Proiect la Fundamente de BIG DATA

Abordarea Absenteismului în Industria Curieratului: O Analiză Profundă

Student: Godja Vasile
Specializarea: IE
Grupa: 3

1. Introducere

Am ales drept temă principală analiza absențelor în industria curieratului și impactul acestora asupra eficienței operaționale. Drept sursă de date vom folosi un set de date pus la dispoziție de către firma de curierat XYZ, care se confruntă cu problema absenteismului. Prin studierea acestui set, vom încerca să răspundem la următoarele întrebări:

1.Care dintre factorii prezenți în setul de date influențează cel mai mult nivelul de absenteism?
2.Cu câtă acuratețe se poate anticipa nivelul de absenteism al unei persoane, bazându-ne pe diverși factori personali prezenți în setul de date?

Odată stabilită direcția de cercetare a acestui proiect, ne concentrăm pe răspunsurile la întrebările noastre de business, aplicând diverse metode de clasificare studiate în cadrul disciplinei Fundamente de Big Data. Scopul este de a oferi o comparație a rezultatelor metodelor alese și, desigur, de a identifica metoda cea mai eficace.

Întrebările noastre de afaceri nu au fost selectate la întâmplare. Există o serie de motive care au stat la baza selecției acestora, însă două aspecte ne-au captat în mod deosebit atenția, având în vedere impactul lor asupra eficienței operaționale a unei companii și înțelegerea perspectivei angajaților.

În primul rând, am identificat absenteismul ca o problemă majoră în cadrul multor companii. Pentru majoritatea organizațiilor, prezența continuă a personalului este o necesitate vitală pentru a asigura un serviciu de înaltă calitate și satisfacția clienților. Acest lucru este adesea afectat profund de absenteism, fenomen care poate duce la o scădere a productivității, costuri suplimentare și disfuncționalități în cadrul fluxului de muncă.

În al doilea rând, înțelegerea factorilor care contribuie la absenteism poate ajuta la dezvoltarea unor strategii mai eficiente de management al personalului. Astfel, ajustarea politicilor de resurse umane poate deveni esențială pentru a combate acest fenomen.

Având aceste două aspecte în minte, vom încerca în acest proiect să analizăm și să anticipăm nivelul de absenteism. Ne vom baza pe diferiți factori personali și alți parametri disponibili în setul de date. Scopul nostru este să identificăm ce anume influențează cel mai mult absenteismul. Rezultatele acestei analize ar putea beneficia nu numai compania de curierat în cauză, ci și alte companii care se confruntă cu probleme similare, furnizându-le o bază solidă pentru a dezvolta strategii de gestionare a absenteismului.

2. Setul de date

Setul de date folosit poate fi accesat la adresa https://www.kaggle.com/datasets/tonypriyanka2913/employee-absenteeism iar in forma lui initiala acesta cuorinde urmatoarele attribute:

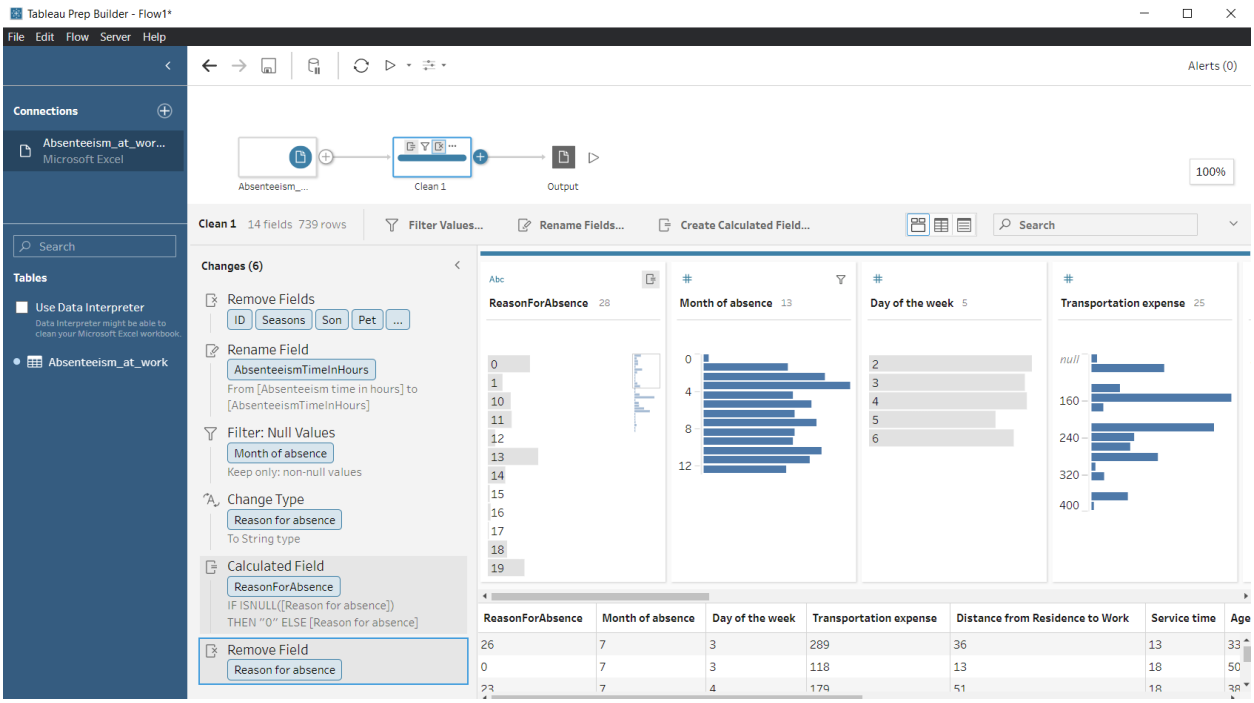
Atribut	Tip	Descriere
ID	Număr	Numărul de identificare al angajatului
Reason for absence	String	Motivul absenței, codificat în 28 de categorii diferite (ex. boli, concediu, etc.)
Month of absence	String	Luna anului când a avut loc absența
Day of the week	String	Ziua săptămânii când a avut loc absența
Seasons	String	Anotimpul în care a avut loc absența
Transportation expense	Număr	Costul transportului pentru angajat
Distance from Residence to Work	String	Distanța de la locuința angajatului la locul de muncă
Service time	String	Durata serviciului angajatului în cadrul companiei
Age	String	Vârsta angajatului
Work load Average/day	Număr	Volumul mediu de muncă pe zi
Hit target	Număr	Realizarea obiectivului de muncă
Disciplinary failure	String	A existat sau nu o încălcare disciplinară
Education	Număr	Nivelul de educație al angajatului
Son	String	Numărul de fii ai angajatului
Social drinker	Număr	Indică dacă angajatul este sau nu un consumator social de alcool
Social smoker	Număr	Indică dacă angajatul este sau nu un fumător social
Pet	Număr	Numărul de animale de companie ale angajatului
Weight	Număr	Greutatea angajatului
Height	Număr	Înălțimea angajatului
Body mass index	Număr	Indicele de masă corporală al angajatului
Absenteeism time in hours	Număr	Durata absenței în ore

Înainte de a ne aprofunda în analiza datelor propriu-zisă, folosind limbajul de programare R, am ales să explorăm și să preprocesăm datele brute cu ajutorul Tableau Prep. Această etapă inițială a inclus numeroși pași de curățare a datelor, printre care eliminarea unor atribute care fie erau identificatori (de exemplu, ID), fie nu exercitau un impact semnificativ asupra variabilei țintă (Absenteeism time in hours). Astfel, am încercat să evităm riscul de a ne abate de la obiectivul principal al analizei.

In acelasi timp au existat si alti pasi de curatare precum:

- eliminarea coloanelor cu valori nule
- folosirea unor formule de calcul pentru a complete unele spatii nule (media/mediana/val 0)
- formatarea capetelor de table pentru a corespunde formatului standard (Absenteeism time in hours → AbsenteeismTimeInHours)

La final s-a realizat un output de tip csv , fisier pe care l-am important in sriptul R.



Setul de date importat in R studio (varianta finala dupa cleaning) are urmatoare structura:

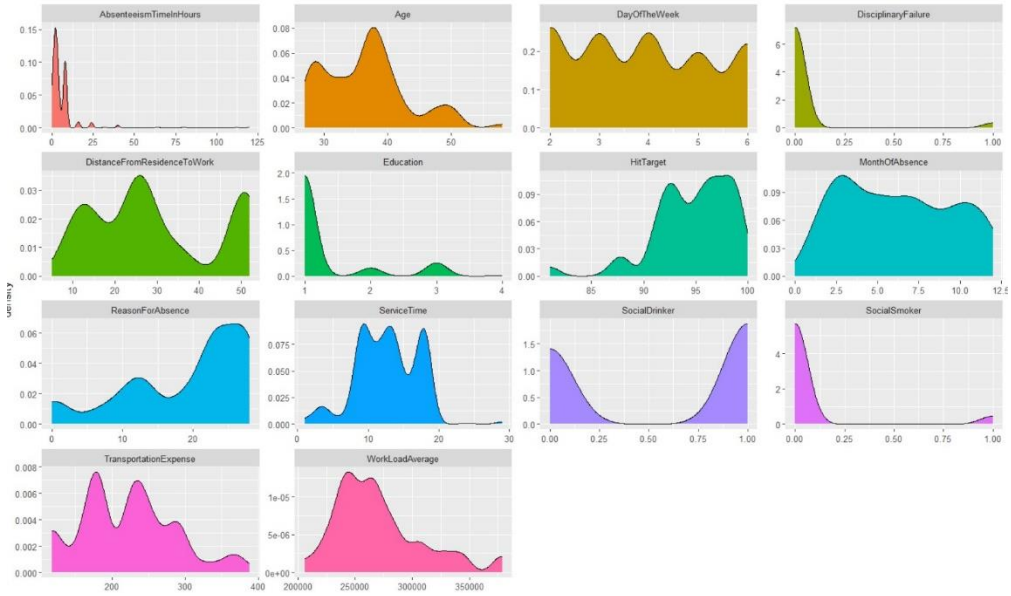
```
$ ReasonForAbsence           : num [1:663] 26 0 23 7 23 23 19 22 1 1 ...
$ MonthOfAbsence             : num [1:663] 7 7 7 7 7 7 7 7 7 7 ...
$ DayOfTheWeek               : num [1:663] 3 3 4 5 5 6 2 2 2 3 ...
$ TransportationExpense      : num [1:663] 289 118 179 279 289 260 155 235 260 260 ...
$ DistanceFromResidenceToWork: num [1:663] 36 13 51 5 36 50 12 11 50 50 ...
$ ServiceTime                : num [1:663] 13 18 18 14 13 11 14 14 11 11 ...
$ Age                        : num [1:663] 33 50 38 39 33 36 34 37 36 36 ...
$ WorkLoadAverage            : num [1:663] 239554 239554 239554 239554 239554 ...
$ HitTarget                  : num [1:663] 97 97 97 97 97 97 97 97 97 97 ...
$ DisciplinaryFailure        : num [1:663] 0 1 0 0 0 0 0 0 0 0 ...
$ Education                  : num [1:663] 1 1 1 1 1 1 1 3 1 1 ...
$ SocialDrinker              : num [1:663] 1 1 1 1 1 1 1 0 1 1 ...
$ SocialSmoker               : num [1:663] 0 0 0 1 0 0 0 0 0 0 ...
$ AbsenteeismTimeInHours     : num [1:663] 4 0 2 4 2 4 40 8 8 8 ...
```

3. Rezultate și discuții

În această secțiune, vom detalia metodele alese pentru a răspunde la întrebările de afaceri stabilite, in principal ne focusam spre identificarea acelor factori care au cea mai mare influenta aspra numarului de ore absentate. Am abordat problema clasificării, în care variabila dependentă este calitativă. În cazul nostru, variabila calitativă din setul de date este "AbsenteeismTimeInHours", care determină dacă angajații sunt frecvent absenți sau nu. Valorile acesteia au fost împărțite în "High" și "Low" fiind considerati cu grad mare de absenteeism (High) cei care au un numar de ore ce depaseste media de 7 ore absentate iar restul se incadreaza in categoria Low.

La o primă inspecție a setului de date, în care ne-am concentrat pe variabilele numerice, am identificat mai multe atribute care sunt mai bine reprezentate ca factori, nu ca valori numerice.

```
absente %>%
  select_if(is.numeric) %>%
  gather(metric,value) %>%
  ggplot(aes(value, fill=metric)) +
  geom_density(show.legend = FALSE) +
  facet_wrap(~metric, scales = "free")
```



Urmatoarele atribute reprezintă categorii sau clase distincte, mai degrabă decât valori continue:

1. **ReasonForAbsence:** Acest atribut reprezintă diferite motive de absență, fiecare codificat ca un număr astfel prin transformarea acestuia într-un factor va permite o mai bună interpretare și analiză a acestui atribut.

2. **MonthOfAbsence:** Deși acest atribut este în prezent numeric, reprezentarea sa ca factor este mai adecvată.

3. **DayOfTheWeek:** Similar cu MonthOfAbsence, zilele săptămânii sunt mai bine reprezentate ca factori, permițând astfel analiza modelului de absență în funcție de ziua săptămânii.

4. **Education:** Acest atribut reprezintă diferite niveluri de educație și, prin urmare, ar trebui să fie tratat ca un factor.

5. **SocialSmoker & SocialDrinker :** Acestea sunt atribute binare care reprezintă dacă o persoană este fumătoare sau nu oridaca consuma alccool sau nu.

6. **DisciplinaryFailure:** Si acest atribut este un indicator binar al faptului dacă o persoană a avut sau nu o defecțiune disciplinară.

În cadrul proiectului, am folosit diverse metode de clasificare pentru a analiza și rezolva problema absenteismului în rândul angajaților, cum ar fi:

```
1.Arbore de Decizie: Implementat cu "rpart".
2.Arbore Tăiat: Tot cu "rpart".
3.Arbore cu Gini: Folosind "tree" și "ipred".
3.Arbore Bagging: Folosind .
4.Random Forest: Utilizând "randomForest" și "ranger".
5.Naive Bayes: Implementat cu "caret".
```

Înainte de a demara procesul de creare a modelelor, am efectuat un pas crucial, acela de a diviza setul de date în două subseturi esențiale pentru orice model, setul de antrenament (absente_train) și setul de test (absente_test). Folosind parametrul `strata = "AbsenteeismTimeInHours"` ne-a permis să ne asigurăm că proporția dintre categoriile "Low" și "High" se păstrează în mod constant în ambele subseturi, pentru a menține integritatea și reprezentativitatea datelor în cadrul modelului de învățare automată.

3.1 Arbori de decizie

Arborii de decizie reprezintă un model de învățare automată utilizat într-o varietate de domenii fiind recunoscuți pentru simplitatea interpretării lor devenind accesibili persoanelor fără cunoștințe avansate în domeniu. Totuși, din cauza lipsei lor de robustețe, este adesea necesară implementarea unor metode de îmbunătățire a acurateței, cum ar fi bagging și random forest.

Arbore 1

În cazul primului arbore, se utilizează libraria rpart pe setul de date de test, absente_test, folosind metoda "class". Variabila dependentă este AbsenteeismTimeInHours, iar restul variabilelor sunt considerate independente iar prin interpretarea modelului tragem urmatoarele concluzii.

- Arborele de decizie începe cu 463 de înregistrări, unde majoritatea (64%) sunt etichetate ca "High". Deciziile ulterioare se bazează pe variabila ReasonForAbsence.

- Dacă ReasonForAbsence este în {1,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,21,22,24,26}, există o probabilitate mai mare (72%) de a avea eticheta "High". Dacă TransportationExpense este mai mare sau egal cu 234, probabilitatea crește la 90%.

- Dacă ReasonForAbsence este în {0,16,23,25,27,28}, majoritatea înregistrărilor (93%) sunt etichetate ca "Low".

- Alte variabile precum ServiceTime, MonthOfAbsence și DayOfTheWeek ajută la rafinarea predicțiilor în anumite condiții insa cele mai influente variabile par sa fie ReasonForAbsence, MonthofAbsente si TransportationExpense.

Se poate observa ca modelul a efectuat 7 împărțiri, cu un cost de tăiere (cost complexity) minim de 0.32. Cu toate acestea, trebuie să fim conștienți că modelul poate fi suprara-ajustat (overfitting) după un anumit număr de împărțiri, conform erorii de validare încrucișată (cross-validation error).

```
n= 463

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 463 168 Low (0.36285097 0.63714903)
   2) ReasonForAbsence=1,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,21,22,24,26 210  59 High (0.71904762 0.28095238)
     4) TransportationExpense≥234 104  16 High (0.84615385 0.15384615)
       8) ReasonForAbsence=1,6,7,10,11,13,15,18,19,21,22,24,26 92   9 High (0.90217391 0.09782609) *
       9) ReasonForAbsence=4,12,14 12   5 Low (0.41666667 0.58333333) *
     5) TransportationExpense< 234 106  43 High (0.59433962 0.40566038)
      10) ServiceTime≥11.5 54  15 High (0.72222222 0.27777778)
        20) ReasonForAbsence=6,7,10,12,13,17,18,19,22 34   4 High (0.88235294 0.11764706) *
        21) ReasonForAbsence=1,5,8,11,14,21,26 20   9 Low (0.45000000 0.55000000)
          42) MonthOfAbsence=2,3,6,7,9 11   3 High (0.72727273 0.27272727) *
          43) MonthOfAbsence=1,5,8,10,11 9   1 Low (0.11111111 0.88888889) *
      11) ServiceTime< 11.5 52  24 Low (0.46153846 0.53846154)
        22) DayOfTheWeek=2,3 23   7 High (0.69565217 0.30434783) *
        23) DayOfTheWeek=4,5,6 29   8 Low (0.27586207 0.72413793) *
   3) ReasonForAbsence=0,16,23,25,27,28 253  17 Low (0.06719368 0.93280632) *
> summary(arb1)
Call:
rpart(formula = AbsenteeismTimeInHours ~ ., data = absente_train,
    method = "class")
  n= 463

         CP nsplit rel error    xerror       xstd
1 0.54761905      0 1.0000000 1.0000000 0.06158371
2 0.02579365      1 0.4523810 0.5178571 0.05003268
3 0.02083333      4 0.3750000 0.6250000 0.05363353
4 0.01190476      6 0.3333333 0.6309524 0.05381301
5 0.01000000      7 0.3214286 0.6369048 0.05399047

Variable importance
          ReasonForAbsence             MonthOfAbsence       TransportationExpense DistanceFromResidenceToWork               ServiceTime            WorkLoadAverage
                        53                         10                          9                          7                          7                          6
             DayOfTheWeek                        Age             SocialDrinker                  HitTarget
                         3                          2                          2                          1
```

Un urma predictiilor pe setul de date absente_test au fost prezise corect 163 din cele 200 de înregistrări de test, ceea ce rezultă într-o acuratețe de 81.5%. Sensibilitatea (capacitatea de a detecta corect clasa “High”) este de 75.34%, iar specificitatea (capacitatea de a detecta corect clasa “Low”) este de 85.04%. Rezultatele sugerează că modelul are o performanță moderată în clasificarea corectă a absenteismului, cu o oportunitate de îmbunătățire a detectării absenteismului ridicat.

In cee ace priveste p de 2.109e-08, aceasta este extrem de mică, ceea ce sugerează că modelul tău are o performanță semnificativ mai bună decât o predicție aleatorie, în ceea ce privește acuratețea.

```
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High   55  19
      Low    18 108

               Accuracy : 0.815
                 95% CI : (0.7541, 0.8663)
    No Information Rate : 0.635
    P-Value [Acc > NIR] : 2.109e-08

                  Kappa : 0.6021

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.7534
            Specificity : 0.8504
         Pos Pred Value : 0.7432
         Neg Pred Value : 0.8571
             Prevalence : 0.3650
         Detection Rate : 0.2750
   Detection Prevalence : 0.3700
      Balanced Accuracy : 0.8019

       'Positive' Class : High
```

Arbore 1 Pruned

Arborele de decizie pruned (tăiat) este o versiune simplificată a arborelui de decizie inițial, care a fost realizată prin eliminarea unor ramuri și împărțiri care nu contribuie semnificativ la performanța modelului. Prin tăierea arborelui, se elimină riscul de suprarajustare și se obține o structură mai ușor de interpretat.

În cazul arborelui pruned, avem următoarele concluzii:

- Arborele este format din 5 noduri și 4 ramuri. Nodul rădăcină are 463 de observații și prezice clasa "Low" cu o acuratețe de 63.79% și clasa "High" cu o acuratețe de 36.29%.

- Primul criteriu de împărțire este variabila "ReasonForAbsence". Dacă motivul absenței este în intervalul {1,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,21,22,24,26}, există o probabilitate mai mare (71.9%) ca absentele să fie clasificate ca "High". Dacă valoarea variabilei "TransportationExpense" este mai mare sau egală cu 234, probabilitatea crește la 84.61%. În caz contrar, absentele sunt clasificate ca "Low".

- Dacă motivul absenței este în intervalul {0,16,23,25,27,28}, majoritatea absențelor sunt clasificate ca "Low" (probabilitate de 93.28%).

- Alte variabile precum "ServiceTime", "MonthOfAbsence" și "DayOfTheWeek" contribuie și ele la rafinarea predicțiilor.

```
n= 463

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 463 168 Low (0.36285097 0.63714903)
   2) ReasonForAbsence=1,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,21,22,24,26 210  59 High (0.71904762 0.28095238)
     4) TransportationExpense≥234 104  16 High (0.84615385 0.15384615) *
     5) TransportationExpense< 234 106  43 High (0.59433962 0.40566038)
      10) ServiceTime≥11.5 54  15 High (0.72222222 0.27777778) *
      11) ServiceTime< 11.5 52  24 Low (0.46153846 0.53846154)
        22) DayOfTheWeek=2,3 23   7 High (0.69565217 0.30434783) *
        23) DayOfTheWeek=4,5,6 29   8 Low (0.27586207 0.72413793) *
   3) ReasonForAbsence=0,16,23,25,27,28 253  17 Low (0.06719368 0.93280632) *
> summary(arb1.pruned)
```

Acuratețea arborelui pruned pe setul de date de test (absente_test) este de 82.5%. Sensibilitatea (capacitatea de a detecta corect clasa "High") este de 79.45%, iar specificitatea (capacitatea de a detecta corect clasa "Low") este de 84.25%. Valorile preconizate pozitive și negative sunt de 74.36% și, respectiv, 87.70%. Aceste rezultate sugerează că arborele pruned are o performanță moderată în clasificarea corectă a absenteismului.

```
n= 463

         CP nsplit rel error    xerror       xstd
1 0.54761905      0  1.000000 1.0000000 0.06158371
2 0.02579365      1  0.452381 0.5178571 0.05003268
3 0.02500000      4  0.375000 0.6250000 0.05363353

Variable importance
          ReasonForAbsence       TransportationExpense
ServiceTime DistanceFromResidenceToWork            WorkLoadAve
                        54                             10
8                        7                              6
             DayOfTheWeek                SocialDrinker
                        3                              2
```

```
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High   58  20
      Low    15 107

               Accuracy : 0.825
                 95% CI : (0.7651, 0.875)
    No Information Rate : 0.635
    P-Value [Acc > NIR] : 3.02e-09

                  Kappa : 0.6279

 Mcnemar's Test P-Value : 0.499

            Sensitivity : 0.7945
            Specificity : 0.8425
         Pos Pred Value : 0.7436
         Neg Pred Value : 0.8770
             Prevalence : 0.3650
         Detection Rate : 0.2900
   Detection Prevalence : 0.3900
      Balanced Accuracy : 0.8185

       'Positive' Class : High
```

Arbore 2 CP=0

În cadrul arborelui 2, am utilizat o valoare a parametrului de cost de tăiere (cost complexity parameter - CP) egală cu zero. Acest lucru înseamnă că nu s-a realizat nicio tăiere a arborelui, iar acesta este complet dezvoltat.

Următoarele concluzii pot fi trase din arborele de decizie:

- ReasonForAbsence este variabila care inițiază separarea în arborele de decizie. În funcție de motivele absenței, observațiile se împart în două grupuri majore: unul în care predomină categoria "High" și unul în care predomină categoria "Low".

- Pentru motivele de absență grupate în prima categorie, următoarea variabilă care adâncește separarea este TransportationExpense. Observațiile în care TransportationExpense este mai mare sau egală cu 234 au o tendință și mai mare să fie clasificate ca "High", în special în cazul anumitor motive de absență (1,6,7,10,11,13,15,18,19,21,22,24,26).

- În cazul observațiilor în care TransportationExpense este sub 234, se produce o nouă separare pe baza ServiceTime. Observațiile cu un ServiceTime de 11.5 sau mai mare au o tendință mai mare de a fi clasificate ca "High", în timp ce cele cu ServiceTime sub 11.5 au o tendință de a fi clasificate ca "Low". În fiecare dintre aceste grupe, separarea continuă pe baza altor variabile, cum ar fi ReasonForAbsence, MonthOfAbsence, și DayOfTheWeek.

- În ceea ce privește grupul de observații în care ReasonForAbsence este din cealaltă categorie (0,16,23,25,27,28), majoritatea acestora sunt clasificate ca "Low". Separarea ulterioară în acest grup se face în funcție de TransportationExpense și DistanceFromResidenceToWork. De exemplu, observațiile cu TransportationExpense sub 295.5 și cu DistanceFromResidenceToWork mai mare sau egală cu 10.5 sunt în mare parte clasificate ca "Low".

```
n= 463

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 463 168 Low (0.36285097 0.63714903)
    2) ReasonForAbsence=1,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,21,22,24,26 210  59 High (0.71904762 0.28095238)
      4) TransportationExpense≥234 104  16 High (0.84615385 0.15384615)
        8) ReasonForAbsence=1,6,7,10,11,13,15,18,19,21,22,24,26 92   9 High (0.90217391 0.09782609) *
        9) ReasonForAbsence=4,12,14 12   5 Low (0.41666667 0.58333333) *
      5) TransportationExpense< 234 106  43 High (0.59433962 0.40566038)
       10) ServiceTime≥11.5 54  15 High (0.72222222 0.27777778)
         20) ReasonForAbsence=6,7,10,12,13,17,18,19,22 34   4 High (0.88235294 0.11764706) *
         21) ReasonForAbsence=1,5,8,11,14,21,26 20   9 Low (0.45000000 0.55000000)
           42) MonthOfAbsence=2,3,6,7,9 11   3 High (0.72727273 0.27272727) *
           43) MonthOfAbsence=1,5,8,10,11 9   1 Low (0.11111111 0.88888889) *
       11) ServiceTime< 11.5 52  24 Low (0.46153846 0.53846154)
         22) DayOfTheWeek=2,3 23   7 High (0.69565217 0.30434783)
           44) ReasonForAbsence=1,7,8,9,10,11,18 12   1 High (0.91666667 0.08333333) *
           45) ReasonForAbsence=12,13,19 11   5 Low (0.45454545 0.54545455) *
         23) DayOfTheWeek=4,5,6 29   8 Low (0.27586207 0.72413793)
           46) MonthOfAbsence=3,6,10 13   6 High (0.53846154 0.46153846) *
           47) MonthOfAbsence=2,4,5,7,8,9,11 16   1 Low (0.06250000 0.93750000) *
    3) ReasonForAbsence=0,16,23,25,27,28 253  17 Low (0.06719368 0.93280632)
      6) TransportationExpense≥295.5 9   3 Low (0.33333333 0.66666667) *
      7) TransportationExpense< 295.5 244  14 Low (0.05737705 0.94262295)
       14) DistanceFromResidenceToWork< 10.5 16   3 Low (0.18750000 0.81250000) *
       15) DistanceFromResidenceToWork≥10.5 228  11 Low (0.04824561 0.95175439)
         30) MonthOfAbsence=3,4,8,10 89   9 Low (0.10112360 0.89887640)
           60) TransportationExpense< 207 39   7 Low (0.17948718 0.82051282)
            120) ReasonForAbsence=23,25,28 21   7 Low (0.33333333 0.66666667)
              240) TransportationExpense≥167 13   6 High (0.53846154 0.46153846) *
              241) TransportationExpense< 167 8   0 Low (0.00000000 1.00000000) *
            121) ReasonForAbsence=0,16,27 18   0 Low (0.00000000 1.00000000) *
           61) TransportationExpense≥207 50   2 Low (0.04000000 0.96000000) *
         31) MonthOfAbsence=0,1,2,5,6,7,9,11,12 139   2 Low (0.01438849 0.98561151) *
> summary(arb2)
```

După aplicarea predicțiilor pe setul de date de test folosind modelul bazat pe arbore de decizie cu cp = 0, au fost prezise corect 153 din cele 200 de înregistrări de test, rezultând o acuratețe de 76.5%. Sensibilitatea, care măsoară capacitatea de a detecta corect clasa "High", este de 72.6%. Specificitatea, care măsoară capacitatea de a detecta corect clasa "Low", este de 78.74%.

Valorile preconizate pozitive, adică când modelul prezice clasa "High", sunt corecte în 66.25% din cazuri. Valorile preconizate negative, adică când modelul prezice clasa "Low", sunt corecte în 83.33% din cazuri.

```
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High   53  27
      Low    20 100

               Accuracy : 0.765
                 95% CI : (0.7, 0.8219)
    No Information Rate : 0.635
    P-Value [Acc > NIR] : 5.615e-05

                  Kappa : 0.5032

 Mcnemar's Test P-Value : 0.3815

            Sensitivity : 0.7260
            Specificity : 0.7874
         Pos Pred Value : 0.6625
         Neg Pred Value : 0.8333
             Prevalence : 0.3650
         Detection Rate : 0.2650
   Detection Prevalence : 0.4000
      Balanced Accuracy : 0.7567

       'Positive' Class : High
```

Arbore Gini

Metoda arborelui de decizie bazat pe indicele Gini este o tehnică populară de învățare automată utilizată pentru problemele de clasificare și regresie. Aceasta funcționează prin crearea unui model de decizii bazat pe valori ale atributelor datelor de intrare. Un arbore de decizie construit cu indexul Gini va căuta să minimizeze impuritatea Gini la fiecare pas, alegând divizarea care rezultă în cel mai mic indice Gini pentru nodurile copil. Astfel, arborele de decizie tinde să creeze ramuri care separă cât mai bine clasele.

```
Number of terminal nodes:  35
Residual mean deviance:  0.3963 = 169.6 / 428
Misclassification error rate: 0.09935 = 46 / 463
> pred_arb_gini <- predict(arb_gini, newdata = absente_test, target = "class")
> pred_arb_gini <- as_tibble(pred_arb_gini) %>% mutate(class = ifelse(Low ≥ High, "Low", "High"))
> confusionMatrix(factor(pred_arb_gini$class), factor(absente_test$AbsenteeismTimeInHours))
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High   50  21
      Low    23 106

               Accuracy : 0.78
                 95% CI : (0.7161, 0.8354)
    No Information Rate : 0.635
    P-Value [Acc > NIR] : 7.124e-06

                  Kappa : 0.5226

 Mcnemar's Test P-Value : 0.8802

            Sensitivity : 0.6849
            Specificity : 0.8346
         Pos Pred Value : 0.7042
         Neg Pred Value : 0.8217
             Prevalence : 0.3650
         Detection Rate : 0.2500
   Detection Prevalence : 0.3550
      Balanced Accuracy : 0.7598

       'Positive' Class : High
```

Arborele de decizie bazat pe indicele Gini a generat un model cu 35 de noduri terminale, folosind 9 variabile pentru construcția arborelui. Variabilele utilizate au fost "ReasonForAbsence", "TransportationExpense", "DistanceFromResidenceToWork", "DayOfTheWeek", "ServiceTime", "Age", "WorkLoadAverage", "MonthOfAbsence" și "SocialDrinker".

Eroarea medie a deviației reziduale a fost de 0.3963, sugerând că modelul se potrivește destul de bine datelor. Totuși, rata de eroare a clasificării greșite a fost de 0.09935, ceea ce înseamnă că modelul a făcut erori în aproximativ 10% din cazuri.

Aplicând modelul la setul de testare, am obținut o matrice de confuzie care ne-a permis să evaluăm performanța modelului. Acuratețea modelului a fost de 78%, ceea ce înseamnă că modelul a făcut predicții corecte pentru 78% din cazuri. Acest procent reprezintă o îmbunătățire semnificativă față de rata de bază de 63.5%, care este proporția celei mai mari clase din setul de date.

Valoarea Kappa a fost de 0.5226, indicând o concordanță moderată între predicțiile modelului și valorile reale. Sensibilitatea modelului (proporția de predicții corecte pentru "High") a fost de 68.49%, în timp ce specificitatea (proporția de predicții corecte pentru "Low") a fost de 83.46%. Aceasta înseamnă că

modelul a avut un randament mai bun în predicția perioadelor de absență "Low" comparativ cu perioadele "High".

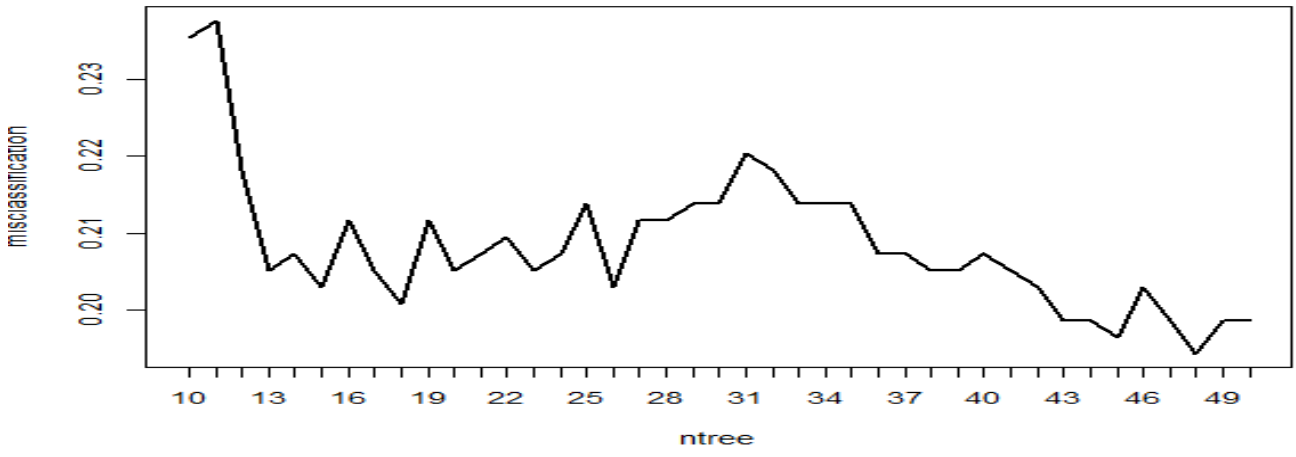
În plus, valoarea P a testului McNemar a fost de 0.8802, sugerând că nu există o diferență semnificativă între numărul de fals pozitive și fals negative, ceea ce indică o performanță echilibrată a modelului.

Arbori Avansati Bagging

Bagging, sau bootstrap aggregating, implică crearea mai multor modele de arbori de decizie și utilizarea predicției cele mai frecvente pentru a reduce variabilitatea și pentru a preveni supraadaptarea.
Conceptul de bagging este bazat pe două componente principale: bootstrap și agregare.
Bootstrap este o tehnică ce ne permite să simulăm procesul de generare a noi seturi de date prin eșantionare repetată cu înlocuire din setul de date de instruire iar bservațiile care nu sunt folosite în timpul acestui proces sunt utilizate pentru validare.

Începem prin a construi un model de bagging inițial, utilizând funcția bagging(), acest model foloeste implicit 25 de "bag-uri". Utilizăm setul de date absente_train pentru a instrui modelul și utilizăm toate variabilele ca predictori pentru AbsenteeismTimeInHours iar prin argumentul coob = TRUE specificăm că dorim să folosim eșantionarea out-of-bag pentru a estima eroarea de predicție. În cele din urmă, evaluăm acuratețea acestui model, însă scopul acestuia este doar de a oferi o perspectivă între un arbore cu bag-uri standard și unul cu un număr specific.

Apoi, explorăm cum variază eroarea de predicție în funcție de numărul de "bagguri". Definim o secvență de numere de la 10 la 50, cu pasul 1, pe care le vom utiliza ca număr de "bagguri". Pentru fiecare număr din această secvență, construim un model de bagging și calculăm eroarea de predicție, pe care o stocăm în vectorul misclassification pe care il vizualizam sub forma de graphic si de unde extragem numarul de baguri cu cea mai mica eroare de predictie.



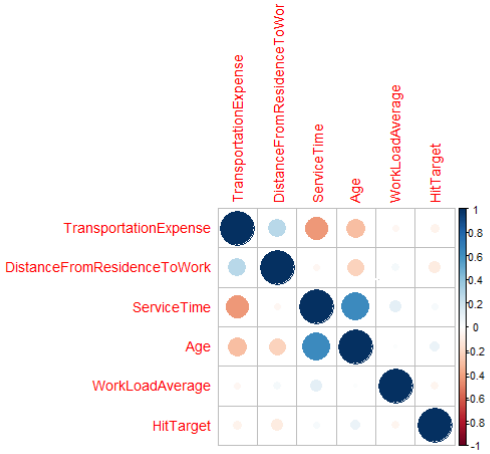
În ultima parte construim un nou model de bagging folosind 48 de "bagguri", număr pe care l-am identificat ca fiind cel care stabilizează rata de eroare. Acest număr a fost selectat în urma analizei graficului de mai sus, iar ca rezultat obtine o acuratete egala cu 82% iar in cee ace priveste Sensibilitatea (capacitatea de a detecta corect clasa "High") este de 75.34%, iar specificitatea (capacitatea de a detecta corect clasa "Low") este de 85.83%. Se observa caci fata de modelele antecedente Sesibilitatea a scazut iar specificitatea a crescut usor.

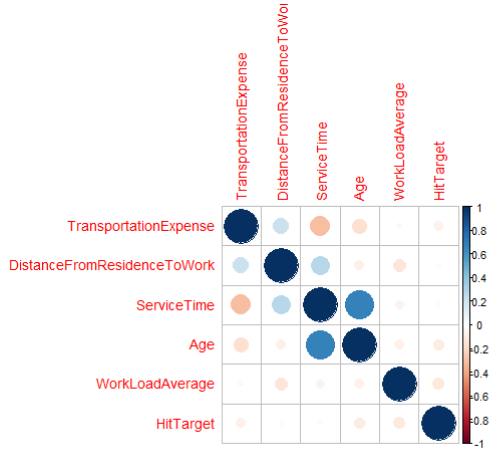
```
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High   55  18
      Low    18 109

               Accuracy : 0.82
                 95% CI : (0.7596, 0.8706)
    No Information Rate : 0.635
    P-Value [Acc > NIR] : 8.113e-09

                  Kappa : 0.6117

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.7534
            Specificity : 0.8583
         Pos Pred Value : 0.7534
         Neg Pred Value : 0.8583
             Prevalence : 0.3650
         Detection Rate : 0.2750
   Detection Prevalence : 0.3650
      Balanced Accuracy : 0.8058

       'Positive' Class : High
```

3.2 Naive Bayse

Algoritmul Naive Bayes este un instrument simplu dar eficient de învățare automată, folosit pentru a clasifica date. Acesta se bazează pe ideea că fiecare caracteristică a datelor este independentă și contribuie în mod egal la rezultat iar pentru a îmbunătăți acuratețea acestui algoritm, adesea se folosesc metode ca netezirea Laplace sau estimatorii de tip kernel.

Pentru început, dorim să verificăm dacă există corelații între atributele numerice, astfel încât am creat două grafice, fiecare pentru una dintre cele două stări ale atributului AbsenteeismTimeInHours. Prima figură reprezintă corelațiile pentru angajații care au un grad de absenteism ridicat (High), iar a doua pentru cei cu grad scăzut (Low).





Se pot observa corelații între anumite atribute, însă am decis să continuăm analiza iar in prima fază, ne-am ocupat de împărțirea datelor, unde variabila "x" reprezintă setul de caracteristici (features), iar variabila "y" reprezintă variabila țintă. De asemenea, am definit și controlul de antrenare folosind validarea încrucișată cu 10 fold-uri.

Pentru început, am antrenat un model (naive_b1) folosind metoda Naive Bayes pe setul de antrenament și am obținut o acuratețe medie egală cu 78.83% insă am preferat să nu testăm în continuare și pe setul de test, ci să definim un grid de căutare pentru a testa diferite stări ale modelului Naive Bayes. Grid-ul de căutare conține opțiuni pentru utilizarea kernelului sau nu, metoda de netezire Laplace și ajustarea cu valori într-o anumită secvență.
Ajunși în acest punct, am antrenat un nou model (naive_b2) folosind grid-ul definit și s-a putut observa o creștere a mediei acurateții la 80.35%, demonstrându-se o performanță îmbunătățită. Pentru aprofundare, am decis să afișăm și top 10 combinații ale setărilor făcute anterior, prezentate în figura următoare.

usekernel	adjust	Accuracy	Kappa
FALSE	0	0.7883441	0.5460144
FALSE	1	0.7883441	0.5460144
FALSE	2	0.7883441	0.5460144
FALSE	3	0.7883441	0.5460144
FALSE	4	0.7883441	0.5460144
FALSE	5	0.7883441	0.5460144
TRUE	0	NaN	NaN
TRUE	1	0.7882054	0.5288307
TRUE	2	0.8143386	0.5945090
TRUE	3	0.8142923	0.5942900
TRUE	4	0.8099907	0.5809331
TRUE	5	0.8100370	0.5770352

In final am aplicat modelul optimizat (naive_b2) pe setul de test obtinand urmatoarele rezultate: Acuratetea de 82% , o sensibilitate de 73.97% care indică că modelul are capacitatea de a identifica corect 73.97% din cazurile cu grad de absenteeism ridicat (clasa "High") iar specificitatea de 86.61% indică capacitatea modelului de a identifica corect 86.61% din cazurile cu grad de absenteeism scăzut (clasa "Low").

3.2 Compararea rezultatelor obtinute

Drept puncte de comparațive am ales Sensitivitatea, Specificitatea și Acuratețea, deoarece acestea reprezintă măsuri cheie în evaluarea performanței unui model de clasificare. Acestea sunt utile pentru a înțelege cât de bine funcționează modelul în predicția celor două clase.

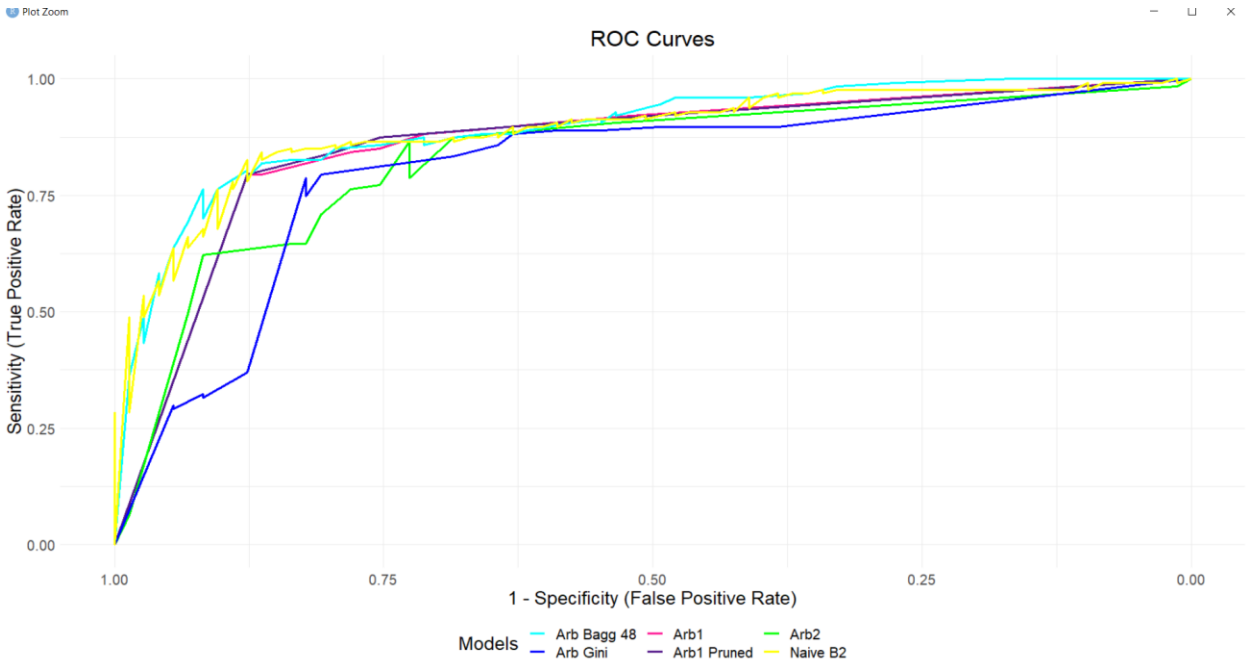
1. Sensibilitatea (cunoscută și sub numele de Recall sau True Positive Rate) măsoară proporția de exemple pozitive reale care au fost identificate correct fiind o măsură esențială atunci când costul ratării exemplelor pozitive este mare.Se poate observa cel mai bine in exemplele cu diagnosticarea medicală, vrem să fim siguri că detectăm cât mai multe cazuri pozitive posibile.

2. Specificitatea (denumită și rata de adevăr negativ) măsoară proporția de exemple negative reale care au fost identificate correct fiind importantă atunci când vrem să fim siguri că modelul nostru nu clasifică incorect exemplele negative. Dacă costul falselor alarme este unui important atunci aceasta devine măsuratoarea cel mai importantă.

3. Acuratețea este cea mai intuitivă măsură de performanță fiind definită ca proporția de predicții corecte (atât pozitive, cât și negative) din totalul de predicții însă acuratetea este o măsură bună când setul de date este echilibrat.

Model	Sensitivitate	Specificitate	Acuratete
Arbore Simplu (arb1)	75.34%	85.04%	81.5%
Arbore Taiat (arb1_pruned)	79.45%	84.25%	82.5%
Arbore CP=0	72.6%	78.74%	76.5%
Arbore Gini	68.49%	83.46%	78%
Arbori Avansati Bagging	75.34%	85.83%	82%
Naive Bayse	73.97%	86.61%	82%

Pentru o mai bună vizualizare a diferențelor, am decis să folosim Curba ROC (Receiver Operating Characteristic), aceasta fiind un instrument de evaluare important pentru a înțelege performanța unui model de clasificare. În graficul nostru ROC, axa orizontală (x) reprezintă rata falselor pozitive (1 – specificitate), în timp ce pe axa verticală (y) se află rata adevăratelor pozitive (sensibilitate). Astfel, fiecare punct de pe curba reprezintă un anumit prag. Cu cât modelul nostrum se apropie de colțul din stânga sus, este considerat un model performant, iar pe parcurs ce se apropie de diagonala, ajunge din ce în ce mai aproape de modelarea random.



Pe baza analizei datelor prezentate mai sus, constatăm că modelul superior, având ca principal criteriu de selecție sensibilitatea, este Arborele Tăiat (arb1_pruned). Acesta se distinge prin sensibilitatea maximă de 79.45%, indicând o performanță deosebită în identificarea angajaților cu absenteism ridicat si chiar dacă arb1_pruned nu prezintă cea mai mare specificitate dintre modele (84.25%, comparativ cu 86.61% pentru Naive Bayse), specificitatea sa rămâne competitive astfel se crează un echilibru între sensibilitate și specificitate care duce la obținerea celei mai mari acurateți dintre modele.

Alte concluzii:

• Se poate observa că Naive Bayes și Arborii avansați bagging au cele mai mari performanțe în ceea ce privește specificitatea, indicând că aceste modele au o eficiență mare în identificarea corectă a exemplelor negative și, în același timp, au un risc mic de a emite alarme false.

• Interesant de remarcat este faptul că Arborele Simplu (arb1) și Arborele CP=0, deși prezintă rezultate scăzute în ceea ce privește Sensitivitatea și Specificitatea, acestea au o acuratețe destul de bună, ceea ce arată că, în ciuda unor erori, ele pot oferi totuși predicții bune într-un număr considerabil de cazuri.

• Se observă că arborii bagging și Naive Bayes sunt cele mai echilibrate modele, astfel sunt considerate robuste deoarece ele nu tind să privilegieze nici clasa pozitivă, nici cea negativă.

4. Concluzia

Analizând rezultatele prezentate și interpretand fiecare model în contextul problemei absenteismului la locul de muncă, putem trage următoarele concluzii:

1. Factori de influență asupra absenteismului:

Variabilele cu cea mai mare influență asupra absenteismului sunt Motivul absenței și Costurile transportului, care sunt destul de intuitive, deoarece motivele unor lungi perioade de absență sunt reprezentate de probleme personale (ex: boli) și costul transportului, care poate fi un impediment pentru prezența la locul de muncă. Acești factori sunt logici și direct legați de absenteism. Alte variabile ar fi și Timpul petrecut la muncă, luna sau ziua săptămânii, acestea contribuind la rafinarea predicțiilor.

2. Acuratețea anticipării nivelului de absenteism:

Modelul Arbore Tăiat a oferit cea mai bună acuratețe în predicția nivelului de absenteism, cu un procent de 82.5%. Astfel, folosind datele disponibile și aplicând acest model, putem anticipa nivelul de absenteism cu o acuratețe relativ ridicată.

Pe de altă parte, este de remarcat că factorii care influențează absenteismul sunt complecși și interconectați, astfel este imposibil să captăm toate nuanțele acestor relații. Așadar, analiza noastră poate fi văzută mai degrabă ca un ghid pentru înțelegerea generală a tendințelor în absentarea de la locul de muncă și pentru identificarea potențialelor domenii de intervenție.