ACADEMIA DE STUDII ECONOMICE DIN BUCUREŞTI

Facultatea de Cibernetică, Statistică și Informatică Economică

PROIECT ICE

ANALIZA COSTURILOR EDUCATIEI INTERNATIONALE

Profesor coordonator:

Cadru asociat drd. DOMENTEANU ADRIAN

Student:

Maria-Alexandra IORDAN

București

2025

CUPRINS

Introducere

În contextul creșterii mobilității educaționale și al interesului tot mai mare pentru studiile universitare în afara granițelor naționale, analiza costurilor asociate educației internaționale devine esențială pentru studenți, consultanți educaționali si economiști. Prin urmare, in aceasta analiza, a fost utilizat un set de date obținut de pe platforma Kaggle, intitulat "International_Education_Costs", care oferă informații detaliate privind cheltuielile asociate studiilor superioare în diverse țări, orașe și universități din întreaga lume.

In ceea ce priveste structura setului de date, acesta a fost prelucrat in prealabil pentru a asigura calitatea analizei. Acesta nu contine valori lipsa, iar valorile extreme (outlierii) au fost identificate și eliminate, deoarece prezenta acestora afecta negativ analiza, distorsionand structura datelor si reducand relevanta rezultatelor obtinute. În forma sa inițială, setul conținea 907 observații, iar după procesul de curățare, au ramas 794 observații valide.

Setul de date este compus din urmatoarele 10 variabile, atat numerice, cat si categoriale:

- **Country** (string) – tara in care este localizata institutia de învățământ superior.
- **City** (string) – orasul în care se află instituția.
- **University** (string) – numele oficial al instituitiei.
- **Program** (string) – denumirea programului educațional sau a specializării.
- **Level** (string) – nivelul de studiu al programului.
- **Duration_Years** (integer) – durata programului exprimată în ani.
- **Tuition_USD** (numeric) – taxa totală de școlarizare a programului, convertită în dolari americani, pentru a facilita comparațiile între țări.
- **Living_Cost_Index** (numeric) – un indice standardizat care reflectă costurile de trai zilnic (precum mâncare, transport, utilități) în orașul respectiv, permițând comparații între regiuni.
- **Rent_USD** (numeric) – valoarea medie lunară a chiriei pentru cazare studențească, exprimată în dolari americani.
- **Visa_Fee_USD** (numeric) – taxa unică de viză percepută studenților internaționali, exprimată în dolari americani.

In ceea ce priveste observatiile, fiecare linie din setul de date corespunde unui program de studii international, oferind o imagine de ansamblu asupra costurilor prin intermediul variabilelor prezentate. Aceasta permite compararea programelor de studii in functie de diferiti indicatori. Acest set de date oferă o bază solidă pentru analiza comparativă a costurilor de educație internațională și pentru investigarea accesibilității financiare a studiilor superioare în diferite părți ale lumii.

Obiectivul general al acestei analize este de a explora, segmenta și modela costurile educației internaționale în funcție de diferite variabile, folosind mai multe metode de învățare automată și analiză statistică. Prin aplicarea unor tehnici precum clusterizarea fuzzy, regresia logistică (binomială și multinomială), arborii de decizie, algoritmul KNN și rețelele neuronale artificiale, se urmărește identificarea tiparelor dominante în structura costurilor,

clasificarea programelor educaționale pe baza caracteristicilor lor și estimarea probabilităților de apartenență la anumite categorii definite de cost sau regiune geografică.

1. *Descrierea datelor. Statistici descriptive si reprezentari grafice.*

Ca un prim pas în cadrul analizei, au fost realizate statistici descriptive și reprezentări grafice pentru a analiza structura setului de date și a înțelege distribuția variabilelor.

Interpretarea statisticilor descriptive:

Duration_Years	Tuition_USD	Living_Cost_Index	Rent_USD	Visa_Fee_USD
Min. :1.000	Min. : 0	Min. :32.50	Min. : 160.0	Min. : 40.0
1st Qu.:2.000	1st Qu.: 2600	1st Qu.:53.62	1st Qu.: 500.0	1st Qu.:100.0
Median :2.000	Median : 6500	Median :66.20	Median : 900.0	Median :150.0
Mean :2.855	Mean :15269	Mean :63.47	Mean : 936.5	Mean :181.4
3rd Qu.:4.000	3rd Qu.:29150	3rd Qu.:71.80	3rd Qu.:1300.0	3rd Qu.:235.0
Max. :5.000	Max. :58000	Max. :95.20	Max. :2400.0	Max. :450.0

Figură 1. Statistici descriptive, utilizand summary

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Duration_Years	1	794	2.86	0.95	2.0	2.80	0.00	1.0	5.0	4.0	0.42	-1.40	0.03
Tuition_USD	2	794	15268.95	16556.20	6500.0	12997.88	9636.90	0.0	58000.0	58000.0	0.92	-0.51	587.56
Living_Cost_Index	3	794	63.47	13.18	66.2	63.92	10.67	32.5	95.2	62.7	-0.40	-0.57	0.47
Rent_USD	4	794	936.54	511.35	900.0	895.14	593.04	160.0	2400.0	2240.0	0.60	-0.34	18.15
Visa_Fee_USD	5	794	181.36	115.79	150.0	161.44	75.61	40.0	450.0	410.0	1.39	0.77	4.11

Figură 2. Statistici descriptive, utilizand describe

Duration_Years: Valoarea medie de 2.86 ani si mediana de 2 ani indica faptul că majoritatea programelor analizate au o durată de aproximativ 2–3 ani. Abaterea standard de 0.95 arată o variabilitate moderată a duratelor, iar valorile variază între un minim de 1 an și un maxim de 5 ani. Distributia este usor asimetrica pozitiv, ceea ce sugerează o concentrare mai mare a programelor cu durată scurtă, dar cu prezența unor programe mai lungi care ridică media. Kurtosisul negativ, de -1.40, indica o distribuție mai plată decât cea normală, cu valori mai puțin concentrate în jurul mediei.

Tuition_USD: Costul total al taxei de scolarizare prezinta o valoare medie de aproximativ 15.269 USD, în timp ce mediana este de 6.500 USD, sugerând o distribuție asimetrică, cu un număr semnificativ de programe cu taxe reduse și câteva programe cu taxe foarte ridicate care trag media în sus. Abaterea standard ridicata indică o variabilitate semnificativă intre programele analizate. Valoarea minimă este 0 USD, semnalând existența unor programe gratuite, iar valoarea maximă este de 58.000 USD, ceea ce evidențiază un interval larg al costurilor. Coeficientul de asimetrie confirmă faptul că distribuția este asimetrică pozitiv, iar cel de kurtosis sugerează o distribuție mai plată decât normalul, cu valori dispersate.

Living_Cost_Index: Valoarea medie a indicelui costului de trai este de 63.47, mediana fiind putin mai mare (66.2), ceea ce indica o distribuție ușor asimetrică. Valorile sunt relativ bine concentrate, după cum indică abaterea standard de 13.18, iar indicele variază de la un minim de 32.5 (cost de trai mic) la un maxim de 95.2 (cost de trai foarte mare), rezultând un interval de 62.7 puncte. Eroarea standard de 0.47 sugerează o estimare precisă a mediei. În ansamblu, majoritatea orașelor din setul de date au un cost de trai moderat, cu variații relativ uniforme.

Rent_USD: Costul mediu al chiriei este de 936.54 USD, iar mediana este de 900 USD, ceea ce sugereaza o distributie ușor asimetrică pozitiv (skew = 0.60), cu unele valori mai ridicate care influențează media. Abaterea standard de 511.35 USD relevă o variație considerabilă a costurilor de cazare în funcție de oraș și țară. Valoarea minimă este de 160 USD, în timp ce chiria maximă atinge 2.400 USD, ceea ce indica diferente semnificative intre locatiile programelor de studii. Eroarea standard de 18.15 USD indică o estimare destul de precisă a mediei. În concluzie, costurile de cazare pentru studenți internaționali variază semnificativ, cu o tendință generală spre chirii moderate, dar cu prezența unor centre universitare cu costuri ridicate.

Visa_Fee_USD: Taxa unica de viza este, in medie, de 181.36 USD. In comparatie cu valoarea medie, mediana este mai scazuta (150 USD), ceea ce sugereaza o distribuție asimetrică pozitiv (asimetrie: 1.39), cu numeroase taxe moderate și câteva semnificativ mai mari care trag media în sus. Valoarea minimă este de 40 USD, iar cea maximă de 450 USD, rezultând un interval de 410 USD. Eroarea standard de 4.11 USD sugerează o estimare destul de precisă a mediei, în ciuda variabilității. Astfel, taxele de viză variază considerabil între destinațiile educaționale, unele fiind mai accesibile, altele impunând costuri administrative mai ridicate.

Matricea de corelatie:

	Duration_Years	Tuition_USD	Living_Cost_Index	Rent_USD	Visa_Fee_USD
Duration_Years	1.00000000	0.1905010	-0.04689269	0.08548305	0.05310248
Tuition_USD	0.19050101	1.0000000	0.41726625	0.74515772	0.47586113
Living_Cost_Index	-0.04689269	0.4172663	1.00000000	0.81007390	0.23357570
Rent_USD	0.08548305	0.7451577	0.81007390	1.00000000	0.37788460
Visa_Fee_USD	0.05310248	0.4758611	0.23357570	0.37788460	1.00000000

Figură 3. Matricea de corelatie

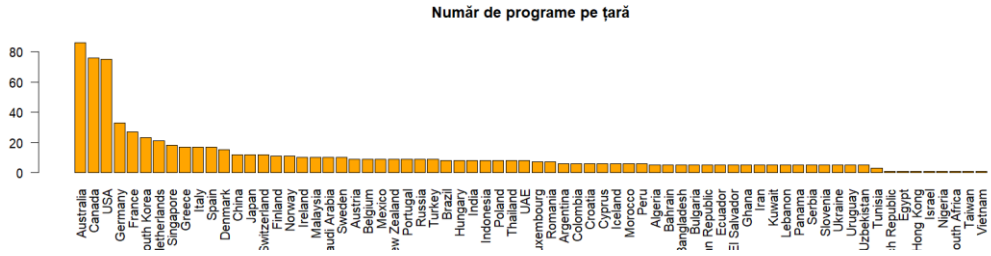
In urma analizarii matricei de corelatie, putem deduce urmatoarele:

- Durata programelor prezinta corelatii destul de slabe (relatii nesemnificative) in raport cu celelalte variabile, insa are o corelatie slab pozitiva cu taxa de scolarizare (0.19), ceea ce sugereaza ca programele mai lungi tind sa aiba taxe de scolarizare mai mari.
- Taxa de scolarizare este puternic corelata pozitiv cu costul chiriei (0.75) si moderat cu indicele costului de trai (0.42) si taxa de viza (0.48), ceea ce sugereaza ca in locatiile cu taxe de scolarizare ridicate, si costurile de trai, chiria si taxele de viza tind sa fie mai mari – o asociere logică în contextul centrelor educaționale scumpe, cum ar fi America.
- Indicele costului de trai este foarte puternic corelat cu chiria (0.81), ceea ce indică faptul că o mare parte din costurile de trai sunt influențate de chirie.
- In concluzie, cele mai puternice relatii sunt intre indicele costului de trai si chirie, si intre chirie si taxa de scolarizare, ceea ce arata efectul de amplificare reciproca intre costurile de trai, chirii si taxele universitare in marile centre educationale internationale.

Frecventele de aparitie:

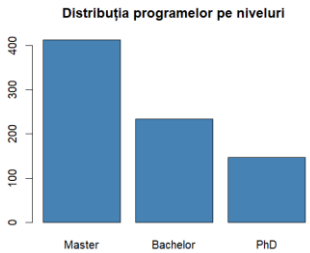
```
table(date_proiect$Country)

     Algeria    Argentina    Australia      Austria      Bahrain   Bangladesh 
           5            6           86            9            5            5 
     Belgium       Brazil     Bulgaria       Canada        China     Colombia 
           9            8            5           76           12            6 
     Croatia       Cyprus Czech Republic      Denmark Dominican Republic      Ecuador 
           6            6            1           15            5            5 
       Egypt  El Salvador      Finland       France      Germany        Ghana 
           1            5           11           27           33            5 
      Greece    Hong Kong      Hungary      Iceland        India    Indonesia 
          17            1            8            6            8            8 
        Iran      Ireland       Israel        Italy        Japan       Kuwait 
           5           10            1           17           12            5 
     Lebanon   Luxembourg     Malaysia       Mexico      Morocco  Netherlands 
           5            7           10            9            6           21 
 New Zealand      Nigeria       Norway       Panama         Peru       Poland 
           9            1           11            5            6            8 
    Portugal      Romania       Russia Saudi Arabia       Serbia    Singapore 
           9            7            9           10            5           18 
    Slovenia South Africa  South Korea        Spain       Sweden  Switzerland 
           5            1           23           17           10           12 
      Taiwan     Thailand      Tunisia       Turkey          UAE      Ukraine 
           1            8            3            9            8            5 
     Uruguay          USA   Uzbekistan      Vietnam 
           5           75            5            1 
```

Figură 4. Frecventa tarilor



Figură 5. Barplot

Din figura 4 se observa ca cele mai multe programe de studii sunt localizate in Australia (86), Canada (76) si USA (75). Acest lucru reflectă statutul acestor țări ca fiind principale centre universitare la nivel internațional, atrăgând un număr mare de studenți străini datorită calității educației, infrastructurii universitare și diversității programelor oferite. La polul opus, țări precum Africa de Sud, Taiwan și Vietnam dispun de un număr redus de opțiuni educaționale disponibile pentru studenții internaționali.



Figură 6. Barplot

```
> table(date_proiect$Level)

Bachelor   Master      PhD 
     234      413      147 
```

Figură 7. Frecventa tipurilor de programe

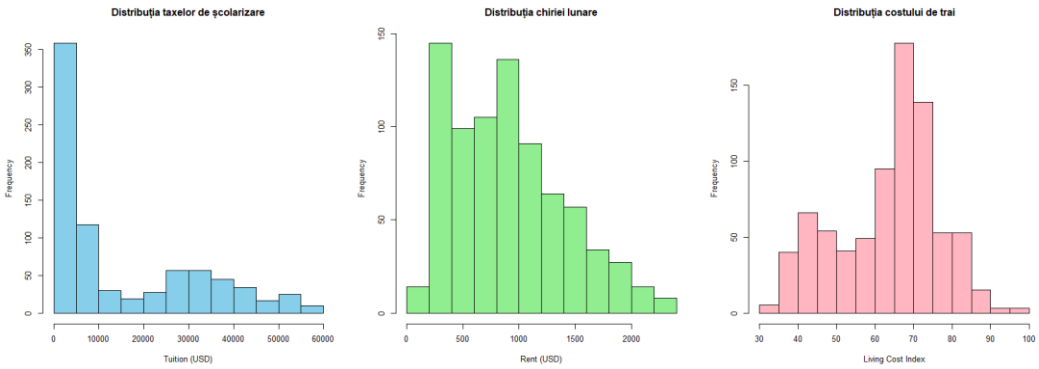
Figurile anterioare ilustreaza distribuția numărului de observații în funcție de nivelul de studii, evidențiind faptul că cele mai multe programe sunt de Master (413), urmate de cele de Bachelor (234), iar cele mai puține programe sunt la nivel de PhD (147). Acest lucru reflecta interesul ridicat pentru studiile postuniversitare de masterat în rândul studenților internaționali.

```
> table(date_proiect$Country, date_proiect$Level)

                    Bachelor Master PhD
Algeria                    2      2   1
Argentina                  2      2   2
Australia                 28     31  27
Austria                    4      4   1
Bahrain                    2      2   1
Bangladesh                 0      3   2
Belgium                    2      4   3
Brazil                     1      5   2
Bulgaria                   1      3   1
Canada                    34     38   4
China                      0     12   0
Colombia                   1      3   2
Croatia                    1      3   2
Cyprus                     2      3   1
Czech Republic             1      0   0
Denmark                    2     11   2
Dominican Republic         2      2   1
Ecuador                    1      2   2
```

Figură 8. Frecventa incrucisata a tarilor si a nivelelor de studii

În această figură este prezentată frecvența nivelelor de studii în funcție de țară, evidențiind cum sunt distribuite programele de Bachelor, Master și PhD în diferite destinatii educationale, oferind o perspectiva asupra ofertei academice specifice fiecarei locatii. De exemplu, țările mai puțin dezvoltate, precum Algeria, Bangladesh și Ecuador, prezintă o ofertă educațională mai restrânsă, având disponibile un număr scăzut de programe de studii. În contrast, țări dezvoltate precum Australia și Canada oferă un număr mult mai ridicat de programe, reflectând atât capacitatea instituțiilor educaționale din aceste țări, cât și atractivitatea lor pentru studenții internaționali.

Reprezentari grafice:



Figură 9. Histograme

Analizand histogramele, observam urmatoarele:

- Distributia taxelor de scolarizare este asimetrica pozitiv, ceea ce indica prezenta unor valori mai ridicate care ridica media. Aceste valori ridicate distorsionează media și sugerează că, deși există programe cu taxe foarte mari, ele nu reprezintă majoritatea. Majoritatea valorilor se concentreaza in intervalul 0-10.000 USD, valoarea 0 reflectand existenta unor state/institutii care ofera educatie gratuita pentru studentii internationali.
- Distributia chiriei lunare prezinta, de asemenea, o usoara asimetrie pozitiva, cauzată de prezența unor valori mai ridicate specifice țărilor cu un cost al vieții mai mare. Cu toate acestea, cea mai mare parte a valorilor se regăsește în intervalul 500–1000 USD, evidențiind un nivel mediu al cheltuielilor de cazare.

- Distribuția indicelui costului de trai prezintă o ușoară asimetrie, însă variațiile sunt moderate. Majoritatea valorilor sunt concentrate în intervalul 60–80, ceea ce indică un nivel relativ uniform al costurilor de trai în rândul orașelor analizate. Totuși, existența câtorva valori mai scăzute sau mai ridicate influențează media, reflectând diversitatea condițiilor economice între diferitele locații.

2. *Clusterizare Fuzzy*

Acest capitol are ca scop identificarea unor tipare comune în cadrul setului de date prin gruparea observațiilor în trei clustere distincte, corespunzătoare nivelurilor de cost ale educației internaționale: accesibil, mediu și scump. Prin aplicarea metodei de *Clusterizare Fuzzy*, se permite atribuirea flexibilă a observațiilor către mai multe grupuri, în funcție de gradul de apartenență, reflectând variațiile și suprapunerile dintre nivelurile de cost. Această abordare oferă o segmentare a programelor educaționale în funcție de variabilele numerice, (durata studiilor, taxa de scolarizare, costurile de trai, chiria si taxa de viza), facilitand interpretarea accesibilității educației în diverse țări și instituții.

In urma aplicarii clusterizarii Fuzzy s-au obtinut urmatoarele rezultate:

```
Fuzzy c-means clustering with 3 clusters

Cluster centers:
  Duration_Years Tuition_USD Living_Cost_Index  Rent_USD Visa_Fee_USD
1       2.730098    3667.928           59.54878  673.3188     130.4761
2       3.132579   29077.257           67.14384 1191.6272     299.4227
3       3.134523   46979.804           75.41636 1776.4254     224.7908

Memberships:
                1             2             3
1   2.350638e-02 9.073445e-02 8.857592e-01
3   3.171996e-02 4.329247e-01 5.353554e-01
4   1.457312e-02 1.282277e-01 8.571991e-01
5   9.830475e-01 1.230259e-02 4.649902e-03
6   9.196477e-01 6.273955e-02 1.761277e-02
7   5.024116e-01 4.211754e-01 7.641302e-02
8   2.821016e-02 7.785711e-01 1.932187e-01
9   9.973082e-01 2.016699e-03 6.750954e-04
10  9.877618e-01 8.942818e-03 3.295370e-03
```

```
Closest hard clustering:
  1   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  24  25  26  27  28  29  30
  3   3   3   1   1   1   2   1   1   1   1   1   1   2   1   2   1   1   3   3   3   2   2   1   1   1   1   3
 31  34  35  36  37  38  39  40  41  42  43  45  46  47  48  49  50  51  53  54  55  56  57  58  59  60  61  62
  3   1   1   1   1   2   2   1   3   3   2   1   1   1   2   1   1   3   2   2   1   1   1   1   1   1   1   1
 63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  1   1   1   1   1   1   1   1   1   2   2   2   2   2   2   2   2   2   2   1   1   1   1   1   1   1   1   1
 91  92  93  94  95  96  97  98  99 100 101 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
  1   3   3   3   3   3   3   3   3   3   3   2   2   2   2   2   2   2   2   2   2   3   3   2   1   1   1   1
129 130 131 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
  1   1   2   3   1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   2   2   2   2   2
158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185
  2   2   3   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 285 286 287 288 289 290 291
  1   3   3   3   3   3   3   3   3   3   3   2   2   2   2   2   2   2   2   2   2   3   2   3   2   3   2   3
```

Figură 10. Rezultat clusterizare

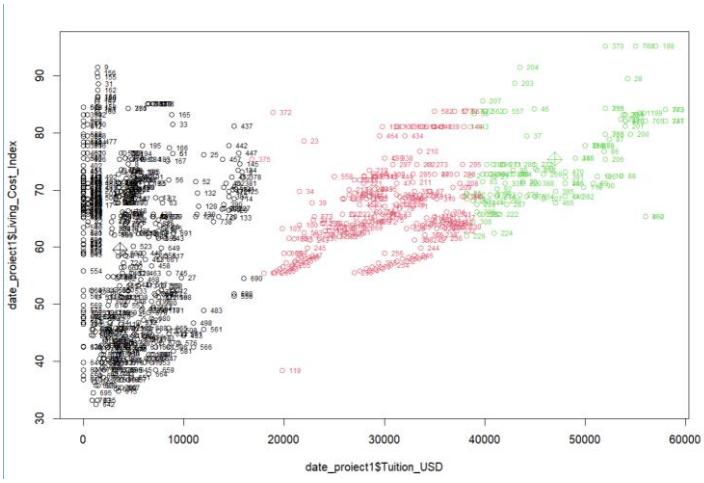
Clusterizarea fuzzy a permis gruparea programelor educaționale internaționale în trei categorii, în funcție de costurile asociate educației: accesibile, medii și ridicate. Analiza centroizilor (mediile clusterelor) evidențiază faptul că durata medie a studiilor este relativ constantă între clustere, situându-se în jurul a 3 ani: 2,73 ani pentru clusterul 1, 3,13 ani pentru clusterul 2 și tot 3,13 ani pentru clusterul 3. Diferențele majore apar însă în ceea ce privește costurile. Taxele medii de școlarizare sunt cele mai reduse în clusterul 1 (aproximativ

3.668 USD), urmate de cele din clusterul 2 (29.077 USD), iar cele mai ridicate aparțin clusterului 3 (46.980 USD). Indicele cheltuielilor de trai crește progresiv de la un cluster la altul: 59,54 pentru clusterul 1, 67,14 pentru clusterul 2 și 75,41 pentru clusterul 3. Aceeași tendință ascendentă este vizibilă și în cazul chiriei lunare, cu o valoare medie de 673 USD în primul cluster, 1.191 USD în al doilea și 1.776 USD în al treilea. Costul vizelor urmează o distribuție ușor diferită, fiind cel mai redus în clusterul 1 (130 USD), cel mai mare în clusterul 2 (299 USD), și intermediar în clusterul 3 (225 USD).

Pe baza acestor caracteristici, se pot interpreta astfel cele trei clustere:

- ***Clusterul 1*** - Educatie accesibila/cu costuri reduse - include tari cu cost total redus al educatiei internationale.
- ***Clusterul 2*** - Educatie cu costuri medii - reprezinta un compromis intre cost si calitate, cu taxe si costuri moderate.
- ***Clusterul 3*** - Educatie foarte scumpa/cu costuri ridicate - include tari in care costurile sunt foarte ridicate, reflectand un sistem educational "premium".

În ceea ce privește gradul de apartenență (membership), spre exemplu, prima observație analizată – programul de Master în Computer Science de la Harvard – are un grad foarte ridicat de apartenență la clusterul 3 (0,88), ceea ce confirmă faptul că acesta se încadrează în categoria programelor educaționale cu costuri ridicate.

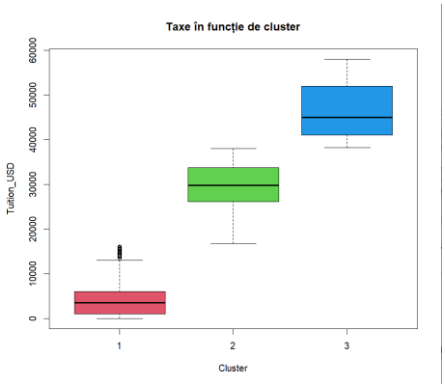


Figură 11. Reprezentarea grafica a observatiilor

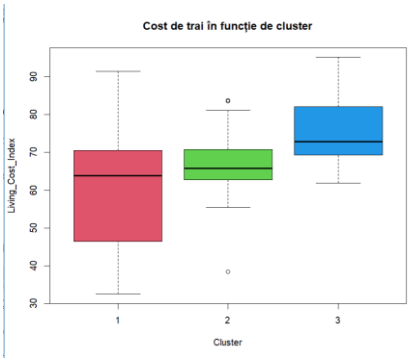
Reprezentarea grafică a observațiilor într-un sistem de axe XOY, având pe axa Ox taxele de școlarizare (Tuition_USD) și pe axa Oy indicele costului de trai (Living_Cost_Index), evidențiază distribuția programelor educaționale în funcție de costuri. Observațiile au fost colorate în funcție de clusterul din care fac parte, iar centroizii au fost marcați distinct prin simboluri de tip romb.

Observațiile din clusterul 1, reprezentate cu negru, se caracterizează prin taxe de școlarizare scăzute și, în majoritatea cazurilor, costuri de trai moderate. Totuși, există câteva puncte izolate care indică valori foarte ridicate ale costurilor de trai, ce pot corespunde unor programe de studii din țări foarte dezvoltate, unde, în ciuda nivelului general ridicat al cheltuielilor, sunt oferite finanțări sau facilități care reduc costurile pentru studenții

internaționali. Clusterul 2, reprezentat cu roșu, include observații cu taxe de școlarizare moderate și un nivel mediu al costurilor de trai, sugerând un echilibru între accesibilitate și cheltuieli. În schimb, observațiile din clusterul 3, marcate cu verde, indică cele mai ridicate taxe de școlarizare și costuri de trai superioare celorlalte două clustere, ceea ce sugerează că acest grup reunește programele educaționale cele mai costisitoare.



Figură 12. Reprezentarea grafica a taxelor in functie de cluster



Figură 13. Reprezentarea grafica a costului de trai in functie de cluster

Figurile 12 și 13 confirmă interpretările anterioare. Primul cluster este caracterizat de taxe de școlarizare foarte reduse, deși apar și câteva valori extreme mai ridicate. Clusterul 2 include universități cu taxe moderate, iar clusterul 3 grupează instituțiile cu cele mai mari taxe de școlarizare. În ceea ce privește costurile de trai, se observă o creștere progresivă a valorilor de la clusterul 1 la clusterul 3, ceea ce sugerează o corelație între nivelul taxelor și costurile generale asociate studiilor.

Pentru a facilita vizualizarea rezultatelor clusterizarii, am ordonat observațiile în funcție de apartenența la cluster, în ordine crescătoare. Această operațiune a fost realizată cu ajutorul funcției order(rez$cluster), care returnează un vector de indici ce rearanjează observațiile în funcție de numărul clusterului atribuit fiecărei unități. În plus, au fost create doua data frame-uri, unul care contine denumirea fiecărei universități, alături de clusterul din care face parte si unul care contine denumirea fiecarei universitati si gradele de apartenenta la cele trei clustere, pentru a evidenția clar distribuția programelor educaționale.

```
  [1]   4   5   6   8   9  10  11  12  13  15  17  18  24  25  26  27  30  31  32  33  36  40  41  42  44  45
 [27]  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  76  77  78  79  80  81  82  83  84
 [53]  85 109 110 111 112 113 114 117 118 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 145 146
 [79] 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
[105] 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 332
[131] 336 340 343 347 351 355 359 363 367 371 374 377 378 380 381 383 384 386 387 389 390 392 393 394 395 396
[157] 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422
[183] 423 424 425 426 427 428 429 430 431 432 433 436 437 438 441 442 443 446 447 448 451 452 453 456 457 458
[209] 461 462 463 466 467 468 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490
[235] 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516
[261] 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542
[287] 543 544 545 546 547 548 549 550 551 552 553 554 555 556 559 560 561 564 565 566 569 570 571 574 575 576
[313] 579 580 581 584 585 586 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607
[339] 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633
[365] 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659
[391] 660 661 662 663 664 665 667 668 669 670 671 672 673 675 676 677 678 679 680 681 682 683 684 685 686 687
[417] 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713
[443] 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739
[469] 740 741 742 743 744 745 746 748 749 750 751 752 754 755 756 757 758 760 761 762 763 764 766 767 768 769
[495] 770 771 772 773 774 775 777 778 779 780 781 783 784 785 786 787 789 790 791 792 793   7  14  16  22  23
[521]  34  35  39  43  47  48  66  67  68  69  70  71  72  73  74  75  96  97  98  99 100 101 102 103 104 105
[547] 108 115 119 135 136 137 138 139 140 141 142 144 208 209 210 211 212 213 214 215 216 217 219 221 223 225
[573] 227 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253
[599] 254 255 256 257 258 259 260 261 262 263 264 265 266 267 270 273 276 279 282 285 288 291 294 295 297 298
[625] 300 301 303 304 306 307 309 310 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329
[651] 330 334 338 341 342 345 348 349 352 353 356 357 360 361 364 365 368 369 372 375 434 439 444 449 454 459
[677] 464 469 558 563 567 568 573 577 578 582 583 587 666 674   1   2   3  19  20  21  28  29  37  38  46  86
[703]  87  88  89  90  91  92  93  94  95 106 107 116 143 198 199 200 201 202 203 204 205 206 207 218 220 222
[729] 224 226 228 268 269 271 272 274 275 277 278 280 281 283 284 286 287 289 290 292 293 296 299 302 305 308
[755] 311 331 333 335 337 339 344 346 350 354 358 362 366 370 373 376 379 382 385 388 391 435 440 445 450 455
[781] 460 465 470 557 562 572 747 753 759 765 776 782 788 794
```

Figură 14. Ordonarea crescatoare a observatiilor dupa cluster

	date_proiect.University.ordine.	rez.cluster.ordine.
5	Technical University of Munich	1
6	University of Tokyo	1
7	University of Amsterdam	1
9	Sorbonne University	1
10	ETH Zurich	1
11	KTH Royal Institute	1
12	University of Copenhagen	1
13	Tsinghua University	1
14	Seoul National University	1
16	Pusan National University	1
18	University of Vienna	1
19	KU Leuven	1
26	University of Lisbon	1
27	Tel Aviv University	1
28	National Taiwan University	1
29	Charles University	1
34	Heidelberg University	1
35	ETH Basel	1

Showing 1 to 18 of 794 entries, 2 total columns

Figură 15. Data frame cu clusterele

	date_proiect.University	X1	X2	X3
1	Harvard University	2.350638e-02	9.073445e-02	8.857592e-01
3	University of Toronto	3.171996e-02	4.329247e-01	5.353554e-01
4	University of Melbourne	1.457312e-02	1.282277e-01	8.571991e-01
5	Technical University of Munich	9.830475e-01	1.230259e-02	4.649902e-03
6	University of Tokyo	9.196477e-01	6.273955e-02	1.761277e-02
7	University of Amsterdam	5.024116e-01	4.211754e-01	7.641302e-02
8	National University of Singapore	2.821016e-02	7.785711e-01	1.932187e-01
9	Sorbonne University	9.973082e-01	2.016699e-03	6.750954e-04
10	ETH Zurich	9.877618e-01	8.942818e-03	3.295370e-03
11	KTH Royal Institute	9.780324e-01	1.588367e-02	6.083953e-03
12	University of Copenhagen	9.778522e-01	1.601367e-02	6.134104e-03
13	Tsinghua University	9.207033e-01	6.191683e-02	1.737988e-02
14	Seoul National University	9.670435e-01	2.530487e-02	7.651641e-03
15	Trinity College Dublin	3.456040e-04	9.989804e-01	6.739600e-04
16	Pusan National University	9.879257e-01	9.159271e-03	2.915053e-03
17	University of Auckland	5.442357e-04	9.984736e-01	9.822001e-04
18	University of Vienna	9.914830e-01	6.227735e-03	2.289261e-03
19	KU Leuven	9.997173e-01	2.100456e-04	7.266768e-05

Showing 1 to 18 of 794 entries, 4 total columns

Figură 16. Data frame cu gradele de apartenenta

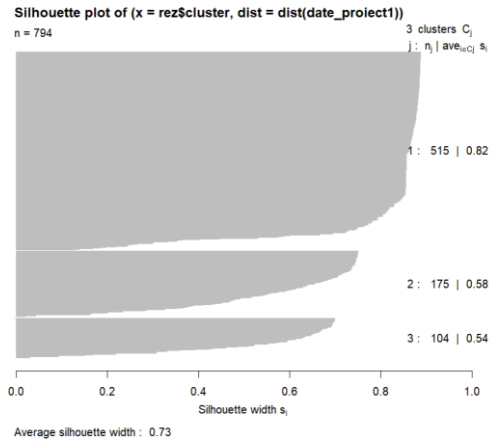
```
          1         2         3
1 0.02350638 0.09073445 0.8857592
3 0.03171996 0.43292469 0.5353554
4 0.01457312 0.12822775 0.8571991
```

Figură 17. Gradele de apartenenta la clustere pentru primele 3 observatii

In figura 14 au fost afisate gradele de apartenență pentru primele trei observații din setul de date. Rezultatele indică faptul că toate cele trei observații prezintă cel mai mare grad de apartenență față de clusterul 3, ceea ce sugerează că aceste programe educaționale se încadrează în categoria cu costuri ridicate – educație premium. Acest lucru reflectă faptul că atât taxele de școlarizare, cât și costurile asociate traiului sunt semnificativ mai mari în cazul acestor unități de învățământ.

Verificarea calitatii clusterizarii

Pentru a verifica calitatea clusterizarii, am utilizat funcția silhouette(), care evaluează coerența fiecărei observații în cadrul clusterului său pe baza distanțelor dintre date. După calcularea indicilor de siluetă pentru fiecare observație, am reprezentat grafic rezultatele pentru a vizualiza clar separarea și omogenitatea clusterelor identificate.



Figură 18. Reprezentarea grafica a siluetei

Analizand figura 16, observam ca clusterizarea este bine definită pentru clusterul 1, cu o apartenență clară a observațiilor. Clusterul 2 și clusterul 3 au o coeziune mai redusă, ceea ce înseamnă că există mai multă variabilitate internă sau suprapunere cu alte clustere.

În general, valoarea medie a siluetei de 0.73 indică o clusterizare reușită, cu o bună separabilitate între cele trei grupuri identificate.

3. Regresie logistica

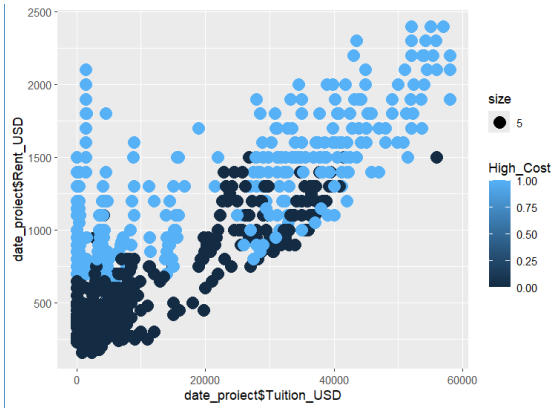
3.1. Regresie logistica binomiala

In acest capitol, voi analiza probabilitatea ca o țară să fie clasificată drept *„high cost”* sau *„low cost”* din perspectiva cheltuielilor educaționale, folosind metoda regresiei logistice binare. Scopul principal al analizei este de a construi un model care să distingă între cele două categorii în funcție de caracteristicile disponibile, având ca variabilă țintă o variabilă binară denumită *High_Cost*. Aceasta a fost definită pe baza valorii mediane a indicelui costului de trai (*Living_Cost_Index*): dacă indicele este mai mare decât mediana, este clasificată ca *high cost* (1), în caz contrar, ca *low cost* (0). Alegerea medianei ca prag permite o separare echilibrată a datelor și facilitează aplicarea modelului de clasificare.

```
> table(date_proiect$High_Cost)

  0   1
399 395
```

Figură 19. Distributia observatiilor in functie de variabila tinta

În urma clasificării pe baza indicelui costului de trai, distribuția observațiilor în funcție de variabila binară *High_Cost* arată că 399 de țări au fost încadrate în categoria *low cost (0)*, în timp ce 395 de țări au fost clasificate ca *high cost (1)*. Această împărțire relativ echilibrată susține utilizarea medianei ca prag de delimitare între cele două clase.



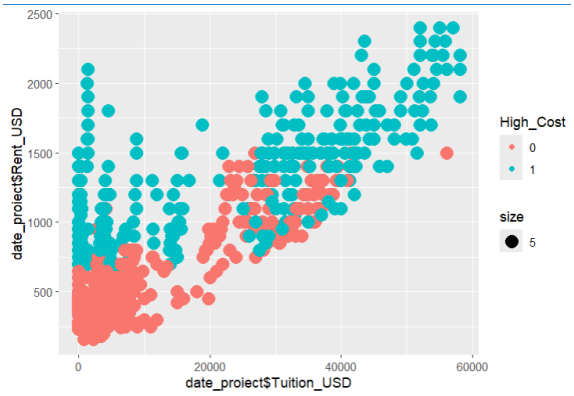
Figură 20. Reprezentarea grafica a datelor

In graficul realizat, variabila dependentă *High_Cost* a fost reprezentată în funcție de două variabile independente: *Tuition_USD* (taxa de școlarizare) și *Rent_USD* (costul chiriei). Se poate observa o evolutie vizibilă de la categoria *low cost* la *high cost*, sugerând o asociere între taxele și chiriile ridicate și apartenența la categoria *high cost*. De asemenea, distribuția

punctelor evidențiază un raport relativ echilibrat între programele educaționale accesibile și cele costisitoare.

Aplicarea modelului de regresie logistica binomiala

În vederea aplicării regresiei logistice binare, variabila dependentă *High_Cost* a fost transformată într-o variabilă de tip factor. Ulterior, setul de date a fost împărțit în două subseturi: unul pentru antrenare (75% din date) și unul pentru testare (25%), utilizând o împărțire aleatorie controlată de un seed. Această împărțire permite evaluarea performanței modelului pe date noi, neutilizate în procesul de învățare. Reprezentarea grafică a fost refăcută pentru a vizualiza distribuția observațiilor în funcție de taxa de școlarizare și costul chiriei, in functie de apartenenta la variabila dependenta, acum de tip factor.



Figură 21. Reprezentarea grafica cu variabila factor

Regresia logistica binomiala a fost aplicata utilizand functia glm, avand ca variabila dependenta *High_Cost* și ca variabile explicative *Tuition_USD* (taxa de școlarizare) și *Rent_USD* (costul chiriei), utilizând setul de antrenare. Rezultatul modelului de regresie este urmatorul:

```
> summary(model_regresie)

Call:
glm(formula = High_Cost ~ Tuition_USD + Rent_USD, family = binomial(),
    data = set_antrenare)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.2355035  0.5111579  -12.20   <2e-16 ***
Tuition_USD -0.0001371  0.0000156   -8.79   <2e-16 ***
Rent_USD     0.0092349  0.0007611   12.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 824.83  on 594  degrees of freedom
Residual deviance: 375.12  on 592  degrees of freedom
AIC: 381.12

Number of Fisher Scoring iterations: 6
```

Figură 22. Rezultatul regresiei logistice binomiale

Modelul de regresie logistică binomială estimat are forma:

$$logit(p) = -6.2355035 - 0.0001371 \times Tuition\_USD + 0.0092349 \times Rent\_USD,$$

unde p reprezintă probabilitatea ca o țară să fie clasificată drept high cost din perspectiva educației ($High\_Cost = 1$), iar $1 - p$ indică probabilitatea ca aceasta să fie low cost ($High\_Cost = 0$).

Devianta reziduală a modelului este 375.12, semnificativ mai mică decât devianta nulă, care este 824.83. Acest lucru indică faptul că modelul construit oferă o predicție mult mai bună decât modelul nul, adica cel care conține doar termenul liber. În plus, toate variabilele din model s-au dovedit a fi semnificative din punct de vedere statistic.

```
> exp(coef(model_regresie))
(Intercept) Tuition_USD    Rent_USD
0.001958643 0.999862874 1.009277665
```

Figură 23. Coeficientii exponentiati ai modelului

Coeficientii modelului ilustreaza impactul variabilelor asupra șanselor ca o țară să fie clasificată ca fiind scumpă din punct de vedere al programelor educaționale pentru studenții internaționali, dupa cum urmeaza:

- O creștere cu 1 USD a taxei de școlarizare este asociată cu o scădere a șanselor ca o țară să fie considerată scumpă cu aproximativ 0.0137% ((0.999862874-1)*100) față de șansele de a fi clasificată ca ieftină. Cu toate ca asteptarea ar fi ca cresterea taxei de scolarizare sa creasca șansele ca o țară să fie considerată scumpă, modelul arată că efectul este ușor negativ. Acest lucru poate sugera că taxa de școlarizare, luată separat, nu este un indicator decisiv al costului general perceput - sau ca în unele țări cu taxe mari, costurile de trai sunt suficient de mici încât per ansamblu să nu fie percepute ca „scumpe”.
- În schimb, o creștere cu 1 USD a costului chiriei determină o creștere a șanselor ca o țară să fie percepută drept scumpă cu aproximativ 0.927% ((1.009277665-1)*100). Acest lucru indica faptul că costul locuirii este un factor mult mai influent în percepția generală asupra costurilor educației pentru studenții internaționali.

In continuare, am factorizat categoriile astfel incat țărilor ieftine să li se atribuie valoarea 0, iar țărilor scumpe valoarea 1. Ulterior, folosind modelul de regresie și setul de testare, s-au prezis probabilitățile ca o țară să fie clasificată drept scumpă sau ieftină.

```
> probabilitati
        6         7         9        11        16        24        29        30        31        34
0.989536570 0.995722475 0.997708457 0.992204378 0.358891547 0.858955438 0.171256070 0.999485606 0.998774734 0.881564692
       37        41        46        48        54        57        58        65        73        78
0.999338156 0.994776777 0.888538549 0.595026112 0.976810259 0.966003461 0.985711925 0.151474309 0.528766540 0.118080150
       84       126       130       131       140       143       149       152       156       159
0.921945305 0.088940063 0.069079433 0.690160698 0.052930868 0.300429860 0.725975662 0.959298287 0.939049833 0.949155544
      162       165       169       172       174       180       182       184       192       198
0.995837720 0.630718508 0.404153400 0.999985047 0.999905714 0.999763914 0.998357747 0.863827490 0.048744855 0.673217607
      201       206       210       212       218       219       220       221       224       225
0.260779122 0.103383981 0.137576238 0.260779122 0.992068622 0.981549554 0.999721116 0.999881367 0.998860800 0.967633948
      229       231       233       234       293       302       303       304       305       307
0.810012124 0.977621928 0.887673934 0.894331178 0.781852465 0.885369121 0.480810361 0.835302498 0.714075448 0.872253374
      310       318       325       330       331       347       352       354       370       373
0.851587333 0.297590191 0.123913253 0.177198409 0.067074042 0.996975395 0.993075698 0.712227820 0.843330342 0.853897224
```

Figură 24. Probabilitati

De exemplu, rezultatul indică faptul că probabilitatea ca prima observatie să fie considerată scumpă este 0.98, iar pentru a doua observatie această probabilitate este 0.99.

Predictii pe setul de antrenare

Am definit un vector de predicții inițial cu valoarea „0” (reprezentând țările ieftine), având lungimea egală cu numărul de observații din setul de antrenare. Ulterior, elementele

corespunzătoare țărilor pentru care probabilitatea previzionată de a fi scumpe depășește 50% au fost actualizate la „1”. Astfel, predicțiile reflectă clasificarea unei țări ca „scumpă” dacă probabilitatea estimată este mai mare decât pragul de 0.5.

Pe baza vectorului de predictii definit, s-a determinat matricea de confuzie:

```
predictie   0   1
        0 138 138
        1 161 158
```

Matricea de confuzie evidentiaza performanta modelului de regresie pe setul de antrenare. Acesta contine 299 de tari ieftine (138 + 161) si 296 de tari scumpe (138 + 158). Dintre cele 296 de țări scumpe, 158 au fost clasificate corect, iar 138 au fost clasificate eronat ca fiind ieftine. În cazul celor 299 de țări ieftine, 138 au fost identificate corect, în timp ce 161 au fost clasificate greșit ca fiind scumpe. Observațiile clasificate corect se regăsesc pe diagonala principală a matricei de confuzie, însumând 296 de cazuri (138 + 158).

Acuratețea modelului este determinată ca raport dintre numărul predicțiilor corecte și totalul observațiilor, adică (138 + 158) / (138 + 138 + 161 + 158), rezultând o acuratețe de 49,7%, ceea ce înseamnă că modelul a clasificat corect 49,7% dintre observațiile din setul de antrenare.

Am introdus două noi observații, reprezentând două țări, pentru care am dorit să previzionez dacă sunt scumpe sau ieftine, pe baza valorilor pentru Tuition_USD (10.000 și 1.000) și Rent_USD (1.000 și 650). Utilizând modelul de regresie, am obținut probabilitățile asociate fiecărei țări de a fi clasificată ca „scumpă”.

```
> predictie_noua
        1         2
0.8359001 0.4085509
```

Figură 25. Predictie pentru doua observatii noi

Pentru evaluare, am comparat aceste probabilități cu pragul de 0.5: dacă probabilitatea este mai mare de 0.5, țara este considerată scumpă, altfel este clasificată ca ieftină.

```
> predictie_noua[1] <= 0.5
    1
FALSE
> # Prima tara este scumpa
> predictie_noua[2] <=0.5
   2
TRUE
```

Figură 26. Evaluarea clasificarii

În urma acestei analize, prima țară a fost clasificată ca fiind scumpă, iar a doua ca fiind ieftină.

Predictii pe setul de testare

Pentru setul de testare, a fost generat, de asemenea, un vector de predictii, pe baza caruia s-a construit matricea de confuzie.

```
predictie1  0  1
         0 84  8
         1 16 91
```

```
> mean(predictie1==set_testare$High_Cost)
[1] 0.879397
```

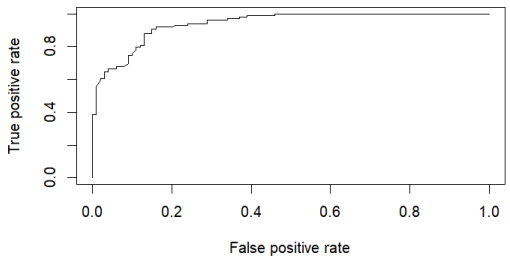
Figură 27. Matricea de confuzie

Din matrice reiese că în setul de testare există 99 de țări scumpe (91 clasificate corect și 8 incorect) și 100 de țări ieftine (84 clasificate corect și 16 incorect). Acuratețea modelului pentru acest set este de 87,9%, ceea ce indică faptul că aproximativ 87% dintre observații au fost etichetate corect.

Analizând indicatorii de performanță ai modelului, specificitatea – adică proporția cazurilor negative (țări ieftine) recunoscute corect (TN/(TN+TP) – este de 52% în timp ce senzitivitatea – proporția cazurilor pozitive (țări scumpe) identificate corect (TP/(TP+FN)) – este de 91%. Aceste valori sugerează că modelul este mult mai eficient în identificarea țărilor scumpe decât în recunoașterea celor ieftine.

Evaluarea performantei modelului de clasificare

Pentru a evalua performanța modelului de clasificare, a fost generată curba ROC, folosind pachetul ROCR. S-au utilizat predicțiile probabilistice obținute din setul de testare și valorile reale ale variabilei țintă.



Figură 28. Curba ROC

Curba ROC a fost trasată prin compararea ratei de adevărate pozitive (True Positive Rate) cu rata de false pozitive (False Positive Rate). Graficul rezultat se apropie de colțul din stânga sus al diagramei, ceea ce indică o performanță foarte bună a modelului.

```
> auc
[1] 0.9405556
```

Figură 29. Valoare AUC

De asemenea, a fost calculată valoarea AUC (Area Under the Curve), care a rezultat a fi 0.94. Această valoare indică o capacitate excelentă de clasificare, sugerând că modelul are 94% șanse să distingă corect între clasele pozitive (țări scumpe) și negative (țări ieftine).

3.2. *Regresie logistica multinomiala*

Regresia logistică multinomială este o metodă statistică utilizată pentru a analiza relația dintre o variabilă dependentă categorială cu mai mult de două niveluri și una sau mai multe variabile independente. Scopul acestei analize este de a estima probabilitatea ca o observație să aparțină uneia dintre mai multe clase posibile, în funcție de valorile variabilelor explicative.

Variabila țintă este *Level*, care conține trei categorii de studii universitare: „Bachelor", „Master" și „PhD".

```
> unique(date_proiect$Level)
[1] "Master"   "Bachelor" "PhD"
```

Figură 30. Valorile variabilei "Level"

Pentru a putea aplica regresia, am transformat variabila într-un factor și am stabilit „Bachelor" ca nivel de referință. Ulterior, am aplicat modelul de regresie logistică multinomială pentru a examina în ce măsură variabilele *Tuition_USD* (taxa de școlarizare) și *Duration_Years* (durata programului) influențează probabilitatea ca un program de studii să fie de tip Master sau PhD, în comparație cu un program de tip Bachelor.

Rezultatele modelului de regresie sunt urmatoarele:

```
> summary(model_regresie_multi)
Call:
multinom(formula = out ~ Tuition_USD + Duration_Years, data = date_proiect,
    trace = FALSE)

Coefficients:
       (Intercept)   Tuition_USD Duration_Years
Master    44.44995 -3.379777e-05     -15.562459
PhD      -16.18454 -4.015812e-05       4.220675

Std. Errors:
         (Intercept)  Tuition_USD Duration_Years
Master 1.200334e-09 2.790265e-05   3.598949e-09
PhD    1.762881e-10 5.542654e-06   7.090700e-10

Residual Deviance: 416.4454
AIC: 428.4454
```

Figură 31. Rezultate regresie multinomiala

Devianta reziduală reprezintă eroarea rămasă în model după estimarea parametrilor, iar pentru ca modelul să fie considerat performant, această valoare trebuie să fie cât mai mică. În cazul de față, devianța reziduală are valoarea 416.4454.

Pe baza coeficienților estimați de modelul de regresie logistică multinomială, pot fi formulate două ecuații care exprimă logaritmul raportului probabilităților (log odds) între categoriile Master și Bachelor, respectiv PhD și Bachelor.

$$\ln\left[\frac{P(Master)}{P(Bachelor)}\right] = 44.44995 - 0.0000337 * Tuition\_USD - 15.532459 * Duration\_Years \quad (1)$$

$$\ln\left[\frac{P(PhD)}{P(Bachelor)}\right] = -16.18454 - 0.0000401 * Tuition\_USD + 4.220675 * Duration\_Years \quad (2)$$

Ecuatia (1) indică faptul că o creștere a taxei de școlarizare (*Tuition_USD*) cu o unitate determină o scădere a log odds-ului cu 0.0000337, ceea ce înseamnă că probabilitatea ca un program să fie de tip Master (comparativ cu Bachelor) scade ușor. De asemenea, o creștere a duratei programului (*Duration_Years*) cu un an determină o scădere a log odds-ului cu 15.562, sugerând că durata mai lungă reduce probabilitatea ca un program să fie clasificat ca Master în raport cu Bachelor.

Ecuatia (2) arată că taxa de școlarizare are, și aici, un efect negativ: o unitate în plus reduce log odds-ul cu 0.0000401, deci scade probabilitatea ca un program să fie de tip PhD în raport cu Bachelor. În schimb, durata programului are un efect pozitiv — fiecare an suplimentar crește log odds-ul cu 4.22, ceea ce semnalează o creștere a probabilității ca un program să fie de tip PhD față de Bachelor.

```
> exp(coef(model_regresie_multi))
        (Intercept) Tuition_USD Duration_Years
Master 2.015426e+19   0.9999662    1.74305e-07
PhD    9.357181e-08   0.9999598    6.80794e+01
```

Figură 32. Coeficientii modelului

Coeficientii modelului indica urmatoarele:

- Șansele ca un program să fie de Master sunt cu 0,00338% mai mici decât șansele ca programul să fie de Bachelor, dacă taxa de școlarizare crește cu o unitate (1 USD).
- Șansele ca un program să fie de Master sunt cu 99,99% mai mici decât șansele ca programul să fie de Bachelor, dacă durata programului crește cu o unitate (1 an).
- Șansele ca un program să fie de PhD sunt cu 0,00402% mai mici decât șansele ca programul să fie de Bachelor, dacă taxa de școlarizare crește cu o unitate (1 USD).
- Șansele ca un program să fie de PhD sunt cu 6707,94% mai mari decât șansele ca programul să fie de Bachelor, dacă durata programului crește cu o unitate (1 an), adică, pentru fiecare an în plus în durata programului, șansele ca programul să fie PhD față de Bachelor cresc de 68 de ori.

Predictii

Am utilizat modelul de regresie pentru a prezice clasele pentru fiecare observație din setul de date, in figura 33 fiind afisate clasele prezise direct, iar in figura 34 fiind returnate probabilitățile asociate fiecărei clase pentru fiecare observație.

```
> predict(model_regresie_multi, date_proiect)
  [1] Master   Master   Master   Master   Master   Master   Master   Master   Master   Master   Master   Master
 [13] Master   Master   PhD      Bachelor Bachelor Bachelor Bachelor Bachelor Bachelor Bachelor Bachelor Bachelor
 [25] Bachelor PhD      Bachelor PhD      PhD      Bachelor PhD      PhD      PhD      Bachelor Bachelor PhD
 [37] Bachelor Master   Bachelor Master   Bachelor Master   Bachelor PhD      PhD      Bachelor Bachelor Bachelor
 [49] Bachelor PhD      Bachelor Bachelor Bachelor Bachelor PhD      Master   PhD      Master   PhD      Master
 [61] Bachelor PhD      Master   PhD      Master   Bachelor Master   Bachelor Master   Bachelor Master   Bachelor
 [73] Master   Bachelor Master   Master   Bachelor Master   Bachelor Master   Bachelor Master   Bachelor Master
 [85] Master   Master   Master   PhD      Bachelor Master   PhD      Bachelor Master   Bachelor Master   Bachelor
 [97] Master   Master   Bachelor Master   Bachelor Bachelor Master   Master   Bachelor Master   PhD      Bachelor
[109] Master   PhD      Master   PhD      Master   PhD      Bachelor PhD      Master   Bachelor Bachelor PhD
[121] PhD      Master   PhD      PhD      Master   Bachelor Master   Bachelor Master   Bachelor Master   Bachelor
[133] Master   Master   Bachelor Bachelor Bachelor Bachelor Master   Master   Master   Master   Bachelor Bachelor
```

Figură 33. Predictii clase

```
> predict(model_regresie_multi, date_proiect, type="prob")
       Bachelor       Master          PhD
1   1.062136e-05 9.999894e-01 4.979154e-10
3   5.999580e-06 9.999940e-01 5.544213e-10
4   6.752960e-06 9.999932e-01 5.422151e-10
5   1.660931e-06 9.999983e-01 7.060052e-10
6   2.206217e-06 9.999978e-01 6.692750e-10
7   4.855551e-13 1.000000e+00 1.639986e-18
8   2.225383e-09 1.000000e+00 2.868518e-14
9   1.901360e-06 9.999981e-01 6.882698e-10
10  1.715705e-06 9.999983e-01 7.017074e-10
11  1.633099e-06 9.999984e-01 7.082540e-10
12  1.633099e-06 9.999984e-01 7.082540e-10
13  5.256535e-03 9.947303e-01 1.315721e-05
14  2.083029e-06 9.999979e-01 6.765510e-10
15  7.560017e-13 1.000000e+00 1.508879e-18
16  3.866943e-01 5.893496e-09 6.133057e-01
17  9.522582e-01 3.879029e-02 8.951463e-03
18  8.855392e-01 8.984359e-02 2.461723e-02
19  8.924756e-01 8.462902e-02 2.289533e-02
20  8.025182e-01 2.549131e-09 1.974818e-01
```

Figură 34. Probabilitati de apartenenta

De exemplu, potrivit figurii 34, pentru primul program, probabilitatea de a fi „Bachelor” este extrem de mică (aproximativ 0.00001), cea de a fi „Master” este aproape 1 (99.9%), iar cea de a fi „PhD” este aproape zero, deci modelul îl clasifică clar ca program de tip „Master”.

```
        Bachelor       Master          PhD
11   1.633099e-06 9.999984e-01 7.082540e-10
217  8.161467e-01 2.406653e-09 1.838533e-01
751  8.914594e-01 8.539412e-02 2.314644e-02
```

Figură 35. Previzionarea probabilitatilor pentru trei observatii aleatoare

Am previzionat probabilitatile pentru trei observatii aleatoare (10, 200 si 650) din setul de date. Rezultatele arată că primul program are cea mai mare probabilitate să fie de tip „Master”, în timp ce al doilea și al treilea sunt clasificate predominant ca programe de tip „Bachelor”.

Evaluarea modelului

Am comparat predicțiile modelului de regresie logistică multinomială cu valorile reale pentru primele 50 de observații, construind o matrice de confuzie.

```
          Bachelor Master PhD
Bachelor       18      0   4
Master          0     17   0
PhD             4      0   7
```

```
> mean(date_proiect$Level[1:50] == predict(model_regresie_multi)[1:50])
[1] 0.84
```

Figură 36. Matricea de confuzie si acuratetea modelului

Din cele 22 programe de Bachelor, 18 au fost corect clasificate, iar 4 au fost greșit etichetate ca PhD. Toate cele 17 programe de Master au fost clasificate corect. În cazul celor 11 programe de PhD, 7 au fost identificate corect, iar 4 au fost confundate cu programe de Bachelor. Astfel, s-a observat o confuzie între clasele Bachelor și PhD, în timp ce programele de Master au fost recunoscute cu precizie totală.

Acuratețea modelului pentru acest subset este de 84%, indicând o performanță excelentă în clasificare.

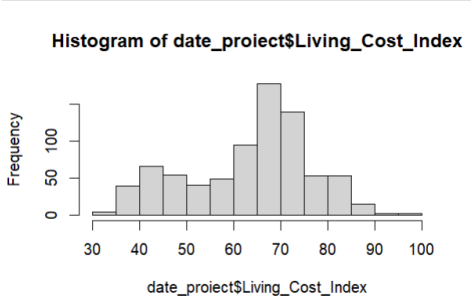
4. Arbori de regresie si clasificare

4.1. Arbori de decizie

Scopul analizei prin arbori de decizie este de a clasifica observatiile în funcție de costul de trai, delimitându-le în două categorii: cost de trai ridicat și cost de trai scăzut. Pentru aceasta, am utilizat variabila numerică Living_Cost_Index, care variază între 32.5 și 95.2, pentru a defini variabila binară denumită High_Living_Cost. Am stabilit pragul la valoarea 65, astfel încât țările cu indicele costului de trai peste această valoare sunt etichetate ca „Yes” (cost ridicat), iar cele cu valori mai mici sau egale ca „No” (cost scăzut). Ulterior, am selectat un set relevant de variabile explicative împreună cu variabila țintă pentru a construi și antrena modelul de arbore decizional, cu scopul de a înțelege factorii care influențează clasificarea costului de trai.

```
> range(date_proiect$Living_Cost_Index) # valori intre 32.5 si 95.2
[1] 32.5 95.2
```

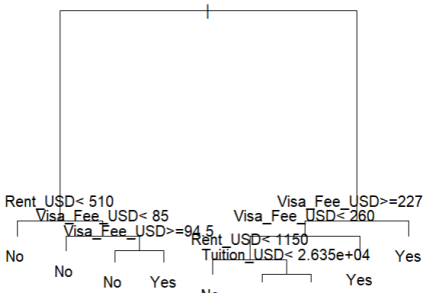
Figură 37. Interval de valori pentru indicele costului de trai



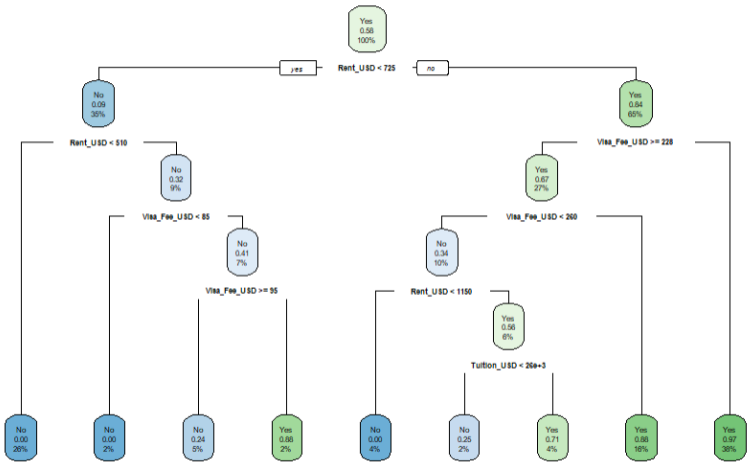
Figură 38. Histograma costului de trai

Am transformat variabila țintă într-un factor, pentru a o putea folosi ca variabilă de clasă în modelul de clasificare. Apoi, am împărțit setul de date în două eșantioane egale: unul pentru antrenarea modelului și unul pentru testarea sa.

Am construit un arbore de decizie folosind funcția rpart, în care variabila țintă este High_Living_Cost (cost ridicat sau scăzut), iar celelalte variabile sunt folosite ca predictori. În final, am vizualizat arborele de decizie pentru a observa structura regulilor de clasificare generate pe baza datelor de antrenament.



Figură 39. Reprezentare grafica cu plot



Figură 40. Reprezentare grafica cu rpart.plot

```
> table(set_antrenare$High_Living_Cost)

 No Yes
168 229
```

În nodul 1, dintr-un total de 397 de observații, 229 sunt etichetate ca aparținând clasei „Yes” (cost de trai ridicat), iar 168 sunt clasificate în clasa „No” (cost de trai scăzut). Vectorul de probabilitate este (0.42, 0.58), ceea ce înseamnă că probabilitatea ca o observație să fie în categoria „Yes” este de 58%. Clasa dominantă în acest nod este „Yes” (cost de trai ridicat), cu o probabilitate de 58%.

Predictii pe setul de testare. Evaluarea modelului.

Am realizat predictii pentru observatiile din setul de testare, rezultatele fiind evaluate prin matricea de confuzie.

```
     predictie
       No Yes
No    159  23
Yes    21 194
```

Figură 41. Matrice de confuzie

Din cele 180 de observații etichetate ca „No” (cost scăzut), 159 au fost clasificate corect, în timp ce 21 au fost clasificate greșit. Pentru clasa „Yes” (cost ridicat), din totalul de 217 observații, 194 au fost corect clasificate, iar 23 au fost încadrate eronat. Aceste rezultate reflectă performanța modelului în distingerea corectă a costului de trai ridicat versus scăzut în setul de testare.

```
> mean(predictie!=set_testare$High_Living_Cost)
[1] 0.1108312
```

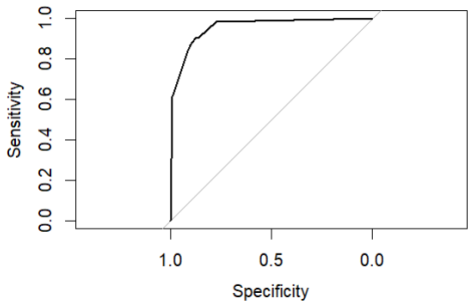
Eroarea de clasificare sugereaza ca aproximativ 11% dintre observații au fost etichetate greșit, ceea ce reflectă o acuratețe de clasificare de 89% pentru modelul utilizat.

	No	Yes
1	0.03311258	0.9668874
3	0.29411765	0.7058824
4	0.12307692	0.8769231
5	0.03311258	0.9668874
7	0.03311258	0.9668874
8	0.03311258	0.9668874
10	0.03311258	0.9668874
16	0.76190476	0.2380952
18	0.03311258	0.9668874

Figură 42. Probabilitatile pentru fiecare observatie

Probabilitățile generate de model pentru fiecare observație din setul de testare indică în ce măsură modelul aloca o observatie intr-o anumita clasa. De exemplu, prima observație are o probabilitate de 0.03 de a aparține clasei „No” (cost scăzut) și 0.96 de a fi clasificată ca „Yes” (cost ridicat), ceea ce sugerează o încredere ridicată a modelului în încadrarea acesteia în clasa „Yes”.

Pentru a evalua performanta modelului, a fost generata curba ROC. Graficul curbei ROC arată că aceasta se apropie de colțul din stânga sus, ceea ce indică o performanță bună a modelului.



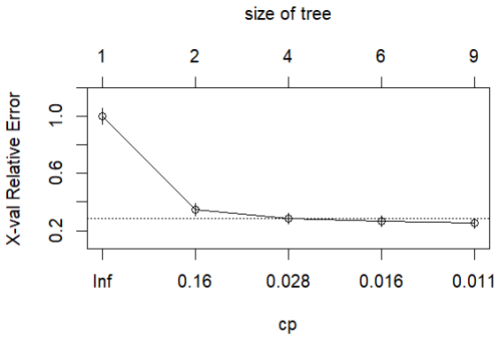
Figură 43. Curba ROC

```
Area under the curve: 0.9551
```

Valoarea AUC este de 0.9551, ceea ce înseamnă că modelul are o capacitate de discriminare de aproximativ 95.51% în a diferenția între costurile de trai ridicate și cele scăzute — un rezultat foarte bun, care reflectă o clasificare eficientă.

Construirea arborelui curatat

Pentru a construi arborele curatat, este necesară identificarea valorii optime a parametrului de complexitate (cp). În acest sens, a fost generat graficul plotcp(arbore), care ilustrează evoluția erorii de validare în funcție de complexitatea modelului. Din grafic se poate observa că valoarea optimă a parametrului cp, care corespunde celei mai mici erori de validare, este 0.01.



Figură 44. Graficul plotcp

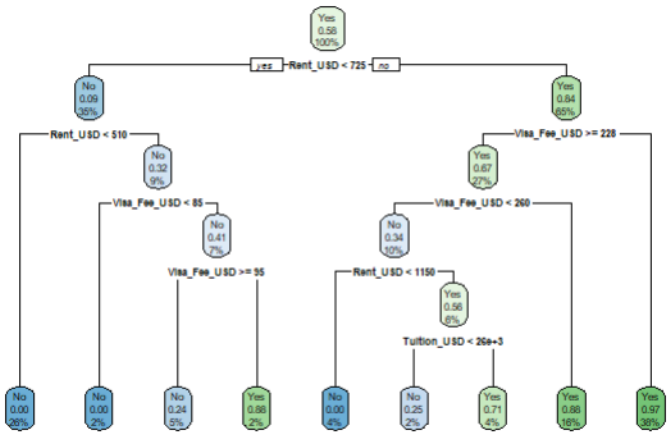
```
Classification tree:
rpart(formula = set_antrenare$High_Living_Cost ~ ., data = set_antrenare,
    method = "class")

Variables actually used in tree construction:
[1] Rent_USD     Tuition_USD  Visa_Fee_USD

Root node error: 168/397 = 0.42317

n= 397

        CP nsplit rel error  xerror     xstd
1 0.690476      0   1.00000 1.00000 0.058596
2 0.038690      1   0.30952 0.34524 0.041890
3 0.020833      3   0.23214 0.28571 0.038666
4 0.011905      5   0.19048 0.26786 0.037599
5 0.010000      8   0.15476 0.25000 0.036478
> mincp
[1] 0.01
```

Figură 45. Output cp

Ulterior, valoarea optimă a parametrului de complexitate (cp) a fost identificată pe baza tabelului de validare încrucișată generat de model. Cea mai mică valoare a erorii relative de validare încrucișată (xerror = 0.25) corespunde unui cp de 0.01. Această valoare reprezintă pragul optim pentru tăierea arborelui, contribuind la reducerea riscului de suprainvățare (overfitting). Conform acestei setări, arborele rezultat conține 8 împărțiri (nsplit = 8), ceea ce înseamnă 9 noduri terminale. Astfel, modelul obținut păstrează un echilibru adecvat între complexitate și capacitatea predictivă.

Arborele curatat (prunat) a fost construit prin aplicarea funcției prune() asupra arborelui inițial, utilizând valoarea cp corespunzătoare celei mai mici erori de validare încrucișată. Această valoare a fost identificată ca fiind cp = 0.013158. Arborele astfel curățat a fost vizualizat cu ajutorul funcției rpart.plot(), iar rezultatul confirmă că acesta conține 9

noduri terminale, asadar tăierea a fost realizată la nivelul optim, asigurând un echilibru între simplitatea modelului și precizia clasificării.



Figură 46. Reprezentarea arborelui curatat

```
Classification tree:
rpart(formula = set_antrenare$High_Living_Cost ~ ., data = set_antrenare,
    method = "class")

Variables actually used in tree construction:
[1] Rent_USD     Tuition_USD  Visa_Fee_USD

Root node error: 168/397 = 0.42317

n= 397

        CP nsplit rel error  xerror     xstd
1 0.690476      0   1.00000 1.00000 0.058596
2 0.038690      1   0.30952 0.34524 0.041890
3 0.020833      3   0.23214 0.28571 0.038666
4 0.011905      5   0.19048 0.26786 0.037599
5 0.010000      8   0.15476 0.25000 0.036478
```

Figură 47. Output cp arbore curatat

Pe baza arborelui curățat, s-a realizat predicția claselor pentru setul de testare. Rezultatele au fost apoi evaluate prin intermediul unei matrici de confuzie, care compară valorile reale cu cele prezise.

```
  1   3   4   5   7   8  10  16  18  19  22  24  27  29  30  36  39  42  47  48  49  50  51  55  59  62  64  65  66  67
Yes Yes Yes Yes Yes Yes Yes  No Yes Yes Yes Yes Yes  No Yes Yes Yes Yes  No Yes  No  No Yes Yes Yes Yes  No  No  No Yes
 68  69  71  74  76  77  81  83  84  85  87  88  92  93  94  98  99 101 113 114 115 117 118 119 123 125 128 140 141 143
 No  No  No Yes Yes Yes Yes Yes Yes  No Yes  No Yes Yes Yes Yes Yes Yes  No Yes Yes  No  No  No Yes  No  No  No  No Yes
145 148 149 150 153 155 156 157 159 161 162 163 164 166 167 171 173 174 178 179 185 186 188 189 190 193 198 199 200 204
Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes  No  No  No  No Yes Yes Yes Yes
205 206 207 209 210 211 215 216 218 220 223 225 230 231 232 233 287 288 294 295 297 298 299 300 306 309 312 313 314 316
```

Figură 48. Predictii pe setul de testare

```
           No        Yes
1   0.03311258 0.9668874
3   0.29411765 0.7058824
4   0.12307692 0.8769231
5   0.03311258 0.9668874
7   0.03311258 0.9668874
8   0.03311258 0.9668874
10  0.03311258 0.9668874
16  0.76190476 0.2380952
18  0.03311258 0.9668874
19  0.03311258 0.9668874
22  0.03311258 0.9668874
24  0.29411765 0.7058824
27  0.03311258 0.9668874
29  0.76190476 0.2380952
30  0.03311258 0.9668874
36  0.03311258 0.9668874
39  0.12307692 0.8769231
42  0.03311258 0.9668874
47  1.00000000 0.0000000
48  0.03311258 0.9668874
49  0.76190476 0.2380952
50  1.00000000 0.0000000
```

Figură 49. Probabilitati

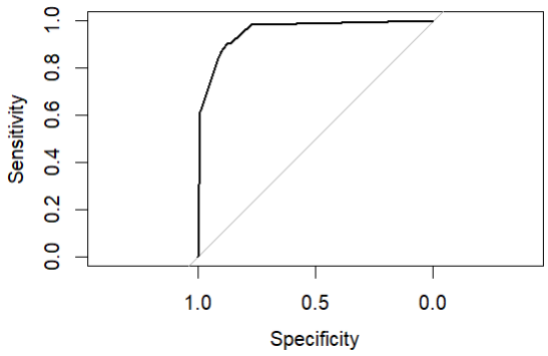
```
   predictie2
     No Yes
No  159  23
Yes  21 194
```

```
> mean(predictie2!=set_testare$High_Living_Cost)
[1] 0.1108312
```

Figură 50. Matrice de confuzie

Rata de eroare de clasificare obținută este de 11.08%, valoare identică cu cea obținută anterior folosind arborele necurățat. Aceasta indică faptul că, deși complexitatea modelului a fost redusă prin tăiere (pruning), performanța sa de clasificare pe setul de testare a fost menținută. Astfel, arborele curățat reușește să păstreze acuratețea predicțiilor, oferind în același timp un model mai simplu și mai ușor de interpretat, fără a compromite calitatea rezultatelor.

Pentru a evalua performanta arborelui curatat, a fost construita curba ROC pe baza probabilităților de apartenență la clasa "Yes". Graficul rezultat arată că modelul are o capacitate foarte bună de discriminare între clase, întrucât curba se apropie de colțul din stânga sus – zona ideală în clasificare binară. Valoarea AUC (Area Under the Curve) obținută este 95.51%, ceea ce indică o performanță excelentă a modelului în diferențierea între costurile de trai ridicate și cele scăzute.



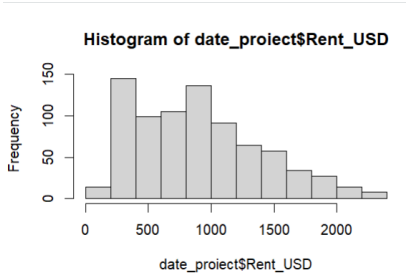

```
Area under the curve: 0.9551
```

Figură 51. Curba ROC si valoarea AUC

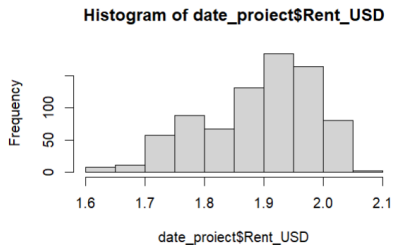
4.2. Arbori de regresie

Arborii de regresie sunt modele de tip arbore decizional utilizați pentru a prezice o variabilă continuă, pe baza unuia sau mai multor predictori. Spre deosebire de arborii de clasificare, care atribuie observațiile unor clase discrete, arborii de regresie oferă o estimare numerică pentru fiecare observație nouă.

In exemplul de față, arborii de regresie sunt utilizați pentru a estima costul chiriei în USD. Pentru a asigura o distribuție mai apropiată de normală, necesară pentru o analiză eficientă, valorile chiriei au fost transformate logaritmic. Această transformare îmbunătățește stabilitatea modelului și acuratețea predicțiilor.

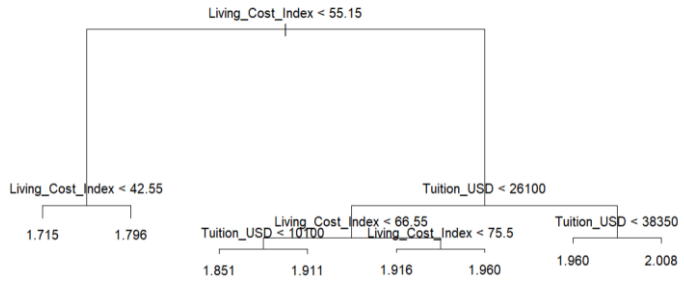


Figură 52. Histograma inainte de logaritmare



Figură 53. Histograma dupa logaritmare

După ce setul de date a fost împărțit în două subseturi egale – unul pentru antrenare și unul pentru testare – a fost construit un arbore de regresie folosind funcția tree(), având ca variabilă dependentă chiria logaritmată (Rent_USD) și ca predictori toate celelalte variabile disponibile. Am reprezentat grafic arborele rezultat, fiecare nod terminal indicând valoarea estimată a chiriei (în formă logaritmată) pentru observațiile care corespund acelui segment al datelor. Astfel, modelul oferă o interpretare clară și intuitivă a modului în care variabilele explicative contribuie la predicția costului chiriei.



Figură 54. Reprezentarea grafica a arborelui

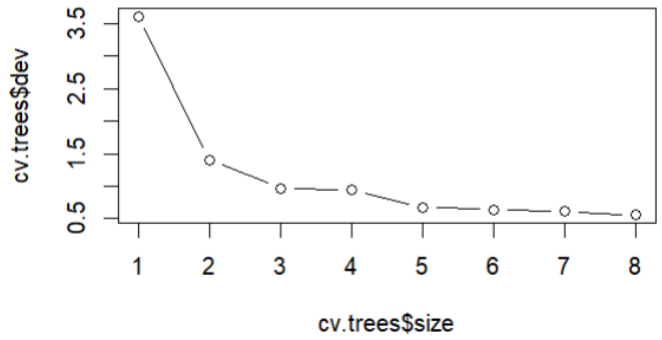
<u>Curatarea arborelui de regresie</u>

Pentru a imbunatati arborele si a evita suprainvatarea, s-a aplicat o procedură de validare încrucișată pentru curățarea arborelui de regresie.

```
> cv.trees$size
[1] 8 7 6 5 4 3 2 1
> cv.trees$dev
[1] 0.5559998 0.6195980 0.6415280 0.6788524 0.9468821 0.9671126 1.4058798 3.6153475
```

Figură 55. Rezultate validare incrucisata

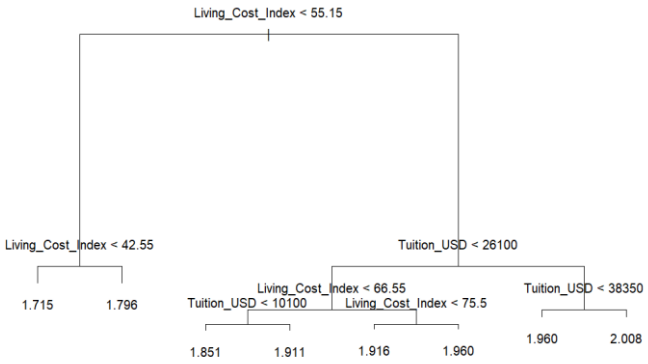
Rezultatele arată că eroarea minimă de 0.55 este obținută atunci când arborele are 8 noduri terminale, ceea ce sugerează că acesta este cel mai potrivit compromis între complexitatea modelului și capacitatea sa de generalizare. Astfel, un arbore cu 8 noduri oferă predicții mai stabile, fără a supra ajusta datele din setul de antrenare.



Figură 56. Reprezentarea grafica

Pentru a vizualiza relația dintre complexitatea arborelui și eroarea modelului, a fost realizat un grafic în care eroarea este reprezentată în funcție de numărul de noduri terminale. Curba obținută arată cum evoluează performanța modelului pe măsură ce acesta devine mai complex. Din grafic se observă clar că eroarea atinge valoarea minimă atunci când arborele are 8 noduri terminale. Acest punct marchează structura optimă a arborelui, în care se obține cel mai bun echilibru între sub ajustare și supra ajustare.

A m redus dimensiunea arborelui inițial prin tăiere, păstrând doar 8 noduri terminale, obținând astfel arborele curățat. Ulterior, l-am reprezentat grafic pentru o interpretare vizuală mai clară. Această versiune optimizată asigură un model mai simplu și mai robust, capabil să ofere predicții mai stabile.



Figură 57. Reprezentarea grafica a arborelui curatat

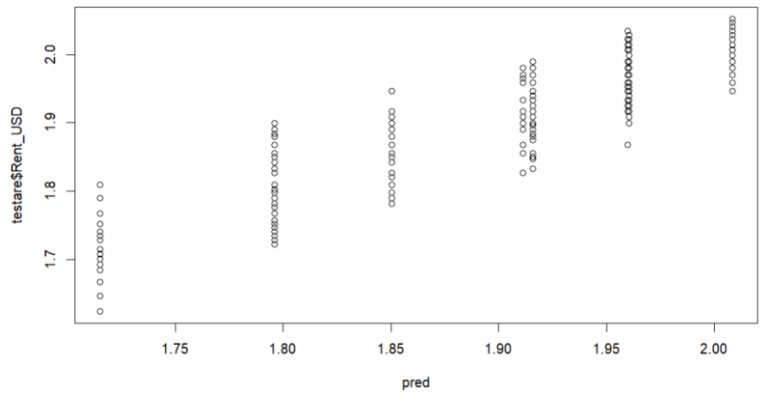
Predictii pe setul de testare

Pe baza arborelui curățat, am realizat predicții pentru chiria logaritmată pe setul de testare, obținând astfel valorile estimate pentru fiecare observație din acest set.

```
       8        9       10       13       14       15       16       17       21       26       27       31       34
1.960398 1.915655 1.959686 1.796199 1.915655 1.960398 1.850543 1.960398 2.008312 1.850543 1.959686 2.008312 1.915655
      45       49       51       54       57       58       61       68       71       73       75       76       77
1.796199 1.850543 2.008312 1.960398 1.850543 1.915655 1.915655 1.850543 1.850543 1.960398 1.960398 1.960398 1.960398
      80       81       84       87       88       91       95       99      101      113      115      117      121
1.960398 1.960398 1.915655 1.915655 1.850543 1.915655 2.008312 2.008312 2.008312 1.960398 1.960398 1.960398 1.911411
     124      126      127      130      131      134      136      138      141      144      146      148      151
1.960398 1.796199 1.796199 1.796199 1.960398 1.850543 1.715246 1.796199 1.796199 1.915655 1.911411 1.915655 1.915655
     154      156      160      161      162      165      166      169      172      174      176      178      179
1.960398 1.960398 2.008312 1.960398 1.915655 1.850543 1.915655 1.850543 1.959686 1.959686 1.959686 1.959686 1.959686
     181      182      185      186      187      188      190      192      194      197      200      202      203
1.959686 1.959686 1.796199 1.796199 1.796199 1.796199 1.796199 1.715246 1.715246 1.959686 1.915655 1.915655 1.915655
     206      210      212      215      216      218      222      224      225      226      227      230      231
1.915655 1.915655 1.959686 2.008312 2.008312 2.008312 2.008312 2.008312 1.960398 1.960398 1.960398 1.960398 1.960398
```

Figură 58. Valorile prezise pe setul de testare

Pentru a evalua performanța modelului, am creat un grafic de dispersie care compară valorile previzionate ale chiriei cu valorile reale (tot în formă logaritmică). Acest plot oferă o imagine vizuală a acurateței predicțiilor, evidențiind cât de bine se potrivesc estimările modelului cu datele observate.



Figură 59. Reprezentarea grafica a valorilor previzionate

Analizând graficul, observăm că predicțiile iau doar câteva valori distincte, indicând că modelul nu diferențiază foarte bine observațiile. Această grupare a predictiilor sugerează că modelul este prea simplu pentru complexitatea datelor analizate și ar putea beneficia de o ajustare.

```
> mean((pred-testare$Rent_USD)^2)
[1] 0.001360875
```

Figură 60. Eroarea de predictie

Cu toate acestea, eroarea de predictie de 0.0013 indică faptul că modelul are o performanță bună, deoarece valorile estimate sunt foarte apropiate de cele reale. O eroare atât de mică sugerează că modelul reușește să surprindă în mod fidel comportamentul variabilei țintă în setul de testare.

5. *Algoritmul de clasificare KNN*

KNN este un algoritm de clasificare care atribuie o clasă unei observații pe baza claselor celor mai apropiați vecini din setul de date. Scopul acestui algoritm este de a clasifica programele educaționale în două categorii: „Program accesibil” și „Program inaccesibil”, pe baza costurilor totale estimate pentru fiecare program.

Pentru aceasta, mai întâi calculăm costul total al programului, care include taxa de școlarizare, costul chiriei pe durata studiilor și taxa pentru viză.

	date_proiect.University	date_proiect.Total_Cost
1	Harvard University	55608.9775
2	University of Toronto	38782.9633
3	University of Melbourne	42497.5249
4	Technical University of Munich	621.7124
5	University of Tokyo	9167.2781
6	University of Amsterdam	16003.8762
7	National University of Singapore	35126.3869
8	Sorbonne University	4646.5249
9	ETH Zurich	1596.8320
10	KTH Royal Institute	157.0087
11	University of Copenhagen	167.2781
12	Tsinghua University	9096.9942
13	Seoul National University	7376.0146
14	Trinity College Dublin	29073.9816
15	Pusan National University	6120.2222
16	University of Auckland	28815.5131

wing 1 to 16 of 794 entries, 2 total columns

Figură 61. Data frame cu costurile totale

```
> date_proiect[which.max(date_proiect$Total_Cost),]
    Country   City         University                Program  Level Duration_Years Tuition_USD Living_Cost_Index
850     USA Boston Harvard University Artificial Intelligence Master              2       58000              82.1
    Rent_USD Visa_Fee_USD Cluster High_Cost LevelF    out High_Living_Cost Total_Cost
850 2.040728         160       3         1 Master Master              Yes   58208.98

> date_proiect[which.min(date_proiect$Total_Cost),]
      Country    City                    University             Program  Level Duration_Years Tuition_USD
748 Argentina Rosario National University of Rosario Computer Engineering Master              2           0
    Living_Cost_Index Rent_USD Visa_Fee_USD Cluster High_Cost LevelF    out High_Living_Cost Total_Cost
748              38.5 1.715721           90       1         0 Master Master               No   131.1773
```

Figură 62. Valorile min si max pt Total_Costs

Observăm că programul cu cel mai mare cost total (58208.98 USD) este Master in Artificial Intelligence de la Harvard University, situat în Boston, SUA. La polul opus, cel mai accesibil program (131.1773 USD) este Master in Computer Engineering, oferit de National University din Rosario, Argentina. Această diferență subliniază impactul semnificativ pe care îl au factorii geografici și prestigiul instituției asupra costurilor totale de studiu, influențând accesibilitatea programelor pentru studenți internaționali.

Apoi, am definit variabila țintă „Accessible” astfel încât să fie etichetată ca accesibilă (1) dacă costul total este mai mic sau egal cu 15.000 USD, iar inaccesibilă (0) în caz contrar. Astfel, din totalul programelor, 498 sunt clasificate ca accesibile, iar 296 ca inaccesibile.

```
  0   1
296 498
```

Figură 63. Distributia programelor

Pentru a putea fi utilizata in clasificare, variabila tinta a fost transformata in variabila de tip factor. Algoritmul KNN va atribui fiecărui program o clasă în funcție de proximitatea sa față de programele similare în setul de date, folosind distanța între observații.

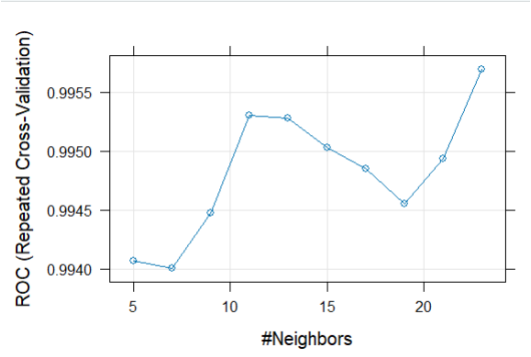
Setul de date a fost împărțit aleatoriu în două subseturi: 70% pentru antrenare și 30% pentru testare. Ulterior, nivelurile variabilei dependente *Accessible* (0 și 1) au fost transformate în denumiri valide, compatibile cu algoritmii de clasificare.

Modelul KNN a fost antrenat folosind validare încrucișată repetată (10 fold-uri repetate de 3 ori), pe baza a 5 variabile numerice relevante. Procesul a inclus standardizarea datelor și căutarea celui mai performant model în funcție de aria sub curba ROC.

Rezultatul a indicat că cel mai bun model este cel cu 23 de vecini, având o valoare ROC de aproximativ 0.996 — ceea ce sugerează o capacitate excelentă de clasificare între programele accesibile și cele inaccesibile.

```
k-Nearest Neighbors

557 samples
  5 predictor
  2 classes: 'X0', 'X1'

Pre-processing: centered (5), scaled (5)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 501, 502, 501, 501, 501, 502, ...
Resampling results across tuning parameters:

  k   ROC        Sens       Spec
   5  0.9940720  0.9630592  0.9823529
   7  0.9940073  0.9630592  0.9833333
   9  0.9944752  0.9584416  0.9833333
  11  0.9953028  0.9570707  0.9921569
  13  0.9952805  0.9538961  0.9941176
  15  0.9950323  0.9478355  0.9941176
  17  0.9948519  0.9431457  0.9941176
  19  0.9945569  0.9416306  0.9941176
  21  0.9949389  0.9416306  0.9941176
  23  0.9956922  0.9432179  0.9941176

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 23.
```

Figură 64. Output model knn



Figură 65. Reprezentare grafica

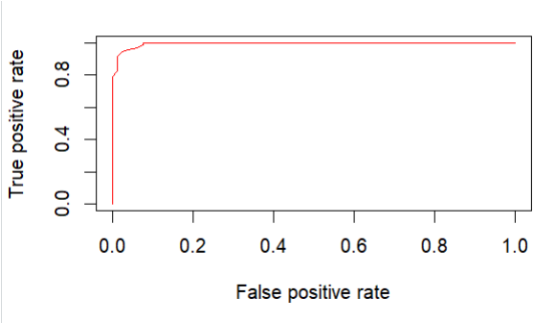
Predictii pe setul de testare. Evaluarea modelului.

Pe baza setului de testare, am realizat predicții utilizând modelul KNN, afișând rezultatele pentru primele șase observații sub forma probabilităților de apartenență la clasele „accesibil” și „inaccesibil”. Analiza arată că primul și al cincilea program sunt previzionate ca fiind inaccesibile, iar celelalte patru ca accesibile. Clasificarea s-a realizat în funcție de probabilitatea maximă estimată pentru fiecare observație.

```
> head(valid_pred)
          X0        X1
1 1.00000000 0.0000000
2 0.00000000 1.0000000
3 0.04347826 0.9565217
4 0.00000000 1.0000000
5 1.00000000 0.0000000
6 0.00000000 1.0000000
```

Figură 66. Probabilitatile pentru primele 6 observatii

Pentru evaluarea performanței modelului KNN pe setul de testare, am reprezentat grafic curba ROC. Graficul generat se apropie de colțul din stânga sus, ceea ce indică o capacitate bună de clasificare. Aria de sub curba ROC (AUC) este de 0.9947, ceea ce înseamnă că modelul are o acuratețe de aproximativ 99,4% în separarea programelor accesibile de cele inaccesibile — un rezultat foarte bun pentru un model de clasificare.



Figură 67. Reprezentarea ariei de sub curba ROC

```
> auc
[1] 0.9947524
```

Figură 68. Valoarea AUC

6. Retele neuronale

Rețelele neuronale artificiale (RNA) reprezintă o metodă de modelare inspirată din structura și funcționarea creierului uman, având capacitatea de a învăța relații complexe dintre variabile. Scopul utilizării rețelelor neuronale în această lucrare este dublu: pe de o parte, clasificarea unei variabile calitative multiclasă, iar pe de altă parte, realizarea unei regresii pentru a previziona valorile unei variabile numerice. Astfel, se va demonstra performanței rețelelor neuronale în contexte diferite de analiză a datelor.

6.1. Utilizarea RNA pentru clasificarea unei variabile calitative

In aceasta parte, voi construi o rețea neuronală pentru clasificarea unei variabile dependente calitative - Level, care indică nivelul studiilor și poate avea una dintre următoarele trei valori: *Bachelor*, *Master* sau *PhD*. Scopul acestei analize este de a determina în ce măsură caracteristicile numerice asociate programelor de studii pot fi utilizate pentru a prezice nivelul acestora.

```
> table(date_proiect$Level)

Bachelor   Master      PhD
     234      413      147
```

Figură 69. Categoriile variabilei "Level"

In primul rand, am eliminat din setul de date inițial toate variabilele de tip calitativ care nu sunt utile pentru antrenarea rețelei. Si apoi am extras un eșantion aleatoriu de 500 de observații pentru a construi setul de antrenare.

```
> head(train)
       Level Duration_Years Tuition_USD Living_Cost_Index Rent_USD Visa_Fee_USD
509   Master              2           0              69.8      800          350
557   Master              2        5900              55.4      480           75
196   Master              2        7000              85.1      920           90
620   Master              2        3600              75.4      900          180
212   Master              2        5900              77.8      650           90
135 Bachelor              3        5900              49.2      600          140
```

Figură 70. Esantionul aleatoriu

Pentru a permite rețelei neuronale să învețe apartenența fiecărei observații la una dintre cele trei clase (Bachelor, Master, PhD), am creat câte o variabilă binară pentru fiecare categorie si am eliminat variabila Level, deoarece a fost înlocuită de cele trei variabile binare corespunzătoare claselor.

```
> head(train)
    Duration_Years Tuition_USD Living_Cost_Index Rent_USD Visa_Fee_USD Bachelor Master   PhD
509              2           0              69.8      800          350    FALSE   TRUE FALSE
557              2        5900              55.4      480           75    FALSE   TRUE FALSE
196              2        7000              85.1      920           90    FALSE   TRUE FALSE
620              2        3600              75.4      900          180    FALSE   TRUE FALSE
212              2        5900              77.8      650           90    FALSE   TRUE FALSE
135              3        5900              49.2      600          140     TRUE  FALSE FALSE
```

Figură 71. Noua structura a setului de date

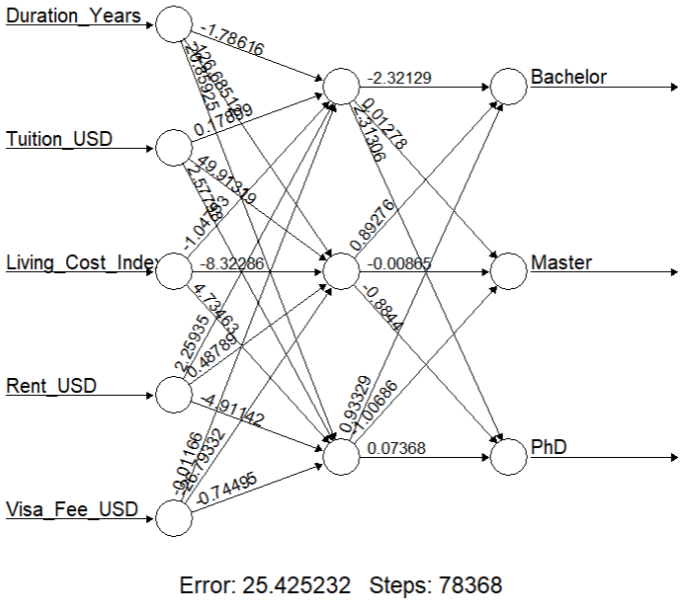
Deorece exista diferente mari de scara intre variabilele numerice, le-am standardizat calculând media și deviația standard pe setul de antrenare, apoi am aplicat această transformare atât pe datele de antrenare, cât și pe întregul set de date folosit pentru predicție.

Pentru antrenarea rețelei neuronale, am utilizat pachetul neuralnet, definind un model care are ca variabile de ieșire cele trei categorii binare corespunzătoare nivelului studiilor (Bachelor, Master, PhD), iar ca variabile explicative au fost incluse: Duration_Years, Tuition_USD, Living_Cost_Index, Rent_USD și Visa_Fee_USD, acestea reprezentând principalele caracteristici cantitative ale programelor de studii. Am configurat reteaua cu un singur strat ascuns ce conține 3 neuroni, alegere frecvent utilizată pentru modele simple de clasificare. Parametrul lifesign = "full" a permis afișarea detaliată a procesului de învățare, iar stepmax = 1e6 a fost setat pentru a asigura un număr suficient de pași în procesul de optimizare.

```
68000 min thresh: 0.0107907268889246
69000 min thresh: 0.0107907268889246
70000 min thresh: 0.0100383077102855
71000 min thresh: 0.0100383077102855
72000 min thresh: 0.0100383077102855
73000 min thresh: 0.0100383077102855
74000 min thresh: 0.0100383077102855
75000 min thresh: 0.0100383077102855
76000 min thresh: 0.0100383077102855
77000 min thresh: 0.0100383077102855
78000 min thresh: 0.0100383077102855
78368 error: 25.42523 time: 14.12 secs
```

Figură 72. Antrenarea modelului

Dupa antrenarea modelului, am reprezentat grafic structura retelei:



Figură 73. Structura retelei neronale

În urma antrenării modelului, observăm că rețeaua neuronală construită are o arhitectură formată din 5 neuroni în stratul de intrare (corespunzători celor 5 variabile explicative), un strat ascuns cu 3 neuroni și un strat de ieșire cu 3 neuroni, aferenți celor trei clase ale variabilei țintă. Dupa 78.368 de iteratii, s-a obtinut o eroare totală de aproximativ 25.425. Această valoare a erorii reprezintă suma pătratelor diferențelor dintre valorile prezise și cele reale pentru fiecare ieșire, și este un prim indicator al capacității de generalizare a rețelei pe setul de antrenare.

Pentru a evalua performanta retelei neuronale, am generat predicții asupra întregului set de date, după eliminarea coloanei Level, care conținea valorile reale ale variabilei țintă. Apoi, am afisat valorile nete calculate pentru fiecare dintre cele trei ieșiri ale rețelei. Aceste rezultate exprimă gradul de apartenență al fiecărui program de studii la una dintre cele trei categorii posibile.

```
> predictie$net.result[1:10,]
           [,1]      [,2]        [,3]
1  -0.005098643 0.9994124 0.005691020
3  -0.004926786 0.9994116 0.005519689
4  -0.004696220 0.9994103 0.005289965
5  -0.002818246 0.9994000 0.003418603
6  -0.003713867 0.9994049 0.004311061
7  -0.005062471 0.9994124 0.005654861
8  -0.005084518 0.9994125 0.005676830
9  -0.004312999 0.9994082 0.004908050
10 -0.004981465 0.9994119 0.005574144
11 -0.003467805 0.9994036 0.004065851
```

Figură 74. Valorile nete

Analizând rezultatele generate pentru primele 10 observații, se poate observa că valorile corespunzătoare clasei a doua — cea asociată programelor de master — sunt cele mai mari dintre cele trei ieșiri, adica modelul atribuie o probabilitate mai mare acestor observații de a aparține categoriei Master.

Am creat o variabilă denumită rezultat care conține, pentru fiecare observație, clasa cu cea mai mare probabilitate estimată de rețeaua neuronală. Ulterior, aceste valori prezise au fost adăugate într-o coloană nouă, denumită Predicted, în cadrul unui nou set de date numit comparatie, care conține atât valorile reale ale variabilei Level, cât și cele estimate de rețea. Analizand rezultatul afisat, se poate observa ca toate valorile au fost prezise corect.

```
> head(comparatie)
   Level Duration_Years Tuition_USD Living_Cost_Index Rent_USD Visa_Fee_USD Predicted
1 Master              2       55400              83.5     2200          160    Master
3 Master              2       38500              72.5     1600          235    Master
4 Master              2       42000              71.2     1400          450    Master
5 Master              2         500              70.5     1100           75    Master
6 Master              2        8900              76.4     1300          220    Master
7 Master              1       15800              73.2     1500          180    Master
```

Figură 75. Observatiile din setul de date "comparatie"

```
> comparatie[1:10, c(1,7)]
    Level Predicted
1  Master    Master
3  Master    Master
4  Master    Master
5  Master    Master
6  Master    Master
7  Master    Master
8  Master    Master
9  Master    Master
10 Master    Master
11 Master    Master
```

Figură 76. Comparatie intre valorile reale si cele previzionate

Evaluând matricea de confuzie din figura 77, observăm că dintre cele 208 observații clasificate ca fiind in categoria „Bachelor”, 184 au fost clasificate corect, 8 au fost etichetate eronat ca „Master”, iar 16 ca „PhD”. În cazul categoriei „Master”, din cele 407 observații, doar 2 au fost clasificate greșit ca „Bachelor”, restul fiind corect identificate. Pentru categoria „PhD”, dintr-un total de 179 observații, 48 au fost clasificate incorect ca „Bachelor”, restul

fiind atribuite corect. Aceste rezultate indică o performanță bună a rețelei, în special în recunoașterea programelor de tip „Master”.

	Bachelor	Master	PhD
Bachelor	184	2	48
Master	8	405	0
PhD	16	0	131

Figură 77. Matricea de confuzie

Pentru a evalua mai precis performanța modelului, au fost calculate mai multe statistici relevante. Indicatorul diag reprezintă procentul de clasificări corecte, adică 90,68% dintre observații au fost atribuite corect claselor. Coeficientul Kappa, cu valoarea de 0.85, reflectă reflectă un acord foarte bun între valorile reale și cele prezise. Indicele Rand (0.91) arată că majoritatea deciziilor de clasificare (atât clasificări corecte, cât și respingeri corecte ale apartenenței) sunt in conformitate cu realitatea. Valoarea crand (82,4%) confirmă că rețeaua neuronală are o performanță ridicată, chiar și după ajustare.

```
> classAgreement(tab)
$diag
[1] 0.906801

$kappa
[1] 0.8483168

$rand
[1] 0.9165431

$crand
[1] 0.8239793
```

Figură 78. Statistici

În concluzie, rețeaua neuronală antrenată reușește să clasifice în mod eficient cele trei niveluri de studii (Bachelor, Master, PhD), oferind o acuratețe ridicată și o bună concordanță cu datele reale.

6.2. Utilizarea RNA pentru previzionarea valorilor unei variabile numerice

Aceasta sectiune are ca scop dezvoltarea unui model de regresie neurală pentru estimarea costului total (Total_Cost) al unui program de studii. Motivul alegerii acestei variabile tinta este importanța sa practică în luarea deciziilor educaționale, iar modelul este construit utilizând doar două variabile explicative: durata programului (Duration_Years) și indicele costului de trai (Living_Cost_Index), evitând includerea altor variabile care intră direct în calculul costului total pentru a preveni colinearitatea.

În prima etapă a analizei, datele au fost curățate prin eliminarea coloanelor nenumerice și apoi împărțite într-un set de antrenare (75% dintre observații) și unul de testare (25%), utilizând funcția sample.split(). Ulterior, variabilele numerice au fost standardizate folosind metoda *min-max scaling*, pentru a asigura o scală unitară între 0 și 1, necesară pentru antrenarea eficientă a rețelei neuronale. Astfel, datele au fost pregatite pentru construcția și evaluarea modelului RNA de regresie.

	Duration_Years	Tuition_USD	Living_Cost_Index	Rent_USD	Visa_Fee_USD	Total_Cost
1	2.0	55400	83.5	2200	160	108360
3	2.0	38500	72.5	1600	235	77135
4	2.0	42000	71.2	1400	450	76050
5	2.0	500	70.5	1100	75	26975
6	2.0	8900	76.4	1300	220	40320
7	1.0	15800	73.2	1500	180	33980
8	1.5	35000	81.1	1900	90	69290
9	2.0	4500	74.6	1400	99	38199
10	2.0	1460	91.5	2100	88	51948
11	2.0	0	71.8	1200	110	28910
12	2.0	0	73.4	1300	120	31320
13	2.5	8900	52.3	800	140	33040
14	2.0	7200	68.7	900	130	28930
15	1.0	28900	72.9	1600	150	48250
16	4.0	5900	62.4	700	130	39630
17	3.0	28500	69.8	1200	245	71945
18	3.0	1500	67.4	950	160	35860
19	3.0	3500	68.9	1000	180	39680
20	4.0	52300	72.5	1800	160	138860
21	4.0	43800	71.5	1700	160	125560
22	4.0	40900	72.4	1600	160	117860

owing 1 to 21 of 794 entries, 6 total columns

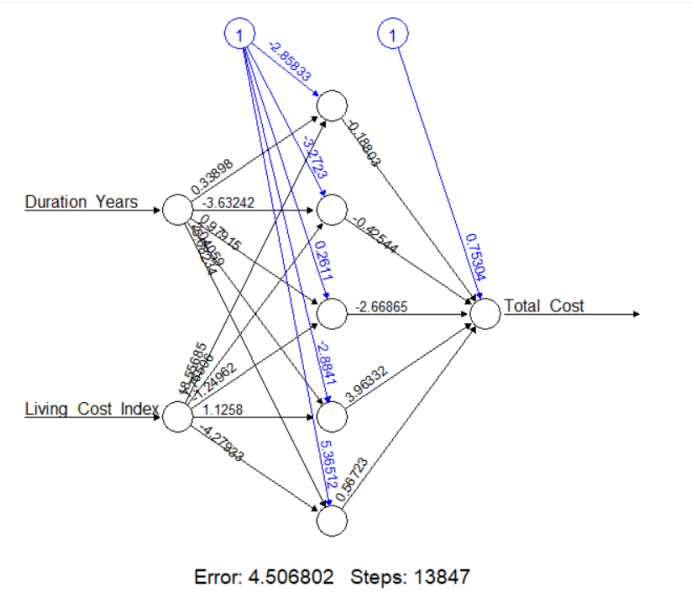
Figură 79. Datele inainte de standardizare

	Duration_Years	Tuition_USD	Living_Cost_Index	Rent_USD	Visa_Fee_USD	Total_Cost
1	0.250	0.955172414	0.81339713	0.9107143	0.29268293	0.55070603
3	0.250	0.663793103	0.63795853	0.6428571	0.47560976	0.38369170
4	0.250	0.724137931	0.61722488	0.5535714	1.00000000	0.37788832
5	0.250	0.008620690	0.60606061	0.4196429	0.08536585	0.11539902
6	0.250	0.153448276	0.70015949	0.5089286	0.43902439	0.18677792
7	0.000	0.272413793	0.64912281	0.5982143	0.34146341	0.15286692
8	0.125	0.603448276	0.77511962	0.7767857	0.12195122	0.34173085
9	0.250	0.077586207	0.67145136	0.5535714	0.14390244	0.17543325
10	0.250	0.025172414	0.94098884	0.8660714	0.11707317	0.24897304
11	0.250	0.000000000	0.62679426	0.4642857	0.17073171	0.12574882
12	0.250	0.000000000	0.65231260	0.5089286	0.19512195	0.13863928
13	0.375	0.153448276	0.31578947	0.2857143	0.24390244	0.14783911
14	0.250	0.124137931	0.57735247	0.3303571	0.21951220	0.12585580
15	0.000	0.498275862	0.64433812	0.6428571	0.26829268	0.22919341
16	0.750	0.101724138	0.47687400	0.2410714	0.21951220	0.18308729
17	0.500	0.491379310	0.59489633	0.4642857	0.50000000	0.35593175
18	0.500	0.025862069	0.55661882	0.3526786	0.29268293	0.16292255
19	0.500	0.060344828	0.58054226	0.3750000	0.34146341	0.18335473
20	0.750	0.901724138	0.63795853	0.7321429	0.29268293	0.71384253
21	0.750	0.755172414	0.62200957	0.6875000	0.29268293	0.64270432
22	0.750	0.705172414	0.63636364	0.6428571	0.29268293	0.60151904

ring 1 to 21 of 794 entries, 6 total columns

Figură 80. Datele dupa standardizare

Apoi, am construit modelul de rețea neuronală pentru a estima valoarea variabilei numerice Total_Cost în funcție de doi predictori: Duration_Years și Living_Cost_Index. Am antrenat reteaua utilizând setul de date standardizat pentru antrenare, specificând un strat ascuns format din 5 neuroni. Deoarece scopul este regresia (nu clasificarea), a fost setată opțiunea linear.output = TRUE, care permite modelului să producă rezultate continue. După antrenare, rețeaua a fost reprezentată grafic pentru a vizualiza structura și conexiunile dintre nodurile de intrare, stratul ascuns și ieșire.



Figură 81. Reprezentare grafica

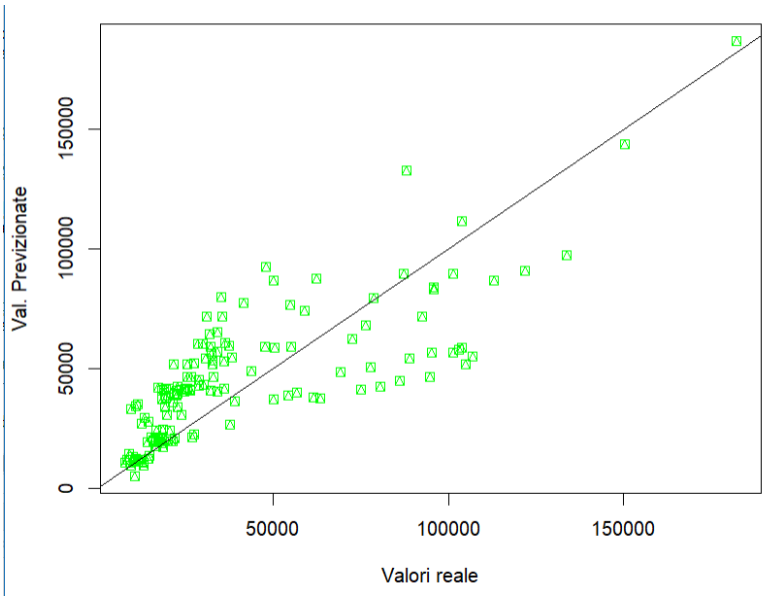
În urma antrenării modelului, observăm că rețeaua neuronală construită are o arhitectură formată din 2 neuroni în stratul de intrare (corespunzători celor 2 variabile explicative), un strat ascuns cu 5 neuroni și un strat de ieșire cu un neuron, aferent variabilei țintă. Dupa 13.847 de iteratii, s-a obtinut o eroare totală de aproximativ 4.506, ceea ce sugerează o capacitate rezonabilă a modelului de a învăța relația dintre variabilele analizate.

Pentru a evalua performanța modelului de rețea neuronală, am previzionat valorile utilizand setul de testare. Rezultatele generate au fost inițial în formă standardizată, astfel că a fost necesară rescalarea lor la dimensiunile reale. În final, valorile previzionate au fost vizualizate pentru a putea fi comparate cu cele reale.

```
> head(predictie)
        [,1]
26  41068.29
27  59442.40
34  53756.15
35 132952.10
48  79510.07
53  86809.27
```

Figură 82. Predictii

Pentru a vizualiza performanta modelului, au fost reprezentate grafic valorile reale ale costului total față de valorile previzionate de rețeaua neuronală.



Figură 83. Reprezentarea grafica a valorilor

Valorile reale sunt reprezentate pe axa Ox, iar valorile previzionate pe axa Oy. Linia trasată cu ajutorul funcției abline(0,1) marchează poziția ideală în care s-ar afla toate punctele în cazul unor predicții perfecte. Comparând distribuția punctelor cu această linie, putem observa existența unei erori în previzionare, indicând că modelul nu reproduce perfect valorile reale.

Pentru a evalua performanța modelului pe setul de testare, am calculat eroarea medie pătratică rădăcină (RMSE).

```
> eroare #eroarea este de 57840.79
[1] 57840.79
```

Figură 84. Eroarea modelului

Valoarea obținută a fost de aproximativ 57,840.79, ceea ce indică abaterea medie a predicțiilor față de valorile reale ale costului total. Aceasta ofera o estimare a preciziei modelului în previzionarea variabilei numerice țintă.

Iar pentru a vizualiza diferentele, am creat un data frame care conține valorile reale ale costului total din setul de testare și valorile previzionate de model.

	set_testare.Total_Cost	predictie
26	32420	41068.288
27	55350	59442.404
34	32975	53756.146
35	87988	132952.096
48	78750	79510.071
53	113035	86809.268
57	43920	49219.818
60	50535	58931.438
61	50220	86809.268
62	30320	43251.340
79	54450	39152.154
85	25775	52023.134
87	27525	52486.921
117	72635	62370.516
131	87370	89944.672
135	27640	22889.371
153	103890	111518.280
167	24699	40476.566
170	32799	51831.310
171	22199	39220.459
176	37548	59481.889
180	59068	74307.605
185	37840	26719.320

wing 1 to 24 of 147 entries, 2 total columns

Figură 85. Data frame cu valori reale si previzionate

In concluzie, observăm că modelul, deși în unele cazuri tinde să supraestimeze sau să subestimeze valorile, reușește totuși să previzioneze valori destul de apropiate de cele reale, indicând o performanță rezonabilă, dar cu o eroare specifică oricărui model predictiv.

Concluzii

Proiectul a avut ca scop explorarea, segmentarea și modelarea costurilor educației internaționale prin aplicarea unor metode variate de învățare automată și analiză statistică. Prin utilizarea clusterizării fuzzy, regresiei logistice (binare și multinomiale), arborilor de decizie, algoritmului KNN și rețelelor neuronale artificiale, s-au identificat tipare relevante privind structura costurilor și s-a realizat clasificarea programelor educaționale în funcție de accesibilitate, nivel de studiu și costuri.

Analiza a evidențiat diferențe clare în structura costurilor educației internaționale, reflectând diverse niveluri de accesibilitate economică. Modelele de clasificare și segmentare au demonstrat o capacitate ridicată de a distinge între grupuri de țări și programe educaționale cu costuri diferite, subliniind impactul factorilor economici locali asupra accesului la educație. Algoritmii utilizați au identificat variabilele cheie care influențează deciziile financiare ale studenților și instituțiilor, cum ar fi durata studiilor și costul vieții, evidențiind o corelație puternică între aceste elemente și costurile totale suportate. Rezultatele indică faptul că, deși există o anumită variabilitate în estimări, modelele pot susține luarea deciziilor economice prin furnizarea unor previziuni utile asupra costurilor educației internaționale.

În ansamblu, aceste metode au oferit o imagine detaliată și robustă asupra costurilor educației internaționale, susținând deciziile fundamentate privind accesibilitatea și planificarea programelor educaționale. Performanțele ridicate obținute la clasificare și predicție subliniază potențialul aplicării tehnicilor moderne de învățare automată în domeniul educației și al analizei socio-economice.

Anexe

Sursa bazei de date: https://www.kaggle.com/code/brunodvulhatka/international-education-costs-analysis/input

```
#############

# PROIECT ICE

#############

# Import fisierul csv "International_Education_Costs", care contine date despre costurile de educatie pentru studentii internationali


date_proiect <- read.csv("D:/Downloads (D)/International_Education_Costs.csv")

View(date_proiect)

boxplot(date_proiect$Duration_Years) # fara outlieri

boxplot(date_proiect$Tuition_USD) # fara

boxplot(date_proiect$Living_Cost_Index) # multi outlieri

boxplot(date_proiect$Rent_USD) # un outlier

boxplot(date_proiect$Visa_Fee_USD) # un outlier

# Gestionarea outlierilor
```

```
# Selectez doar coloanele numerice
numeric_vars <- sapply(date_proiect, is.numeric)
date_numerice <- date_proiect[, numeric_vars]
date_fara_outlieri <- date_proiect
# Pentru fiecare coloana numerica, aplic regula IQR si setez outlierii pe NA
for (var in names(date_numerice)) {
  Q1 <- quantile(date_fara_outlieri[[var]], 0.25, na.rm = TRUE)
  Q3 <- quantile(date_fara_outlieri[[var]], 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  date_fara_outlieri[[var]][date_fara_outlieri[[var]] < (Q1 - 1.5 * IQR) |
                            date_fara_outlieri[[var]] > (Q3 + 1.5 * IQR)] <- NA
}
# Elimin toate randurile care au cel puțin un NA (adica cel putin un outlier)
date_fara_outlieri <- na.omit(date_fara_outlieri)
dim(date_proiect)          # Dimensiunea initiala
dim(date_fara_outlieri)    # Dupa eliminarea outlierilor
date_proiect <- date_fara_outlieri
dim(date_proiect)
# Statistici descriptive
install.packages("psych")
library(psych)
summary(date_proiect[-c(1,2,3,4,5)])
describe(date_proiect[-c(1,2,3,4,5)])
# Matricea de corelatie
matrice_corelatie <- cor(date_proiect[-c(1,2,3,4,5)])
# Frevente
table(date_proiect$Country)
table(date_proiect$Level)
table(date_proiect$Country, date_proiect$Level)
# Reprezentari grafice
par(mfrow=c(1,3))
hist(date_proiect$Tuition_USD, main = "Distribuția taxelor de școlarizare", xlab = "Tuition (USD)", col = "skyblue")
hist(date_proiect$Rent_USD, main = "Distribuția chiriei lunare", xlab = "Rent (USD)", col = "lightgreen")
hist(date_proiect$Living_Cost_Index, main = "Distribuția costului de trai", xlab = "Living Cost Index", col = "lightpink")
barplot(sort(table(date_proiect$Country), decreasing = TRUE), las=2, col="orange", main="Număr de programe pe țară")
barplot(sort(table(date_proiect$Level), decreasing = TRUE), col="steelblue", main="Distribuția programelor pe niveluri")
####################
# Clusterizare Fuzzy
####################
```

```
# Elimin variabilele categoriale/de tip string
date_proiect1 <- date_proiect[,-c(1,2,3,4,5)]
View(date_proiect1)
install.packages("e1071")
library(e1071)
# Voi clusteriza/grupa datele in 3 clustere (3 categorii privind costul educatiei: accesibil, mediu, scump)
set.seed(123)
rez <- cmeans(date_proiect1, 3, 100, m=2, method = "cmeans")
rez
# Centroizii clusterelor (mediile clusterelor):
# Primul cluster prezinta o durata medie a studiilor de 2.73 ani (aproximativ 3 ani), in timp ce clusterul 2 are o durata medie de
# 3.13 ani, iar clusterul 3 o durata de 3.13 ani => duratele medii sunt de aproximativ 3 ani pentru toate clusterele identificate
# In ceea ce priveste taxele de scolarizare, clusterul 1 prezinta cea mai mica valoare medie, de 3667.928 USD, urmata de clusterul 2
# cu valoarea de 29077.257 USD, iar, in cele din urma, cea mai mare valoare medie o prezinta clusterul 3, fiind de 46979.804 USD
# Indicele cheltuielilor de trai (mancare, transport, utilitati) pentru clusterul 1 este de 59.54, urmat de clusterul 2 cu 67.14 si
# clusterul 3 cu valoarea medie de 75.41
# Chiria medie in cazul clusterului 1 este de 673.318 USD, pentru clusterul 2 este de 1191.627 USD, iar pentru clusterul 3 este de 1776.425 USD
# Pentru clusterul 1 costul mediu al vizei (in USD) este de 130.476 USD, pentru clusterul 2 este de 299.422 USD, iar pentru clusterul 3 este de 224.7908 USD
# Analizand aceste rezultate, putem ajunge la urmatoarele concluzii:
# Clusterul 1 - Educatie accesibila/cu costuri reduse - include tari cu cost total redus al educatiei internationale
# Clusterul 2 - Educatie cu costuri medii - reprezinta un compromis intre cost si calitate, cu taxe si costuri moderate
# Clusterul 3 - Educatie foarte scumpa/cu costuri ridicate - include tari in care costurile sunt foarte ridicate, reflectand un sistem educational "premium"
# Interpretare Memberships (gradele de apartenenta la fiecare cluster):
# Pentru prima observatie:
# Gradul de apartenenta la primul cluster este de 0.023, pentru al doilea cluster este de 0.090, iar pentru al treilea cluster este de 0.88
# In concluzie, prima observatie (Programul de Master in Computer Science la Harvard, in Cambridge, USA) apartine celui de-al treilea cluster
# Deci acest program educational face parte din categoria cu costuri ridicate, fiind un sistem educational "premium"
# Reprezentarea grafica a observatiilor intr-un sistem de axe xOy, unde Tuition_USD va fi pe Ox si Living_Cost_Index pe Oy
plot(date_proiect1$Tuition_USD, date_proiect1$Living_Cost_Index, col = rez$cluster)
points(x = rez$centers[,"Tuition_USD"],
       y = rez$centers[,"Living_Cost_Index"],
       col =1:3, pch=9, cex=2)
text(x=date_proiect$Tuition_USD, y =date_proiect1$Living_Cost_Index,
     col=rez$cluster, cex = 0.6, pos = 4, offset = 0.5)
# Centroizii au fost reprezentati prin romburi
# Observatiile reprezentate cu negru au taxe de scolarizare scazute si costurile de trai moderate in majoritatea cazurilor,
# existand si anumite observatii cu costuri de tari foarte ridicate (cele mai ridicate din intreg graficul) => clusterul 1
# Observatiile reprezentate cu rosu au taxe de scolarizare moderate si costuri de trai medii => clusterul 2
```

```
# Observatiile reprezentate cu verde au cele mai mari taxe de scolarizare si costuri de trai mai mari decat in cazul celorlalte

# 2 clustere => clusterul 3

# Reprezentarea grafica a variabilelor in functie de cluster

date_proiect$Cluster <- rez$cluster

boxplot(Tuition_USD ~ Cluster, data = date_proiect, main = "Taxe în funcție de cluster", col=2:4)

boxplot(Living_Cost_Index ~ Cluster, data = date_proiect, main = "Cost de trai în funcție de cluster", col=2:4)

# Ordonarea crescatoare a observatiilor dupa cluster

ordine <- order(rez$cluster)

ordine

# Afisarea denumirii fiecarei universitati si clusterul din care face parte

df_clustere <- data.frame(date_proiect$University[ordine], rez$cluster[ordine])

View(df_clustere)

# Afisarea gradelor de apartenenta pentru fiecare observatie

df_membership <- data.frame(date_proiect$University, rez$membership)

View(df_membership)

# Afisarea gradelor de apartenenta la cele 3 clustere pentru primele 3 observatii

rez$membership[1:3,]

# Observam ca toate cele 3 observatii apartin clusterului 3 - educatie foarte scumpa, premium

# Verificarea clusterizarii

library(cluster)

sil <- silhouette(rez$cluster, dist(date_proiect1))

plot(sil, border = NA)

####################

# Regresie logistica

####################

# 1. Regresia logistica binomiala

# Pentru a clasifica tarile in high cost/low cost (dpdv al educatiei), voi crea o variabila binara (variabila tinta)

# O voi denumi High_Cost si va depinde de indicele costului de trai (Living_Cost_Index)

# Daca acesta este mai mare decat mediana, atunci high cost (1), altfel low cost (0)

# Voi folosi mediana ca prag

mediana <-  median(date_proiect$Living_Cost_Index, na.rm = TRUE)

mediana

# Creez variabila binara

date_proiect$High_Cost <- ifelse(date_proiect$Living_Cost_Index > mediana, 1, 0)

table(date_proiect$High_Cost)

# 399 de tari sunt low cost, 395 de tari sunt high cost

install.packages("ggplot2")

library(ggplot2)

# Reprezentarea grafica a datelor
```

```
grafic <- ggplot(data=date_proiect, aes(x=date_proiect$Tuition_USD, y=date_proiect$Rent_USD, col=High_Cost))

grafic <- grafic+geom_point(aes(size=5))

grafic

# Variabila dependenta este High_Cost, ce poate avea valorile 0 (low cost) si 1 (high cost)

# Variabilele independente sunt: Tuition_USD (taxa de scolarizare) si Rent_USD (costul chiriei)

# Observam, pe baza graficului, o evolutie a variabilei High_Cost de la 0 la 1, putem observa ca exista un raport echilibrat intre

# numarul de programe educationale scumpe si accesibile

# Transformarea variabilei dependente in variabila de tip factor

date_proiect$High_Cost <- factor(date_proiect$High_Cost)

install.packages("caTools")

library(caTools)

set.seed(88)

# Impartirea setului de date in antrenare si testare

# 75% antrenare

# 25% testare

impartire <- sample.split(date_proiect$High_Cost, SplitRatio = 0.75)

impartire

# Definirea setului de antrenare

set_antrenare <- subset(date_proiect, impartire == TRUE)

set_antrenare

# Definirea setului de testare

set_testare <- subset(date_proiect, impartire == FALSE)

set_testare

# Refac graficul anterior

grafic <- ggplot(data=date_proiect, aes(x=date_proiect$Tuition_USD, y=date_proiect$Rent_USD, col=High_Cost))

grafic <- grafic+geom_point(aes(size=5))

grafic

# Aplicarea regresiei logistice binomiale, utilizand glm

# Voi alege ca variabile explicative taxa de scolarizare si costul chiriei

model_regresie <- glm(High_Cost~Tuition_USD+Rent_USD, data=set_antrenare, family=binomial())

summary(model_regresie)

# Interpretare:

# Ecuatia regresiei logistice binomiale:

# logit(p) = -6.2355035 - 0.0001371 × Tuition_USD + 0.0092349 × Rent_USD

# unde p reprezinta probabilitatea ca tara sa fie scumpa din punct de vedere al educatiei (High_Cost = 1)

# 1-p = probabilitatea ca tara sa fie ieftina din punct de vedere al educatiei (High_Cost = 0)

# Varianta reziduala este de 375.12, fiind mai mica decat devianta nula (824.83), asadar modelul actual este mai bun decat modelul nul

# Toate variabilele sunt semnificative dpdv statistic

# Impactul variabilelor analizate asupra sanselor ca o tara sa fie scumpa dpdv al programelor educationale pentru studenti internationali
```

```
exp(coef(model_regresie))

# Interpretare:

# Sansele ca o tara sa fie scumpa scad cu aproximativ 0.0137% ((0.999862874-1)*100) relativ la sansele ca tara

# sa fie ieftina, daca taxa de scolarizare creste cu o unitate (1 USD)

# Sansele ca o tara sa fie scumpa cresc cu aproximativ 0.927% ((1.009277665-1)*100) relativ la sansele ca tara

# sa fie ieftina, daca chiria creste cu o unitate (1 USD)

# Factorizarea categoriilor

contrasts(date_proiect$High_Cost)

# Se atribuie valoarea 0 tarilor ieftine si 1 tarilor scumpe

# Previzionarea probabilitatilor ca o tara sa fie scumpa/ieftina, utilizand setul de testare

probabilitati <- predict(model_regresie, set_testare, type='response')

probabilitati

# Probabilitatea ca prima tara sa fie scumpa este 0.98, pe cand probabilitatea ca a doua tara sa fie scumpa este 0.99

# Voi defini un vector de valori pentru "0" (ieftina) corespunzator numarului de observatii din setul de antrenare

# Elementul 0 este convertit la 1 atunci cand probabilitatea previzionata ca o tara sa fie scumpa este mai mare de 50%

predictie <- rep("0", nrow(set_antrenare))

predictie[probabilitati>.5] = "1"

# Matricea de confuzie pt setul de antrenare

table(predictie, set_antrenare$High_Cost)

# Interpretare:

# Setul de antrenare contine 138+161=299 tari ieftine si 138+158=296 tari scumpe

# Din cele 296 tari scumpe, 158 au fost clasificate in mod corect ca fiind scumpe, iar 138 au fost clasificate eronat ca fiind scumpe

# Din cele 299 tari ieftine, 138 au fost clasificate corect, iar 161 au fost clasificate in mod eronat ca fiind ieftine

# Pe diagonala principala se regasesc observatiile previzionate corect, si anume 138+158=296 observatii

# Acuratetea clasificarii se determina ca raportul dintre suma elementelor de pe diagonala principala si totalul elementelor:

(138+158)/(138+138+161+158)

# Acuratetea modelului este de 49.7% (adica 49.7% din elementele din setul de antrenare au fost clasificate in mod corect)

# Voi introduce inca 2 tari si voi previziona daca acestea sunt scumpe sau ieftine

predictie_noua <- predict(model_regresie, newdata=data.frame(Tuition_USD=c(10000,1000), Rent_USD=c(1000,650)), type='response')

predictie_noua

# Evaluez daca cele doua tari sunt scumpe, comparand probabilitatile previzionate cu 0.5

predictie_noua[1] <= 0.5

# Prima tara este scumpa

predictie_noua[2] <=0.5

# A doua tara este ieftina

# Predictia pe setul de testare

predictie1 <- rep("0", nrow(set_testare))

predictie1[probabilitati>.5] = "1"

# Matricea de confuzie pt setul de testare
```

```
table(predictie1, set_testare$High_Cost)

# In setul de testare avem 8+91=99 tari scumpe si 84+16=100 tari ieftine

# Din cele 99 tari scumpe, 91 au fost etichetate corect, iar 8 incorect

# Din cele 100 tari ieftine, 84 au fost etichetate corect, iar 16 incorect

mean(predictie1==set_testare$High_Cost)

# Acuratetea este de 87.9% pentru setul de testare (aproximativ 87% din observatii au fost etichetate corect)

# Indicatorii matricei de confuzie:

#Specificitate: TN/(TN+TP)

91/(91+84) # Modelul clasificator recunoaste in proportie de 52% toate cazurile negative observate

#Senzitivitate: TP/(TP+FN)

84/(84+8) # Modelul clasificator recunoaste in proportie de 91% toate cazurile pozitive observate

# Curba ROC

install.packages("ROCR")

library(ROCR)

p <- predict(model_regresie, newdata = set_testare, type='response')

pr <- prediction(p, set_testare$High_Cost)

prf <- performance(pr, measure = "tpr", x.measure="fpr")

plot(prf)

# Graficul se apropie de coltul din stanga sus, ceea ce este ideal

auc <- performance(pr, measure="auc")

auc <- auc@y.values[[1]]

auc

# Valoarea de 0.94 a AUC indica o clasificare excelenta, existand 94% sanse ca modelul sa poata distinge clasa pozitiva de cea negativa

##################################

# Regresie logistica multinomiala

##################################

# Variabila tinta categoriala (multinomiala) utilizata va fi Level, cu 3 valori posibile: "Master", "Bachelor", "PhD"

unique(date_proiect$Level)

# Transform variabila Level in variabila factor

date_proiect$LevelF <- factor(date_proiect$Level)

# Voi seta ca nivel de referinta valoarea "Bachelor"

date_proiect$out <- relevel(date_proiect$LevelF, ref="Bachelor")

# Aplicam modelul de regresie logistica multinomiala

install.packages("nnet")

library(nnet)

model_regresie_multi <- multinom(out~Tuition_USD+Duration_Years, data=date_proiect, trace=FALSE)

summary(model_regresie_multi)

# Devianta reziduala este eroarea ramasa in model, iar valoarea acesteia trebuie sa fie cat mai mica

# In acest caz, este de 416.4454
```

Conform tabelului de mai sus, se pot determina ecuatiile urmatoare pentru determinarea probabilitatilor:

ln[P(Master)/P(Bachelor)] = 44.44995 - 0.0000337*Tuition_USD -15.562459*Duration_Years (1)

Interpretare: Logaritmul probabilitatii ca un program sa fie de Master raportata la probabilitatea ca un un program sa fie de Bachelor se numeste "log odds"

Coeficientul negativ -0.0000337 indica faptul ca variabila "Tuition_USD" are un impact negativ asupra raportului

O crestere cu o unitate a taxei de scolarizare conduce la scaderea logaritmului log odds cu 0.0000337

De asemenea, coeficientul negativ de -15.562 arata ca variabila "Duration_Years" are un impact negativ, o crestere cu o unitate a duratei programului educational

determinand o scadere a logaritmului cu 15.562

ln[P(PhD)/P(Bachelor)] = -16.18454 - 0.0000401*Tuition_USD + 4.220675*Duration_Years (2)

Interpretare:

Coeficientul negativ -0.0000401 indica un impact negativ al taxei de scolarizare asupra raportului, o crestere cu o unitate a taxei de scolarizare

determinand o scadere a log cu 0.0000401

Coeficientul pozitiv 4.22 indica un impact pozitiv asupra raportului al duratei, o crestere cu o unitate a duratei determinand o crestere a log cu 4.22

exp(coef(model_regresie_multi))

Interpretare:

Sansele ca un program sa fie de Master sunt cu 0.00338% mai mici decat sansele ca programul sa fie de Bachelor, daca taxa de scolarizare creste cu o unitate (1 USD)

Sansele ca un program sa fie de Master sunt cu 99.99% mai mici decat sansele ca programul sa fie de Bachelor, daca durata prorgamului creste cu o unitate (1 an)

Sansele ca un program sa fie de PhD sunt cu 0.00402% mai mici decat sansele ca programul sa fie de Bachelor, daca taxa de scolarizare creste cu o unitate (1 USD)

Sansele ca un program sa fie de PhD sunt cu 6707.94% mai mari decat sansele ca programul sa fie de Bachelor, daca durata prorgamului creste cu o unitate (1 an)

adica pt fiecare an in plus in durata programului, sansele ca programul sa fie PhD fata de Bachelor cresc de 68 ori

Suma probabilitatilor este egala cu 1:

P(Bachelor)+P(Master)+P(PhD)=1

Determinarea probabilitatilor folosind predict

predict(model_regresie_multi, date_proiect)

predict(model_regresie_multi, date_proiect, type="prob")

Probabilitatea ca primul program sa fie de Bachelor este de 1.06*10^-5, sa fie de Master 9.99*10^-1 si sa fie PhD 4.97*10^-10 => programule este de Master

Previzionarea valorilor pentru observatiile 10, 200 si 650

predict(model_regresie_multi, date_proiect[c(10,200,650),], type = "prob")

Primul este program de Master, al doilea si al treilea sunt programe de Bachelor

Compararea predictiilor modelului cu date reale pentru primele 50 de observatii

matrice_confuzie <- table(date_proiect$Level[1:50], predict(model_regresie_multi)[1:50])

matrice_confuzie

Interpretare:

Din 22 programe de Bachelor, 18 au fost etichetate corect, iar 4 au fost etichetate in mod eronat ca fiind PhD

Din 17 programe de Master, toate au fost clasificate corect

Din 11 programe de PhD, 7 au fost identificate in mod corect, iar 4 in mod eronat fiind clasificate ca Bachelor

```
# Observam ca a existat o confuzie intre Bachelor si PhD, iar programele de Master au fost toate identificate corect

mean(date_proiect$Level[1:50] == predict(model_regresie_multi)[1:50])

# Acuratetea modelului este de 84%, ceea ce indica o clasificare excelenta a datelor

##################

#Arbori de decizie

##################

install.packages("ISLR")

install.packages("rpart")

install.packages("pROC")

library(ISLR)

library(rpart)

library(pROC)

# Crearea variabilei tinta binare

range(date_proiect$Living_Cost_Index) # valori intre 32.5 si 95.2

hist(date_proiect$Living_Cost_Index)

# Transform variabila Living_Cost_Index in variabila binara "High_Living_Cost" cu valorile "Yes" daca > 70 si "No" in caz contrar

date_proiect$High_Living_Cost <- ifelse(date_proiect$Living_Cost_Index > 65,'Yes','No')

# Voi pastra doar variabilele dorite si variabila tinta

date_model <- date_proiect[, c("Level", "Duration_Years", "Tuition_USD", "Rent_USD",

                               "Visa_Fee_USD", "High_Living_Cost")]

# Transform variabila in variabila de tip factor

date_model$High_Living_Cost <- as.factor(date_model$High_Living_Cost)

# Extrag doua esantioane egale, unul pt antrenare si unul pt testare

set.seed(123)

antrenare <- sample(1:nrow(date_model), nrow(date_model)/2)

set_antrenare <- date_model[antrenare,]

set_antrenare

set_testare <- date_model[-antrenare,]

set_testare

# Definirea si vizualizarea arborelui de clasificare

arbore <- rpart(set_antrenare$High_Living_Cost~., data=set_antrenare, method="class")

# Relatia de dependenta liniara este intre eticheta High Living Cost cu valorile yes(cost de trai ridicat) sau no (cost de trai scazut) si restul atributelor setului de
date

# Reprezentarea grafica a arborelui

plot(arbore)

text(arbore, pretty=0)

install.packages("rpart.plot")

library(rpart.plot)

rpart.plot(arbore, extra=106)
```

```
table(set_antrenare$High_Living_Cost)

# In nodul 1, din 397 de observatii, 229 sunt etichetate ca fiind clasa "Yes" (cost de trai ridicat),

# iar 168 sunt clasificate ca fiind clasa "No" (cost de trai scazut)

# Vectorul de probabilitate este (0.42, 0.58), indicand faptul ca cele 229 de observatii sunt in categoria "yes"

# cu o probabilitate de 58%

# Clasa dominanta in nodul 1 este "Yes" (costuri de trai ridicate), cu o probabilitate de 58%

# Predictia observatiilor din setul de testare

predictie <- predict(arbore, set_testare, type="class")

confuzie <- table(set_testare$High_Living_Cost, predictie)

confuzie

# Interpretare matrice de confuzie:

# Din 180 observatii etichetate in clasa "No" (cost scazut), 159 au fost clasificate corect, iar 21 gresit

# Din 217 observatii ectihetate in clasa "Yes" (cost ridicat), 194 au fost clasificate corect, iar 23 gresit

#Eroarea de clasificare

mean(predictie!=set_testare$High_Living_Cost)

# 11.08% din observatii au fost eronat clasificate

predictie1 <- predict(arbore, set_testare, type="prob")

predictie1

# Reprezentarea curbei ROC

curbaroc <- roc(set_testare$High_Living_Cost, predictie1[,"Yes"])

plot(curbaroc)

auc(curbaroc)

# Clasificatorul distinge in proportie de 95.51% costurile de trai ridicate de cele scazute

# Determinarea parametrului de complexitate optim (cp)

plotcp(arbore)

mincp <- arbore$cptable[which.min(arbore$cptable[,"xerror"]), "CP"]

mincp

printcp(arbore)

# Eroarea relativa de validare incrucisata este minima (xerror = 0.25) pt o valoare a parametrului de

# complexitate de 0.01, iar arborele curatat are aproximativ 9 noduri terminale (nsplit=8)

# Construirea arborelui curatat

arbore_curat <- prune(arbore, cp=arbore$cptable[which.min(arbore$cptable[,'xerror']),"CP"])

rpart.plot(arbore_curat, extra=106) # arborele curatat are 9 noduri terminale

printcp(arbore_curat) # se confirma ca cp minim este 0.013158, cu 9 noduri terminale

# Predictia pe arborele curatat

predictie2 <- predict(arbore_curat, set_testare, type="class")

predictie2

# Matricea de confuzie

confuzie <- table(set_testare$High_Living_Cost, predictie2)
```

```
confuzie

mean(predictie2!=set_testare$High_Living_Cost)

# 11.08% observatii au fost eronat clasificate (eroare a scazut de la 18.7%)

predictie3 <- predict(arbore_curat, set_testare, type="prob")

predictie3

# Curba ROC

curbaroc1 <- roc(set_testare$High_Living_Cost, predictie3[,"Yes"])

plot(curbaroc1)

auc(curbaroc1)  # Acuratete de 95.51%

####################

# Arbori de regresie

####################

install.packages("tree")

install.packages("ISLR")

library(tree)

library(ISLR)

# Scopul este de a estima costul chiriei

# Histograma chiriilor

hist(date_proiect$Rent_USD)

# Pentru avea o distributie normala a chiriilor, vom logaritma variabila

date_proiect$Rent_USD <- log(date_proiect$Rent_USD)

# Refacem histograma

hist(date_proiect$Rent_USD)

install.packages("caret")

library(caret)

# Impartirea setului de date in antrenare si testare (50% antrenare, 50% testare)

impartire <- createDataPartition(y=date_proiect$Rent_USD, p=0.5, list=FALSE)

impartire

# Setul de antrenare

antrenare <- date_proiect[impartire,]

antrenare

# Setul de testare

testare <- date_proiect[-impartire,]

testare

# Definirea arborelui de decizie

arbore <- tree(Rent_USD~., antrenare)

plot(arbore)

text(arbore, pretty = 0)

# Fiecare nod terminal al arborelui prezinta chiria logaritmata pentru observatiile din nodul respectiv
```

```
# Curatarea arborelui

cv.trees <- cv.tree(arbore)

cv.trees$size

cv.trees$dev

# In acest caz, cea mai mica eroare este 0.55 aferenta numarului de 8 noduri

# Reprezentare grafica a deviantei in functie de nr de noduri

plot(cv.trees$size, cv.trees$dev, type = "b")

# Din grafic deducem faptul ca eroarea minima se atinge cand avem 8 noduri terminale

# Construim arborele curatat

arbore1 <- prune.tree(arbore, best = 8)

plot(arbore1)

text(arbore1, pretty = 0)

# Predictii pe setul de testare

pred <- predict(arbore1, testare)

pred # in acest rezultat sunt afisate valorile previzionate pentru chirie (forma logaritmata)

# Reprezentarea grafica a valorilor previzionate pentru chirii din setul de testare

plot(pred, testare$Rent_USD)

# Eroarea de predictie

mean((pred-testare$Rent_USD)^2)

# 0.0013% din observatii au fost previzionate eronat => deci modelul de previziune este bun

################

# Algoritmul KNN

################

library(caret)

library(e1071)

# Scopul este de a clasifica programele ca "Program accesibil" vs "Program inaccesibil", pe baza costurilor totale estimate

# Calcularea costului total

date_proiect$Total_Cost <- date_proiect$Tuition_USD + (date_proiect$Rent_USD*12*date_proiect$Duration_Years)+date_proiect$Visa_Fee_USD

df_totalcost <- data.frame(date_proiect$University, date_proiect$Total_Cost)

View(df_totalcost)

# Valoarea maxima pt Total Cost

date_proiect[which.max(date_proiect$Total_Cost),]

# Valoarea minima pt Total Cost

date_proiect[which.min(date_proiect$Total_Cost),]

# Variabila tinta va fi Accesible si va avea valorile: accesibil (1) daca costul total <= 15000, inaccesibil (0) altfel

date_proiect$Accessible <- ifelse(date_proiect$Total_Cost <= 15000, 1, 0)

table(date_proiect$Accessible) # 296 programe sunt inaccesibile, 498 sunt accesibile

# Transform variabila in variabila de tip factor
```

```
date_proiect$Accessible <- as.factor(date_proiect$Accessible)

set.seed(101)

# Impartirea setului de date in antrenare si testare

index <- sample(2, nrow(date_proiect), replace = TRUE, prob = c(0.7,0.3))

index

set_antrenare <- date_proiect[index==1,]

set_antrenare

set_testare <- date_proiect[index==2,]

set_testare

# In mod implicit, nivelurile variabilei dependente sunt 0 si 1

# Le voi transforma in nume de variabile valide

levels(set_antrenare$Accessible) <- make.names(levels(factor(set_antrenare$Accessible)))

levels(set_testare$Accessible) <- make.names(levels(factor(set_testare$Accessible)))

# Validare incrucisata

# Voi defini repeats=3, numbers=10

repeats=3

numbers=10

set.seed(1234)

x=trainControl(method="repeatedcv", number=numbers, repeats = repeats, classProbs = TRUE, summaryFunction = twoClassSummary)

tunel = 10

# Definirea modelului KNN

# Voi utilza doar variabilele numerice in model

model_knn <- train(Accessible~Duration_Years+Tuition_USD+Living_Cost_Index+Rent_USD+Visa_Fee_USD, data= set_antrenare, method="knn",
preProcess=c("center", "scale"), trControl=x, metric='ROC', tuneLength = tunel)

model_knn

# Cea mai mare valoare a ariei de sub curba ROC este 0.9956922, aferenta modelului optim cu cei mai apropiati 23 vecini

# Valoarea curbei ROC in acest caz este apropiata de 1, fiind un model bun pentru separarea programelor accesibile de cele inaccesibile

plot(model_knn)

# Predictii pe setul de testare

valid_pred <- predict(model_knn, set_testare, type="prob")

head(valid_pred)

# Pentru primele 6:

# Primul program este previzionat a fi inaccesibil

# Al doilea accesibil

# Al treilea accesibil

# Al patrulea acc

# Al cincilea inacc

# Al saselea acc

# Reprezentare grafica a ariei de sub curba ROC pt setul de testare
```

```
library(ROCR)

pred_val <- prediction(valid_pred[,2], set_testare$Accessible)

perf_val <- performance(pred_val, "tpr", "fpr")

plot(perf_val, col="red", lwd=1.5)

# Graficul se apropie de coltul din stanga sus, ceea ce este ideal

auc <- performance(pred_val, measure="auc")

auc <- auc@y.values[[1]]

auc

# Aria de sub curba ROC este de 0.9947524, indicand 99.4% sanse de a separa clasa pozitiva de clasa negativa

##################

# Retele neuronale

##################

# 1. Retele neuronale cand variabila dependenta are > 2 clase

# Variabila calitativa utilizata va fi 'Level', care ia valorile: 'Bachelor', 'Master' sau 'PhD'

table(date_proiect$Level)

# Vom elimina celelalte variabile calitative

date_rn <- date_proiect[,-c(1,2,3,4)]

View(date_rn)

# Extrag in mod aleatoriu 500 observatii pt setul de antrenare:

set.seed(123)

train <- date_rn[sample(1:794,500),]

head(train)

# Adaug in setul de antrenare atribute ce convin valoarea de adevar a apartenentei fiecarei observatii la cele 3 categorii

train$Bachelor <- c(train$Level=="Bachelor")

train$Master <- c(train$Level=="Master")

train$PhD <- c(train$Level=="PhD")

# Elimin variabila 'Level' din setul de date

train$Level <- NULL

head(train)

# Voi standardiza variabilele numerice, deoarece au scale diferite

var_exp <- c("Duration_Years", "Tuition_USD", "Living_Cost_Index", "Rent_USD", "Visa_Fee_USD")

# Media si deviatia standard pe setul de antrenare

means <- apply(train[, var_exp], 2, mean)

sds <- apply(train[, var_exp], 2, sd)

# Standardizez datele din train

train[, var_exp] <- scale(train[, var_exp], center = means, scale = sds)

# Standardizez toate datele pentru predicție (date_rn), folosind media și sd din train

date_rn_scaled <- date_rn

for (v in var_exp) {
```

```
  date_rn_scaled[, v] <- (date_rn_scaled[, v] - means[v]) / sds[v]

}

# Antrenarea retelei neuronale ce contine 3 noduri in stratul ascuns:

library(neuralnet)

retea_neuronala <- neuralnet(Bachelor+Master+PhD~Duration_Years+Tuition_USD+Living_Cost_Index+Rent_USD+Visa_Fee_USD,train,hidden = 3,lifesign =
"full",stepmax = 1e6)

plot(retea_neuronala, rep="best", intercept=FALSE

# Predictii pentru nivelul programului de studii, dupa eliminarea coloanei 1 ('Level')

predictie <- compute(retea_neuronala,date_rn_scaled[,-c(1)])

predictie

# Gradul de apartenenta al fiecarui program la cele 3 categorii

predictie$net.result[1:10,]

# Variabila rezultat va contine clasele "Bachelor", "Master" si "PhD", pe baza predictiei

rez <- 0

for(i in 1:794){rez[i] <- which.max(predictie$net.result[i,])}

for(i in 1:794){if(rez[i]==1){rez[i]="Bachelor"}}

for(i in 1:794){if(rez[i]==2){rez[i]="Master"}}

for(i in 1:794){if(rez[i]==3){rez[i]="PhD"}}

# Comparam rezultatele cu valorile reale

comparatie <- date_rn

comparatie$Predicted <- rez

comparatie

head(comparatie)

# Variabila denumita comparatie cuprinde col 1 (val reale) si 7 (val previzonate )

comparatie[1:10, c(1,7)]

# Matricea de confuzie

tab <- table(comparatie$Level, comparatie$Predicted)

tab

install.packages("e1071")

library(e1071)

classAgreement(tab)

# 2. Retele neuronale in regresie (previzionarea valorilor unei variabile numerice)

# Variabila numerica tinta va fi Total_Cost (costul total) pe care imi doresc sa o previzionez in

# functie de durata programului si indicele costului de trai (nu voi folosi celelalte variabile deoarece

# sunt componente directe ale costului total)

View(date_proiect)

set.seed(123)

install.packages("caTools")

library(caTools)
```

```
# Voi elimina coloanele care nu sunt numerice:
date_proiect1 <- date_proiect[,-c(1:5)]
View(date_proiect1)
# Voi imparti setul de date in 75% antrenare si 25% testare
split <- sample.split(date_proiect1$Tuition_USD, SplitRatio = 0.75)
set_antrenare <- subset(date_proiect1,split==TRUE)
set_testare <- subset(date_proiect1,split==FALSE)
View(set_testare)
# Voi standardiza datele folosind metoda max-min:
maxim <- apply(date_proiect1,2,max)
minim <- apply(date_proiect1,2,min)
data_std <- as.data.frame(scale(date_proiect1,center = minim,scale=maxim-minim))
View(data_std)
data_std$split <- split
# Definirea seturilor de antrenare si testare ale retelei
antrenare_retea <- subset(data_std, split==TRUE)
testarere_retea <- subset(data_std, split==FALSE)
# Verificăm numele variabilelor
names(antrenare_retea)
# Construirea retelei neuronale:
retea_neuronala1 <- neuralnet(Total_Cost ~ Duration_Years+Living_Cost_Index,
                              data = antrenare_retea, hidden = 5, linear.output = TRUE)
# Reprezentare grafica
plot(retea_neuronala1)
# Previzionarea valorilor taxei de scolarizare
predictie <- compute(retea_neuronala1,testarere_retea[,c(1,3)])
# Vizualizarea rezultatelor standardizate
predictie$net.result
# Aducem valorile la dimensiunile reale pentru a le putea compara cu valorile reale
predictie <- (predictie$net.result*(max(date_proiect$Total_Cost)-min(date_proiect$Total_Cost)))+min(date_proiect$Total_Cost)
# Vizualizam predictiile
head(predictie)
# Voi reprezenta grafic val reale si cele previzionate:
plot(set_testare$Total_Cost, predictie, col="green", pch=14,ylab="Val. Previzionate", xlab="Valori reale")
abline(0,1)
# Determin eroarea RMSE(Root Mean Squared Error) pt setul de testare conform formulei:
eroare <- (sum(set_testare$Total_Cost-predictie)^2/nrow(set_testare))^0.5
eroare #eroarea este de 57840.79
```

```
df_rn <- data.frame(set_testare$Total_Cost,predictie) #Valori reale vs previzionate

View(df_rn)
```