu 30% și, respectiv, 32%.
- Agricultura a fost principala sursă de amoniac (94% din emisiile totale) și metan (56%) în 2020. Emisiile de amoniac au scăzut cu doar 8% din 2005 până în 2020. Aceasta a fost cea mai scăzută reducere procentuală dintre toți poluanții.
- Transportul rutier a fost principala sursă de oxizi de azot în 2020, reprezentând 37% dintre emisii. O reducere semnificativă de până la 48% în emisiile de oxizi de azot a fost între 2005 și 2020.
- Sectorul aprovizionării cu energie a fost principala sursă de dioxid de sulf, responsabil pentru 41% din emisiile din 2020. Emisiile de dioxid de sulf au scăzut cu 79% între 2005 și 2020.
- Industriile extractive și producătoare și sectorul aprovizionării cu energie au fost principalele surse de emisii de metale grele în 2020. Între 2005 și 2020, cele mai mari reduceri ale emisiilor au fost pentru nichel (64%) și arsenic (62%).

D.1 TIPURI DE POLUANȚI ȘI RISCURILE PENTRU SĂNĂTATE

Particule fine (PM)

Particulele fine (PM) reprezintă un indicator comun pentru poluarea aerului. Principalele componente ale fracțiilor de PM sunt sulfații, nitrații, amoniacul, clorura de sodiu, cărbunele negru, praful mineral și apa.

Riscurile pentru sănătate asociate cu particulele fine mai mici de 10 și 2,5 microni în diametru (PM_{10} și $PM_{2,5}$) sunt, în special, foarte bine documentate. PM pot pătrunde adânc în plămâni și intra în sistemul circulator cauzând boli cardiovasculare (boala cardiacă ischemică), cerebrovasculare (accident vascular cerebral) și respiratorii. Atât expunerea pe termen lung, cât și cea pe termen scurt la PM sunt asociate cu morbiditatea și mortalitatea provocate de boli cardiovasculare și respiratorii. Expunerea pe termen lung este legată de rezultatele perinatale adverse și cancerul pulmonar.

Monoxidul de carbon (CO)

Monoxidul de carbon este un gaz toxic incolor, fără miros și gust produs de arderea incompletă a combustibililor carbonici precum lemnul, benzina, cărbunele, gazul natural și kerosenul. Monoxidul de carbon se răspândește în țesutul pulmonar și în sistemul circulator, făcând dificil pentru celulele din corp să se lege de oxigen. Această lipsă de oxigen distruge țesuturile și celulele. Expunerea la monoxidul de carbon poate cauza dificultăți de respirație, oboseală, amețeală și alte simptome similare celor de gripă. Expunerea la niveluri ridicate de monoxid de carbon poate fi fatală.

Ozonul (O_3)

Ozonul la nivelul solului este unul dintre componentele majore ale smogului fotochimic. Apare prin reacția cu gazele în prezența luminii solare. Este important de menționat că ozonul poate fi generat de aparatura din gospodării, cum ar fi purificatoarele de aer portabile. Expunerea la ozon în cantități mari poate cauza probleme de respirație, declanșa astm, reduce funcția pulmonară și duce la boli pulmonare.

Dioxidul de azot (NO_2)

NO_2 este un gaz generat de arderea combustibilor în sectoarele de transporturi și industrial. Sursele de oxizi de azot (NO_x) din gospodării includ echipamentele care ard combustibilii, cum ar fi cuptoarele, șemineele, plitele pe gaz. Expunerea la dioxidul de azot poate irita căile aeriene și agrava bolile respiratorii.

Dioxidul de sulf (SO_2)

SO_2 este un gaz incolor cu miros ascuțit, produs prin arderea combustibililor fosili (cărbune și petrol) și topirea minereurilor minerale ce conțin sulf. Expunerea la SO_2 este asociată cu internările pentru astm bronșic și vizitele la urgență.

Hidrocarburile aromatice pociclice (PAH)

Hidrocarburile aromatice pociclice (PAH) sunt prezente în atmosferă sub formă de particule. Sunt un grup de chimicale format în principal prin arderea incompletă a materiei organice (de exemplu, gătirea cărnii) și combustibili fosili în cuptoare de cocs, motoare diesel și sobe cu lemne. De asemenea, pot fi

prezente în fumul de țigară. Expunerea pe termen scurt poate irita ochii și căile respiratorii. Expunerea pe termen lung la PAH este legată de cancerul pulmonar.

D.2 RECOMANDĂRILE OMS PRIVIND CALITATEA AERULUI

Din 1987, OMS a emis periodic recomandări pentru calitatea aerului bazate pe sănătate pentru a sprijini guvernele și societatea civilă în reducerea expunerii umane la poluarea aerului și efectelor sale adverse. Principalul obiectiv este de a oferi recomandări cantitative bazate pe sănătate pentru managementul calității aerului, exprimate sub formă de concentrații pe termen scurt sau lung pentru câțiva poluanți cheie ai aerului. Depășirea nivelurilor recomandate pentru calitatea aerului (air quality guideline - AQG) este asociată cu importante riscuri pentru sănătatea publică. Aceste recomandări nu sunt standarde obligatorii legal. Totuși, oferă statelor membre OMS un instrument bazat pe dovezi pe care îl pot implementa în programe naționale pentru a reduce nivelurile poluanților aerului și povara enormă pentru sănătate generată de expunerea globală la poluarea aerului.

Tabelul D.1 – Recomandările pentru calitatea aerului pentru fiecare poluant [5]

Poluant	**Valoarea recomandată**	**Timpul mediu**	**Ghidul de referință**
$PM_{2,5}$	5 μg/m^3	Anual	OMS 2021
	15 μg/m^3	24 ore	
PM_{10}	15 μg/m^3	Anual	OMS 2021
	45 μg/m^3	24 ore	
Monoxidul de carbon (CO)	4 μg/m^3	24 ore	OMS 2021
Dioxidul de azot (NO_2)	10 μg/m^3	Anual	OMS 2021
	25 μg/m^3	24 ore	
Dioxidul de sulf (SO_2)	40 μg/m^3	24 ore	OMS 2021
Formaldehida	0,1 μg/m^3	30 minute	OMS 2010
Hidrocarburile aromatice policiclice	8,7 × 10^{-5} per ng/m^3		OMS 2010
Radonul	100 Bq/m^3		OMS 2010
Plumbul	0,5 μg/m^3	Anual	OMS, Biroul regional pentru Europa, 2000

D.3 STUDIILE PRIVIND EFECTUL POLUĂRII AMBIENTALE A AERULUI ASUPRA SĂNĂTĂȚII

În ultimul secol, creșterea arderii combustibilor fosili și intensificarea continuă a traficului sunt responsabile pentru schimbarea progresivă în compoziția atmosferei, ceea ce afectează negativ calitatea vieții.

Unul dintre principalele aspecte îl reprezintă influența gazelor de evacuare ale vehiculelor asupra sănătății, la care copiii sunt foarte sensibili. Gazele de evacuare conțin peste 200 de tipuri de poluanți, printre care: CO_2, NO_x, CO, SO_x, hidrocarburile cu greutate moleculară mică, aldehidele (formaldehidă, acetaldehidă, acroleină), benzenul, 1,3 butadiena, hidrocarburile policiclice, particulele componente oxidative (carbon elementar, hidrocarburi aromatice adsorbite, cantități mici de sulfat, nitrat, metale și alte elemente), etc.

Deși activitatea economică redusă în timpul recesiunii a dus la reducerea emisiilor atmosferice, se consideră în mod general că transportul automobilelor în Europa este responsabil pentru nivelurile periculoase de poluanți ai aerului și un sfert din emisiile gazelor cu efect de seră din Uniunea Europeană. Standardele „Euro" pentru vehicule au avut un oarecare success, dar nu au redus semnificativ NO_2.

Poluanții aerului, cum ar fi monoxidul de carbon (CO), dioxidul de sulf (SO_2), oxizii de nitrogen (NOx), compușii organici volatili (VOCs), ozonul (O_3), metalele grele și particulele fine ($PM_{2,5}$ and PM_{10}), diferă prin compoziția chimică, proprietățile de reacție, timpul de dezintegrare și abilitatea de răspândire pe distanțe scurte sau lungi. Poluarea aerului exterior este o problemă de sănătate a mediului majoră afectând pe toată lumea din țările cu venituri mici, mijlocii sau ridicate deoarece poate cauza boli respiratorii, dar și altele, fiind sursă importante de morbiditate și mortalitate [9]. Aceste efecte ale poluanților aerului asupra sănătății umane și mecanismul lor de acțiune vor fi discutate pe scurt în cele ce urmează.

Poluarea aerului are efecte acute și cronice asupra sănătății umane, afectând diferite sisteme și organe. Variază de la o minoră iritație a căilor respiratorii superioare la boli cardiace și respiratorii cronice, cancer pulmonar, infecții respiratorii acute la copii și bronșită cronică la adulți, agravarea bolilor pulmonare și cardiace pre-existente sau atacurilor astmatice. În plus, expunerea pe termen scurt și lung a fost, de asemenea, legată de mortalitatea prematură și speranța redusă de viață.

OMS estimează că, în 2019, 37% din decesele premature provocate de poluarea aerului exterior s-au datorat bolii ischemice cardiace și accidentului vascular cerebral, 18% și 23% din decese s-au datorat bolii pulmonare obstructive cronice și, respectiv, infecțiilor acute ale tractului inferior, iar 11% din decese s-au datorat cancerului de tract respirator. Se estimează că poluarea aerului ambiental (exterior) în orașe și zone rurale a cauzat 4,2 milioane de decese premature la nivel global în anul 2019; această mortalitate se datorează expunerii la particulele fine, care cauzează boli cardiovasculare și respiratorii, dar și cancere.

În UE, este uzuală depășirea nivelurilor poluării aerului comparativ cu ultimele recomandări OMS. Totuși, există semne de îmbunătățire, dar câteva aspecte sunt subliniate mai jos:

- În 2021, 97% din populația urbană a fost expusă la concentrații de particule fine peste nivelul recomandărilor bazate pe sănătate, stabilite de Organizația Mondială a Sănătății.
- Anual, peste 1.200 de decese în rândul celor sub 18 ani sunt estimate a fi cauzate de poluarea aerului în țările membre și partenere EEA [10].
- Datele din 2021 arată că Europa Centrală și de Est și Italia au raportat cele mai ridicate concentrații de particule fine, în special generate de arderea combustibilior solizi pentru încălzirea gospodăriilor și utilizarea lor în industrie.
- Toate țările din UE au raportat niveluri ale ozonului și dioxidului de azot peste cele recomandate de Organizația Mondială a Sănătății.
- În jur de 275.000 de decese premature sunt cauzate de particulele fine și 64.000 de dioxidul de azot (NO_2) în fiecare an.
- La modul general, 97% din populația urbană a UE a fost expusă la niveluri de particule fine peste recomandările cele mai recente stabilite de OMS în 2021.

Efectele adverse ale expunerii la poluarea aerului sunt o problemă globală de sănătate publică atât în țările în curs de dezvoltare, cât și în cele dezvoltate, în care copiii și tinerii sunt mai vulnerabili la efectele poluării aerului.

Studiile epidemiologice sunt indicative pentru evaluarea efectelor poluării aerului asupra sănătății. Cei mai vulnerabili, potriviți pentru studiile contingente, sunt copiii de vârstă preșcolară și școlară timpurie deoarece ei petrec mai mult timp afară, au o intensitate mai ridicată a proceselor metabolice și aspiră un volum relativ mai mare de aer decât adulții. În același timp, nu au dobândit obiceiuri nesănătoase (fumat, consum de alcool, etc.) încă și nu sunt expuși riscurilor industriale. Într-un complex mare de efecte negative asupra sănătății ale emisiilor de evacuare, cel mai mult ies în evidență afectarea funcției respiratorii, a sistemului cardiovascular și a celui imun, hematopoietic și altele.

Un studiu de amploare implicând copiii de vârstă preșcolară și școlară timpurie în 6 orașe din nordul Chinei au arătat o corelație pozitivă puternică între simptomele respiratorii (tuse, dificultate în respirație, șuierat și flegmă) și nivelurile de praf în suspensie totală, dioxid de sulf și azot.

În mod deosebit este subliniată relația dintre dioxidul de azot și ozon și provocarea sau exacerbarea bolilor respiratorii cu sindrom obstructiv, cum ar fi astmul. Un exemplu îl reprezintă cazurile în care chiar și o intensitate redusă temporar a traficului vehiculelor diminuează simptomele respiratorii ale tractului respirator superior.

Poluarea aerului afectează dezvoltarea fizică și mintală a copiilor și agravează condițiile respiratorii precum astmul sau rinita alergică sezonieră, mai uzual numită febra fânului. Febra fânului este cea mai comună condiție cronică la copii și este mai întâlnită în rândul școlarilor. Există tot mai multe dovezi că poluanții aerului precum ozonul (O_3) pot spori alergenitatea polenului, care poate afecta dezvoltarea cognitivă.

D.4 IMPACTUL POLUĂRII AERULUI ASUPRA SĂNĂTĂȚII ȘI MEDIULUI

Calitatea aerului este o problemă majoră pentru europeni, fiind un domeniu în care UE a fost deosebit de activă în ultimii 30 de ani. Principalul obiectiv al UE legat de calitatea aerului este „de a atinge niveluri ale calității aerului care nu au un impact inacceptabil asupra, și nici riscuri, pentru sănătatea oamenilor și mediu". Întrebările din sondajul anual Eurobarometru Flash sunt concepute pentru a sprijini acest obiectiv prin oferirea unei mai bune perspective asupra opiniei publicului european în legătură cu calitatea aerului și poluarea acestuia.

Sondajul Eurobarometru este conceput pentru a examina:

- nivelul de cunoaștere în legătură cu problemele de calitate a aerului;
- seriozitatea percepută în legătură cu problemele de calitatea aerului și schimbările percepute în calitatea aerului în ultimii zece ani;
- impactul perceput al diferitelor sectoare și activități asupra calității aerului;
- principalele amenințări pentru calitatea aerului;
- opțiunile de energie și transport prietenoase cu mediul;
- acțiunile individuale și nu numai pentru reducerea problemelor de calitate a aerului;
- și multe altele.

Rezultatele sondajului din 2022 arată că problema calității aerului este, încă, o preocupare majoră pentru cetățenii europeni. Toate datele brute ale sondajului sunt disponibile gratuit și pot fi accesate online.

- În timp ce majoritatea europenilor nu se simt bine informați (60%), aproape jumătate dintre respondenți cred că, în ultimii zece ani, calitatea aerului s-a deteriorat (47%).

- Majoritatea europenilor cred că afecțiunile precum bolile respiratorii (89%), astmul (88%) și bolile cardiovasculare sunt probleme serioase în țările lor, acestea apărând din cauza poluării aerului. Eurobarometrul arată că cetățenilor le lipsește informația cu privire la problemele de calitate a aerului din țara lor.
- Majoritatea europenilor sunt slab informați în legătură cu standardele de calitate a aerului existente în UE, deoarece doar o minoritate a respondenților (27%) a auzit de ele.
- Totuși, majoritatea respondenților (67%) consideră că standardele de calitate a aerului din UE ar trebui îmbunătățite.

Un chestionar de screening pentru stabilirea percepției asupra poluării aerului și riscului de expunere la poluarea aerului interior și exterior

Pentru dezvoltarea chestionarului, a fost folosit un grup de itemi bazat pe multe recomandări de sondaj standardizat în plus față de mecanismele pentru studii similare legate de poluarea aerului. Itemii au fost scriși cu grijă pentru a minimiza ambiguitatea și crește gradul de înțelegere. În total, grupul a fost compus din 25 de itemi. Chestionarul este un instrument promițător pentru a evalua atitudinile și percepțiile populației în legătură cu poluarea aerului și riscul expunerii la poluarea la interior și exterior. Acest chestionar poate fi utilizat de oameni de știință, cercetători, autorități și experți pentru promovarea sănătății, pentru a dezvolta și implementa programe de protecție a calității aerului.

Chestionarul A – principalii itemi

Vă rugăm citiți toate întrebările și răspundeți prin bifarea căsuței sau oferirea unui răspuns scurt acolo unde este cazul.

Sondajul este anonim și vă asigurăm că păstrăm confidențialitatea răspunsurilor dumneavoastră.

1. Sexul

☐ masculin ☐ feminin

2. Vârsta

☐ sub 3 ani ☐ 3-7 ani ☐ 8-14 ani

☐ 15-20 ani ☐ 21-30 ani ☐ 31-40 ani

☐ 41-50 ani ☐51-60 ani ☐ peste 60 ani

3. În ce regiune locuiți?

Țara..... Regiunea .....

4. Ați spune că locuiți în …

☐ zonă rurală ☐ sat ☐ orășel

☐ oraș mediu ☐ oraș mare

5. Statutul dvs. profesional:

☐ elev ☐ student ☐ liber profesionist ☐ angajat ☐ lucrător manual

☐ fără activitate profesională ☐ refuz să spun ☐ alta.

Vă rugăm specificați.

6. Câți oameni de 15 ani și peste locuiesc în gospodăria dvs., inclusiv dvs.?

☐ 1 ☐ 2 ☐3 ☐ 4 ☐ 5 ☐6 ☐ alt număr ..... Vă rugăm specificați.

7. Care este venitul lunar mediu al familiei dvs. (pe membru)?

☐ sub 300 de euro ☐ 300-600 de euro ☐ 600-1000 de euro
☐ 1000-1500 de euro ☐ peste 1500 de euro ☐ altă sumă ..... ☐ Vă rugăm specificați.

8. Ce tip de încălzire aveți acasă?

☐ încălzire electrică ☐ încălzire pe gaz ☐ aer condiționat ☐ cuptor
☐ încălzire pe lemn /peleți ☐ încălzire solară ☐ alta .....
Vă rugăm specificați.

9. Cât de informat vă simțiți în legătură cu problemele de calitate a aerului din țara dvs.?

☐ foarte bine informat ☐ bine informat ☐ nu foarte bine informat
☐ deloc informat ☐ alta ..... Vă rugăm specificați.

10. Credeți că în ultimii 10 ani, calitatea aerului în țara dvs s-a …?

☐ îmbunătățit ☐ a rămas la fel ☐ deteriorat ☐ alta..... Vă rugăm specificați.

11. Ce impact credeți că au următoarele asupra calității aerului în țara dvs. ?

Are un impact mare, moderat, mic sau niciun impact?

	Impact mare	Impact moderat	Impact mic	Niciun impact
Folosirea rezidențială a energiei (de exemplu, cărbune și lemn pentru încălzirea gospodăriilor individuale)				
Agricultură - emisii de la ferme, fertilizatori și arderea deșeurilor agricole				
Emisii de la mașini și camioane				
Emisii de la transportul internațional (de exemplu, nave și avioane)				
Emisii din producția industrială (oțel, ciment, celuloză, hârtie, etc.) și centralele electrice pe combustibili fosili				
Peisajul				
Râuri / lacuri				
Aer curat				
Alta.....				

12. Care credeți că sunt principale trei tipuri de emisii care amenință calitatea aerului în țara dvs.?

☐ emisii de peste graniță, provenite din alte țări/ regiuni
☐ activitățile de transport
☐ producția de electricitate și căldură
☐ poluanții naturali (sarea de mare, nisipul de deșert, cenușa vulcanică)
☐ activitățile industriale
☐ emisii de la gospodăriile individuale
☐ emisii de la ferme
☐ altele ..... Vă rugăm specificați.

13. Care sunt primele două sisteme de alimentare a mașinii pe care le considerați ca fiind cele mai prietenoase cu mediul din perspectiva calității aerului?

☐ benzină ☐ diesel ☐ biocombustibil
☐ mașini hibride/pe benzină ☐ mașini hibride/diesel
☐ mașini electrice ☐ alta ..... Vă rugăm specificați.

14. Care sunt primele două sisteme de energie pentru încălzirea casei pe care le considerați ca fiind cele mai prietenoase cu mediul din perspectiva calității aerului?

☐ petrol ☐ gaz ☐ cărbune ☐ biomasă (lemn)
☐ biomasă (peleți) ☐ electricitate ☐ termoficare
☐ alta ..... Vă rugăm specificați.

15. Sunt diferite moduri pentru reducerea emisiilor dăunătoare aerului. Pentru a reduce aceste probleme, ați făcut oricare dintre următoarele în ultimii doi ani? Vă rugăm alegeți toate variantele care vi se aplică.

☐ Ați schimbat sistemul de de încălzire a locuinței de la unul cu emisii ridicate (de exemplu, cărbune, petrol sau lemn de foc) la unul cu emisii scăzute (de exemplu, gaz natural, peleți, electricitate)
☐ Ați înlocuit echipamentele vechi consumatoare de energie (boiler pentru apă fierbinte, cuptor, mașină de spălat vase, etc.) cu unele noi, cu o clasă energetică mai bună (de exemplu, A+++ pentru eficiență energetică)
☐ Ați utilizat frecvent transportul public, bicicleta sau mersul pe jos în locul mașinii
☐ Ați cumpărat o mașină cu emisii reduse
☐ Ați cumpărat produse cu emisii scăzute pentru a aprinde focul sau grătarul (de exemplu, brichete în loc de cărbune)
☐ alta ..... Vă rugăm specificați.

16. Ați spune că următoarele reprezintă o problemă foarte serioasă, o problemă destul de serioasă, o problemă nu foarte serioasă sau că nu reprezintă o problemă serioasă în țara dvs.?

	O problemă foarte serioasă	O problemă destul de serioasă	O problemă nu foarte serioasă	Nu reprezintă deloc o problemă serioasă
Bolile respiratorii (de exemplu, bolile pulmonare)				
Bolile cardiovasculare (boli ale inimii)				
Astmul și alergia				
Acidificarea (ploi acide, afectarea pădurilor etc.)				
Eutrofizarea (o creștere a materiei organice într-un ecosistem, cum ar fi creșterea excesivă a algelor care provoacă moartea peștilor în râuri sau lacuri)				

22. Locuința dvs . este situată …?
☐ într-o zonă liniștită, trafic auto redus
☐ într-o zonă gălăgioasă, trafic auto intens
☐ într-o zonă gălăgioasă, din alte cauze decât traficul
☐ alta ..... Vă rugăm specificați.

23. Puteți mirosi gaze de evacuare de la traficul auto în casa dvs.?
☐ da, zilnic ☐ da, des ☐ rar ☐ alta ..... Vă rugăm specificați.

24. În ce măsură experimentați poluarea de la vehicule (zgomot, gaze de evacuare, etc.) în casa dvs.?
☐ foarte mare ☐ medie ☐ scăzută ☐ alta ..... Vă rugăm specificați.

25. Familia dvs are probleme când doarme noaptea (se trezește din cauza zgomotului produs de trafic)?
☐ da, foarte des ☐ des ☐ rar ☐ alta ..... Vă rugăm specificați.

Partea B – o secțiune specială despre sănătatea copiilor

Vă rugăm citiți toate întrebările și răspundeți prin bifarea căsuței sau oferirea unui răspuns scurt acolo unde este cazul.

Sondajul este anonim și vă asigurăm că păstrăm confidențialitatea răspunsurilor dumneavoastră.

1. Sexul
☐ masculin ☐ feminin

2. Vârsta
☐ sub 3 ani ☐ 3-7 ani ☐ 8-15 ani ☐ peste 15 ani

3. Greutatea copiilor
☐ la naștere …… ☐ în prezent …….

4. Vârsta mamei la nașterea copilului
☐ sub 20 ani ☐ 21-30 ani ☐ 31-40 ani
☐ 41-50 ani ☐ peste 50 ani

5. Cât de mult a fost bebelușul alăptat (în luni)?
☐ sub 1 lună ☐ 1-3 luni ☐ 3-6 luni
☐ 6-9 luni ☐ 9-12 luni ☐ alta ..... Vă rugăm specificați.

6. Sunt fumători în familie? Cât de mulți?
☐ 1 ☐ 2 ☐ 3 ☐ alt număr ..... ☐ Vă rugăm specificați dacă mama fumează.

7. Sunt animale de companie în casă? Cât de multe?
☐ da ☐ nu ☐ alta ..... ☐ Vă rugăm specificați.

8. Au vreun părinte sau vreunul din frați/surori o boală alergică?
☐ da ☐ nu ☐ alta ..... ☐ Vă rugăm specificați.

9. Are copilul dvs. (respondent) o boală alergică?
☐ da ☐ nu ☐ alta ..... ☐ Vă rugăm specificați.

10. A avut copilul dvs. (respondent) vreo boală serioasă până acum (nevoie de spitalizare)?
☐ da ☐ nu ☐ alta ..... ☐ Vă rugăm specificați.

11. Suferă copilul dvs. de boli respiratorii (curge nasul, bronșită, pneumonie) mai des de patru ori pe an? ☐ da ☐ nu ☐ alta ..... ☐ Vă rugăm specificați.			
12. A avut copilul dvs. (respondent) vreunul din următoarele simptome în ultimele șase luni?			
	da	nu	Nu știu
Tuse persistentă			
Respirație șuierătoare			
Tuse uscată noaptea			
Febra fânului			
Atacuri de respirație dificilă (astm)			
Gripă sau altă boală ce afectează sistemul respirator			
Ochi injectați (conjunctivită)			

E-CURRICULUM

Această parte a manualului reprezintă curriculumul pentru cursul "Tehnologii avansate de procesare a datelor mari". Acesta este oferită de echipa de la Facultatea de Științele Naturii de la Universitatea Matej Bel University din Banská Bystrica, Slovacia. Partenerul de proiect a implementat deja cursul și toți partenerii îl vor implementa în timpul perioadei de sustenabilitate a proiectului.

Universitatea: Universitatea Matej Bel, Banská Bystrica, Slovacia
Facultatea: Facultatea de Științele Naturii
Cod: DEK FPV/2d-fpv-401
Titlul cursului: Tehnologii avansate de procesare a datelor mari în Științele Naturii
Tipul, gradul de încărcare și metodele activităților educaționale: Tipul cursului: opțional Gradul de încărcare recomandat: 2 ore de seminar/săptămână Metoda de studiu: combinat Forma de studiu: la zi Număr credite: 3 Semestrul recomandat: al doilea semestru al studiilor programului de Master
Programul de studiu: Master anul II
Cursuri prealabile: fără cerințe prealabile
Condiții pentru promovarea și finalizarea cursului: a) evaluarea continuă: participarea activă la exerciții, finalizarea sarcinilor primite: 100 % b) evaluarea finală: 0 % Evaluarea subiectului este în acord cu scala de clasificare determinată de regulamentele de studiu ale UMB.
Rezultatele învățării: 1. Studenții vor dobândi cunoștințe și abilități legate de: 2. Introducere în analiza și procesarea datelor 3. Introducere în sarcinile de bază din analiza datelor – regresie și clasificare 4. Introducere în lucrul cu datele mari – metode de eșantionare a datelor 5. Metode statistice în analiza datelor 6. Bazele Analizei Exploratorii a datelor – teorie și practică 7. Introducere în seturile fuzzy 8. Seturile fuzzy și sarcina de regresie 9. Seturile fuzzy și sarcina de clasificare 10. Introducere în rețelele neuronale 11. În timpul cursului, studentul va dobândi experiență în lucrul cu: 12. Instrumentul software MATLAB 13. Instrumentul software R

BIBLIOGRAFIE

References for sections 1 – 4:

- C.J. Date. An Introduction to Database Systems (8th. ed.). Addison-Wesley Longman Publishing Co., 2003. ISBN: 978-0-321-19784-9
- Felix Kutsanedzie, Sylvester Achio, Edmund Ameko. Practical Approaches to Measurements, Sampling Techniques and Data Analysis. Science Publishing Group, 2016. ISBN: 978-1-940366-58-6.
- William J. Lammers, Pietro Badia. Fundamentals of Behavioral Research Textbook. Online: https://uca.edu/psychology/fundamentals-of-behavioral-research-textbook/
- Jimin Quian et al. Introducing self-organized maps (SOM) as a visualization tool for materials research and education. Results in Materials, Volume 4, 2019, ISSN 2590-048X.
- Naseer Raheem. Big Data: A tutorial-based approach. Chapman and Hal-l/CRC, 2019. ISBN: 978-0-367-67024-5
- Lior Rokach, Oded Maimon. Data mining with decision trees. 2015.
- Steven S. Skiena. The Data Science Design Manual. Springer, 2017. ISBN: 978-3-319-55443-3
- Karthik Ramasubramanian, Abhishek Singh. Machine Learning Using R. Springer, 2019. ISBN: 978-1-4842-4214-8
- Patrik Očenáš. Parallel and distributed methods of big data sampling (in Slovak). 2023.
- Bianka Modrovičová. Decision trees for sizable graph datasets (in Slovak). 2023.
- Aneta Szoliková. Explorative data analysis in document databases (in Slovak). 2023.
- Adam Dudáš, Bianka Modrovičová. Decision Trees in Proper Edge k-coloring of Cubic Graphs. In Proceedings of 33rd FRUCT conference. 2023.

References for sections 5 – 8:

- ZADEH, L. A. Fuzzy Sets. In: Information and Control, 8, 1965, 338-353.
- MICHALÍKOVÁ, A.: Fuzzy množiny v informatike. rec. Mirko Navara, Martin Kalina, Martin Klimo. Belianum. Matej Bel University in Banská Bystrica, 1, 2020, 206p. ISBN 978-80-557-1707-4
- Sendai Subway. Japan Visitor [cit. 2023-02-02]. Online: https://www.japanvisitor.com/japan-transport/sendai-subway
- RUAN D.: Fuzzy Logic Applications in Nuclear Industry. Fuzzy Logic Foundations and Industrial Applications. 1996, 8, ISBN 978-1-4612-8627-1.
- TAKAGI, T., SUGENO, M. Fuzzy Identifications of Fuzzy Systems and its Applications to Modelling and Control. In: IEEE Transactions on Systems, Man, and Cybernetics, 15(1), 1985, 116-132.
- ROSS, T. J. Fuzzy Logic with Engineering Applications. John Wiley & Sons, 2005, 585s., ISBN 9780470743768.
- ZADEH, L. A., The Concept of a Linguistic Variable and its Application to Approximate Reasoning - 1, In: Information Sciences, 8, 1975, 199–249.

References for sections 9

- Ahmed, Z. H. (2010). Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. International Journal of Biometrics & Bioinformatics (IJBB), 3(6), 96.
- Aktaş, M., Yetgin, Z., Kılıç, F., & Sünbül, Ö. (2022). Automated test design using swarm and evolutionary intelligence algorithms. Expert Systems, 39(4), e12918.
- Bartz-Beielstein, T., Branke, J., Mehnen, J., & Mersmann, O. (2014). Evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), 178-195.
- Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. Statistical science, 8(1), 10-15.
- Blickle, T. (2000). Tournament selection. Evolutionary computation, 1, 181-186.
- Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). Multi-objective optimization methods and application in energy saving. Energy, 125, 681-704.
- De La Iglesia, B. (2013). Evolutionary computation for feature selection in classification problems. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(6), 381-407.
- Gaivoronski, A. A., Lisser, A., Lopez, R., & Xu, H. (2011). Knapsack problem with probability constraints. Journal of Global Optimization, 49, 397-413.
- Glover, F., & Laguna, M. (1998). Tabu search (pp. 2093-2229). Springer US.
- Hansen P, Mladenovi'c N (1999) An introduction to variable neighborhood search. In: Voß S, Martello S, Osman IH, Roucairol C (eds) Metaheuristics: advances and trends in local search paradigms for optimization, chapter 30. Kluwer Academic Publishers, Dordrecht, pp 433–458
- Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, 103249.

- Hinson, J. M., & Staddon, J. E. R. (1983). Matching, maximizing, and hill-climbing. Journal of the experimental analysis of behavior, 40(3), 321-331.
- Holland JH. Outline for a logical theory of adaptive systems. J ACM. 1962;9(3):297–314
- Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. SIAM journal on computing, 2(2), 88-105.
- Hoos, H. H., & Stützle, T. (2004). Stochastic local search: Foundations and applications. Elsevier.
- I. Rechenberg, Cybernetic solution path of an experimental problem. Royal Air-craft Establishment, Library Translation 1122, Farnborough, Reprint in: D.B. Fogel (Ed.), Evolutionary Computation, The Fossil Record, IEEE Press, Piscataway, NJ, 1965, pp. 301–309
- I. Rechenberg, Evolutionsstrategie—Optimisierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973
- Kılıç, F., Yılmaz, İ. H., & Kaya, Ö. (2021). Adaptive co-optimization of artificial neural networks using evolutionary algorithm for global radiation forecasting. Renewable Energy, 171, 176-190.
- Kiliç, F., & Gök, M. (2013). A public transit network route generation algorithm. IFAC Proceedings Volumes, 46(25), 162-166.
- Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K., & China, H. (2013). Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. gene, 7(33), 8.
- Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133.
- Rossi, F., Van Beek, P., & Walsh, T. (Eds.). (2006). Handbook of constraint programming. Elsevier.
- Salkin, H. M., & De Kluyver, C. A. (1975). The knapsack problem: a survey. Naval Research Logistics Quarterly, 22(1), 127-144.
- Sharifi, A. A., & Aghdam, M. H. (2019). A novel hybrid genetic algorithm to reduce the peak-to-average power ratio of OFDM signals. Computers & Electrical Engineering, 80, 106498.
- Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 114, 105082.
- Yang, J., & Soh, C. K. (1997). Structural optimization by genetic algorithms with tournament selection. Journal of computing in civil engineering, 11(3), 195-200.

References for section 10:

- Basic Neural Networks 1 - https://docs.google.com/a/atu.edu.tr/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxpaHNhbnlhc3NpbjJ8Z3g6NGY4MjNjN2Y4ZTdhNWM2MQ
- Basic Neural Networks 2 - http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/
- Basic Neural Networks 3

  https://www.cs.bham.ac.uk/~jxb/inn.html
- Basic Neural Network 4

  https://www.fer.unizg.hr/en/course/neunet_a/lecture_notes
- Basic Neural Network 5

  http://users.monash.edu/~cema/courses/FIT3094/lecturePDFs/

References for section 11:

- Paluszek, M., Thomas, S. Matlab machine learning recepies. 2019. Plainsboro, NJ, USA. ISBN-13 (pbk): 978-1-4842-3915-5. DOI 10.1007/978-1-4842-3916-2.
- Kim, P. MATLAB Deep Learning. With Machine Learning, Neural Networks and Artificial Intelligence. 2017. Apress Korea ISBN-13 (pbk): 978-1-4842-2844-9. DOI 10.1007/978-1-4842-2845-6.
- Get Started with Matlab. https://www.mathworks.com/help/matlab/getting-started-with-matlab.html
- Iris Clustering. https://www.mathworks.com/help/deeplearning/ug/iris-clustering.html

References for Appendices:

- Fisher, R.A. (1936) "The use of multiple measurements in taxonomic problems". Annual Eugenics, 7, Part II, pages 179-188
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, pages 431-433
- Duda, R.O., Hart, P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1, page 218
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recogni-

tion in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, pages 67-71
- https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3/data-products
- https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-4/data-products
- https://climexp.knmi.nl/
- https://www.uradmonitor.com/
- Velea L, Udriștioiu MT, Puiu S, Motișan R, Amarie (2023)D. A Community-Based Sensor Network for Monitoring the Air Quality in Urban Romania. Atmosphere; 14(5):840. https://doi.org/10.3390/atmos14050840
- https://bookdown.org/floriandierickx/bookdown-demo/climate-data-from-models.html#differences-between-climate-projections-predictions-and-scenarios
- https://ec.europa.eu/eurostat/web/climate-change/database
- https://ourworldindata.org/
- https://ourworldindata.org/data-review-air-pollution-deaths
- •https://ourworldindata.org/outdoor-air-pollution#outdoor-air-pollution-deaths-by-age
- •https://www.who.int/health-topics/air-pollution#tab=tab_1
- •https://www.eea.europa.eu/en/topics/in-depth/air-pollution
- •https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/health-impacts/types-of-pollutants
- https://www.who.int/publications/i/item/9789240034228
- •https://apps.who.int/iris/bitstream/handle/10665/345329/9789240034228-eng.pdf
- EEA, 2012, The contribution of transport to air quality, EEA Report no. 10/2012, European Environment Agency.
- EEA. A closer look at urban transport TERM 2013: transport indicators tracking progress towards environmental targets in Europe EEA Report No 11/2013 Copenhagen, ISSN 1725-9177.
- http://dx.doi.org/10.1016/j.envpol.2007.06.012
- https://www.who.int/health-topics/air-pollution#tab=tab_1
- Report no. 05/2022, Air quality in Europe 2022. doi: 10.2800/488115. https://www.eea.europa.eu/publications/air-quality-in-europe-2022
- Xin Zhang, X. Chen, Xiaobo Zhang. The impact of exposure to air pollution on cognitive performance. Proc. Natl. Acad. Sci. Unit. States Am., 115 (2018), pp. 9193-9197, 10.1073/pnas.1809474115
- J. Currie, J.S.G. Zivin, J. Mullins, M.J. Neidell. What do we know about short and long term effects of early life exposure to pollution? NBER Work. Pap., 6 (2013), pp. 217-247, 10.3386/w19571
- Escamilla-Nuñez M-C., Barraza-Villarreal A., Hernandez-Cadena L., Moreno-Macias H., Ramirez-Aguilar M., Sienra-Monge J-J., Cortez-Lugo M., Texcalac J-L., del Rio-Navarro B., Romieu I. Traffic-Related Air Pollution and Respiratory Symptoms Among Asthmatic Children, Resident in Mexico City: The EVA Cohort Study. http://www.medscape.com/viewarticle/585875.
- Juvin P., Fournier T., Boland S. et al. Diesel particles are taken up by alveolar type II tumor cells and alter cytokines secretion. Arch Environ Health. 2002; 57(1):53-60.
- Le Tertre A., S. Medina, E. Samoli et al: Short term effects of particulate air pollution on cardiovascular disease in eight European cities. J. Epidemiol Community Health, 2002; 56, (10):773-9.
- Nordling E., Berglind N., Melén E., Emenius G., Hallberg J., Nyberg F., Pershagen G., Svartengren M., Wickman M., Bellander T. Traffic related air pollution and childhood respiratory symptoms, function and allergies. Epidemiology. 2008; 19(3):401-8.
- Pan G., Zhang S., Feng Y., Takahashi K., Kagawa J., Yu L., Wang P., Liu M., Liu Q., Hou S., Pan B., Li J. Air pollution and children's respiratory symptoms in six cities of Northern China. Respiratory Medicine 2010;104(12):1903–11.
- Richardson E.A., Pearce J., Tunstall H., Mitchell R., Shortt N.K.: Particulate air pollution and health inequalities: a Europe-wide ecological analysis. Int J Health Geogr 2013;12:34
- I. Jáuregui, J. Mullol, I. Dávila, M. Ferrer, J. Bartra, A. Del Cuvillo, J. Montoro, J. Sastre, A. Valero. Allergic rhinitis and school performance. J Investig. Allergol. Clin. Immunol., 19 (2009), pp. 32-39
- D.P. Skoner. Allergic rhinitis: definition, epidemiology, pathophysiology, detection, and diagnosis. J. Allergy Clin. Immunol., 108 (2001), pp. 2-8, 10.1067/mai.2001.115569
- I. Beck, S. Jochner, S. Gilles, M. McIntyre, J.T.M. Buters, C. Schmidt-Weber, H. Behrendt, J. Ring, A. Menzel, C. Traidl-Hoffmann. High environmental ozone levels lead to enhanced allergenicity of birch pollen. PloS One, 8 (2013), 10.1371/journal.pone.0080147
- P. Sturdy, S. Bremner, G. Harper, L. Mayhew, S. Eldridge, J. Eversley, A. Sheikh, S. Hunter, K. Boomla, G. Feder, K. Prescott, C. Griffiths. Impact of asthma on educational attainment in a socioeconomically deprived population: a study linking health, education and social care datasets. PloS One, 7 (2012), pp. 1-8, 10.1371/journal.pone.0043977
- https://europa.eu/eurobarometer/surveys/detail/2660
- https://data.europa.eu/data/datasets/s2660_97_2_sp524_eng?locale=en
- https://www.surveymonkey.com/r/airpollutionperceptionsurvey
- https://apps.who.int/iris/rest/bitstreams/1350812/retrieve
- https://www.ab.gov.tr/files/ardb/evt/Attitudes_of_Europeans_towards_air_quality_2013.pdf