Iordan Maria-Alexandra
Grupa 1080-A

TEMA 4

ALGORITMI DE CLASIFICARE – INVATARE SUPERVIZATA

***Primele 5 cerinte au fost preluate din temele anterioare.**

Analiza a fost realizata pe un set de date ce contine informatii despre 200 de companii listate la bursa, obtinute din sursa Yahoo Finance, sectiunea Financials (Income Statement & Cash Flow). Datele se refera la TTM (Trailing Twelve Months), oferind o privire de ansamblu asupra performantei financiare a acestor companii pe parcursul ultimelor 12 luni disponibile.

In ceea ce priveste structura setului de date, acesta nu prezinta valori lipsa, iar variabilele au fost notate de la X1 la X10, cu denumiri sugestive in fisierul Excel. In ceea ce priveste outlierii/valorile extreme, acestia au fost eliminati din setul de date, deoarece prezenta acestora afecta negativ analiza, distorsionand structura datelor si reducand relevanta rezultatelor obtinute. Dupa eliminarea valorilor extreme, in setul de date am ramas cu 81 de observatii.

Setul de date este compus din urmatoarele variabile, notate de la X1 la X10 si exprimate in mii USD ($):

- **Total Revenue/Venit total** – ce reprezinta suma totală a veniturilor generate de o companie din vânzarea bunurilor sau serviciilor sale, fără a ține cont de costurile asociate.

- **Gross Profit/Profitul brut** – este calculat prin scăderea costurilor directe asociate producției bunurilor sau serviciilor din venitul total si reflectă eficiența în generarea profitului din activitatea principală.

- **Operating Income/Venitul operational** – este venitul generat din activitățile de bază ale companiei, excluzând veniturile și cheltuielile non-operaționale; acesta indica profitabilitatea operațiunilor zilnice ale companiei.

- **Net Income/ Venitul net** - reprezintă profitul total obținut de companie după scăderea tuturor cheltuielilor, inclusiv taxe și cheltuieli non-operaționale; este un indicator esențial al sănătății financiare a unei companii.

- **Earnings before interest and taxes (EBIT)** - acest indicator arată profitul companiei înainte de deducerea cheltuielilor cu dobânzile și impozitele, fiind util pentru compararea performanței între companii, indiferent de structura lor de capital.

- **Earnings per share (EPS)** - reprezintă venitul pe acțiune și oferă o măsură a profitabilității unei companii pe acțiune.

- **Operating Cash Flow/Fluxul de numerar operational** - acesta măsoară capacitatea companiei de a genera numerar din activitățile sale operaționale.

- **Investing Cash Flow/Fluxul de numerar din activitati de investitii** - reflectă numerarul cheltuit sau generat din activitățile de investiții ale companiei, inclusiv achiziții de active sau vânzări de active.

- **Financing Cash Flow/Fluxul de numerar din activitati de finantare** - acesta arată fluxurile de numerar rezultate din activitățile de finanțare, cum ar fi emisiunea de acțiuni, împrumuturile și rambursările de datorii.

- **Free Cash Flow/Fluxul de numerar liber** - este un indicator important al capacității unei companii de a genera numerar după ce a acoperit toate cheltuielile necesare.

In ceea ce priveste **observatiile**, fiecare linie din setul de date corespunde unei companii listate la bursă, oferind o imagine de ansamblu asupra performanței financiare a acestora prin intermediul variabilelor de mai sus. Aceasta permite compararea companiilor în funcție de diferiti indicatori financiari.

Obiectivul general al analizei este de a explora și interpreta relațiile dintre variabilele financiare ale unui set de 200 de companii listate la bursă, cu scopul de a evidenția factorii esențiali care influențează performanța financiară. Analiza datelor permite reducerea dimensiunii setului de informații, sintetizarea indicatorilor-cheie și identificarea tiparelor relevante, contribuind astfel la o înțelegere aprofundată a dinamicii financiare și la sprijinirea deciziilor informate în domeniul investițional și managerial.

Interpretarea indicatorilor statistici

```
> summary(tema)
   Companie              X1                X2                X3                X4                X5                X6          
 Length:81          Min.   :    4131   Min.   :   3966   Min.   :-636864   Min.   :-762367   Min.   :-727188   Min.   :-4.7200  
 Class :character   1st Qu.: 1275994   1st Qu.: 705928   1st Qu.: -14466   1st Qu.: -83497   1st Qu.: -24339   1st Qu.:-0.4200  
 Mode  :character   Median : 2562440   Median :1107379   Median : 281000   Median : 116261   Median : 213000   Median : 0.5900  
                    Mean   : 4474807   Mean   :1647343   Mean   : 293014   Mean   :  89902   Mean   : 239688   Mean   : 0.9617  
                    3rd Qu.: 5320059   3rd Qu.:2123393   3rd Qu.: 506000   3rd Qu.: 280000   3rd Qu.: 439514   3rd Qu.: 2.4400  
                    Max.   :23813905   Max.   :9646000   Max.   :1550863   Max.   : 837880   Max.   :1362945   Max.   : 8.8600  
       X7                X8                X9                X10         
 Min.   :-586000   Min.   :-1209300   Min.   :-919000   Min.   :-614000  
 1st Qu.: 131885   1st Qu.: -383000   1st Qu.:-335144   1st Qu.:  23889  
 Median : 384670   Median : -147000   Median :-157094   Median : 181000  
 Mean   : 449780   Mean   : -242236   Mean   :-194710   Mean   : 233963  
 3rd Qu.: 712000   3rd Qu.:  -51200   3rd Qu.: -12411   3rd Qu.: 383000  
 Max.   :1796100   Max.   :  314000   Max.   : 414345   Max.   : 823000  
```

Figura 1. Rezultatul comenzii *summary*

```
> describe(tema[-1])
    vars  n       mean         sd     median    trimmed        mad        min         max       range  skew kurtosis        se
X1     1 81 4474806.54 5230485.32 2562440.00 3345078.97 2334521.23     4131.00 23813905.00 23809774.00  2.07     3.92 581165.04
X2     2 81 1647342.80 1646814.18 1107379.00 1354668.31  975246.87     3966.00  9646000.00  9642034.00  2.25     6.56 182979.35
X3     3 81  293013.65  398911.76  281000.00  272393.32  391702.92  -636864.00  1550863.00  2187727.00  0.52     0.38  44323.53
X4     4 81   89902.35  313272.53  116261.00  100658.46  257585.44  -762367.00   837880.00  1600247.00 -0.32     0.27  34808.06
X5     5 81  239687.85  426923.00  213000.00  228556.34  351878.80  -727188.00  1362945.00  2090133.00  0.33     0.18  47435.89
X6     6 81       0.96       2.41       0.59       0.95       1.76       -4.72        8.86       13.58  0.41     1.27      0.27
X7     7 81  449780.25  447403.06  384670.00  410150.88  382605.69  -586000.00  1796100.00  2382100.00  0.84     0.78  49711.45
X8     8 81 -242235.63  282768.47 -147000.00 -199593.78  172682.87 -1209300.00   314000.00  1523300.00 -1.28     1.43  31418.72
X9     9 81 -194709.60  284965.00 -157094.00 -180020.72  236108.50  -919000.00   414345.00  1333345.00 -0.49     0.16  31662.78
X10   10 81  233962.54  298504.74  181000.00  228332.17  249804.76  -614000.00   823000.00  1437000.00  0.13    -0.13  33167.19
```

Figura 2. Rezultatul comenzii *describe*

Iordan Maria-Alexandra
Grupa 1080-A

Analiza variabilelor financiare evidențiază diferențe semnificative între companiile analizate, indicând o piață eterogenă dominată de câțiva actori majori. Variabila X1 - Venitul total prezintă o variație largă între 4.131 USD și 23.813.905 USD, cu o medie de 4.474.807 USD influențată de valori extreme și o mediana de 2.562.440 USD, reflectând o distribuție asimetrică spre dreapta. Similar, X2 - Profitul brut variază între 3.966 USD și 9.646.000 USD, având o medie de 1.647.343 USD și o mediana de 1.107.379 USD, cu o dispersie semnificativă indicată de devierea standard. X3 - Venitul operațional variază între -636.864 USD și 1.550.863 USD, cu o medie de 293.014 USD și o mediană apropiată de 281.000 USD, reflectând o ușoară asimetrie spre dreapta și o distribuție mai plată. În ceea ce privește X4 - Venitul net, valorile se situează între -762.367 USD și 837.880 USD, iar o medie de 89.902 USD și o mediană de 116.261 USD indică o ușoară asimetrie spre stânga. Variabila X5 - EBIT variază moderat, între -727.188 USD și 1.362.945 USD, cu o medie de 239.688 USD, o mediană de 213.000 USD și o ușoară asimetrie spre dreapta. X6 - Venitul pe acțiune (EPS) prezintă o variabilitate mare, între -47.200 USD și 88.600 USD, cu o mediană de 0.59 USD și o distribuție moderat leptocurtică. X7 - Fluxul de numerar operațional variază între -586.000 USD și 1.796.100 USD, cu o medie de 449.780 USD influențată de valori extreme și o mediană de 384.670 USD, indicând o piață diversificată. X8 - Fluxul de numerar din investiții are valori între -1.209.300 USD și 314.000 USD, cu o distribuție asimetrică spre stânga, în timp ce X9 - Fluxul de numerar din finanțare variază între -919.000 USD și 414.345 USD, majoritatea companiilor raportând valori negative, indicând o tendință generală de finanțare negativă. Aceste rezultate subliniază variația mare a performanțelor financiare între companiile analizate, de la pierderi semnificative până la performanțe remarcabile, reflectând o piață cu strategii diverse și diferențe marcante în competitivitate.

Matricea de corelatie si matricea de covarianta

Pentru a observa mai bine rezultatele si a fi mai usor de interpretat, am standardizat datele utilizand functia scale. Observam ca dupa standardizarea datelor, matricea de covarianta este egala cu matricea de corelatie, toate variabilele avand aceeasi deviatie standard (1) si media 0, ceea ce le face comparabile direct intre ele. Asadar, valorile observate reflectă doar relațiile dintre variabile, fără influența unității de măsură, astfel încât covariațele și corelațiile devin mai ușor de interpretat.

Iordan Maria-Alexandra
Grupa 1080-A

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X1	1.0000000	0.9045192	0.5718650	0.2988523	0.4717430	0.12605371	0.6700836	-0.53350678	-0.4915897	0.4187086
X2	0.9045192	1.0000000	0.5752673	0.2847471	0.4123124	0.11844546	0.6509549	-0.50014086	-0.4603218	0.4554152
X3	0.5718650	0.5752673	1.0000000	0.6695824	0.8032117	0.39782999	0.7554865	-0.66274626	-0.5823679	0.5261587
X4	0.2988523	0.2847471	0.6695824	1.0000000	0.9158698	0.74327828	0.5107917	-0.28602568	-0.5455754	0.5513663
X5	0.4717430	0.4123124	0.8032117	0.9158698	1.0000000	0.63149859	0.6425682	-0.45358593	-0.6531158	0.5789148
X6	0.1260537	0.1184455	0.3978300	0.7432783	0.6314986	1.00000000	0.2245358	-0.02663007	-0.3068137	0.3263840
X7	0.6700836	0.6509549	0.7554865	0.5107917	0.6425682	0.22453583	1.0000000	-0.71682955	-0.6125110	0.8096187
X8	-0.5335068	-0.5001409	-0.6627463	-0.2860257	-0.4535859	-0.02663007	-0.7168295	1.00000000	0.2874937	-0.3145959
X9	-0.4915897	-0.4603218	-0.5823679	-0.5455754	-0.6531158	-0.30681372	-0.6125110	0.28749371	1.0000000	-0.5262104
X10	0.4187086	0.4554152	0.5261587	0.5513663	0.5789148	0.32638399	0.8096187	-0.31459593	-0.5262104	1.0000000

Figura 3. Matricea de corelatie

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X1	1.0000000	0.9045192	0.5718650	0.2988523	0.4717430	0.12605371	0.6700836	-0.53350678	-0.4915897	0.4187086
X2	0.9045192	1.0000000	0.5752673	0.2847471	0.4123124	0.11844546	0.6509549	-0.50014086	-0.4603218	0.4554152
X3	0.5718650	0.5752673	1.0000000	0.6695824	0.8032117	0.39782999	0.7554865	-0.66274626	-0.5823679	0.5261587
X4	0.2988523	0.2847471	0.6695824	1.0000000	0.9158698	0.74327828	0.5107917	-0.28602568	-0.5455754	0.5513663
X5	0.4717430	0.4123124	0.8032117	0.9158698	1.0000000	0.63149859	0.6425682	-0.45358593	-0.6531158	0.5789148
X6	0.1260537	0.1184455	0.3978300	0.7432783	0.6314986	1.00000000	0.2245358	-0.02663007	-0.3068137	0.3263840
X7	0.6700836	0.6509549	0.7554865	0.5107917	0.6425682	0.22453583	1.0000000	-0.71682955	-0.6125110	0.8096187
X8	-0.5335068	-0.5001409	-0.6627463	-0.2860257	-0.4535859	-0.02663007	-0.7168295	1.00000000	0.2874937	-0.3145959
X9	-0.4915897	-0.4603218	-0.5823679	-0.5455754	-0.6531158	-0.30681372	-0.6125110	0.28749371	1.0000000	-0.5262104
X10	0.4187086	0.4554152	0.5261587	0.5513663	0.5789148	0.32638399	0.8096187	-0.31459593	-0.5262104	1.0000000

Figura 4. Matricea de covarianta

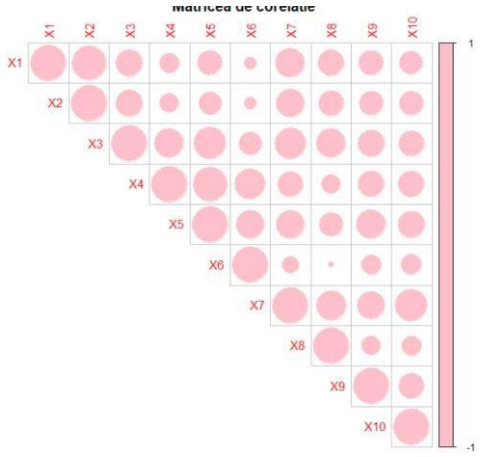


Figura 5. Reprezentarea grafica a matricei de corelatie

Analiza matricei de corelație relevă relații complexe între variabilele financiare, evidențiind legături semnificative între venitul total (X1), profitul brut (X2), fluxul de numerar operațional (X7) și venitul operațional (X3), ceea ce sugerează că performanța financiară generală este strâns legată de creșterea veniturilor și a fluxurilor de numerar. Variabilele precum EBIT (X5) și venitul net (X4) joacă un rol determinant în generarea profitabilității, având corelații puternice cu majoritatea indicatorilor. În schimb, fluxurile de numerar din activități de investiții (X8) și finanțare (X9) prezintă corelații negative cu alte variabile financiare, indicând un impact negativ asupra performanței financiare pe termen scurt, dar un potențial pentru investiții pe termen lung. Matricea de covarianță confirmă aceste relații, sugerând că majoritatea variabilelor financiare cresc împreună, cu excepția fluxurilor de investiții și finanțare, care influențează în sens opus profitabilitatea și veniturile pe termen scurt.

ALGORITMI DE CLASIFICARE – INVATARE SUPERVIZATA

Variabila de clasificare utilizată în analiză a fost generată prin intermediul analizei de tip cluster, realizata la tema anterioara, având ca scop gruparea companiilor pe baza similarităților dintre variabilele financiare. **Obiectivul general al analizei** il reprezinta identificarea unor grupuri distincte de companii care prezintă caracteristici financiare similare, astfel încât să fie posibilă o clasificare simplificată în două clase.

Deși analiza cluster (de la tema 3) a determinat formarea a șase clustere, pentru a facilita interpretarea și utilizarea practică a rezultatelor, datele vor fi grupate în doar două clase utilizând metoda **k-means** cu un număr de clustere predefinit (k=2).

```
> k_means
K-means clustering with 2 clusters of sizes 34, 47

Cluster means:
         X1         X2         X3         X4         X5         X6         X7         X10
1  0.6819965  0.6612618  0.8227916  0.7658124  0.8394279  0.6005639  0.7893532  0.7225224
2 -0.4933591 -0.4783596 -0.5952109 -0.5539919 -0.6072457 -0.4344504 -0.5710214 -0.5226758

Clustering vector:
 RDDT  SPOT   DJT  SOUN  HIMS  IONQ   NGD  PTON  WULF  HOOD   RIG   KGC  GRAB   BTG  RELY  CFLT  PAYC  TTMI  CVNA   NXT   MCW  MBLY
    2     1     2     2     2     2     2     2     2     2     2     1     2     2     2     2     1     2     1     1     2     2
 ETSY  LAUR    IP   DNB   KTB  COMP   SFM DJTWW   BTU  MYRG   VNT  PCOR  DRVN  PCTY   SCI   PEN  CNMD   LAZ   HLI    SW  MMSI  FLNC
    1     2     1     2     1     2     1     2     1     2     1     2     2     2     1     2     2     2     1     1     2     2
  CLX  VSCO  CWEN  TRUP  ALVO   WGS   WEN   LYV  PINS FWONA   WMG    ZG  PLTK   SKX   HAS  WING  BIRK   MHK  GLBE  SHAK     M   HOG
    1     1     1     2     2     2     2     1     1     1     1     2     1     1     2     2     2     1     2     2     1     1
  IGT   FUN   PII   AEO   JWN   UAA  URBN  SHOO   VAC    GT  CAKE    FL   MBC  TRIP   KSS
    1     2     1     1     1     2     1     2     1     1     2     2     2     2     1

Within cluster sum of squares by cluster:
[1] 235.8804 147.6885
 (between_SS / total_SS =  40.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

Figura 6. Rezultatul aplicarii algoritmului K-means cu 2 clustere

Iordan Maria-Alexandra
Grupa 1080-A

Analizand rezultatele, putem observa ca algoritmul K-means a efectuat o clusterizare a observatiilor in 2 clustere, cu cate 34, respectiv 47 de elemente.

Cluster means/Mijloacele pentru fiecare cluster reprezintă mediile variabilelor pentru fiecare cluster în parte. Analizand valorile, putem deduce ca clusterul 1 include companiile cu performante financiare puternice, avand valori ridicate pentru toate variabilele analizate, pe cand clusterul 2 are valori negative ale mediilor, ceea ce inseamna ca acest cluster contine companiile mai slabe, cu performanta financiara scazuta.

In cadrul vectorului de clusterizare sunt atribuite fiecărei observații (companie) un număr de cluster corespunzător. Spre exemplu, companiile SPOT si ETSY sunt atribuite clusterului 1, asadar performeaza bine din punct de vedere financiar, iar companiile RDDT si DJT sunt in clusterul 2, intampinand dificultati pe piata financiara.

Suma pătratică internă a erorii indică un anumit grad de dispersie în interiorul fiecărui cluster, adică cât de mult variază observațiile din interiorul fiecărui cluster față de centrul acestuia. Cu cât valoarea este mai mică, cu atât observațiile din cluster sunt mai apropiate între ele. Putem observa ca ambele clustere prezinta valori ridicate pentru acest indicator, ceea ce este explicabil avand in vedere numarul mare de componente din cadrul gruparilor, existand o diversitate ridicata a conditiilor financiare. Totusi, clusterul 2 este mai compact decat primul, avand o valoare mai scazuta.

Procentul de 40.1% indică faptul că doar 40.1% din variabilitatea totală a datelor este explicată de diferențele dintre clusterele identificate. Acest rezultat sugerează că modelul K-means a realizat o clusterizare suboptimală, ceea ce poate indica fie că algoritmul nu a reușit să separe clar grupurile în funcție de structura datelor, fie că datele au o variație ridicată care nu poate fi explicată adecvat folosind doar două clustere.

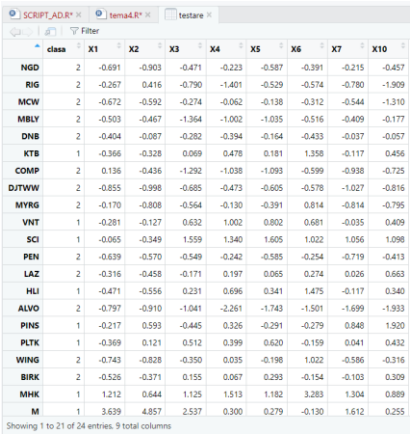
In continuare, am adaugat clasificarea generata anterior la setul de date analizat. In primul rand, clasificările atribuite fiecărei observații sunt extrase și stocate în variabila clasa. Ulterior, această variabilă este combinată cu datele standardizate, utilizând funcția cbind, iar valorile numerice ale setului de date sunt rotunjite la trei zecimale. Rezultatul este un nou set de date, care include atât clasificarea, cât și valorile standardizate ale variabilelor. În final, acest set de date este transformat într-un obiect de tip data frame, facilitând astfel analiza ulterioara.

Iordan Maria-Alexandra
Grupa 1080-A

	clasa	X1	X2	X3	X4	X5	X6	X7	X10
RDDT	2	-0.668	-0.473	-2.331	-2.119	-2.053	-1.842	-0.926	-0.708
SPOT	1	1.791	1.207	-1.040	-0.638	-0.742	-0.649	0.854	1.973
DJT	2	-0.855	-0.998	-0.775	-0.473	-0.605	-0.578	-1.027	-0.816
SOUN	2	-0.845	-0.977	-0.919	-0.639	-0.756	-0.558	-1.172	-1.035
HIMS	2	-0.652	-0.468	-0.705	-0.229	-0.533	-0.366	-0.722	-0.484
IONQ	2	-0.850	-0.990	-1.232	-0.810	-1.026	-0.732	-1.220	-1.200
NGD	2	-0.691	-0.903	-0.471	-0.223	-0.587	-0.391	-0.215	-0.457
PTON	2	-0.339	-0.268	-1.751	-2.049	-1.591	-1.277	-1.153	-1.071
WULF	2	-0.833	-0.955	-0.786	-0.451	-0.618	-0.475	-0.888	-1.120
HOOD	2	-0.428	-0.105	0.030	0.639	0.153	-0.271	-2.315	-2.841
RIG	2	-0.267	0.416	-0.790	-1.401	-0.529	-0.574	-0.780	-1.909
KGC	1	0.008	-0.237	1.457	1.287	1.516	-0.233	3.009	1.510
GRAB	2	-0.361	-0.362	-1.181	-0.967	-0.819	-0.570	0.233	0.781
BTG	2	-0.484	-0.531	0.941	-0.736	-0.179	-0.450	1.428	-0.388
RELY	2	-0.649	-0.619	-0.994	-0.618	-0.775	-0.628	-1.022	-0.854
CFLT	2	-0.690	-0.620	-1.769	-1.467	-1.528	-0.890	-1.037	-0.902
PAYC	1	-0.516	-0.106	0.731	1.220	0.856	3.042	0.143	0.245
TTMI	2	-0.413	-0.727	-0.386	-0.232	-0.343	-0.333	-0.576	-0.705
CVNA	1	1.375	0.330	0.296	1.992	2.552	0.914	0.816	1.689

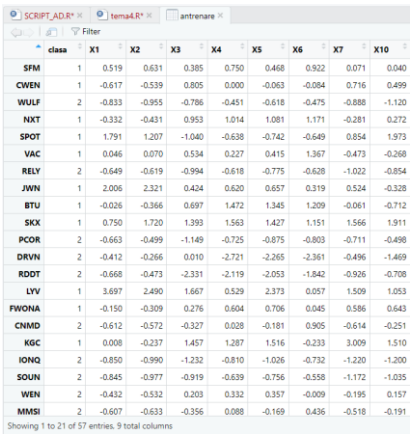
Figura 7. Noul set de date utilizat in analiza

Apoi, am impartit noul set de date in doua subseturi: unul pentru antrenare (70% din date) si unul pentru testare (30% din date). In primul rand, am calculate numărul de observații care vor face parte din setul de antrenare, iar apoi se selectează aleator observațiile corespunzătoare folosind funcția sample.



	clasa	X1	X2	X3	X4	X5	X6	X7	X10
NGD	2	-0.691	-0.903	-0.471	-0.223	-0.587	-0.391	-0.215	-0.457
RIG	2	-0.267	0.416	-0.790	-1.401	-0.529	-0.574	-0.780	-1.909
MCW	2	-0.672	-0.592	-0.274	-0.062	-0.138	-0.312	-0.544	-1.310
MBLY	2	-0.503	-0.467	-1.364	-1.002	-1.035	-0.516	-0.409	-0.177
DNB	2	-0.404	-0.087	-0.282	-0.394	-0.164	-0.433	-0.037	-0.057
KTB	1	-0.366	-0.328	0.069	0.478	0.181	1.358	-0.117	0.456
COMP	2	0.136	-0.436	-1.292	-1.038	-1.093	-0.599	-0.938	-0.725
DJTWW	2	-0.855	-0.998	-0.685	-0.473	-0.605	-0.578	-1.027	-0.816
MYRG	2	-0.170	-0.808	-0.564	-0.130	-0.391	0.814	-0.814	-0.795
VNT	1	-0.281	-0.127	0.632	1.002	0.802	0.681	-0.035	0.409
SCI	1	-0.065	-0.349	1.559	1.340	1.605	1.022	1.056	1.098
PEN	2	-0.639	-0.570	-0.549	-0.242	-0.585	-0.254	-0.719	-0.413
LAZ	2	-0.316	-0.458	-0.171	0.197	0.065	0.274	0.026	0.663
HLI	1	-0.471	-0.556	0.231	0.696	0.341	1.475	-0.117	0.340
ALVO	2	-0.797	-0.910	-1.041	-2.261	-1.743	-1.501	-1.699	-1.933
PINS	1	-0.217	0.593	-0.445	0.326	-0.291	-0.279	0.848	1.920
PLTK	1	-0.369	0.121	0.512	0.399	0.620	-0.159	0.041	0.432
WING	2	-0.743	-0.828	-0.350	0.035	-0.198	1.022	-0.586	-0.316
BIRK	2	-0.526	-0.371	0.155	0.067	0.293	-0.154	-0.103	0.309
MHK	1	1.212	0.644	1.125	1.513	1.182	3.283	1.304	0.889
M	1	3.639	4.857	2.537	0.300	0.279	-0.130	1.612	0.255

Figura 8. Setul de testare



	clasa	X1	X2	X3	X4	X5	X6	X7	X10
SFM	1	0.519	0.631	0.385	0.750	0.468	0.922	0.071	0.040
CWEN	1	-0.617	-0.539	0.805	0.000	-0.063	-0.084	0.716	0.499
WULF	2	-0.833	-0.955	-0.786	-0.451	-0.618	-0.475	-0.888	-1.120
NXT	1	-0.332	-0.431	0.953	1.014	1.081	1.171	-0.281	0.272
SPOT	1	1.791	1.207	-1.040	-0.638	-0.742	-0.649	0.854	1.973
VAC	1	0.046	0.070	0.534	0.227	0.415	1.367	-0.473	-0.268
RELY	2	-0.649	-0.619	-0.994	-0.618	-0.775	-0.628	-1.022	-0.854
JWN	1	2.006	2.321	0.424	0.620	0.657	0.319	0.524	-0.328
BTU	1	-0.026	-0.366	0.697	1.472	1.345	1.209	-0.061	-0.712
SKX	1	0.750	1.720	1.393	1.563	1.427	1.151	1.566	1.911
PCOR	2	-0.663	-0.499	-1.149	-0.725	-0.875	-0.803	-0.711	-0.498
DRVN	2	-0.412	-0.266	0.010	-2.721	-2.265	-2.361	-0.496	-1.469
RDDT	2	-0.668	-0.473	-2.331	-2.119	-2.053	-1.842	-0.926	-0.708
LYV	1	3.697	2.490	1.667	0.529	2.373	0.057	1.509	1.053
FWONA	1	-0.150	-0.309	0.276	0.604	0.706	0.045	0.586	0.643
CNMD	2	-0.612	-0.572	-0.327	0.028	-0.181	0.905	-0.614	-0.251
KGC	1	0.008	-0.237	1.457	1.287	1.516	-0.233	3.009	1.510
IONQ	2	-0.850	-0.990	-1.232	-0.810	-1.026	-0.732	-1.220	-1.200
SOUN	2	-0.845	-0.977	-0.919	-0.639	-0.756	-0.558	-1.172	-1.035
WEN	2	-0.432	-0.532	0.203	0.332	0.357	-0.009	-0.195	0.157
MMSI	2	-0.607	-0.633	-0.356	0.088	-0.169	0.436	-0.518	-0.191

Figura 9. Setul de antrenare

Ulterior, setul de antrenare este transformat intr-un data frame, iar valorile variabilei clasa au fost redenumite pentru a indica apartenenta companiilor la “clasa 1” sau “clasa 2”. Acest data frame va fi utilizat in continuare in cadrul analizei.

Iordan Maria-Alexandra
Grupa 1080-A

SCRIPT_AD.R* × tema4.R* × df_a ×

Filter

	clasa	X1	X2	X3	X4	X5	X6	X7	X10
SFM	clasa1	0.519	0.631	0.385	0.750	0.468	0.922	0.071	0.040
CWEN	clasa1	-0.617	-0.539	0.805	0.000	-0.063	-0.084	0.716	0.499
WULF	clasa2	-0.833	-0.955	-0.786	-0.451	-0.618	-0.475	-0.888	-1.120
NXT	clasa1	-0.332	-0.431	0.953	1.014	1.081	1.171	-0.281	0.272
SPOT	clasa1	1.791	1.207	-1.040	-0.638	-0.742	-0.649	0.854	1.973
VAC	clasa1	0.046	0.070	0.534	0.227	0.415	1.367	-0.473	-0.268
RELY	clasa2	-0.649	-0.619	-0.994	-0.618	-0.775	-0.628	-1.022	-0.854
JWN	clasa1	2.006	2.321	0.424	0.620	0.657	0.319	0.524	-0.328
BTU	clasa1	-0.026	-0.366	0.697	1.472	1.345	1.209	-0.061	-0.712
SKX	clasa1	0.750	1.720	1.393	1.563	1.427	1.151	1.566	1.911
PCOR	clasa2	-0.663	-0.499	-1.149	-0.725	-0.875	-0.803	-0.711	-0.498
DRVN	clasa2	-0.412	-0.266	0.010	-2.721	-2.265	-2.361	-0.496	-1.469
RDDT	clasa2	-0.668	-0.473	-2.331	-2.119	-2.053	-1.842	-0.926	-0.708
LYV	clasa1	3.697	2.490	1.667	0.529	2.373	0.057	1.509	1.053
FWONA	clasa1	-0.150	-0.309	0.276	0.604	0.706	0.045	0.586	0.643
CNMD	clasa2	-0.612	-0.572	-0.327	0.028	-0.181	0.905	-0.614	-0.251
KGC	clasa1	0.008	-0.237	1.457	1.287	1.516	-0.233	3.009	1.510
IONQ	clasa2	-0.850	-0.990	-1.232	-0.810	-1.026	-0.732	-1.220	-1.200
SOUN	clasa2	-0.845	-0.977	-0.919	-0.639	-0.756	-0.558	-1.172	-1.035
WEN	clasa2	-0.432	-0.532	0.203	0.332	0.357	-0.009	-0.195	0.157
MMSI	clasa2	-0.607	-0.633	-0.356	0.088	-0.169	0.436	-0.518	-0.191

Showing 1 to 21 of 57 entries, 9 total columns

Figura 10. Data frame-ul cu datele de testare

CLASIFICATORUL NAIV – BAYESIAN

In urma aplicarii clasificatorului NAIV BAYESIAN, am obtinut urmatoarele:

SCRIPT_AD.R × tema4.R* × model ×

Show Attributes

Name	Type	Value
model	list [5] (S3: naiveBayes)	List of length 5
apriori	integer [2] (S3: table)	23 34
tables	list [8]	List of length 8
levels	character [2]	'clasa1' 'clasa2'
isnumeric	logical [8]	TRUE TRUE TRUE TRUE TRUE TRUE ...
call	language	naiveBayes.default(x = X, y = Y, laplace = laplace)

Figura 11. Vizualizarea modelului

```
> summary(model)
          Length Class  Mode
apriori   2      table  numeric
tables    8      -none- list
levels    2      -none- character
isnumeric 8      -none- logical
call      4      -none- call
```

Figura 12. Rezultatul comenzii summary

Rezultatul obtinut indica detaliile structurii modelului antrenat. Variabila *apriori* este un tabel numeric de lungime 2, care reprezintă probabilitățile a priori pentru cele două clase identificate. Variabila *tables* conține un obiect de tip listă, cu 8 elemente, fiecare referindu-se la distribuțiile condiționate ale caracteristicilor pentru fiecare dintre cele două clase. Variabila *levels* indică cele două niveluri/clase ale variabilei de clasificare, iar *isnumeric* este un vector logic care indică dacă variabilele de intrare sunt numerice. În cele din urmă, *call* oferă informații despre apelul funcției care a fost folosit pentru a construi modelul.

Probabilitatile apriori si conditionate

```
> model$apriori
Y
clasa1 clasa2
    23     34
```

Figura 13. Probabilitatile apriori

Rezultatul arata distribuția probabilităților a priori pentru cele două clase. Valoarile 23, respectiv 34, indică numărul de exemple din setul de date care au fost atribuite fiecărei clase în procesul de antrenare. Aceste valori nu reprezintă probabilitățile propriu-zise, ci mai degrabă frecvențele fiecărei clase în setul de date. Astfel, se poate observa că, în setul de antrenare, există mai multe companii cu performanțe financiare slabe decât companii cu performanțe bune.

```
> model$tables
$X1
        X1
Y              [,1]      [,2]
  clasa1  0.7342609 1.1754094
  clasa2 -0.4925000 0.3113722

$X2
        X2
Y              [,1]      [,2]
  clasa1  0.6180000 0.8974854
  clasa2 -0.4551471 0.4201749

$X3
        X3
Y              [,1]      [,2]
  clasa1  0.8818696 0.8497586
  clasa2 -0.5970294 0.6776710

$X4
        X4
Y              [,1]      [,2]
  clasa1  0.8043043 0.8483075
  clasa2 -0.5621176 0.8150427

$X5
        X5
Y              [,1]      [,2]
  clasa1  0.9578261 0.8681459
  clasa2 -0.6420588 0.6676610

$X6
        X6
Y              [,1]      [,2]
  clasa1  0.4915652 0.9164059
  clasa2 -0.5062647 0.8412958

$X7
        X7
Y              [,1]      [,2]
  clasa1  0.8775217 0.9690456
  clasa2 -0.5587353 0.6635384

$X10
        X10
Y              [,1]      [,2]
  clasa1  0.7349565 0.9549173
  clasa2 -0.4891176 0.7469841
```

Figura 14. Probabilitatile conditionate

În acest rezultat, pentru fiecare variabilă sunt prezentate două valori asociate fiecărei clase: media (în prima coloană) și abaterea standard (în a doua coloană). Aceste tabele indică distribuțiile fiecărei variabile explicative condiționate de clasele "clasa1" și "clasa2". Acestea ne permit să observăm diferențele dintre cele două clase în ceea ce privește fiecare variabilă financiară. De exemplu, pentru X1, media din clasa 1 este mult mai mare decât în clasa 2, indicând faptul că companiile din clasa 1 au un venit total mult mai ridicat decât cele din clasa 2, care înregistrează chiar și valori negative. În ceea ce privește abaterea standard, clasa 1 prezintă o valoare mai mare pentru X1, ceea ce sugerează o variabilitate mai mare a valorilor în comparație cu clasa 2. Variabilele cu diferențe semnificative între medii și abateri standard între clase sunt mai relevante pentru clasificatorul Naiv Bayesian, deoarece ele furnizează informații care ajută la distingerea celor două clase. In cazul nostru, toate variabilele prezinta diferente semnificative intre cele doua clase, clasele continand companii cu caracteristici extrem de distincte in ceea ce priveste performanta pe piata financiara. Astfel, se poate observa clar ca prima clasa indica companiile performante, iar cea de-a doua companiile mai slabe din punct de vedere financiar.

Realizarea de predictii pe setul de testare. Matricea de confuzie.

In aceasta parte, am realizat predictii pe setul de testare. Utilizand functia *predict,* am prezis probabilitatile aposteriorice de apartenenta la clasa (*type=”class”),* respectiv probabilitatile aposteriorice de apartenenta la grupe *(type=”raw”).*

```
> pred_test <- predict(model,testare[,-1],type="class")
> pred_test
 [1] clasa2 clasa2 clasa2 clasa2 clasa2 clasa2 clasa2 clasa2 clasa2 clasa1 clasa1 clasa2 clasa2 clasa2 clasa2 clasa1 clasa1 clasa2
[19] clasa2 clasa1 clasa1 clasa1 clasa1 clasa1
Levels: clasa1 clasa2
```

Figura 15. Predictia probabilitatilor aposteriorice de apartenenta la clasa

```
> pred_test2 <- predict(model,testare[,-1],type="raw")
> pred_test2
            clasa1       clasa2
 [1,] 3.272749e-05 9.999673e-01
 [2,] 6.165084e-06 9.999938e-01
 [3,] 5.011123e-05 9.999499e-01
 [4,] 6.382698e-07 9.999994e-01
 [5,] 6.936294e-04 9.993064e-01
 [6,] 2.697302e-01 7.302698e-01
 [7,] 1.504971e-06 9.999985e-01
 [8,] 2.871482e-06 9.999971e-01
 [9,] 1.344279e-04 9.998656e-01
[10,] 9.659995e-01 3.400051e-02
[11,] 9.999998e-01 1.503396e-07
[12,] 1.462226e-05 9.999854e-01
[13,] 3.095024e-02 9.690498e-01
[14,] 4.560240e-01 5.439760e-01
[15,] 8.465531e-10 1.000000e+00
[16,] 9.846772e-01 1.532280e-02
[17,] 6.949749e-01 3.050251e-01
[18,] 8.117686e-04 9.991882e-01
[19,] 1.833172e-02 9.816683e-01
[20,] 1.000000e+00 4.143728e-16
[21,] 1.000000e+00 1.883925e-72
[22,] 9.990698e-01 9.301572e-04
[23,] 9.641214e-01 3.587864e-02
[24,] 1.000000e+00 4.476170e-39
>
```

Figura 16. Predictia probabilitatilor aposteriorice de apartenenta la grupe

Apoi, am generat matricea de confuzie, utilizand functia *table(),* pentru a compara predictiile cu valorile reale din setul de testare pentru a evalua performantele modelului.

```
> table(pred_test,testare[,1],dnn=c("Prediction","Actual"))
          Actual
Prediction  1  2
    clasa1  9  0
    clasa2  2 13
```

Figura 17. Matricea de confuzie

Rândurile reprezintă predicțiile modelului, iar coloanele reprezintă valorile reale. Astfel, din matricea de confuzie reies urmatoarele informatii:

- Modelul a prezis corect 9 observatii ca apartinand clasei 1 si nicio observatie nu a fost gresit clasificata ca apartinand clasei 2.
- Modelul a prezis gresit 2 observatii ca apartinand clasei 1, dar a prezis corect 13 observatii ca apartinand clasei 2.

Aceste rezultate sugereaza că modelul are o performanță bună în a clasifica observațiile în clasa 2, dar prezintă o ușoară eroare în clasificarea celor din clasa 1, avand o anumita confuzie intre cele două clase.

```
> acuratete <- sum(diag(conf)) / sum(conf)
> acuratete
[1] 0.9166667
```

Figura 18. Acuratetea modelului NAIV BAYES

In ceea ce priveste acuratetea sau gradul de clasificare corecta, aceasta se calculează prin raportul dintre numărul de predicții corecte/suma pe diagonala principala (9+13) și suma totala (9+0+2+13), fiind egala cu 91.67%, ceea ce inseamna ca modelul a realizat o clasificare corecta in 91.67% din cazuri.

Metoda clasificatorului KNN

Pentru metoda clasificatorului KNN (K-Nearest Neighbors), am folosit aceeasi distributie ca in cazul anterior, setul de date fiind impartit in date de testare, respective date de antrenament. Apoi, am convertit variabila de clasă într-un factor pentru a permite procesarea corectă.

Am definit o validare încrucișată repetată cu 3 repetiții și 10 fold-uri pentru a evalua performanța modelului, utilizând funcția trainControl.

Am testat modelul pentru mai multe valori ale parametrului k (3, 7 si 10), iar pentru fiecare valoare am antrenat modelul utilizand o bucla for. După antrenare, am folosit setul de testare pentru a face predicții și am construit matricea de confuzie pentru fiecare valoare k. Acuratețea modelului a fost calculată ca raportul dintre numărul de predicții corecte și totalul

predicțiilor, iar rezultatele au fost stocate într-un vector pentru a compara performanța pentru diferite valori ale lui k.

```
> ac_val
[1] 0.8333333 0.8750000 0.8333333
```

Figura 19. Acuratetea celor 3 modele antrenate

Observăm că valorile acurateței sunt foarte apropiate, primul și al treilea model având rezultate identice, în timp ce al doilea model prezintă o valoare ușor mai mare. Prin urmare, vom alege modelul cu k = 7, deoarece acesta are cea mai mare valoare a acurateței și, implicit, a realizat clasificarea cea mai precisă.

Am antrenat modelul cu k = 7, obtinand urmatorul output:

```
k-Nearest Neighbors

57 samples
 8 predictor
 2 classes: 'X1', 'X2'

Pre-processing: centered (8), scaled (8)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 51, 50, 51, 50, 52, 51, ...
Resampling results across tuning parameters:

  k   ROC        Sens       Spec
   5  0.9916667  0.8555556  1
   7  0.9972222  0.8000000  1
   9  1.0000000  0.7500000  1
  11  0.9967593  0.7333333  1
  13  0.9972222  0.7277778  1
  15  0.9953704  0.7388889  1
  17  0.9916667  0.7333333  1

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

Figura 20. Modelul KNN cu k = 7

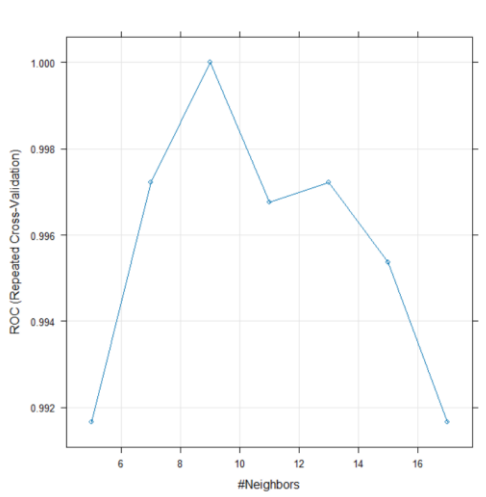


Figura 21. Vizualizarea modelului

Modelul a fost antrenat pe un set de 57 de observatii, 8 predictori si 2 clase (‘X1’ si ‘X2’). Rezultatele arata ca, pe masura ce k creste, valoarea ROC rămâne ridicată, însă sensibilitatea (Sens) tinde să scadă. Specificitatea (Spec) a fost constantă și perfectă (1) pentru toate valorile lui k. Modelul optim a fost selectat pe baza celei mai mari valori ROC, iar k = 9 a fost considerată valoarea finală optimă. Acest model oferă o bună separare între clase, dar cu o ușoară scădere a sensibilității comparativ cu alte valori mai mici ale lui k.

De asemenea, din grafic se poate observa ca numarul optim de cei mai apropiati vecini este de 9.

Realizarea de predictii pe setul de testare. Matricea de confuzie.

In continuare, am realizat predictii pe setul de testare. Utilizand functia *predict,* am prezis probabilitatile de apartenenta la clasa, pentru a observa clasele prezise pentru fiecare observatie din setul de testare, cat si probabilitatile de apartenenta la fiecare clasa.

```
> pred_clase
 [1] X2 X2 X2 X2 X2 X2 X2 X2 X2 X1 X1 X2 X2 X2 X2 X1 X2 X2 X2 X1 X1 X1 X1 X1
Levels: X1 X2
```

Figura 22. Predictia apartenentei la clasa

```
> head(pred_prob)
         X1        X2
1 0.0000000 1.0000000
2 0.0000000 1.0000000
3 0.0000000 1.0000000
4 0.0000000 1.0000000
5 0.0000000 1.0000000
6 0.3333333 0.6666667
```

Figura 23. Predictia probabilistica

Rezultatul afisat in figura 23 reprezinta probabilitatile probabilistice pentru primele 6 observatii din setul de testare. Fiecare rând corespunde unei observații, iar fiecare coloană indică probabilitatea ca observația respectivă să aparțină fiecărei clase. Putem observa ca primele 5 observatii au probabilitatea 0 pentru clasa 1 si probabilitatea 1 pentru clasa 2, ceea ce inseamna ca modelul este foarte sigur ca aceste observatii apartin clasei 2. Insa, pe de alta parte, in ceea ce priveste observatia 6, aceasta are o probabilitate de 33.33% sa faca parte din clasa 1 si 66.67% sa faca parte din clasa 2, asadar putem spune ca modelul este mai incert in privinta clasificarii acestei observatii, dar considera totusi ca apartine clasei 2 (conform figurii 22), avand o probabilitate mai mare pentru aceasta clasa.

Apoi, am generat matricea de confuzie, utilizand functia *table(),* pentru a compara predictiile cu valorile reale din setul de testare pentru a evalua performantele modelului.

```
            Actual
Prediction  1  2
        X1  8  0
        X2  3 13
```

Figura 24. Matricea de confuzie

Din matricea de confuzie reies urmatoarele informatii:

- Modelul a prezis corect 8 observatii ca apartinand clasei 1 si nicio observatie nu a fost gresit clasificata ca apartinand clasei 2.
- Modelul a prezis gresit 3 observatii ca apartinand clasei 1, dar a prezis corect 13 observatii ca apartinand clasei 2.

Aceste rezultate sugereaza că modelul are o performanță bună în a clasifica observațiile în clasa 2, dar prezintă o ușoară eroare în clasificarea celor din clasa 1.

```
> print(acuratete)
[1] 0.875
```

Figura 25. Acuratetea modelului KNN

In ceea ce priveste, aceasta se calculează prin raportul dintre numărul de predicții corecte/suma pe diagonala principala (8+13) și suma totala (8+0+3+13), fiind egala cu 87.5%, ceea ce inseamna ca modelul a realizat o clasificare corecta in 85.7% din cazuri.

Pentru a evalua mai în profunzime modelul realizat, vom utiliza curba ROC și valoarea AUC (area under the curve), care oferă o măsură a capacității modelului de a diferenția între clase. Mai intai, am generat un obiect de predictii pe baza probabilitatilor estimate pentru clasa pozitivă și a etichetelor reale din setul de testare, iar ulterior am calculat performanta modelului, inclusiv valoarea AUC, ce reprezinta probabilitatea ca modelul să clasifice corect o pereche aleatorie formată dintr-un exemplu pozitiv și unul negativ. Aceasta este egala cu 1, ceea ce indică o performanță ridicată a modelului, separand perfect cazurile pozitive de cele negative.

```
> pred_val
A prediction instance
  with 24 data points
```

Figura 26. Obiectul de predictii

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

Figura 27. Valoarea AUC

De asemenea, am reprezentat grafic curba ROC, care ilustreaza rata de adevărate pozitive (TPR) versus rata de false pozitive (FPR). Vizualizand graficul, putem concluziona că modelul realizează o clasificare perfectă, reușind să distingă corect între clasele pozitive și cele negative.

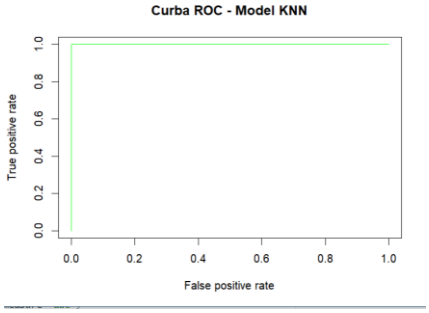


Figura 28. Curba ROC

In final, avand in vedere valoarea acuratetii de 87.5% si valoarea AUC de 1, putem spune despre modelul KNN ca este capabil sa faca distincție între clase într-un mod foarte clar, dar poate avea un prag de decizie care face ca majoritatea predicțiilor pentru clasele negative să fie corecte, dar să rateze un număr semnificativ de exemple pozitive, scăzând astfel acuratețea.

Metoda Arbore de decizie/Random Forest

Metoda arborelui de decizie este o tehnică de învățare automată folosită pentru clasificare și regresie, care construiește un model bazat pe structura unui arbore. Acest arbore este compus din noduri de decizie și frunze, fiecare reprezentând o condiție de împărțire a datelor sau o clasă finală. În acest caz, am utilizat librăria rpart pentru a crea un arbore de decizie, iar analiza se bazează pe setul de date împărțit în două: un set de antrenare și unul de testare.

In primul rand, am creat modelul si am utilizat functia summary pentru a obține o descriere detaliată a arborelui, incluzând nodurile de decizie și condițiile de împărțire.

```
> summary(arbore_decizie)
Call:
rpart(formula = clasa ~ ., data = df_a, method = "class")
  n= 57

          CP nsplit rel error    xerror      xstd
1 0.78260870      0 1.0000000 1.0000000 0.1610416
2 0.04347826      1 0.2173913 0.5652174 0.1377313
3 0.01000000      2 0.1739130 0.5652174 0.1377313

Variable importance
 X5  X3  X4  X7  X1  X6 X10  X2
 21  18  17  14  14   9   3   3

Node number 1: 57 observations,    complexity param=0.7826087
  predicted class=2  expected loss=0.4035088  P(node) =1
    class counts:    23    34
   probabilities: 0.404 0.596
  left son=2 (18 obs) right son=3 (39 obs)
  Primary splits:
      X5 < 0.3725  to the right, improve=18.72065, (0 missing)
      X3 < 0.226   to the right, improve=18.31652, (0 missing)
      X1 < -0.3355 to the right, improve=16.96860, (0 missing)
      X7 < -0.1025 to the right, improve=16.96860, (0 missing)
      X4 < 0.378   to the right, improve=16.85965, (0 missing)
  Surrogate splits:
      X4 < 0.378   to the right, agree=0.947, adj=0.833, (0 split)
      X3 < 0.226   to the right, agree=0.930, adj=0.778, (0 split)
      X1 < -0.168  to the right, agree=0.860, adj=0.556, (0 split)
      X6 < 0.9095  to the right, agree=0.825, adj=0.444, (0 split)
      X7 < -0.0785 to the right, agree=0.825, adj=0.444, (0 split)

Node number 2: 18 observations
  predicted class=1  expected loss=0  P(node) =0.3157895
    class counts:    18     0
   probabilities: 1.000 0.000

Node number 3: 39 observations,    complexity param=0.04347826
  predicted class=2  expected loss=0.1282051  P(node) =0.6842105
    class counts:     5    34
   probabilities: 0.128 0.872
  left son=6 (9 obs) right son=7 (30 obs)
  Primary splits:
      X7  < -0.1025 to the right, improve=4.273504, (0 missing)
      X2  < 0.0965  to the right, improve=3.351877, (0 missing)
      X1  < -0.332  to the right, improve=2.782465, (0 missing)
      X5  < -0.116  to the right, improve=1.698468, (0 missing)
      X10 < -0.1585 to the right, improve=1.698468, (0 missing)
  Surrogate splits:
      X10 < 0.314   to the right, agree=0.923, adj=0.667, (0 split)
      X2  < 0.0965  to the right, agree=0.897, adj=0.556, (0 split)
      X1  < -0.332  to the right, agree=0.872, adj=0.444, (0 split)
      X3  < 0.3385  to the right, agree=0.872, adj=0.444, (0 split)

Node number 6: 9 observations
  predicted class=1  expected loss=0.4444444  P(node) =0.1578947
    class counts:     5     4
   probabilities: 0.556 0.444

Node number 7: 30 observations
  predicted class=2  expected loss=0  P(node) =0.5263158
    class counts:     0    30
   probabilities: 0.000 1.000
```

Figura 29. Rezultatul comenzii summary

Rezultatele ofera o analiză detaliată a modelului de arbore de decizie construit. Modelul a fost antrenat pe un set de 57 de observații, iar complexitatea sa este prezentată prin trei valori de parametru CP. La început, cu 0 diviziuni (nsplit = 0), eroarea relativă este 1, indicând faptul că modelul este încă neantrenat. După prima împărțire (nsplit = 1), eroarea relativă scade la 0.217, iar după încă o împărțire (nsplit = 2), aceasta devine 0.173, sugerând o îmbunătățire a performanței modelului. Valorile xerror și xstd indică eroarea și abaterea standard calculată prin validare încrucișată.

Analiza importanței variabilelor arată că variabila X5 are cea mai mare contribuție la împărțirile arborelui (21%), urmată de X3 (18%) și X4 (17%), iar celelalte variabile au contribuții mai mici.

Arborele începe cu nodul rădăcină (Node 1), care cuprinde toate cele 57 de observații, prezicând clasa „2” cu o probabilitate de 59.6%. Din acest nod, arborele se împarte în două: nodul stâng (Node 2) și nodul drept (Node 3). Nodul 2 include 18 observații, prezicând clasa „1” cu o precizie de 100%. Nodul 3, care conține 39 de observații, prezice clasa „2” cu o eroare de 12.8%. Acest nod este împărțit ulterior în două noduri fiice: Node 6, care include 9 observații cu o probabilitate de 55.6% pentru clasa „1”, și Node 7, care conține 30 de observații, prezicând clasa „2” cu o precizie de 100%.

În procesul de împărțire, variabilele X5, X3, X7 și X1 au fost principalele utilizate, iar variabilele surogate, precum X4 și X10, au oferit soluții alternative pentru împărțiri similare. Acest lucru subliniază relația dintre variabilele predictive și clasa țintă în contextul arborelui de decizie.

Apoi, am utilizat functiile printcp si plotcp pentru a afișa și vizualiza complexitatea modelului. Acestea ajută la determinarea celui mai potrivit nivel de complexitate al arborelui, pe baza valorii de penalizare pentru complexitate (cp).

```
> printcp(arbore_decizie)

Classification tree:
rpart(formula = clasa ~ ., data = df_a, method = "class")

Variables actually used in tree construction:
[1] X5 X7

Root node error: 23/57 = 0.40351

n= 57

        CP nsplit rel error  xerror    xstd
1 0.782609      0   1.00000 1.00000 0.16104
2 0.043478      1   0.21739 0.56522 0.13773
3 0.010000      2   0.17391 0.56522 0.13773
```

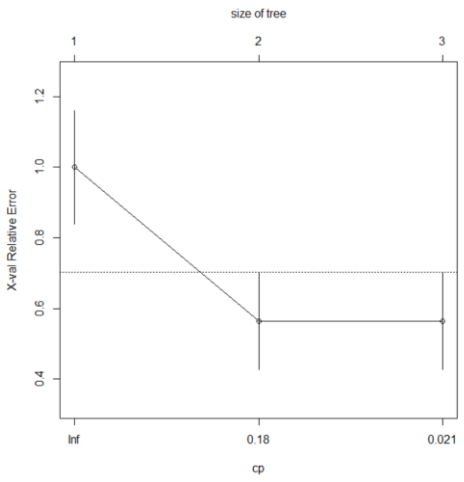


Figura 30. Rezultatul comenzii printcp

Figura 31. Legatura dintre CP si eroarea modelului

Rezultatele prezentate in figura 30 oferă o privire de ansamblu asupra complexității și performanței arborelui de decizie. În construcția arborelui au fost utilizate doar două variabile predictive, X5 și X7, ceea ce indică faptul că acestea sunt cele mai relevante pentru predicția variabilei țintă. Eroarea de la nodul rădăcină (Root node error) este de 23/57 = 0.40351, ceea ce înseamnă că, înainte de orice împărțire, aproximativ 40.35% dintre observații ar fi clasificate greșit. Tabelul evidențiază relația dintre numărul de împărțiri ale arborelui (nsplit), eroarea relativă (rel error), eroarea estimată prin validare încrucișată (xerror) și abaterea standard asociată (xstd). În cazul arborelui inițial, fără ramificare (nsplit = 0), eroarea relativă este

maximă, având valoarea 1.00000. După prima împărțire (nsplit = 1), eroarea relativă scade semnificativ la 0.21739, iar eroarea estimată prin validare încrucișată (xerror) se reduce la 0.56522, ceea ce reflectă o îmbunătățire semnificativă a performanței modelului. Cu toate acestea, după a doua împărțire (nsplit = 2), eroarea relativă continuă să scadă ușor la 0.17391, însă eroarea de validare încrucișată rămâne constantă la 0.56522, indicând că această împărțire suplimentară nu contribuie în mod semnificativ la îmbunătățirea performanței modelului.

Graficul din figura 31 ilustrează legătura dintre parametrul de complexitate (Complexity Parameter - CP) și eroarea modelului, măsurată prin validare încrucișată. Pe baza acestui grafic, vom selecta prima valoare a CP-ului care se află sub linie, indicând punctul optim de tăiere al arborelui, adică acel moment în care adăugarea de complexitate suplimentară nu mai aduce îmbunătățiri semnificative în performanța modelului. Conform analizei graficului, valoarea corespunzătoare este CP = 0.18.

Astfel, voi construi un alt model de arbore de decizie, aplicand un parametru de complexitate (CP) setat la valoarea 0.18 pentru a controla complexitatea arborelui. Dupa antrenarea modelului, am afisat un rezumat detaliat al arborelui de decizie, cat si reprezentarea grafica:

```
> summary(arbore_decizie)
Call:
rpart(formula = clasa ~ ., data = df_a, method = "class", control = rpart.control(cp = 0.18))
  n= 57

         CP nsplit rel error    xerror      xstd
1 0.7826087      0 1.0000000 1.0000000 0.1610416
2 0.1800000      1 0.2173913 0.3913043 0.1196952

Variable importance
X5 X4 X3 X1 X6 X7
25 21 19 14 11 11

Node number 1: 57 observations,    complexity param=0.7826087
  predicted class=2  expected loss=0.4035088  P(node) =1
    class counts:    23    34
   probabilities: 0.404 0.596
  left son=2 (18 obs) right son=3 (39 obs)
  Primary splits:
      X5 < 0.3725  to the right, improve=18.72065, (0 missing)
      X3 < 0.226   to the right, improve=18.31652, (0 missing)
      X1 < -0.3355 to the right, improve=16.96860, (0 missing)
      X7 < -0.1025 to the right, improve=16.96860, (0 missing)
      X4 < 0.378   to the right, improve=16.85965, (0 missing)
  Surrogate splits:
      X4 < 0.378   to the right, agree=0.947, adj=0.833, (0 split)
      X3 < 0.226   to the right, agree=0.930, adj=0.778, (0 split)
      X1 < -0.168  to the right, agree=0.860, adj=0.556, (0 split)
      X6 < 0.9095  to the right, agree=0.825, adj=0.444, (0 split)
      X7 < -0.0785 to the right, agree=0.825, adj=0.444, (0 split)

Node number 2: 18 observations
  predicted class=1  expected loss=0  P(node) =0.3157895
    class counts:    18     0
   probabilities: 1.000 0.000

Node number 3: 39 observations
  predicted class=2  expected loss=0.1282051  P(node) =0.6842105
    class counts:     5    34
   probabilities: 0.128 0.872
```

Figura 32. Rezultatul comenzii summary

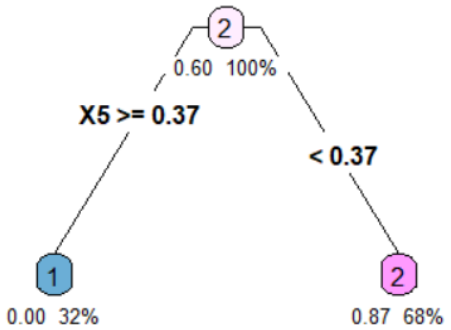


Figura 33. Reprezentarea grafica a arborelui de decizie

Din rezultatele obtinute, observam ca arborele de decizie a fost impartit in doua ramuri. La început, arborele a avut o eroare relativă de 1.0000 și o eroare de validare încrucișată (xerror) de 1.0000. După prima împărțire, eroarea relativă a scăzut semnificativ la 0.2174, iar eroarea de validare a scăzut la 0.3913, ceea ce indică o îmbunătățire semnificativă a modelului. Importanța variabilelor este evidențiată în continuare, cu variabila „X5” având cea mai mare importanță (25), urmată de „X4” (21) și „X3” (19), iar celelalte variabile, precum „X1”, „X6” și „X7”, au valori mai scăzute, dar totuși relevante pentru model.

În ceea ce privește structura arborelui, nodul principal conține 57 de observații și are o probabilitate de 0.596 pentru clasa 2, cu o pierdere estimată de 0.4035. Acest nod se împărțește în două ramuri: nodul 2 (cu o probabilitate de 325), cu 18 observații, unde predicția este 100%

corectă pentru clasa 1 (probabilitatea 1.0000), și nodul 3 (cu o probabilitate de 68%), cu 39 de observații, care prezice clasa 2 cu o probabilitate de 0.872. În nodul 3, distribuția claselor este destul de dezechilibrată, cu 5 observații din clasa 1 și 34 din clasa 2.

Realizarea de predictii pe setul de testare. Matricea de confuzie si acuratetea modelului.

```
> pred_clase_arbore
  NGD   RIG   MCW  MBLY   DNB   KTB  COMP DJTWW  MYRG   VNT   SCI   PEN   LAZ   HLI  ALVO  PINS  PLTK  WING  BIRK
    2     2     2     2     2     2     2     2     2     1     1     2     2     2     2     2     1     2     2
  MHK     M   PII   AEO   KSS
    1     2     2     1     1
Levels: 1 2
```

Figura 34. Predictiile pentru clasele tinta

```
> head(pred_prob_arbore)
             1         2
NGD  0.1282051 0.8717949
RIG  0.1282051 0.8717949
MCW  0.1282051 0.8717949
MBLY 0.1282051 0.8717949
DNB  0.1282051 0.8717949
KTB  0.1282051 0.8717949
```

Figura 35. Predictiile probabilistice

In continuare, am realizat predictii pe setul de testare, utilizand modelul de arbore de decizie antrenat anterior. Mai intai, am generat predicțiile pentru clasele țintă, iar ulterior am obtinut si predicțiile probabilistice, adică probabilitățile corespunzătoare fiecărei clase. Astfel putem observa atat predicțiile exacte ale claselor, cât și incertitudinea asociată acestora, exprimată prin probabilitățile calculate de model. Conform figurii 35, observam ca probabilitatile pentru primele 6 observatii (cel putin) sunt identice, avand probabilitatea de 12.8% sa apartina clasei 1 si probabilitatea de 87.1% sa apartina clasei 2, iar verificand figura 34, observam ca acestea au fost asociate clasei 2 deoarece prezinta o probabilitate mult mai mare.

Astfel, folosind predictiile pentru clasele tinta, am generat matricea de confuzie:

```
> print(matrice_confuzie_arbore)
          Actual
Prediction  1  2
         1  6  0
         2  5 13
```

Figura 36. Matricea de confuzie

În matricea de confuzie, liniile reprezintă predicțiile modelului, iar coloanele reprezintă clasele reale. Matricea de confuzie indica urmatoarele:

- 6 observații au fost corect clasificate în clasa 1, fiind prezise ca aparținând aceleași clase (true positive).
- 5 observații din clasa 2 au fost greșit clasificate în clasa 1 (false negative).

- 13 observații au fost corect clasificate în clasa 2 (true positive).
- 0 observații din clasa 1 au fost greșit clasificate în clasa 2 (false positive).

Astfel, modelul de arbore de decizie a avut o performanță bună în predicțiile pentru clasa 2, cu o rată de corectitudine de 100% în acest caz, dar a avut o rată mai scăzută de corectitudine pentru clasa 1, cu 5 greșeli de clasificare în clasa 2.

```
> print(acuratete_arbore)
[1] 0.7916667
```

Figura 37. Acuratetea modelului

In ceea ce priveste acuratetea modelului, aceasta este egala cu 0.7916, ceea ce înseamnă că aproximativ 79,17% din observațiile din setul de testare au fost corect clasificate. În acest caz, modelul a fost capabil să prezică corect aproape 80% din cazuri, ceea ce sugerează o performanță destul de bună.

Evaluarea modelului, utilizand ROC si AUC

In primul rand, am obtinut probabilitatea de clasă pozitivă (clasa 2) pentru setul de testare, apoi am creat un obiect pentru predictii, folosind probabilitățile prezise și valorile reale ale clasei din setul de date de antrenament. Acesta a fost utilizat pentru a calcula performanța modelului în termeni de true positive rate (tpr) și false positive rate (fpr), care sunt apoi vizualizate sub forma unui grafic cu funcția plot. Graficul reprezinta curba ROC, iar valorile de performanță sunt colorizate pentru o vizualizare mai clară. Analizand graficul, putem deduce performanta modelului, acesta indicand faptul ca modelul nu este foarte bun în a identifica clasele pozitive la început, insa acesta se imbunatateste, reusind sa separe claselor pozitive de cele negative, având un raport bun între adevărate pozitive și false pozitive.

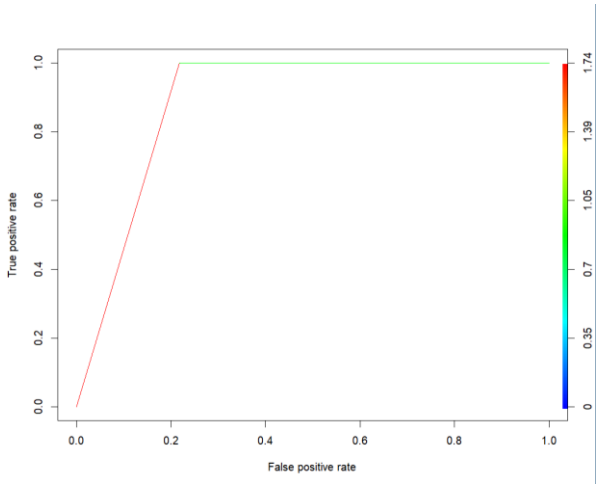


Figura 38. Curba ROC

În final, am calculat valoarea AUC, ce reprezinta probabilitatea ca modelul să clasifice corect o pereche aleatorie formată dintr-un exemplu pozitiv și unul negativ. Aceasta este egala cu 89%, ceea ce indică o performanță ridicată a modelului, separand cazurile pozitive de cele negative destul de bine, insa se pot efectua imbunatatiri.

```
> auc@y.values[[1]]
[1] 0.8913043
```

Figura 39. Valoarea AUC

Reprezentarea grafica a importantei variabilelor

Am reprezentat grafic importanța variabilelor pentru modelul de arbore decizional. Conform graficului și rezultatului comenzii summary, variabilele X5, X4 și X3 sunt cele mai influente în predicțiile modelului. Aceasta înseamnă că aceste variabile au cel mai mare impact asupra clasificării realizate pe arbore.

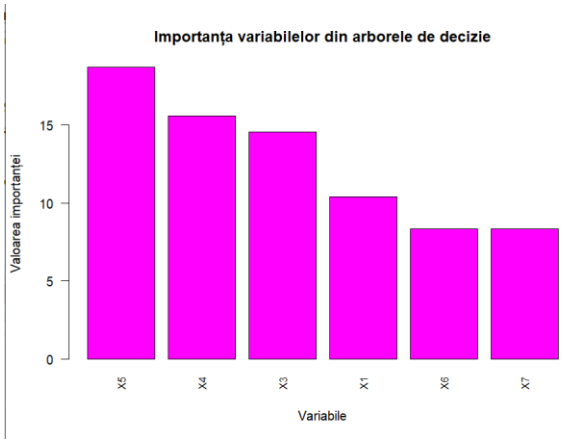


Figura 40. Reprezentarea grafica a importantei variabilelor

Modelul de regresie logistica (binomiala)

Am utilizat un model de regresie logistica binomiala pentru a evalua impactul profitului înainte de dobânzi și impozite (EBIT) și al profitului pe acțiune (EPS) asupra performanței financiare a unei companii. Performanța financiară a fost categorizată în două clase: clasa 1 – performanta ridicată și clasa 2 – performanta scăzută. Scopul principal al acestui model este de a identifica dacă și în ce măsură EBIT și EPS sunt factori determinanți în clasificarea unei companii ca fiind performantă sau nu.

In primul rand, am generat un grafic ce afiseaza distribuția punctelor de date in funcție de valorile variabilelor X4 si X5, cu fiecare categorie din variabila clasa reprezentata printr-o culoare distincta. Astfel, putem vizualiza relația dintre variabile si putem identifica grupari de date.

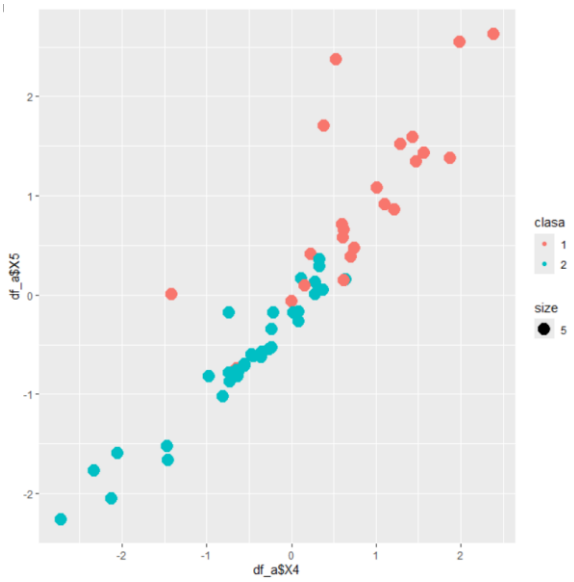


Figura 41. Reprezentarea grafica a relatiei dintre variabile

Analizand graficul putem observa ca elementele din clasa 1 se situeaza in partea superioara a graficului, reprezentand clusterul companiilor performante, iar cele din clasa 2 in partea inferioara, reprezentand clusterul companiilor slab performante, observandu-se o separare clara a observatiilor, a clasei pozitive de cea negativa.

Apoi, am antrenat modelul de regresie si am generat un rezumat al acestuia utilizand comanda summary:

```
Call:
glm(formula = clasa ~ X4 + X5, family = binomial, data = df_a)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.6475     0.4849   1.335  0.18177
X4            1.3032     1.2392   1.052  0.29297
X5           -5.3875     1.8001  -2.993  0.00276 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 76.883  on 56  degrees of freedom
Residual deviance: 30.158  on 54  degrees of freedom
AIC: 36.158

Number of Fisher Scoring iterations: 7
```

Figura 42. Rezultatul comenzii summary

Rezultatele indică faptul că variabila X5 are o asociere statistic semnificativă cu variabila dependentă clasa, ce reprezinta performanta financiara a companiilor (p-value < 0.05). Coeficientul negativ al lui X5 sugerează că o creștere a valorii lui X5 este asociată cu o scădere a probabilității de apariție a clasei 1, si anume a companiilor de succes. În schimb, influența variabilei X4 asupra variabilei clasa nu este semnificativă statistic la nivelul de semnificație de 0.05.

Devianta nula este egala cu 76.883, ceea ce indica faptul ca, fără a lua în considerare niciun predictor, există o incertitudine semnificativă în a prezice corect clasa. In ceea ce priveste devianta reziduala, valoarea de 30.158 indica o discrepanta mare intre valorile observate și cele prezise de model. Totusi, comparand cei doi indicatori, observam ca devianta reziduala este semnificativ mai mica decat devianta nula, ceea ce sugereaza faptul ca adăugarea predictorilor X4 si X5 în model a îmbunătățit semnificativ capacitatea modelului de a explica variația în date.

Realizarea de predictii pe setul de testare. Matricea de confuzie si acuratetea modelului.

Mai intai, am utilizat modelul antrenat pentru a genera probabilitati de apartenenta la clasa pozitiva (1) – clasa companiilor performante financiar, pentru fiecare observatie din setul de testare.

```
> prob
        NGD         RIG         MCW        MBLY         DNB         KTB        COMP       DJTWW         MYRG
0.971233356 0.841810251 0.787540875 0.992737904 0.734503647 0.573281612 0.994422161 0.964102743 0.929863303
        VNT         SCI         PEN         LAZ         HLI         ALVO        PINS        PLTK        WING
0.085681752 0.001920506 0.970223741 0.635067249 0.429801417 0.999168531 0.933396846 0.102218537 0.853187395
       BIRK         MHK           M         PII         AEO         KSS
0.300746699 0.023002497 0.385875080 0.329328407 0.241268042 0.019636743
```

Figura 43. Probabilitatile de apartenenta la clasa pozitiva

Apoi, am generat predictiile clasei intr-un vector pred, iar pentru fiecare observatie, daca probabilitatea calculata anterior este > 0.5, se considera ca observatia apartine clasei negative, altfel va apartine clasei pozitive.

```
> print(pred)
 [1] "2" "2" "2" "2" "2" "2" "2" "2" "2" "1" "1" "2" "2" "1" "2" "2" "1" "2" "1" "1" "1" "1" "1" "1"
```

Figura 44. Predictiile claselor

```
pred  1  2
   1  9  1
   2  2 12
```

Figura 45. Matricea de confuzie

```
> print(acuratete)
[1] 0.875
```

Figura 46. Acuratetea modelului

Apoi, am generat matricea de confuzie, din care reies urmatoarele informatii:

- Modelul a prezis corect 9 observatii ca apartinand clasei 1 si o singura observatie a fost gresit clasificata ca apartinand clasei 2.
- Modelul a prezis gresit 2 observatii ca apartinand clasei 1, dar a prezis corect 12 observatii ca apartinand clasei 2.

Asadar, modelul a clasificat corect 21 din 24 de observatii, ceea ce reprezinta o acuratetea de 87.5%, deci modelul are o performanta buna.

Pentru a evalua mai detaliat performanta modelului, am realizat curba ROC si am calculate valoarea AUC. Din figura 47, observam ca modelul prezinta o performanta scazuta la inceput, deoarece nu reuseste sa separe foarte clar elementele claselor pozitive de cele negative, insa treptat se imbunatateste, ajungand sa aiba o performanta buna in separarea datelor. Iar, in ceea ce priveste valoarea AUC, aceasta este egala cu 92%, ceea ce indica o performanta ridicata a modelului, separand cazurile pozitive de cele negative destul de bine, insa se pot efectua imbunatatiri.

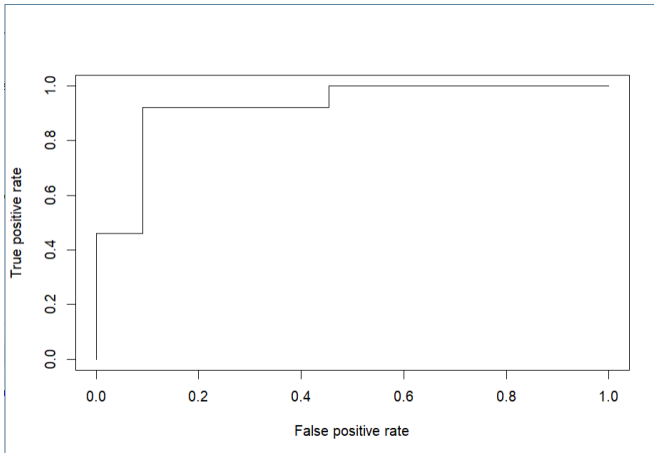


Figura 47. Curba ROC

```
> auc
[1] 0.9230769
```

Figura 48. Valoarea AUC

Tabel comparativ – Acuratetea modelelor

Algoritmul de clasificare	Acuratetea modelului
Clasificatorul Naiv Bayesian	91.67%
Clasificatorul KNN	85.7%
Arbore de decizie	79.17%
Regresie logistica	87.5%

Analizand tabelul realizat, observam ca modelul Naiv Bayesian a obtinut cea mai mare acuratețe, urmat îndeaproape de regresia logistică. Aceasta înseamnă că, în medie, clasificatorul Naiv Bayesian a făcut cele mai puține erori în a prezice clasa corectă a observațiilor din setul de testare, asadar acesta este cel mai eficient model pentru a clasifica setul de date.

Iordan Maria-Alexandra
Grupa 1080-A

Concluziile analizei

In concluzie, analiza evidențiază importanța utilizării algoritmilor de clasificare în învățarea supravegheată, pentru a obține o înțelegere mai profundă a performanței financiare a companiilor listate la bursă. Fiecare dintre metodele analizate - K-means, Naiv Bayesian, KNN, arborele de decizie și regresia logistică - oferă perspective unice și contribuie la identificarea caracteristicilor cheie care influențează clasificarea companiilor în funcție de performanța lor financiară.

Modelul Naiv Bayesian a demonstrat cea mai mare acuratețe, ceea ce sugerează că este cel mai eficient în a diferenția între companiile performante și cele cu performanțe mai slabe. Acest lucru subliniază puterea acestui algoritm în gestionarea variabilității și complexității datelor financiare. Regresia logistică a arătat, de asemenea, o performanță bună, indicând că variabilele precum EBIT și EPS sunt factori esențiali în determinarea performanței financiare. Această metodă oferă o interpretare ușoară și clară a impactului variabilelor individuale asupra probabilității de succes financiar. Alte metode, cum ar fi KNN și arborele de decizie, au prezentat rezultate promițătoare, dar cu unele limitări în precizia clasificării. Totuși, acestea sunt utile pentru a explora relații non-liniare și pentru a oferi vizualizări intuitive ale procesului de decizie.

În concluzie, utilizarea combinată a acestor algoritmi poate oferi o analiză robustă și cuprinzătoare a setului de date, contribuind la decizii mai informate și strategii mai eficiente în managementul financiar. Abordarea unei astfel de analize permite exploatarea punctelor forte ale fiecărui model, optimizând astfel capacitatea de predicție și interpretare a datelor financiare complexe.

ANEXA

```
tema <- Date_AD

View(tema)


# Eliminarea outlierilor:

# Iterăm prin coloanele setului de date (excluzând prima coloană)

for (col in colnames(tema)[-1]) {  # Excludem prima coloană folosind [-1]

  # Identificăm outlierii pentru fiecare coloană

  outliers <- boxplot(tema[[col]], plot = F)$out


  # Excludem liniile care conțin outlieri

  tema <- tema[-which(tema[[col]] %in% outliers), ]
```

```
}

View(tema)

# Calcularea indicatorilor statistici

summary(tema)

install.packages("psych")
library(psych)

describe(tema[-1])

# Matricea de corelatie si matricea de covarianta
cor(tema[-1])
cov(tema[-1])

#Pentru a observa mai bine rezultatele, vom standardiza datele
tema_std = scale(tema[-1], scale = T)
View(tema_std)

# Recalculam corelatia si covarianta

matrice_corelatie <- cor(tema_std)
matrice_covarianta <- cov(tema_std)
View(matrice_corelatie)
View(matrice_covarianta)

# Reprezentarea grafica a matricei de corelatie
```

Iordan Maria-Alexandra
Grupa 1080-A

```
install.packages("corrplot")

library(corrplot)


windows()

corrplot(matrice_corelatie, method = "circle", type = "upper", col ="pink", title = "Matricea de corelatie" )


#Generarea variabilei de clasificare utilizand analiza cluster


tema3 = tema

cor(tema3[-1])


#Elimin pe X8 si X9


date_3 = cbind(tema3[,2:8], tema3[,11])


# Standardizarea datelor


date_3_std = scale(date_3, scale=TRUE)

rownames(date_3_std)=tema3$Companie

View(date_3_std)


#Generarea variabilei de clasificare


k_means = kmeans(date_3_std, 2)

k_means


#Adaugarea clasificarii in setul de date


clasa = k_means$cluster
```

Iordan Maria-Alexandra
Grupa 1080-A

```
dataset = cbind(clasa, round(date_3_std,3))

dataset


df = data.frame(dataset)


#Impartirea setului de date in date de antrenare si date de testare


nr=round(nrow(df)*.70) #70% date-set de antrenare,30%-set testare

a <- sample(seq_len(nrow(df)),size=nr)


antrenare <- df[a,] #setul de antrenare

testare <- df[-a,] #setul de testare

round(antrenare,3)

round(testare,3)


df_a=data.frame(antrenare)

df_a$clasa[df_a$clasa==1]<-"clasa1"

df_a$clasa[df_a$clasa==2]<-"clasa2"

cbind(round(df_a[,2:9],3),df_a[,1])


#Variabila dependenta este clasa


#Clasificatorul NAIV BAYESIAN


install.packages("e1071")

library(e1071)

model <- naiveBayes(as.factor(df_a[,1])~.,data=df_a[,-1])

summary(model)


model$apriori
```

Iordan Maria-Alexandra
Grupa 1080-A

```
model$tables

model$levels

#Realizarea de predictii pe setul de testare

#Class=probabilitatile aposteriorice de apartenenta la clasa
#Raw=probabilitatile aposteriorice de apartenenta la grupe
#Frecventa incrucisata=matricea de confuzie
#Gradul de clasificare corecta(acuratetea )=suma pe diagonala principala/suma totala*

pred_test <- predict(model,testare[,-1],type="class")
pred_test

pred_test2 <- predict(model,testare[,-1],type="raw")
pred_test2

#Matricea de confuzie

conf <- table(pred_test,testare[,1],dnn=c("Prediction","Actual"))

#Acuratetea modelului
acuratete <- sum(diag(conf)) / sum(conf)
acuratete

#Metoda KNN

install.packages("caret")
install.packages("MLmetrics")
```

Iordan Maria-Alexandra
Grupa 1080-A

```
library(MLmetrics)

library(caret)

library(e1071)


#Vom utiliza aceeasi distributie


df_a=data.frame(antrenare)

df_t=data.frame(testare)

df_a$clasa <- as.factor(df_a$clasa)

df_t$clasa <- as.factor(df_t$clasa)


levels(antrenare$clasa) <- make.names(levels(factor(antrenare$clasa)))

levels(testare$clasa) <- make.names(levels(factor(testare$clasa)))


#Setarea parametrilor pt validarea incrucisata repetata

repeats = 3

numbers = 10


set.seed(1234)


#Definirea controlului pentru validarea încrucișată repetată


x=trainControl(method="repeatedcv",number=numbers,repeats = repeats,

               classProbs = TRUE,summaryFunction = twoClassSummary)


#Definirea unor valori pt k

k_val = c(3,7,10)


#Vector pt acuratete

ac_val = c()
```

Iordan Maria-Alexandra
Grupa 1080-A

```
#Antrenarea modelului pt fiecare valoare a lui k

for (k in k_val) {
  model_knn <- train(clasa ~ .,data=antrenare,method="knn",
                     preProcess=c("center","scale"),trControl=x,metric="ROC",tuneLength=tunel)

  pred_test <- predict(model_knn,df_t[,-1])

  matrice_conf <- table(Predicted = pred_test, Actual = df_t$clasa)

  acuratete = sum(diag(matrice_conf)) / sum(matrice_conf)
  ac_val = c(ac_val, acuratete)
}

ac_val

#Vom alege modelul cu k=7 deoarece are cea mai mare valoare pentru acuratete

tunel = 7

model_knn_7 <- train(clasa ~ .,data=antrenare,method="knn",
                     preProcess=c("center","scale"),trControl=x,metric="ROC",tuneLength=tunel)
model_knn_7

#Vizualizarea modelului
windows()
plot(model_knn_7)

#Realizarea de predictii pe setul de testare
```

Iordan Maria-Alexandra
Grupa 1080-A

```
#Predictii pentru clase
pred_clase <- predict(model_knn_7, testare)
pred_clase

#Predictii probabilistice
pred_prob <- predict(model_knn_7, testare, type ="prob")
pred_prob
head(pred_prob)

#Matricea de confuzie
matrice_confuzie <- table(pred_clase,testare[,1],dnn=c("Prediction","Actual"))
print(matrice_confuzie)

#Acuratetea modelului
acuratete <- sum(diag(matrice_confuzie))/sum(matrice_confuzie)
print(acuratete)

# Evaluarea modelului cu ROC și AUC

install.packages("ROCR")
library(ROCR)

pred_val <- prediction(pred_prob[,2],testare$clasa)
pred_val

perf_val <- performance(pred_val,"auc")
perf_val

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

```
plot(perf_val,col="green",lwd=1.5, main = "Curba ROC - Model KNN")


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

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

auc


#Metoda arbore de decizie/Random forest


#Vom utiliza aceeasi distributie


df_a=data.frame(antrenare)

df_t=data.frame(testare)


install.packages("rpart")

library(rpart)

install.packages("rpart.plot")

library(rpart.plot)


arbore_decizie <- rpart(clasa~.,data=df_a,method="class")

summary(arbore_decizie)

printcp(arbore_decizie)


windows()

plotcp(arbore_decizie)


arbore_decizie <- rpart(clasa~.,data=df_a,method="class",control=rpart.control(cp=0.18))

summary(arbore_decizie)

prp(arbore_decizie,type=4,extra=106,box.palette="BuPu",under=T,fallen.leaves =F )


#Realizarea de predictii pe setul de testare
```

Iordan Maria-Alexandra
Grupa 1080-A

```
#Predictii pentru clase
pred_clase_arbore <- predict(arbore_decizie, df_t, type="class")
pred_clase_arbore

#Predictii probabilistice
pred_prob_arbore <- predict(arbore_decizie, df_t, type ="prob")
pred_prob_arbore
head(pred_prob_arbore)

#Matricea de confuzie
matrice_confuzie_arbore <- table(pred_clase_arbore,df_t$clasa,dnn=c("Prediction","Actual"))
print(matrice_confuzie_arbore)

#Acuratetea modelului
acuratete_arbore <- sum(diag(matrice_confuzie_arbore))/sum(matrice_confuzie_arbore)
print(acuratete_arbore)

#Construim curba Roc pt arbore
install.packages("ROCR")
library(ROCR)
yhat2 <- predict(arbore_decizie,type="prob")[,2]
pr2 <- prediction(yhat2,df_a$clasa)
performanta <- performance(pr2,"tpr","fpr")
plot(performanta,colorize=T)
auc <- performance(pr2,"auc")
auc@y.values[[1]]

#Reprezentarea grafica a importantei variabilelor
```

Iordan Maria-Alexandra
Grupa 1080-A

```
importanta_var <- arbore_decizie$variable.importance


barplot(
  importanta_var,
  main = "Importanța variabilelor din arborele de decizie",
  xlab = "Variabile",
  ylab = "Valoarea importanței",
  col = "magenta",
  las = 2,
  cex.names = 0.8 )


#Modelul de regresie logistica(binomiala)


#Scopul problemei: analiza influentei EBIT si a EPS asupra performantei financiare
#Variabila dependenta: clasa (1-performanta ridicata, 2-performanta slaba)
#Variabilele independente: EBIT si EPS


df_a=data.frame(antrenare)
df_t=data.frame(testare)
df_a$clasa <- as.factor(df_a$clasa)
df_t$clasa <- as.factor(df_t$clasa)


install.packages("ggplot2")
library(ggplot2)


plot <- ggplot(data=df_a,aes(x=df_a$X4,y=df_a$X5,col=clasa))
plot <- plot+geom_point(aes(size=5))
windows()
plot
```

Iordan Maria-Alexandra
Grupa 1080-A

```
model <- glm(clasa~X4+X5,data=df_a,family=binomial)
summary(model)

#Realizarea de predictii pe setul de testare

prob <- predict(model,df_t,type="response")
prob

#Matricea de confuzie
pred <- rep("1",dim(df_t)[1])
pred[prob>0.5]="2"
print(pred)
matrice_confuzie_reg <- table(pred,df_t$clasa)
print(matrice_confuzie_reg)

#Acuratetea
acuratete <- sum(diag(matrice_confuzie_reg))/sum(matrice_confuzie_reg)
print(acuratete)

#Curba ROC

install.packages("ROCR")
library(ROCR)

p <- predict(model,newdata=df_t,type="response")
pr <- prediction(p,df_t$clasa)
prf <- performance(pr,measure="tpr",x.measure="fpr")
plot(prf)

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

Iordan Maria-Alexandra
Grupa 1080-A

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

auc
```