Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Specializarea: Cibernetică Economică

INTELIGENȚĂ COMPUTAȚIONALĂ ÎN ECONOMIE

Predictia Acordarii Unui Imprumut

O Abordare Bazată pe Clusterizare și Algoritmi de

Machine Learning



Cadru didactic coordonator:

Cadru asociat drd. DOMENTEANU ADRIAN

Student : Stafie Diana- Flavia

Anul III, Grupa 1081, Seria B

**Introducere**

**Descrierea datelor si scopul cercetarii**

Setul de date pentru aprobarea împrumuturilor conține înregistrări financiare și informații asociate, utilizate pentru a determina eligibilitatea indivizilor pentru obținerea de împrumuturi de la diverse instituții financiare. Acesta include factori precum scorul CIBIL, venitul, statutul de angajare, termenul și suma împrumutului, valoarea activelor și statutul împrumutului. Scopul principal al acestui set de date este de a permite dezvoltarea și evaluarea modelelor de învățare automată pentru a prezice probabilitatea de aprobare a unui împrumut.

Prin utilizarea acestui set de date, instituțiile financiare pot reduce riscul de neîndeplinire a plăților și pot lua decizii mai informate în privința aprobării cererilor de împrumut. Analiza detaliată a datelor despre aplicanți și împrumuturi permite identificarea modelelor și tendințelor relevante, ceea ce este crucial pentru elaborarea politicilor de creditare și gestionarea riscului. Proiectul de față oferă oportunitatea de a aplica diverse tehnici de machine learning pentru a îmbunătăți procesul de luare a deciziilor și pentru a dezvolta strategii de intervenție eficiente pentru clienții cu risc ridicat. Acest lucru nu numai că ajută la menținerea stabilității financiare a instituțiilor, dar și la oferirea unor evaluări mai precise și mai echitabile pentru solicitanții de împrumuturi.

Sursa: <https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/code>

VARIABILELE:

Numerice (8):

1. income\_annum – venitul solicitantului.
2. loan\_amount – suma solicitată pentru împrumut.
3. loan\_term – termenul împrumutului (în luni).
4. cibil\_score – scorul CIBIL al solicitantului.

Scorul CIBIL este un punctaj numeric care reflectă istoricul de credit al solicitantului. Un scor CIBIL mai mare indică un istoric de credit pozitiv și o probabilitate mai mică de neîndeplinire a plăților, în timp ce un scor mai mic sugerează un risc mai mare.

1. residential\_assets\_value – Valoarea activelor rezidențiale ale solicitantului

Descriere: Această variabilă reprezintă valoarea totală a proprietăților rezidențiale deținute de solicitant, cum ar fi case, apartamente sau alte tipuri de locuințe. Aceste active pot fi utilizate ca garanție pentru împrumut.

1. commercial\_assets\_value – valoarea activelor comerciale ale solicitantului.

Această variabilă indică valoarea totală a activelor comerciale deținute de solicitant, cum ar fi clădiri de birouri, magazine, sau alte proprietăți comerciale. Aceste active pot fi utilizate pentru evaluarea capacității financiare a solicitantului.

1. luxury\_assets\_value – valoarea activelor de lux ale solicitantului.
2. bank\_assets\_value– valoarea activelor bancare ale solicitantului.

Calitative (4):

1. no\_of\_dependents – numărul de persoane aflate în întreținerea solicitantului.
2. education – nivelul de educație al solicitantului.

* Graduate = 1
* Not Graduate = 0

1. self\_employed – dacă solicitantul este lucrător pe cont propriu.

* Yes = 1
* No = 0

1. loan\_status – statutul cererii de împrumut.

* Approved = 1
* Rejected = 0

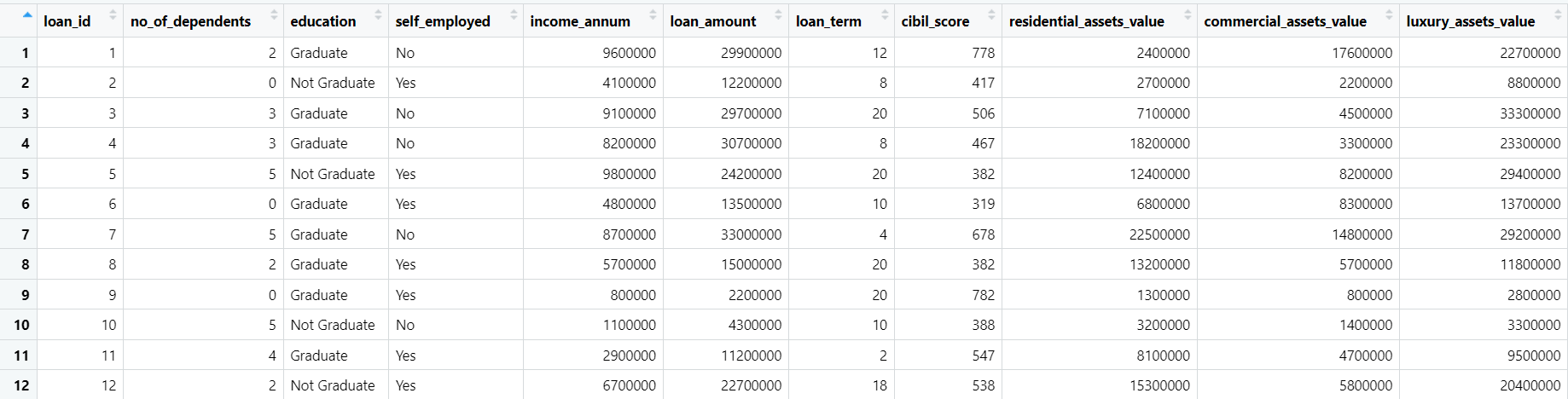


Figure 1: Setul de date cu primele 12 randuri

Am transformat datele într-un format adecvat pentru a fi utilizat într-un model de învățare automată sau într-o analiză statistică. Acest proces este esențial pentru a asigura că toate variabilele din setul de date sunt în formatul corect și pot fi înțelese de algoritmi.( Label Encoding)

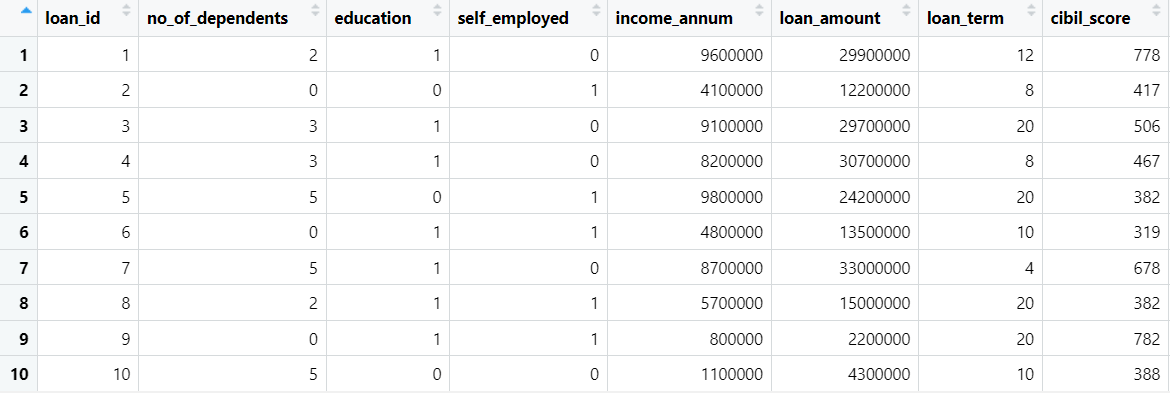


Figure 2:Data Encoding

**Statistici descriptive**

În Figura 3, am aplicat funcția summary pe datele noastre. In prealabil am transformat variabilelor categoriale în factori, fiind un pas important în preprocesarea datelor, în special atunci când lucrezi cu seturi de date care vor fi utilizate în analize statistice sau modele de învățare automată.

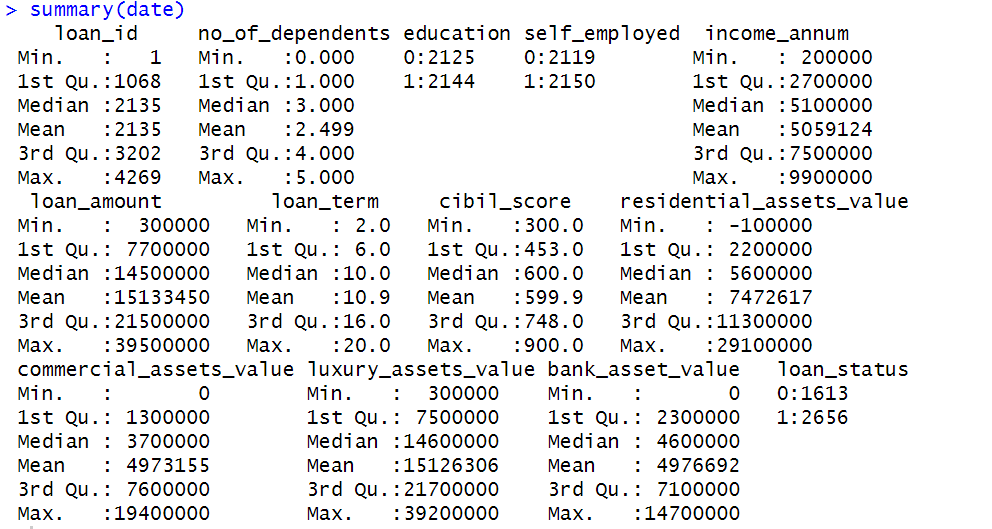


Figura 3: Summary

Analiza acestor variabile oferă o imagine de ansamblu asupra caracteristicilor financiare și demografice ale aplicanților, care sunt esențiale pentru dezvoltarea modelelor de previziune a aprobării împrumuturilor. Înțelegerea acestor variabile și a distribuției lor ajută la construirea unor modele precise de previziune, reducând riscul de neîndeplinire a plăților și îmbunătățind procesul de decizie pentru instituțiile financiare.

Observăm că media pentru venitul anual al aplicanților este de 5.059.124 unități monetare, fiind foarte apropiată de valoarea medianei de 5.100.000 unități monetare, indicând o distribuție aproape normală a datelor. Valoarea minimă a venitului este de 200.000 unități monetare, iar cea maximă este de 9.900.000 unități monetare, arătând o amplitudine destul de mare. Analizând prima cuartilă, rezultă că 25% dintre aplicanți au un venit anual mai mic de 270.000 unități monetare, iar 75% au un venit peste această valoare. Din a treia cuartilă reiese faptul că 75% dintre aplicanți au un venit anual sub 7.500.000 unități monetare și 25% au un venit peste această valoare. Aceasta sugerează o variație semnificativă în veniturile aplicanților, ceea ce ar putea influența deciziile de aprobare a împrumuturilor.

Suma solicitată pentru împrumut variază între 300.000 și 39.500.000 de unități monetare, cu o valoare mediană de 14.500.000, indicând o gamă largă de sume care pot influența riscul asociat cu împrumuturile. Termenul împrumutului variază între 2 și 20 luni, cu o valoare mediană de 10 luni, ceea ce poate afecta capacitatea de rambursare a aplicanților și riscul de neîndeplinire a plăților. Scorul CIBIL, variind între 300 și 900, cu o valoare mediană de 600, este un indicator critic al riscului de credit, aplicanții cu scoruri mai mici având o probabilitate mai mare de a nu îndeplini plățile.

Valoarea activelor rezidențiale variază între -100.000 și 29.100.000 de unități monetare, cu o valoare mediană de 5.600.000, fiind utilizată ca garanție pentru împrumuturi, influențând astfel deciziile de aprobare. Valoarea activelor comerciale variază între 0 și 19.400.000 de unități monetare, cu o valoare mediană de 3.700.000, fiind importantă pentru evaluarea capacității financiare a aplicanților. Valoarea activelor de lux, variind între 300.000 și 39.200.000 de unități monetare, cu o valoare mediană de 1.460.000, oferă informații despre stabilitatea financiară a aplicanților și poate influența deciziile de aprobare. Valoarea activelor bancare variază între 0 și 14.700.000 de unități monetare, cu o valoare mediană de 4.600.000, fiind un indicator al lichidității și al capacității de rambursare a împrumuturilor.

Statutul cererii de împrumut arată că 1265 de împrumuturi au fost aprobate și 2163 respinse, indicând o rată de respingere ridicată, ceea ce sugerează criterii stricte de aprobare sau un profil de risc ridicat al aplicanților.

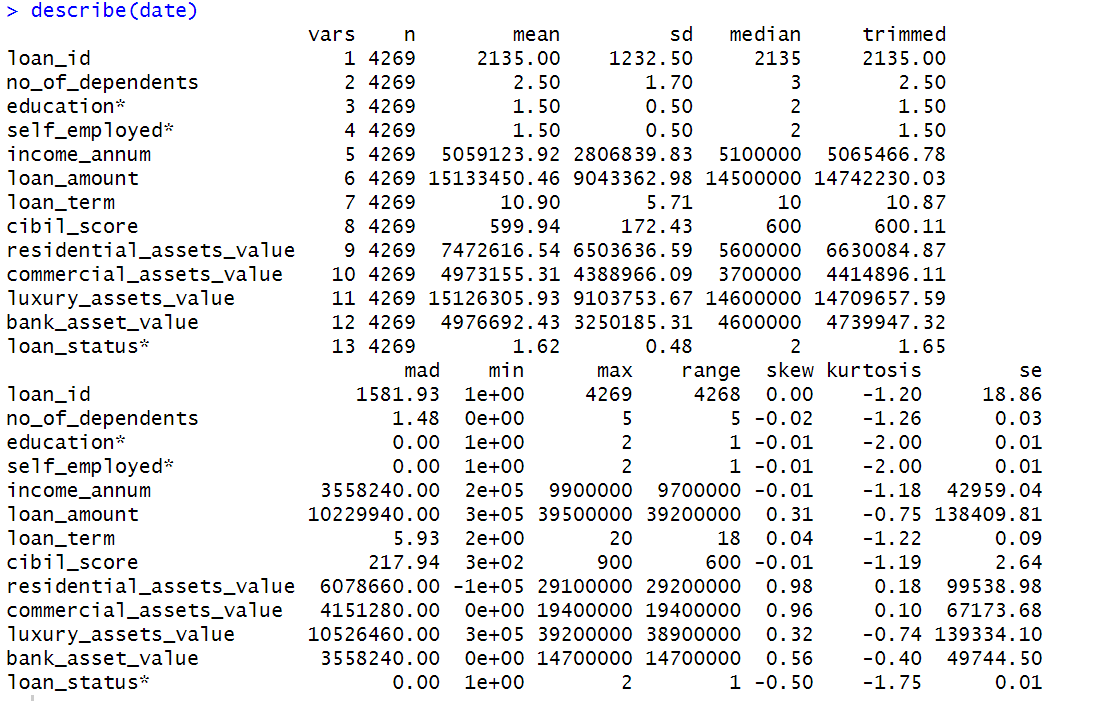


Figura 4: Describe

Interpretările coeficienților de asimetrie și boltire oferă o imagine clară asupra distribuției variabilelor din setul de date, ajutând la înțelegerea caracteristicilor aplicanților și a riscului asociat cu împrumuturile. Asimetriile ușoare la dreapta pentru variabilele financiare indică prezența unor valori mari care pot influența deciziile de aprobare a împrumuturilor. Distribuțiile platocurtice sugerează că datele sunt relativ uniforme, fără valori extreme, ceea ce este important pentru evaluarea riscului și dezvoltarea modelelor predictive precise. Aceste informații sunt esențiale pentru îmbunătățirea proceselor de luare a deciziilor în cadrul instituțiilor financiare.

Venitul anual:

Coeficientul de asimetrie pentru venitul anual are o valoare de -0,01, foarte aproape de 0, indicând o distribuție aproape simetrică a valorilor veniturilor. Aceasta înseamnă că veniturile aplicanților sunt distribuite uniform în jurul valorii medii, fără o tendință pronunțată spre stânga sau dreapta. Coeficientul de boltire (kurtosis) pentru venitul anual este de -1,18, indicând o distribuție platocurtică. Aceasta sugerează că datele nu prezintă valori extreme și distribuția este mai aplatizată decât o distribuție normală, ceea ce înseamnă că majoritatea valorilor sunt concentrate în jurul mediei.

Suma împrumutată:

Coeficientul de asimetrie pentru suma împrumutată are o valoare de 0,31, indicând o ușoară asimetrie la dreapta. Aceasta înseamnă că există câteva valori mari ale împrumuturilor care trag media în sus, indicând prezența unor împrumuturi foarte mari comparativ cu restul. Coeficientul de boltire (kurtosis) pentru suma împrumutată este de -0,75, indicând o distribuție platocurtică.

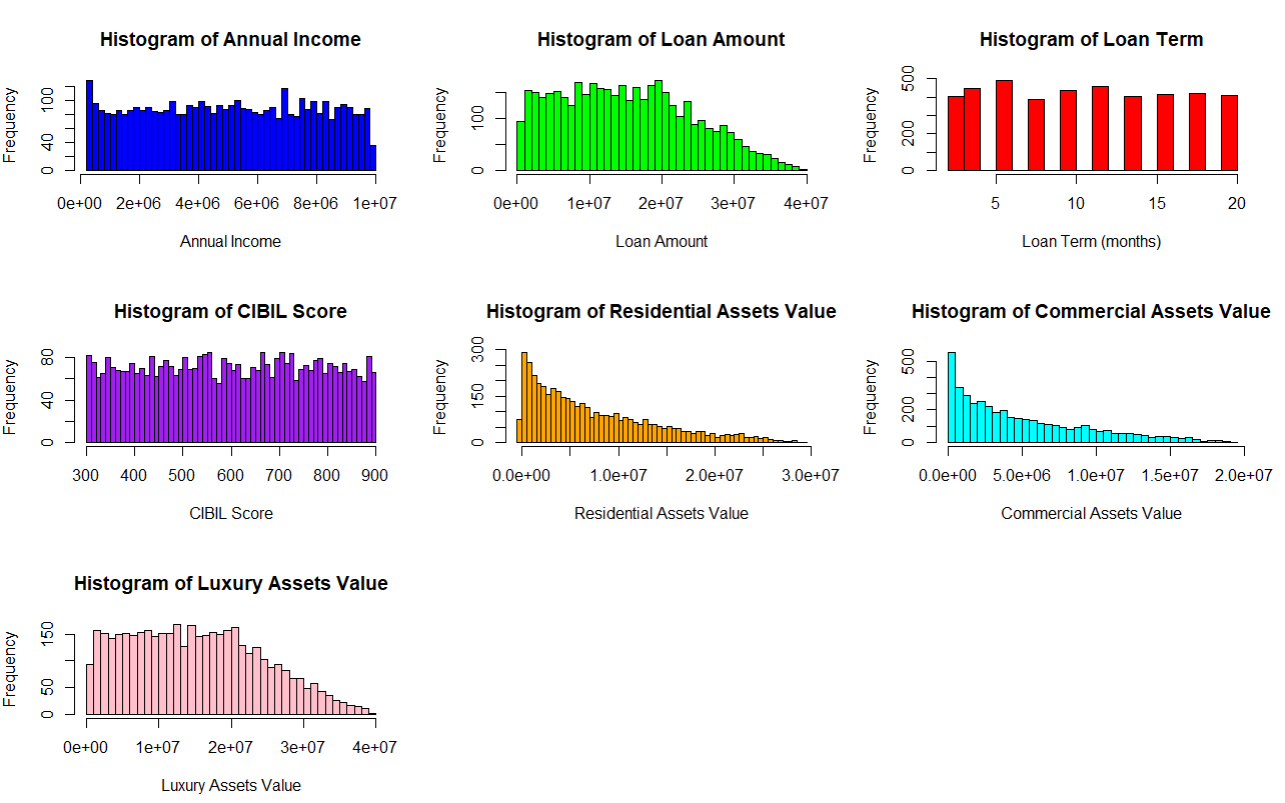


Figura 5: Histograme

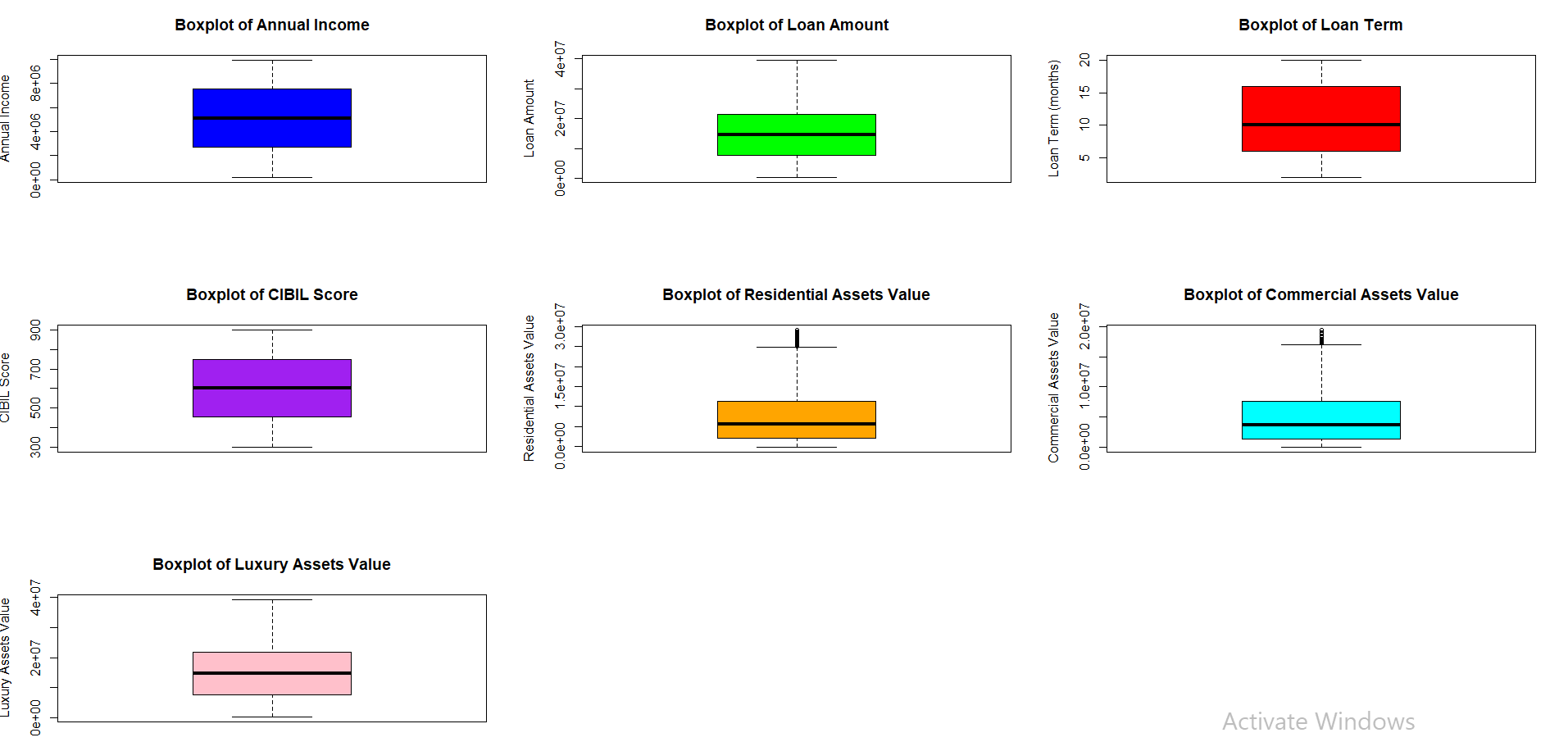


Figura 6: Boxplot-uri

În Figurile 5 si 6 observăm că afirmațiile de mai sus, în privința coeficientului de asimetrie și a celui de boltire sunt evidențiate și în histograma și boxplot-ul venitului si a sumei imprumutate

Venitul anual :

Distribuția veniturilor anuale, vizualizată prin histogramă, este destul de uniformă, fără o concentrație evidentă într-o anumită zonă. Frecvențele sunt similare în majoritatea intervalelor, sugerând o variație constantă a veniturilor între solicitanți. Boxplot-ul confirmă această uniformitate, arătând că veniturile anuale sunt distribuite relativ simetric. Valoarea mediană este centrată, iar whiskers se extind uniform în ambele direcții, indicând o variație echilibrată. Nu există valori extreme evidente, ceea ce indică o stabilitate relativă a veniturilor.

Suma împrumutată :

Distribuția sumelor împrumutate, conform histogramelor, arată o frecvență mai mare pentru sumele mai mici, cu o scădere graduală pe măsură ce sumele cresc. Majoritatea împrumuturilor se situează în intervalul de 0 până la aproximativ 10.000.000 unități monetare, indicând o preferință pentru împrumuturi mai mici. Boxplot-ul arată că mediană este deplasată ușor spre partea inferioară a intervalului, confirmând că majoritatea împrumuturilor sunt pentru sume mai mici. Whiskers se extind mai mult spre valorile mari, sugerând existența unor împrumuturi semnificativ mai mari, dar fără outliers notabili, ceea ce indică o distribuție uniformă a sumelor împrumutate.

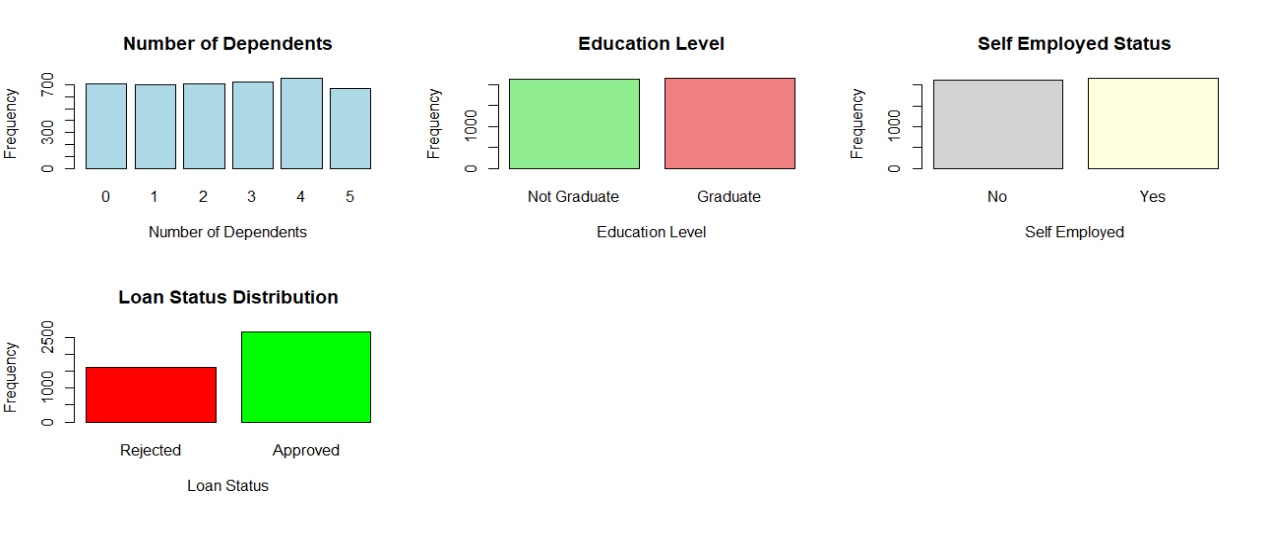


Figura 7: Barplot-uri

1. Numărul de persoane aflate în întreținere

Diagrama de bare arată distribuția numărului de persoane aflate în întreținerea solicitanților. Frecvențele sunt relativ similare pentru toate categoriile, indicând că numărul de persoane aflate în întreținere este distribuit uniform între solicitanți. Aproximativ 700 de solicitanți au între 1 și 4 persoane în întreținere, în timp ce categoriile 0 și 5 au frecvențe puțin mai mici. Aceasta sugerează că majoritatea solicitanților au o responsabilitate familială moderată, ceea ce ar putea influența capacitatea lor de a rambursa împrumuturile.

2. Nivelul de educație

Diagrama de bare pentru nivelul de educație arată o distribuție aproape egală între solicitanții care sunt absolvenți și cei care nu sunt absolvenți. Acest lucru sugerează că setul de date include o reprezentare echilibrată a persoanelor cu și fără studii superioare. Nivelul de educație poate influența capacitatea de a obține un venit stabil și, implicit, capacitatea de a rambursa împrumuturile. O educație superioară este adesea asociată cu venituri mai mari și un risc mai mic de neîndeplinire a plăților.

3. Statutul de lucrător pe cont propriu

Diagrama de bare pentru statutul de lucrător pe cont propriu arată o distribuție aproape egală între solicitanții care sunt angajați și cei care sunt lucrători pe cont propriu.

4. Statutul cererii de împrumut

Diagrama de bare pentru statutul cererilor de împrumut arată că mai multe cereri au fost aprobate decât respinse. Aproximativ 2.500 de cereri au fost aprobate, în timp ce aproximativ 1.700 au fost respinse. Aceasta indică o rată de aprobare relativ ridicată. Această informație este crucială pentru evaluarea politicilor de acordare a împrumuturilor ale instituțiilor financiare și pentru dezvoltarea unor modele predictive care să identifice factorii cheie care influențează deciziile de aprobare a împrumuturilor.

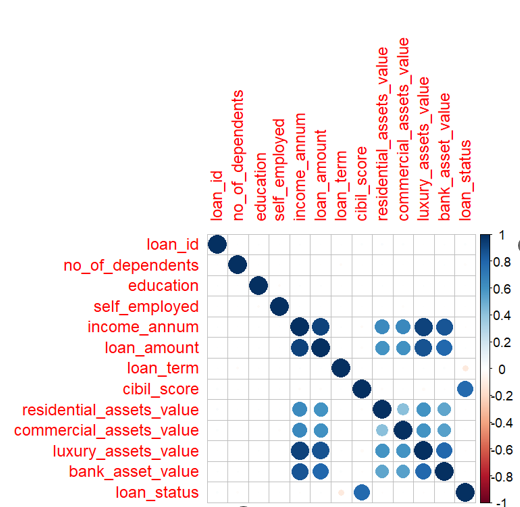


Figure 8: Corelatii

* loan\_amount și income\_annum par să aibă o corelație pozitivă moderată, ceea ce sugerează că persoanele cu venituri anuale mai mari tind să solicite împrumuturi mai mari.
* self\_employed și loan\_amount par să aibă o corelație negativă ușoară sau slabă, indicând că persoanele care sunt self-employed ar putea solicita împrumuturi de valori diferite comparativ cu cei care nu sunt self-employed.
* scorul CIBIL este pozitiv corelat cu starea împrumutului (loan\_status). Un scor CIBIL mai mare indică probabilitatea mai mare ca împrumutul să fie aprobat sau în stare bună.

1. UTILIZAREA METODEI “CLUSTERIZARE FUZZY”

1.Fuzzy clustering cu functia “cmeans”

Adaptam setul nostrum de date, astfel incat toate variabilele sa fie numerice, urmand sa alcatuim o custerizare cu 3 clustere, 100 iteratii si m=2 gradul de fuzzificare.

Centroizii (mediile) celor 3 clustere sunt :

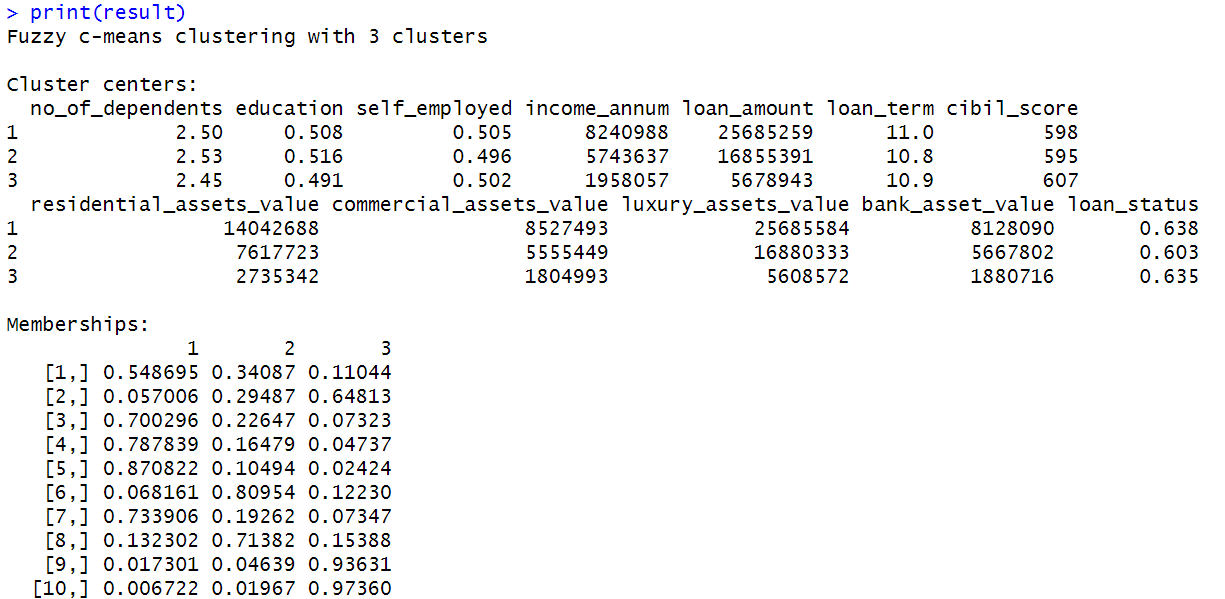


Figure 9: Rezultatele clusterizarii

* Clusterul 1: Are cele mai mari valori medii pentru income\_annum, loan\_amount, loan\_term, residential\_assets\_value, commercial\_assets\_value, luxury\_assets\_value, și bank\_asset\_value. Loan\_status are o valoare medie de 0.638, ceea ce sugerează un status al împrumutului relativ bun.
* Clusterul 2: Are valori medii intermediare pentru majoritatea variabilelor. Loan\_status este ușor mai scăzut decât clusterul 1, sugerând o performanță ușor mai slabă a împrumuturilor.
* Clusterul 3: Are cele mai mici valori medii pentru income\_annum, loan\_amount, loan\_term, residential\_assets\_value, commercial\_assets\_value, luxury\_assets\_value, și bank\_asset\_value. Loan\_status este similar cu cel al clusterului 2, sugerând o performanță a împrumuturilor comparabilă.

Primul membru are cea mai mare apartenență la clusterul 1- 0.548695, dar are și o apartenență semnificativă la clusterul 2 - 0.34087 și o apartenență mai mică la clusterul 3 - 0.11044

Daca luam pe fiecare linie gradul maxim de apartenenta, decidem carui cluster apartine obiectul respective: primul aparine clusterului 1, al doilea obiect clusterului 3, urmatoarele trei clusterului 1, etc.

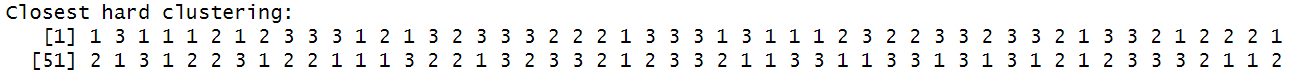


Figure 10: A partenenta obiectelor la clustere

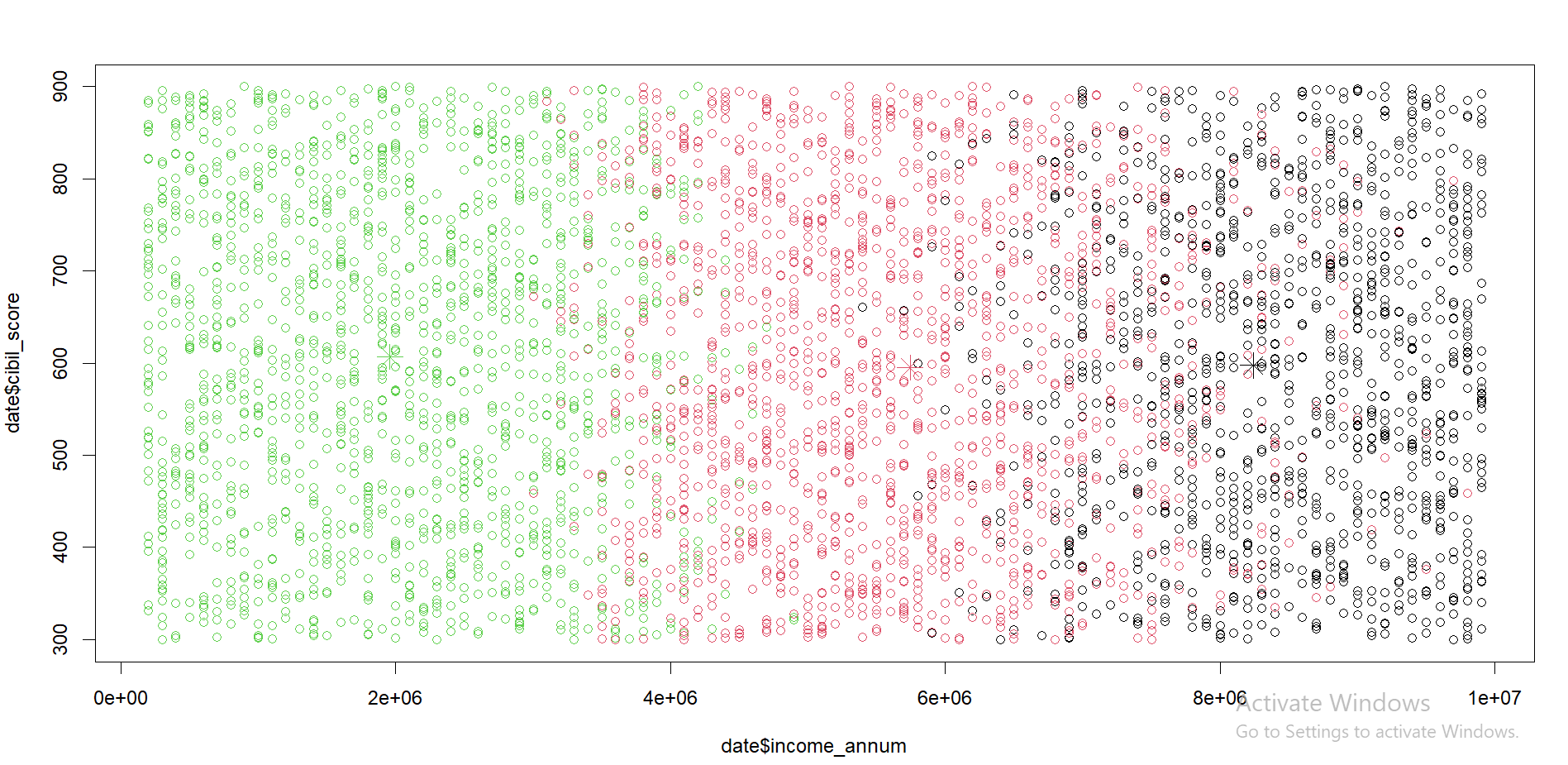


Figure 11: Graficul clusterizarii in functie de scorul cibil si venit

Apar marcati cu \* (pch=8) centroizii celor 3 clustere.

2. Fuzzy clustering cu biblioteca “cluster”

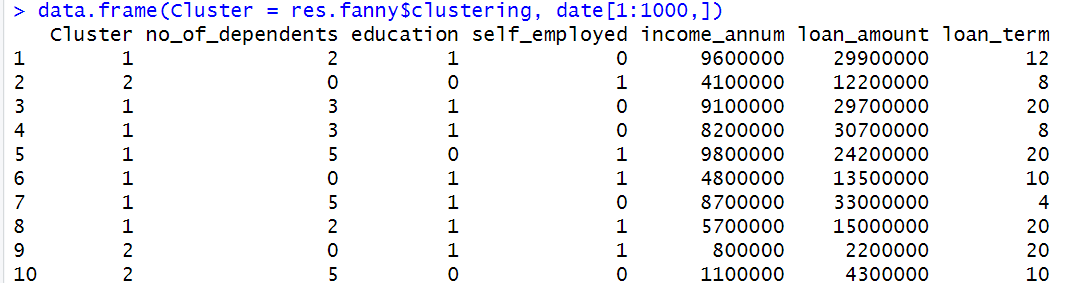


Figure 12: Data frame clustere

Observatii:

* Membrii din clusterul 1 tind să aibă veniturile și sumele împrumutate mai mari comparativ cu clusterul 2.
* Membrii din clusterul 2 par să aibă un nivel mai scăzut de educație și veniturile anuale mai mici.
* Clusterul 3 conține membri cu diverse niveluri de educație și valori variate ale veniturilor și sumelor împrumutate.

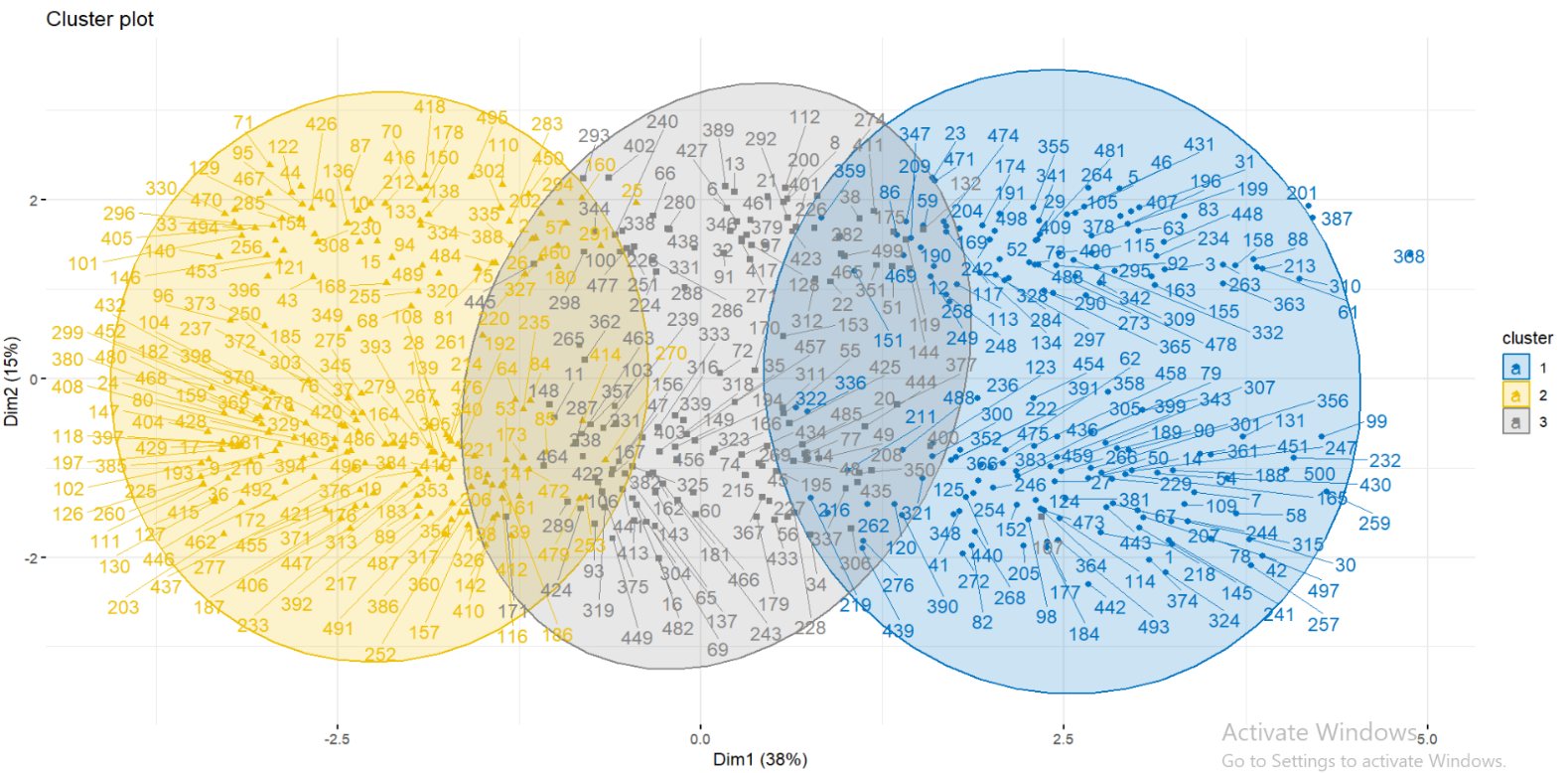


Figure 13: Graficul apartenentelor la clustere dupa PCA

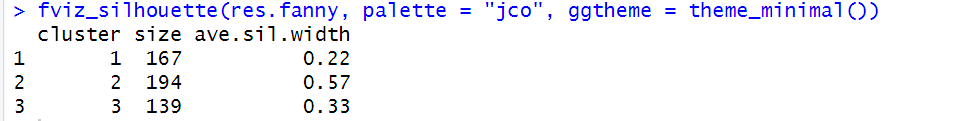


Figure 14: Determinare coeficientilor silueta

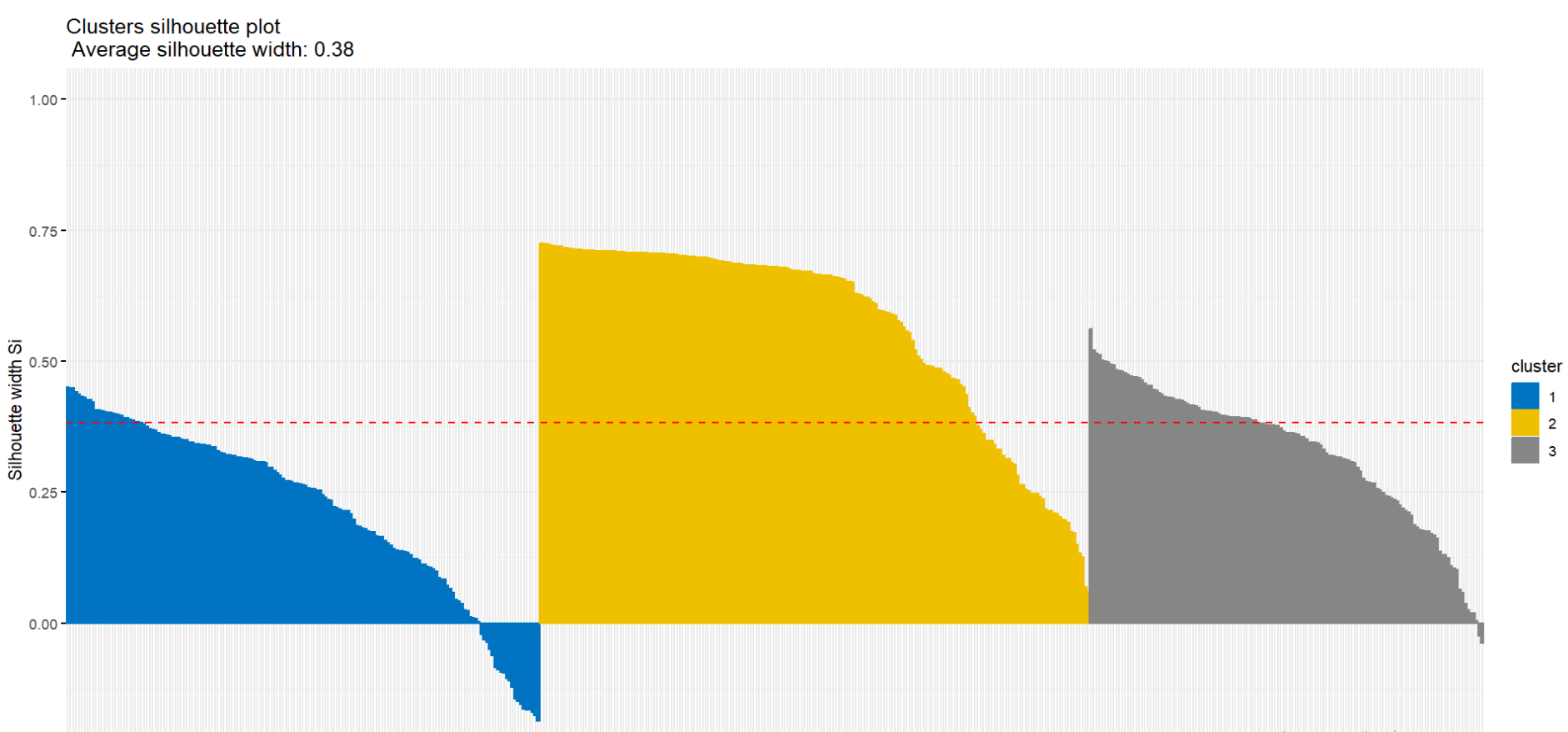


Figure 15: Silhouette plot

Observaii:

Graficul siluetă (Silhouette Plot) afișat oferă informații despre calitatea clusterizării realizate prin algoritmul de clustering. Graficul siluetă evaluează cât de bine sunt grupate punctele de date în clusterele lor respective.

Average silhouette width: 0.38 -- O valoare a lățimii medii a siluetei aproape de 1 indică o clusterizare bună, în timp ce o valoare aproape de 0 indică clustere care se suprapun sau nu sunt bine definite. În acest caz, valoarea de 0.38 sugerează o calitate moderată a clusterizării.

Clusterul 2 este cel mai bine definit, ceea ce indică faptul că datele din acest cluster sunt cele mai omogene și bine separate de celelalte clustere.

Există o nevoie potențială de a revizui sau ajusta clustering-ul pentru a îmbunătăți separarea și omogenitatea membrilor în clusterele 1 și 3.

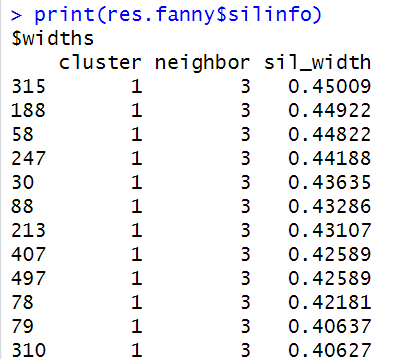


Figure 16: Determinam silueta individuala a fiecarui obiect

Observația 315 are o silhouette width de 0.45009, ceea ce este o valoare pozitivă moderat-înaltă. Acest lucru sugerează că această observație este bine grupată în Clusterul 1, fiind relativ departe de Clusterul 3. Aceasta indică o bună apartenență la Clusterul 1, cu o separare clară de Clusterul 3.

1. Exemplificarea algoritmilor de regresie logistică binomială și multinomială

1.Regresia logistică binomial

Variabila dependenta: loan\_status

Variabile independente: loan\_term si cibil\_score

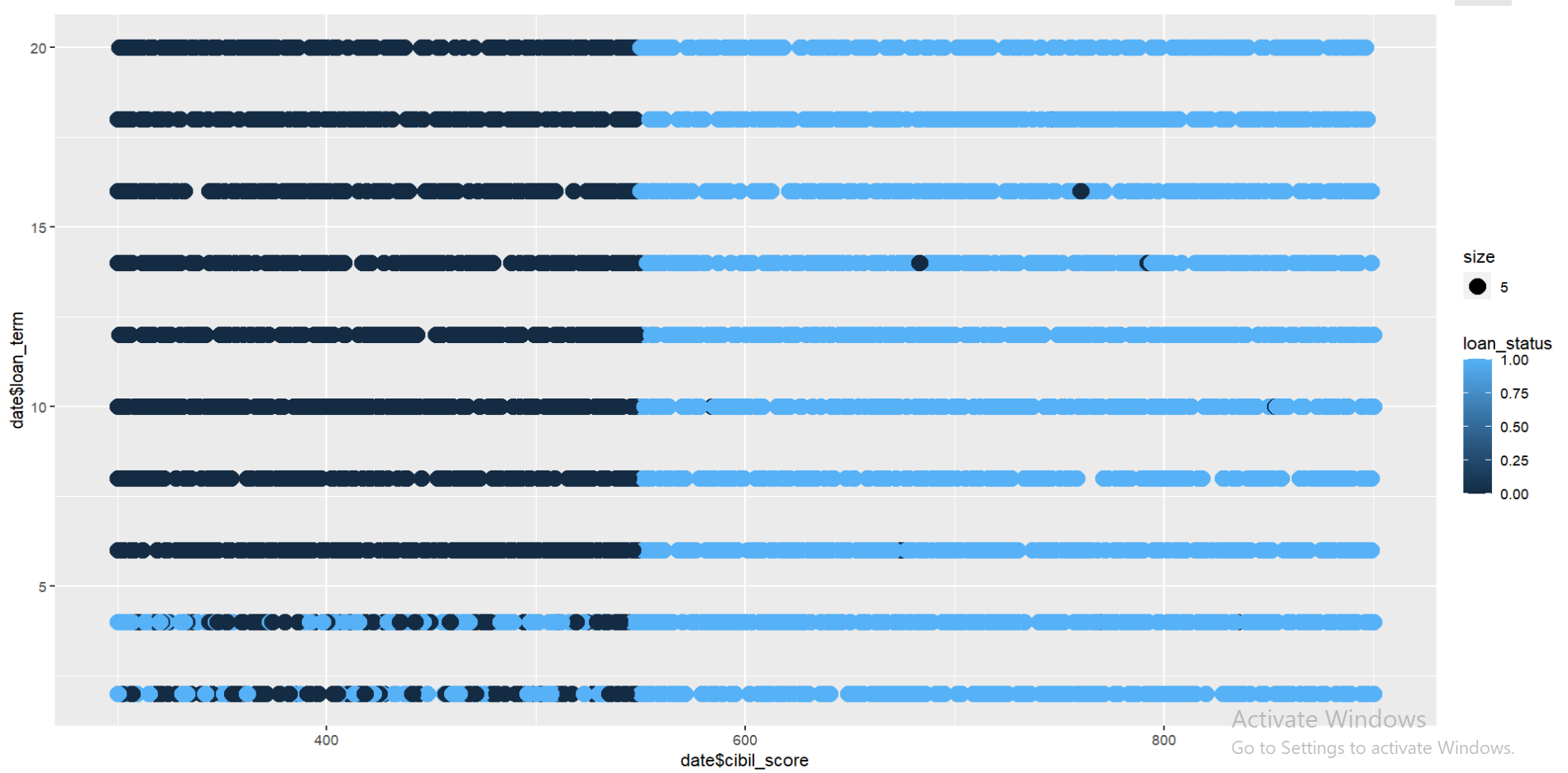


Figure 17: Plot cibil\_scor si loan\_term

Majoritatea împrumuturilor aprobate (albastru deschis) tind să aibă scoruri CIBIL mai mari (aproximativ peste 600), indiferent de termenul împrumutului.

Împrumuturile neaprobate sau cu probabilitate mai mică de aprobare (culori mai închise) sunt distribuite pe întreaga gamă a scorurilor CIBIL, dar par să fie mai frecvente la scoruri CIBIL mai mici (sub 600).

Transformam variabila dependenta in variabila factor:

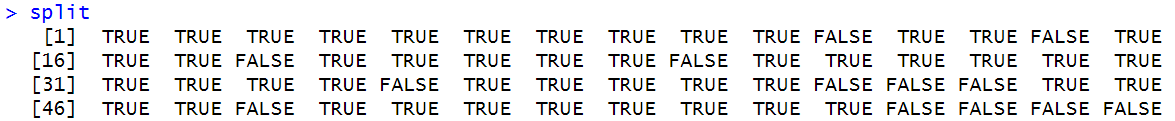


Figure 18: Split

Am împărțit setul de date în două subseturi 80% (setul de antrenare) si 20% (setul de testare) și am refăcut graficul anterior.



Figure 19: Plot cibil\_score si loan\_term

Regresia logistica se construiește cu funcția “glm” - general linear model.

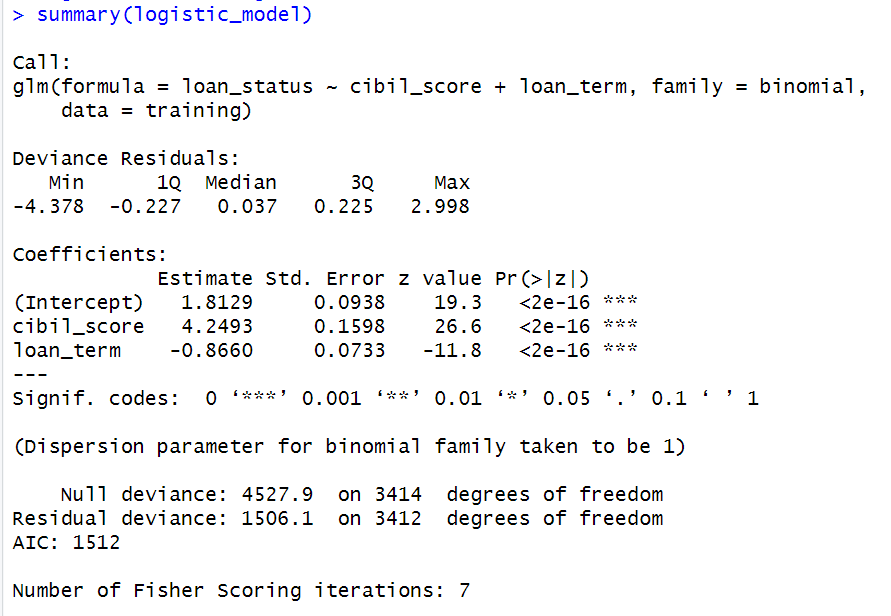


Figure 20: Regresia logistica cu glm

Ecuația de regresie logistică binomială:

logit(p) = ln(p / (1 - p)) = 1.8129 + 4.2493 \* cibil\_score - 0.8660 \* loan\_term

unde: p = probabilitatea ca împrumutul să fie aprobat (loan\_status = 1)

1 - p = probabilitatea ca împrumutul să nu fie aprobat (loan\_status = 0)

Devianța nulă indică cât de bun este răspunsul prognozat de un model în care apare numai termenul liber.

Devianța reziduală (a modelului) indică cât de bun este răspunsul prognozat de un model în care se adaugă variabilele independente.

Devianța reziduală este 1506.1, mai mică decât devianța nulă de 4527.9. Aceasta sugerează că modelul curent, care include variabilele independente, este mult mai adecvat decât modelul nul.

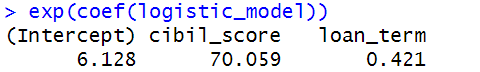


Figure 21: Coeficenti model

* **cibil\_score (70.059)**: Șansele de aprobare a unui împrumut cresc de 70.059 ori pentru fiecare unitate de creștere a scorului CIBIL. Acest coeficient sugerează o influență foarte puternică a scorului CIBIL asupra aprobării împrumutului.

Interpretare: Șansele de aprobare a împrumutului sunt cu 6905.9% mai mari pentru fiecare unitate de creștere a scorului CIBIL, echivalent cu raportul șanselor: p/(1-p) = exp(4.2493) ≈ 70.059.

* **loan\_term (0.421)**: Șansele de aprobare a unui împrumut scad cu aproximativ 57.9% pentru fiecare unitate de creștere a termenului împrumutului. Acest coeficient sugerează o influență negativă a termenului împrumutului asupra aprobării acestuia.

Interpretare: Șansele de aprobare a împrumutului sunt cu 57.9% mai mici pentru fiecare unitate de creștere a termenului împrumutului, echivalent cu raportul șanselor: p/(1-p) = exp(-0.8660) ≈ 0.421.

Aceste interpretări oferă o înțelegere clară a modului în care cibil\_score și loan\_term influențează probabilitatea de aprobare a împrumutului. Modelul sugerează că pentru a crește șansele de aprobare a împrumutului, este esențial să se mențină un scor CIBIL ridicat și să se opteze pentru termene de împrumut mai scurte.

Funcția prob previzionează probabilitățile indivizilor de a li se accepta sau nu imprumutul:

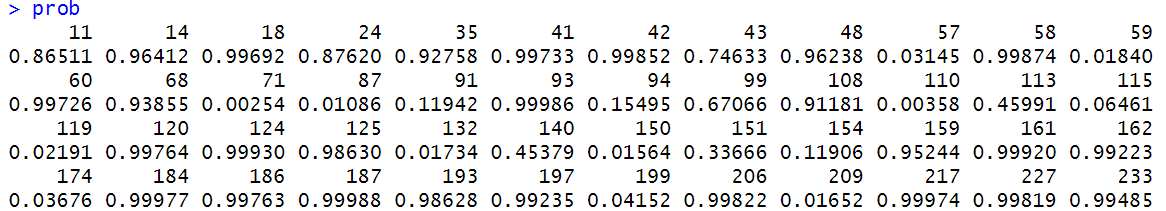


Figure 22: Functia prob

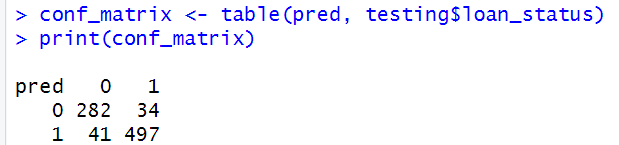


Figure 23: Matricea de confuzie-set de antrenare

Interpretarea valorilor din matricea de confuzie:

True Positives (TP): 497

Numărul de cazuri în care împrumutul a fost aprobat (loan\_status = 1) și modelul a prezis corect aprobarea (pred = 1).

True Negatives (TN): 282

Numărul de cazuri în care împrumutul a fost respins (loan\_status = 0) și modelul a prezis corect respingerea (pred = 0).

False Positives (FP): 41

Numărul de cazuri în care împrumutul a fost respins (loan\_status = 0), dar modelul a prezis incorect aprobarea (pred = 1).

False Negatives (FN): 34

Numărul de cazuri în care împrumutul a fost aprobat (loan\_status = 1), dar modelul a prezis incorect respingerea (pred = 0).

1. **Acuratețea (Accuracy)**:
   * Acuratețea este proporția totală de predicții corecte din toate predicțiile realizate.
   * Calcul: (TP + TN) / (TP + TN + FP + FN) = (497 + 282) / (497 + 282 + 41 + 34) = 779 / 854 ≈ 0.9118 (91.18%)
2. **Precizia (Precision)**:
   * Precizia este proporția predicțiilor pozitive corecte din totalul predicțiilor pozitive.
   * Calcul: TP / (TP + FP) = 497 / (497 + 41) ≈ 0.9238 (92.38%)



Figure 24: Acuratetea modelului

Curba ROC (Receiver Operating Characteristic) este un instrument important pentru evaluarea performanței unui model de clasificare. Aceasta grafică TPR (True Positive Rate) în funcție de FPR (False Positive Rate) pentru diferite valori de prag de decizie.

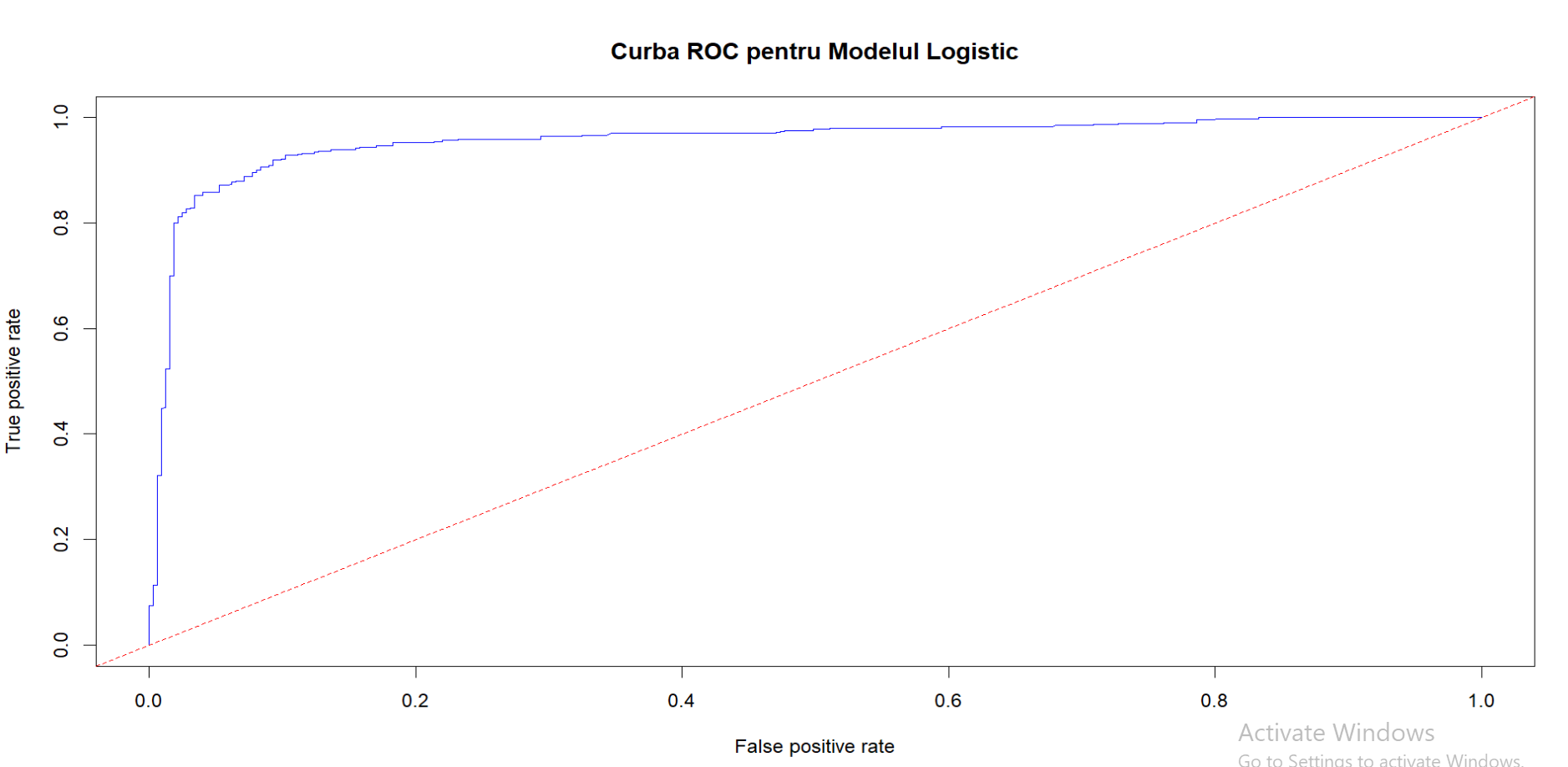


Figure 25: Curba ROC



* Axa X (False Positive Rate - Rata fals pozitivă):

Reprezintă proporția predicțiilor pozitive incorecte din totalul cazurilor negative reale.

Se calculează 

​

* Axa Y (True Positive Rate - Rata adevărat pozitivă):

Reprezintă proporția predicțiilor pozitive corecte din totalul cazurilor pozitive reale.

Se calculează 

* Linia roșie diagonală:

Reprezintă clasificarea aleatorie, unde TPR este egal cu FPR. Este folosită ca linie de referință pentru a evalua performanța modelului.

Interpretari :

Curba ROC a modelului logistic este foarte aproape de colțul din stânga sus al graficului, indicând o performanță excelentă. Aceasta arată că modelul are un TPR ridicat și un FPR scăzut pentru majoritatea valorilor de prag.

Cu cât curba este mai aproape de colțul din stânga sus, cu atât mai bun este modelul în distingerea între clasele pozitive și negative.

Forma curbei sugerează că modelul logistic are o capacitate foarte bună de a distinge între împrumuturile aprobate și cele respinse, cu rate scăzute de fals pozitivi și fals negativi.

2.Regresia logistică multinomiala

Deoarece regresia multinomiala pe varibila mea categoriala pe mai multe niveluri, numarul persoanelor avute in grija (no\_of\_dependents) a avut o acuratete foarte slaba (17%) am decis sa imi transform o varibila numerica corelata puternic cu celelalte variabile, income\_annum, in una categoriala. Income\_annum a fost împărțit în patru categorii (quartile) utilizând funcția cut și quantile.

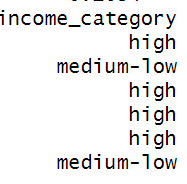


Figure 26: transformarea unei variabile continue într-una categorică pe baza distribuției sale.

* Variabila dependentă este income\_category, Datele sunt împărțite în patru intervale bazate pe cuantile:

low: 10,000 - 32,500

medium-low: 32,501 - 55,000

medium-high: 55,001 - 77,500

high: 77,501 - 100,000

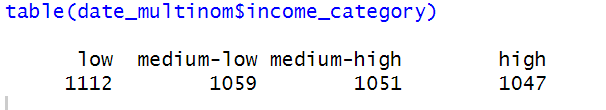


Figure 27: tabela nivelurilor de venit

* Variabilele independente sunt variabilele scalate ale setului de date (fără loan\_status).

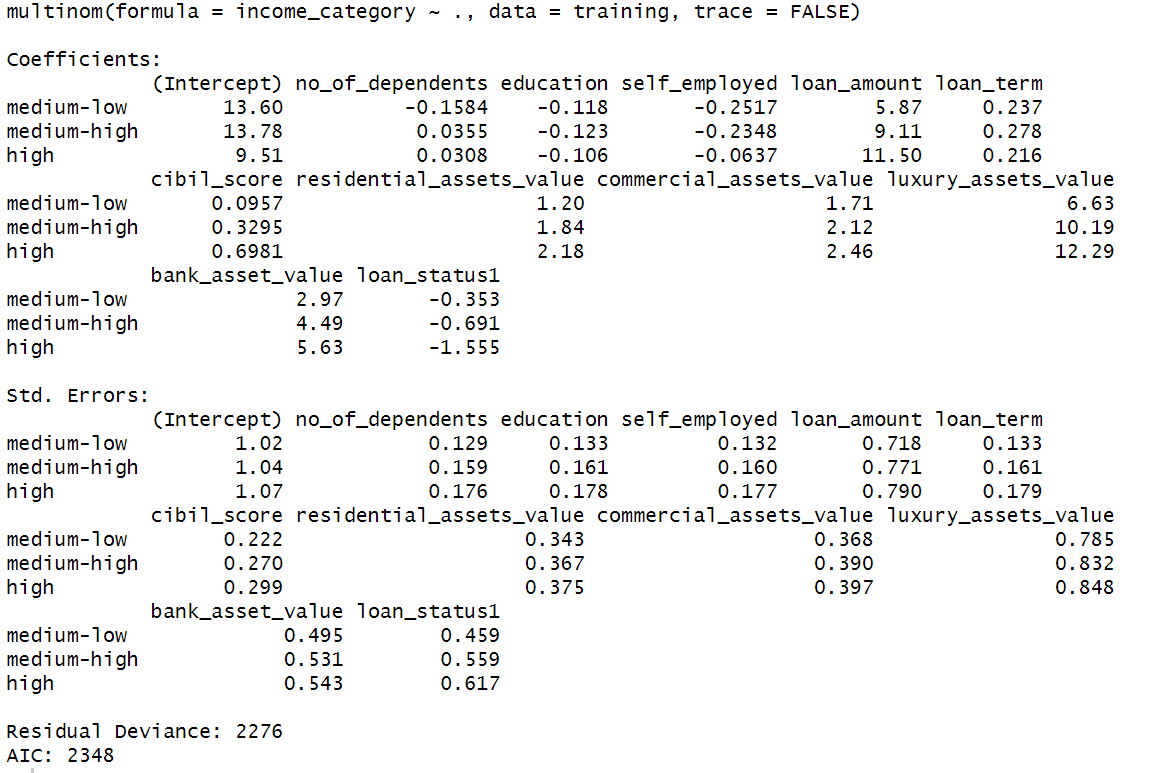


Figure 28: Summary

* **Variabila “loan\_amount”**:

Coeficienții pozitivi (5.87, 9.11, 11.50) indică faptul că o creștere în suma împrumutului mărește log-odds pentru categoriile **medium-low**, **medium-high**, și **high** comparativ cu **low**.

* **Variabila “cibil\_score”**:

Coeficienții pozitivi (0.0957, 0.3295, 0.6981) sugerează că un scor CIBIL mai mare este asociat cu o probabilitate mai mare de a fi în categoriile **medium-low**, **medium-high**, și **high** comparativ cu **low**.

AIC: 2348. Este o măsură a calității modelului care penalizează complexitatea. Un AIC mai mic indică un model mai bun.

Devianţa reziduală (2276) este eroarea rămasă în model, similară cu ceea ce în regresia liniară se numeşte “sum of squares due to errors”. Într-un model de regresie logistică, devianţa reziduală trebuie să fie cât mai mică. Conform tabelului de mai sus, se pot determina ecuaţiile următoare pentru determinarea probabilităţilor:

ln((P(medium-low))/(P(low))) = =13.60−0.1584×no\_of\_dependents−0.118×education−0.2517×self\_employed+5.87×loan\_amount+0.237×loan\_term+0.0957×cibil\_score+1.20×residential\_assets\_value+1.71×commercial\_assets\_value+6.63×luxury\_assets\_value+2.97×bank\_asset\_value−0.353×loan\_status

Logaritmul probabilității ca un individ să fie în categoria medium-low raportată la probabilitatea ca un individ să fie în categoria low se numește “log odds”. Coeficientul negativ -0.118 se poate interpreta astfel: Variabila “education” are un impact negativ asupra raportului O creștere cu o unitate a variabilei “education” conduce la scăderea logarimului “log odds” cu 0.118.

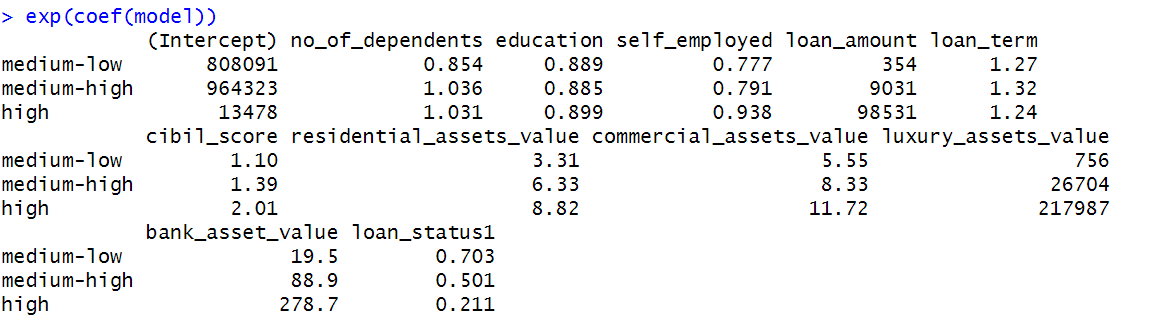


Figure 29: Coeficienți model

Interpretari :

Suma imprumutata:

* Șansele ca un individ să fie în categoria medium-low sunt cu 27% mai mari decât șansele ca un individ să fie în categoria low când suma împrumutului crește cu o unitate (exp(0.24) = 1.27).
* Șansele ca un individ să fie în categoria medium-high sunt cu 32% mai mari decât șansele ca un individ să fie în categoria low când suma împrumutului crește cu o unitate (exp(0.277) = 1.32).
* Șansele ca un individ să fie în categoria high sunt cu 24% mai mari decât șansele ca un individ să fie în categoria low când suma împrumutului crește cu o unitate (exp(0.21) = 1.24).

Determinăm probabilităţile în R prin comanda următoare, realizând o predicție a variabilei de răspuns pentru toate observațiile

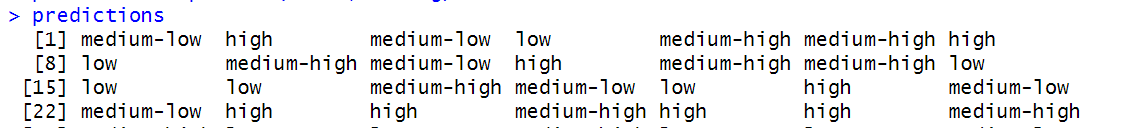


Figure 30: Predictie

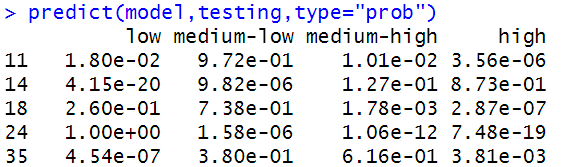


Figure 31: Probabilitati

Pentru fiecare individ din setul de date de testare, modelul de regresie logistică multinomială prezice probabilitățile de a se încadra în fiecare categorie de venit (low, medium-low, medium-high, high). Iată interpretarea pentru câțiva indivizi din setul de testare:

Individul 11:

Probabilitatea de a fi în categoria low: 0.018 (1.80e-02)

Probabilitatea de a fi în categoria medium-low: 0.972 (9.72e-01)

Probabilitatea de a fi în categoria medium-high: 0.0101 (1.01e-02)

Probabilitatea de a fi în categoria high: 0.00000356 (3.56e-06)

Cea mai mare probabilitate este pentru categoria medium-low (0.972), deci predicția este că individul 11 se încadrează în categoria medium-low.

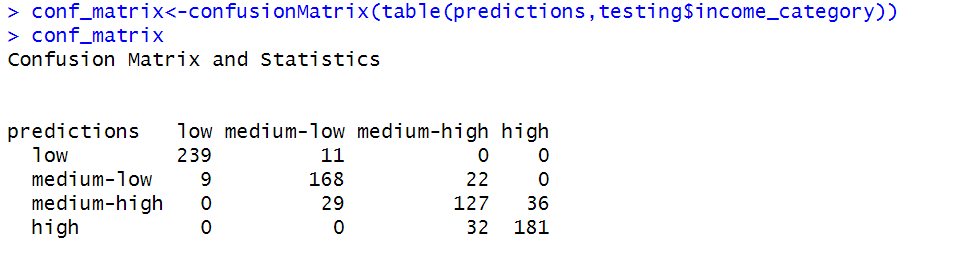


Figure 32: Matricea de confuzie

Conform matricei de confuzie, 239 de indivizi din categoria low și 168 din categoria medium-low au fost corect clasificați. În categoria medium-high, 127 de indivizi au fost corect clasificați, iar 181 de indivizi din categoria high au fost corect clasificați. Erorile notabile includ 11 indivizi din medium-low clasificați ca low, 22 din medium-high clasificați ca medium-low, 29 din medium-low și 36 din high clasificați ca medium-high, și 32 din medium-high clasificați ca high. Modelul are o performanță generală bună, dar cu confuzii între categoriile medium-low, medium-high și high.



Figure 33: Accuracy

Acuratețea modelului de regresie logistică multinomială este de 0.837, ceea ce înseamnă că modelul a clasificat corect 83.7% dintre observațiile din setul de testare. Acest nivel de acuratețe indică o performanță bună a modelului în predicția categoriilor de venit pe baza variabilelor independente incluse în model.

1. Folosirea arborilor de regresie și clasificare + Curățarea arborelui

1.Arbore de regresie

În continuarea analizei noastre privind categoriile de venit, vom utiliza un arbore de regresie pentru a investiga relația dintre variabila dependentă income\_annum și un set de variabile independente. Arborii de regresie sunt modele predictive care împart datele în subseturi bazate pe caracteristicile variabilelor independente, în scopul de a prezice valorile unei variabile continue. Această metodă este intuitivă și ușor de interpretat, oferind o vizualizare clară a modului în care variabilele independente, precum educația, numărul de dependenți, suma împrumutului, scorul CIBIL și altele, influențează venitul anual. Prin aplicarea unui arbore de regresie, ne propunem să identificăm factorii cheie care contribuie cel mai mult la variația venitului anual și să construim un model robust care să ofere predicții precise și utile pentru diferite scenarii financiare.

Din Figurile 5 si 6 putem observa faptul că distribuția este una normală, deci nu va mai fi nevoie de logaritmarea datelor.

Cu ajutorul funcției split , s-a împărțit setul de date în set de antrenare și set de testare. P a fost ales astfel încât datele să fie împărțite în jumătate.

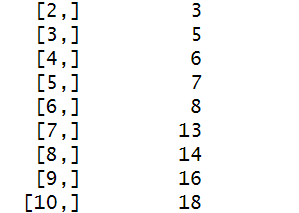


Figure 34: Functia split

Setul de antrenare:

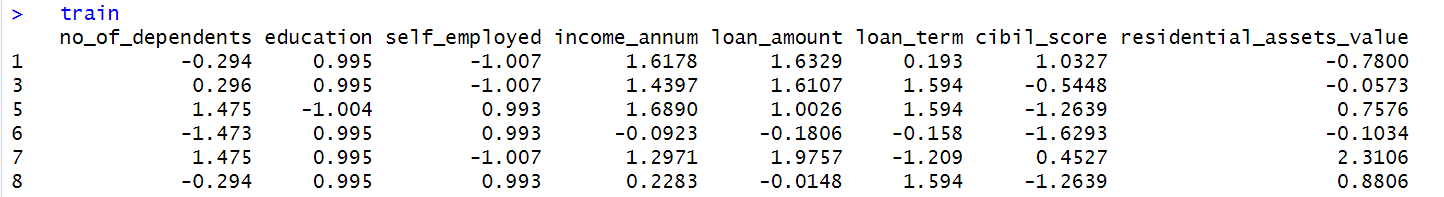


Figure 35: Set de antrenare - arbori de regresie

Setul de testare:

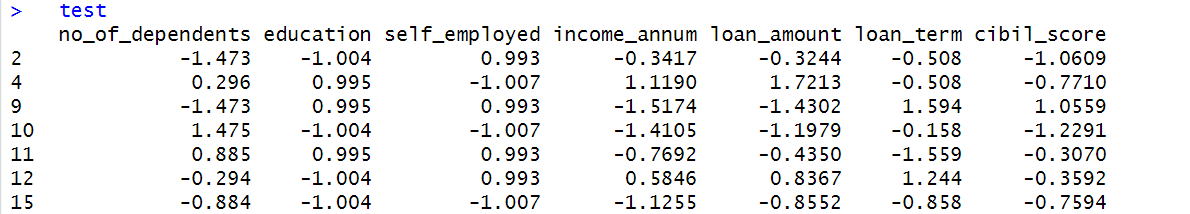


Figure 36: Set de testare - arbori de regresie

Fiecare nod terminal al arborelui are reprezentat clasa dominant.

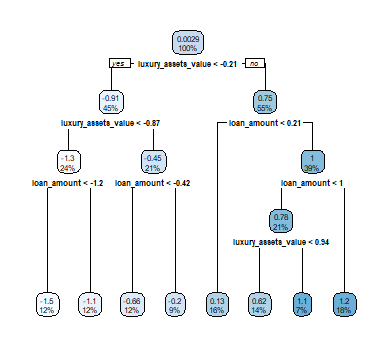


Figure 37: Arbore de regresie

Intrucut cele mai corelate variabile cu income\_ammount sunt luxury\_assets\_value si loan\_amount (Valori mai mari ale activelor de lux și împrumuturilor tind să fie asociate cu venituri anuale mai mari, iar acest lucru este reflectat în diviziunile arborelui), nodurile decizionale vor micsora cadranul in care cautam media pe baza acestor variabile, arborele avand 4 niveluri de parcurgere a graficului, cu 8 frunze, deci 8 posibilitati de valori scalate estimate.

* Pe ramura stângă a arborelui, observăm că indivizii cu valori foarte scăzute ale activelor de lux sunt mai susceptibili de a avea venituri mai mici, iar această ramură se împarte în continuare în funcție de valoarea împrumutului, subliniind că atât valoarea activelor de lux, cât și suma împrumutului sunt critice pentru predicția venitului.
* Pe ramura dreaptă a arborelui, pentru cei cu valori mai mari ale activelor de lux, valoarea împrumutului devine factorul determinant principal. În această ramură, persoanele cu împrumuturi mai mari tind să aibă venituri anuale mai mari, în timp ce o valoare mai mică a împrumutului este asociată cu venituri mai mici.

Urmează etapa de curățare a arborelui. Funcția cv.tree este bazată pe generarea de numere aleatoare și determină în ce loc din arbore se oprește curățarea. Atributul funcției, „size” indică numărul de noduri terminale.



Figure 38: funcția cv.trees$size



Figure 39: funcția cv.trees$dev

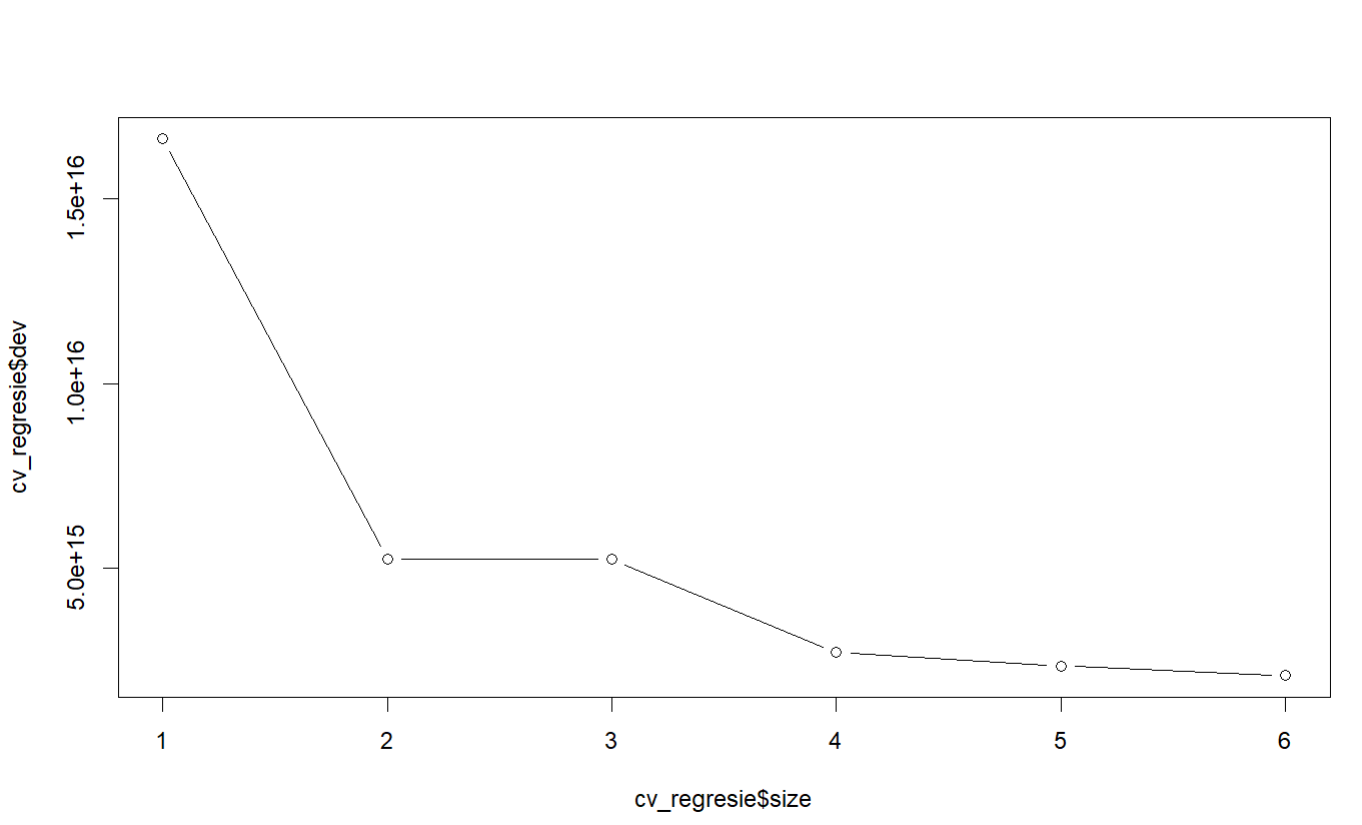


Figure 40: Reprezentăm grafic eroarea în funcție de marimea arborelui

Acesta este graficul erorilor RMSE in functie de dimensiunea maxima pe care o admitem pentru arbore. Observam ca, in acest interval, cu cat marim complexitatea arborelui cu atat obtinem predictii mai bune. Din figura de mai jos putem observa faptul că minimul erorii se atinge pentru o mărime a arborelui egală cu maxim 6 noduri.

Se construiește arborele curățat în figura de mai jos. Se observă că noul arbore are de data aceasta 5 noduri.

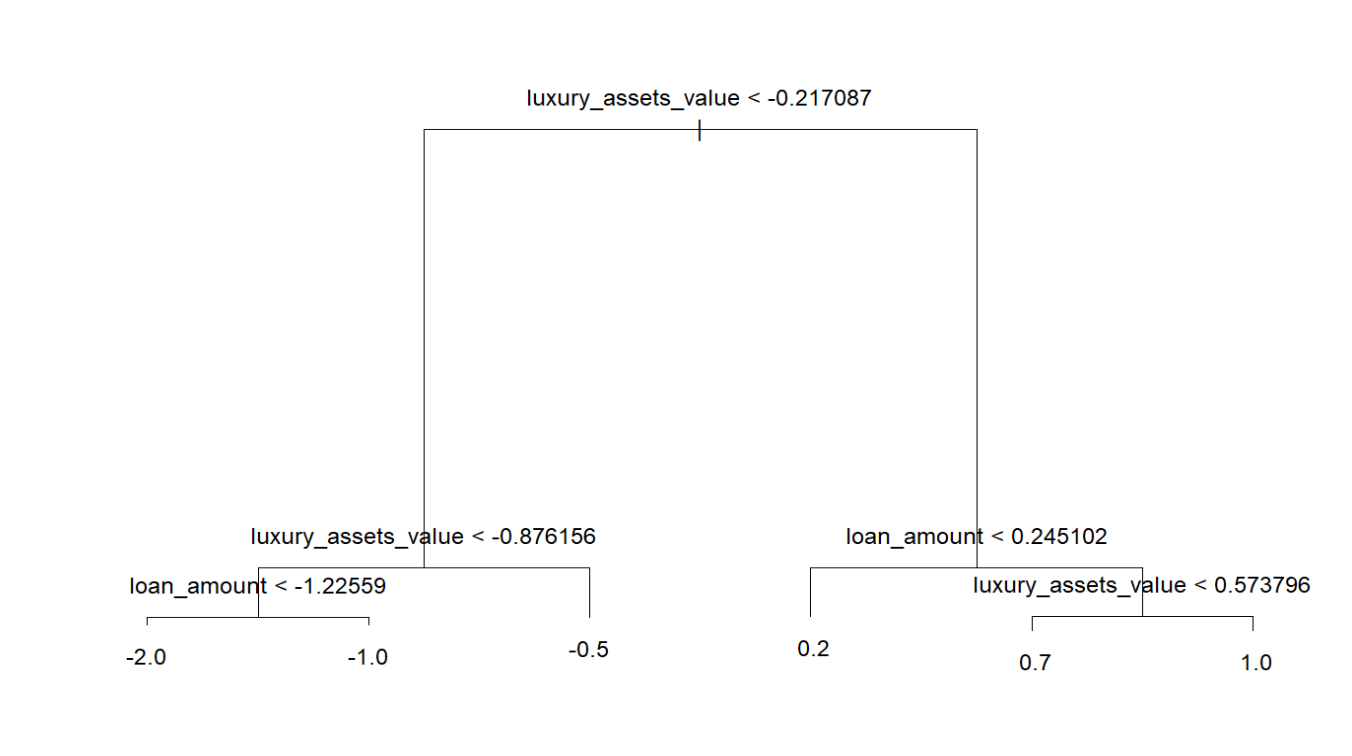


Figure 41: Devianța reprezentată grafic

Se calculează predicția venitului al fiecărui individ:

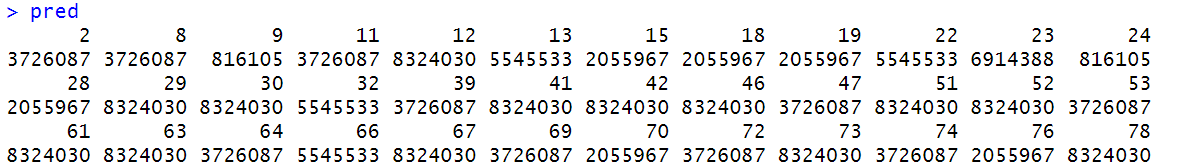


Figure 42: Predictie

In figura 43 de mai jos putem observa:

* Alinierea cu linia de regresie:

Majoritatea punctelor sunt aproape de linia diagonală, ceea ce indică o bună concordanță între valorile prezise și cele reale ale venitului anual (income\_annum). Cu cât punctele sunt mai apropiate de linia diagonală, cu atât predicțiile sunt mai precise.

* Distribuția pe axa x:

Valorile prezise (pred) variază de la aproximativ -1.5 la 1.5. Acest interval este un rezultat al datelor scalate, reflectând variația predicțiilor modelului într-un interval normalizat.

* Distribuția pe axa y:

Valorile reale ale venitului anual (test$income\_annum) sunt, de asemenea, distribuite de la aproximativ -1.5 la 1.5, indicând că modelul captează bine gama veniturilor anuale în datele de testare.

* Devierea față de linia de regresie:

Există câteva puncte care se abat mai mult de la linia diagonală, sugerând erori de predicție pentru acele observații. Aceste devieri pot indica cazuri în care modelul nu a capturat complet complexitatea relațiilor dintre variabilele independente și venitul anual.

Graficul arată că modelul de regresie a reușit în mare măsură să prezică valorile income\_annum destul de precis pentru setul de date de testare. Cu toate acestea, există unele erori de predicție, sugerate de punctele care se abat de la linia diagonală

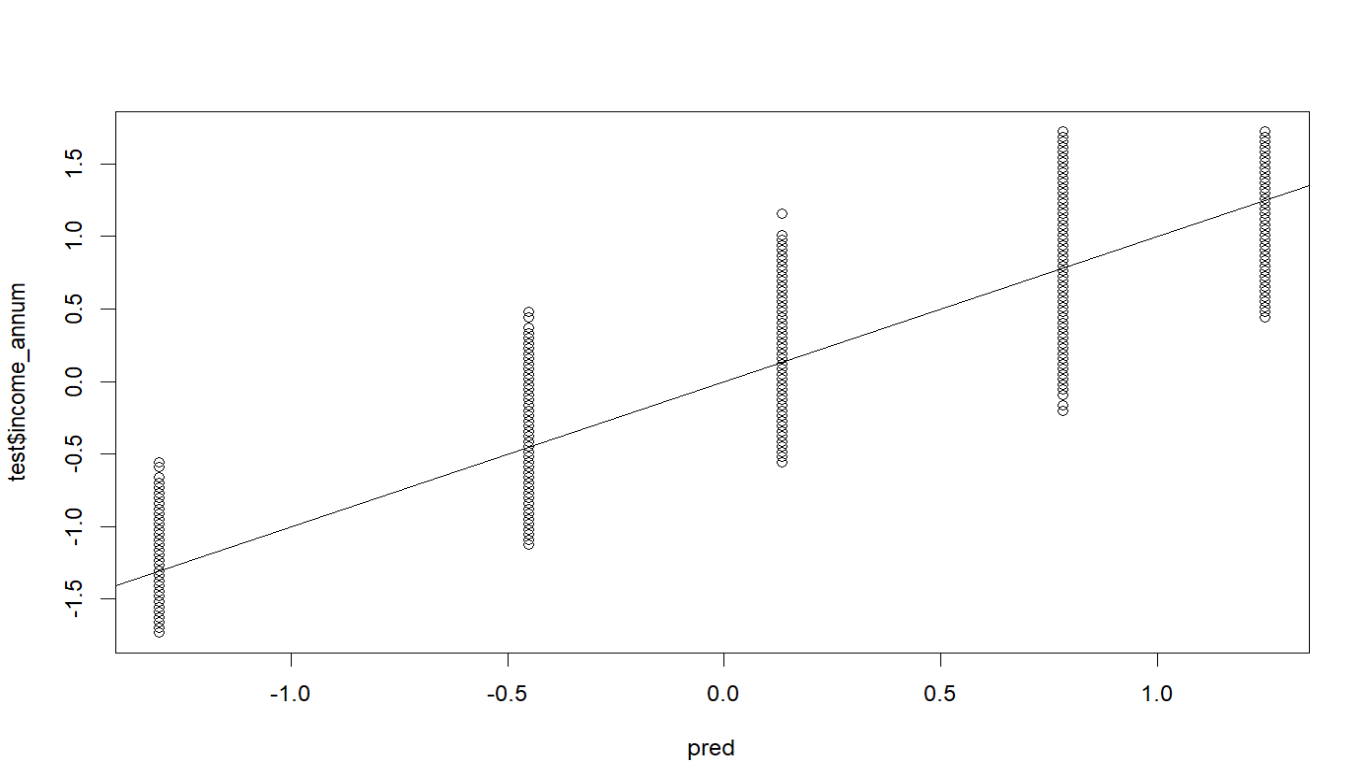


Figure 43: Plot predicții



Figure 44: Eroarea de predicție

Se observă faptul că eroarea de predicție este calculată ca media pătratelor diferențelor dintre venitul previzionat și venitul real. 13.8% din observatiile setului de testare au fost eronat clasificate. Astfel, rezultă eroarea de predicție egală cu 0.138, o valoarea destul de mică, deci setul de testare a fost corect previzionat.

1.Arbore de clasificare

În acest proiect, am construit și analizat un arbore de clasificare multinomial pentru a prezice loan\_status pe baza variabilelor independente. Datele au fost împărțite în seturi de antrenare și testare, și am folosit metoda de validare încrucișată pentru a îmbunătăți performanța modelului.

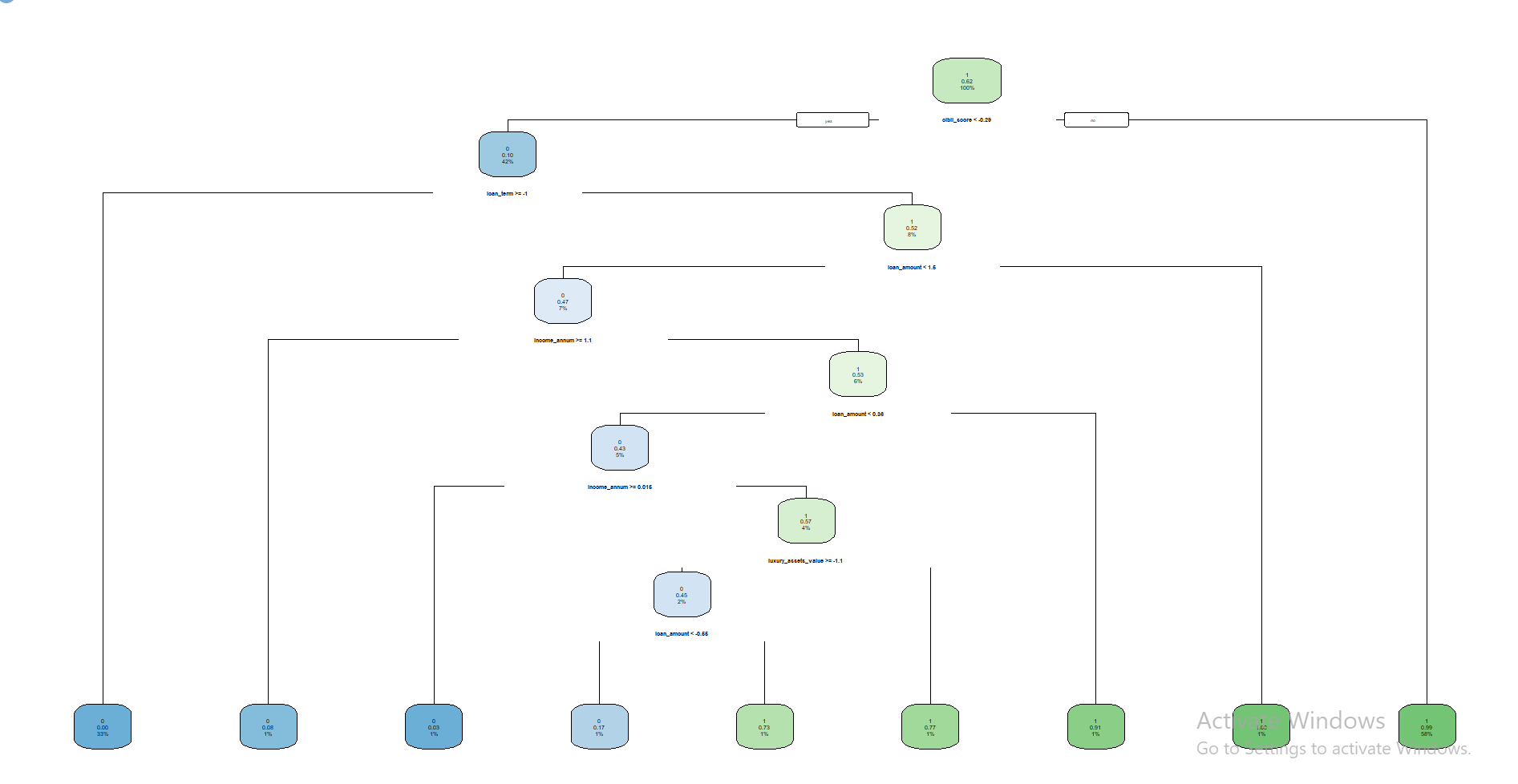


Figure 45: Arborele multinomial initial

Observatiile sunt facute pe date scalate:

Scorul CIBIL:

Este cel mai important factor inițial. Un scor CIBIL sub -0.29 necesită evaluări suplimentare ale termenului împrumutului și ale venitului pentru a decide statusul împrumutului.

Termenul împrumutului:

Un termen foarte scurt al împrumutului (sub -1.4) conduce la neaprobarea împrumutului. Termenii mai lungi necesită evaluări suplimentare.

Suma împrumutului:

Este un factor critic pentru deciziile din ramura dreaptă a arborelui. Împrumuturile mai mari de 1.6 sunt aproape garantat aprobate.

Venitul și activele de lux:

Pentru cazurile în care scorul CIBIL este scăzut, venitul și activele de lux devin factori decisivi pentru aprobarea împrumutului.

Se calculează predicția statusului acordarii de imprumut pentru fiecare individ:

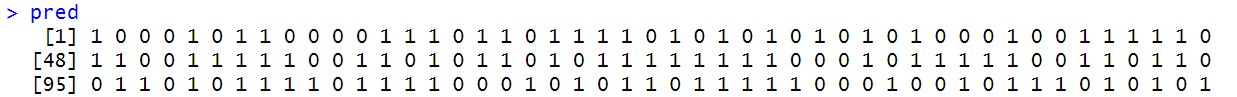


Figure 46: Predicție pentru setul de testare

Matricea de confuzie inițială arată următoarele:

574 de observații au fost corect clasificate ca 0.

1062 de observații au fost corect clasificate ca 1.

71 de observații au fost incorect clasificate ca 1, deși aparțineau clasei 0.

Aceasta indică faptul că modelul are o performanță bună, dar există o ușoară tendință de clasificare greșită a observațiilor din clasa 0.

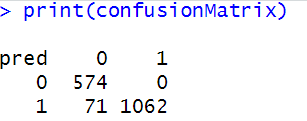


Figure 47: Matricea de confuzie a arborelui de clasificare initial

Arborele de clasificare inițial a fost construit folosind setul de antrenare, iar predicțiile au fost făcute pe setul de testare. Acuratețea inițială a modelului a fost calculată și a fost de 95.84%.



Îmbunătățirea acurateței predictiei utilizând validarea încrucișată:

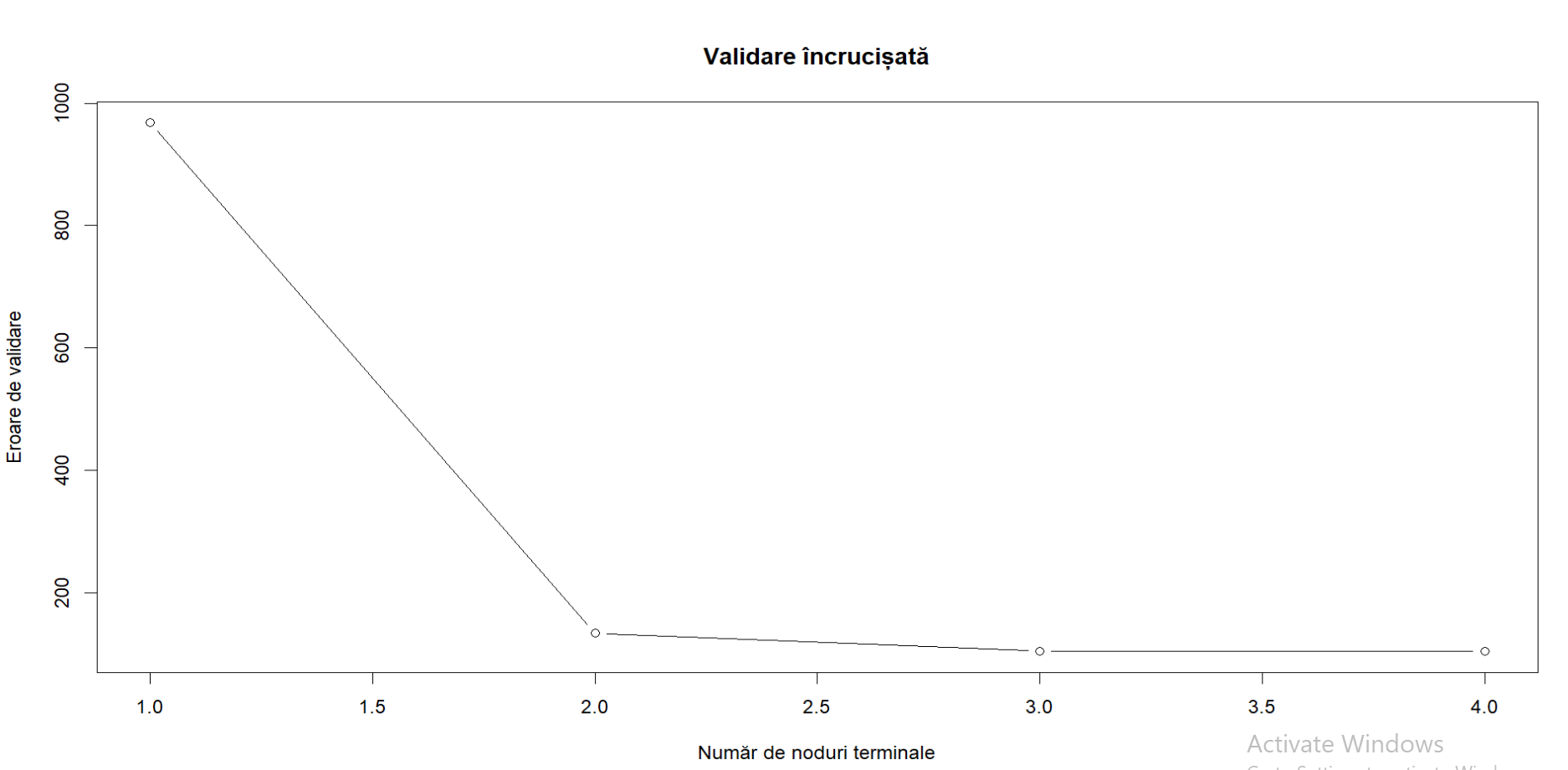


Figure 48: Validare incrucisata

Numărul de noduri terminale vs. eroarea de validare:

La început, când numărul de noduri terminale este 1, eroarea de validare este foarte mare, în jur de 1000.

Pe măsură ce numărul de noduri terminale crește la 2, eroarea de validare scade dramatic, ajungând la aproximativ 100. Aceasta indică o îmbunătățire semnificativă a modelului.

După ce numărul de noduri terminale depășește 2, eroarea de validare se stabilizează, menținându-se aproape constantă și foarte scăzută.

Punctul optim de tăiere:

Graficul sugerează că numărul optim de noduri terminale este în jur de 2 sau 3, unde eroarea de validare este minimă și stabilă. Adăugarea de noduri suplimentare după acest punct nu îmbunătățește semnificativ modelul și poate duce la suprasolicitare.

Curățăm arborele de clasificare:

Funcția prune.misclass curăță arborele inițial (arbore), reducându-l la un arbore cu numărul optim de noduri terminale (best = cv.tree1$size[which.min(cv.tree1$dev)]).

Aceasta elimină ramurile care nu contribuie semnificativ la reducerea erorii de clasificare, lăsând un arbore mai simplu și mai generalizabil.

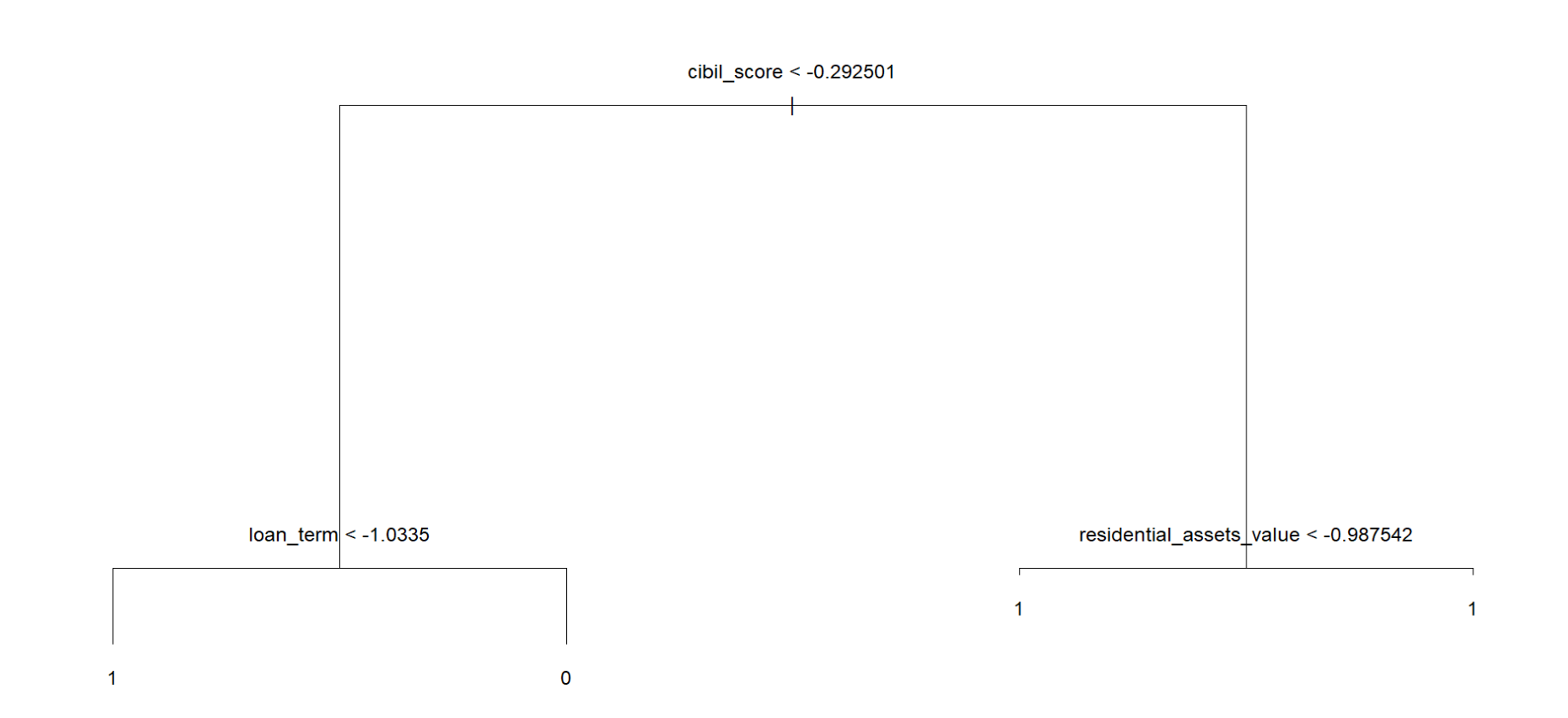


Figure 49: Arborele multinomial curatat

Arborele curățat sugerează că scorul CIBIL și termenul împrumutului sunt factori esențiali în decizia de aprobare a împrumutului. Persoanele cu scoruri CIBIL foarte scăzute au nevoie de termeni de împrumut foarte scurți pentru a obține aprobarea. În schimb, pentru cei cu scoruri CIBIL mai ridicate, aprobarea împrumutului este aproape garantată indiferent de valoarea activelor rezidențiale.

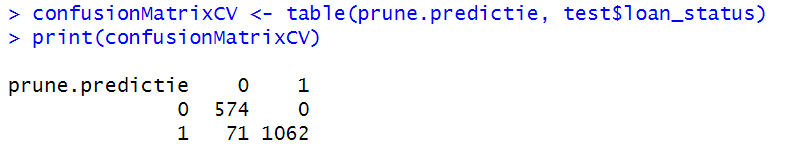


Figure 50: Matricea de confuzie a arborelui multinomial curatat

Performanța Modelului:

Modelul curățat are o rată ridicată de clasificare corectă pentru ambele clase, dar mai ales pentru clasa 1 (aprobat), unde toate observațiile au fost clasificate corect.

Există unele erori de clasificare pentru clasa 0 (neaprobat), unde 71 de observații au fost incorect clasificate ca 1 (aprobat).

Acuratețea Generală:

Acuratețea modelului este calculată ca (numărul de clasificări corecte) / (totalul clasificărilor). În acest caz:

Totalul clasificărilor corecte = 574 + 1062 = 1636.

Totalul observațiilor = 574 + 71 + 1062 = 1707.

Acuratețea = 1636 / 1707 ≈ 0.958, sau 95.8%.



Figure 51: Acuratetea arborelui multinomial curatat

Concluzii:

Faptul că atât arborele inițial, cât și arborele curățat au aceeași acuratețe indică următoarele:

* Robusteză a Modelului:

Modelul este robust, iar variabilele alese sunt suficient de informative pentru a menține performanța chiar și după curățarea arborelui. Acest lucru sugerează că modelul este bine echilibrat și că simplificarea lui nu afectează negativ acuratețea predicțiilor.

* Generalizabilitate:

Curățarea arborelui nu a deteriorat performanța, ceea ce înseamnă că modelul curățat este la fel de eficient ca și modelul inițial în clasificarea datelor de testare. Un model curățat, mai simplu, are un risc mai mic de suprasolicitare și este mai generalizabil pe seturi de date noi.

* Complexitate vs. Simplitate:

Deși modelul inițial și modelul curățat au aceeași acuratețe, modelul curățat este mai simplu și mai ușor de interpretat. Acest lucru poate fi un avantaj în aplicarea practică, deoarece deciziile luate de model sunt mai ușor de explicat și de înțeles.

1. Utilizarea algoritmului KNN pentru clasificare

Variabila etichetă va fi “loan\_status”

Se extrag două eșantioane, corespunzătoare setului de antrenare (80% din date) și setului de testare (20%), folosind funcția createDataPartition.

Antrenarea modelului KNN cu k = 3:

Modelul KNN este antrenat cu k = 3 folosind setul de date de antrenare.

Se calculează matricea de confuzie pentru a evalua performanța modelului și se calculează acuratețea.

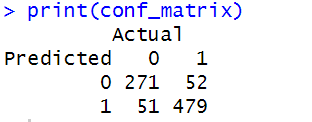


Figure 52: Matrice confuzie

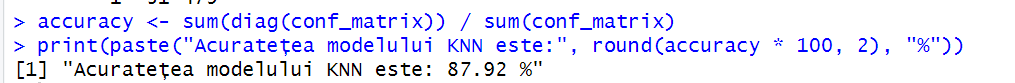
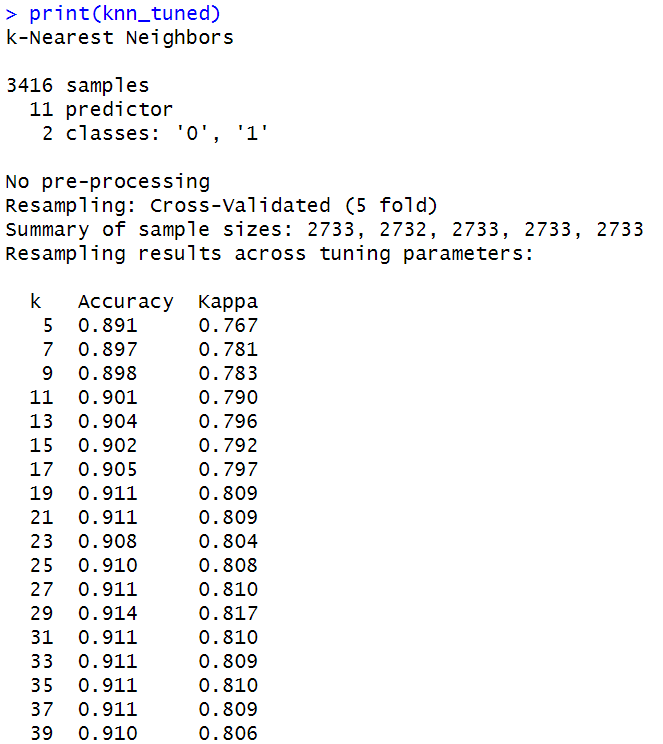


Figure 53: Accuracy model KNN

Se folosește trainControl pentru a determina valoarea optimă a lui k pentru modelul K-Nearest Neighbors (KNN), s-a utilizat validarea încrucișată cu 5 fold-uri. Acest proces a implicat împărțirea setului de date de antrenare în 5 subseturi și evaluarea modelului pe fiecare subset pentru diferite valori ale lui k. Performanța modelului a fost măsurată folosind acuratețea și kappa statistic, care indică gradul de acord între predicțiile modelului și valorile reale.

S-au testat valori impare pentru k între 1 și 60.

Acuratețea modelului pentru k = 29 este de 0.914 (91.4%), iar kappa statistic este 0.817, ceea ce indică un acord foarte bun între predicțiile modelului și valorile reale.



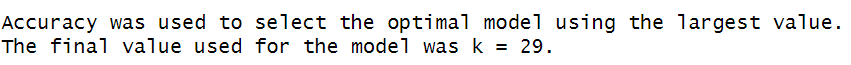


Figure 54: k-Nearest Neighbors

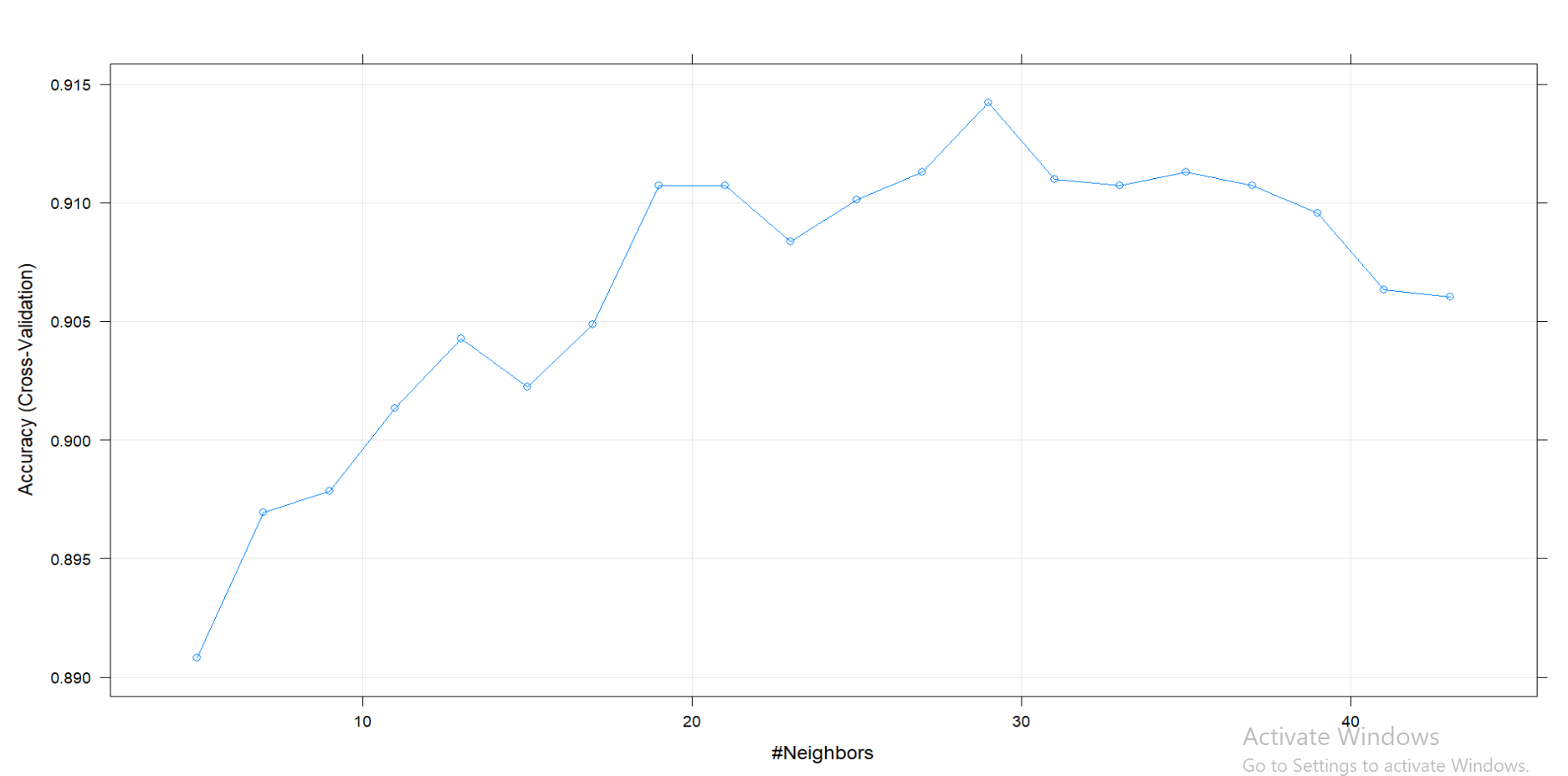


Figure 55: Variația Acurateței Modelului în Funcție de Valorile lui k pentru K-Nearest Neighbors

La valori mici ale lui k, acuratețea este relativ scăzută, în jur de 89%.

Antrenarea și evaluarea modelului optim KNN: Modelul KNN este antrenat din nou folosind valoarea optimă a lui k.

Se calculează matricea de confuzie și acuratețea pentru modelul optim.

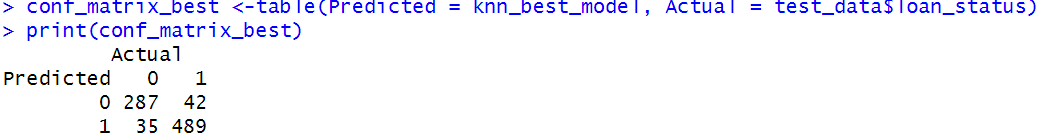


Figure 56: Matricea de confuzie pt modelul cu k optim



Figure 57: Acuratetea

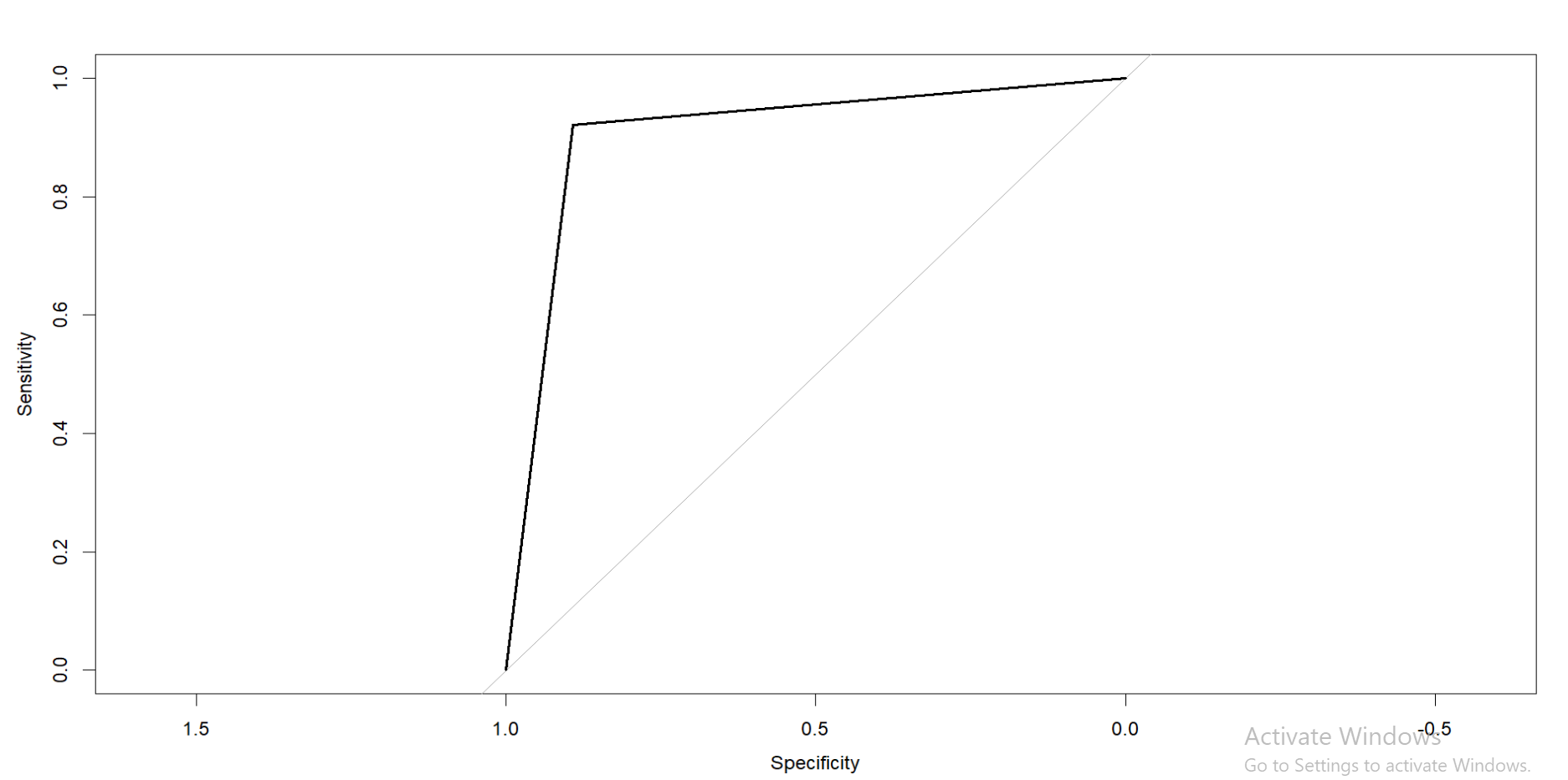


Figure 58: Curba ROC pentru Modelul KNN Optimizat



În acest caz, curba ROC este aproape de colțul din stânga sus, ceea ce indică o performanță excelentă a modelului KNN optimizat. Valoarea AUC (Area Under the Curve) este 0.906, ceea ce înseamnă că modelul are o capacitate foarte bună de a diferenția între clase. Cu cât valoarea AUC este mai aproape de 1, cu atât modelul este mai performant

1. Folosirea rețelelor neuronale pentru clasificare

Rețelele neuronale sunt modele complexe inspirate de structura creierului uman, capabile să învețe relații complexe dintre variabilele de intrare și cele de ieșire. În clasificare, rețelele neuronale sunt folosite pentru a construi modele care pot prezice clasele unei observații pe baza unor caracteristici de intrare

În acest proiect, am utilizat setul de date pentru a construi și antrena o rețea neuronală artificială. Scopul a fost de a prezice statutul de aprobare a unui împrumut (**loan\_status**) pe baza mai multor caracteristici demografice și financiare ale solicitantului. Acest model poate ajuta instituțiile financiare să evalueze mai bine riscul asociat cu acordarea unui împrumut.

Am început prin încărcarea și preprocesarea datelor. Setul de date a fost încărcat și variabilele categorice au fost convertite în factori. Variabilele numerice au fost scalate pentru a asigura o distribuție uniformă și a îmbunătăți performanța modelului. Am împărțit setul de date în seturi de antrenare și testare, utilizând primele 3200 (75%) de observații pentru antrenare și restul pentru testare.

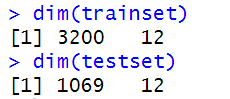


Figure 59: Dimensiunea seturilor de date de antrenare și testare

Antrenăm rețeaua neuronală cu variabila țintă „loan\_status” și predictorii, setând parametrii rețelei: 4 noduri ascunse și prag de oprire 0.1.

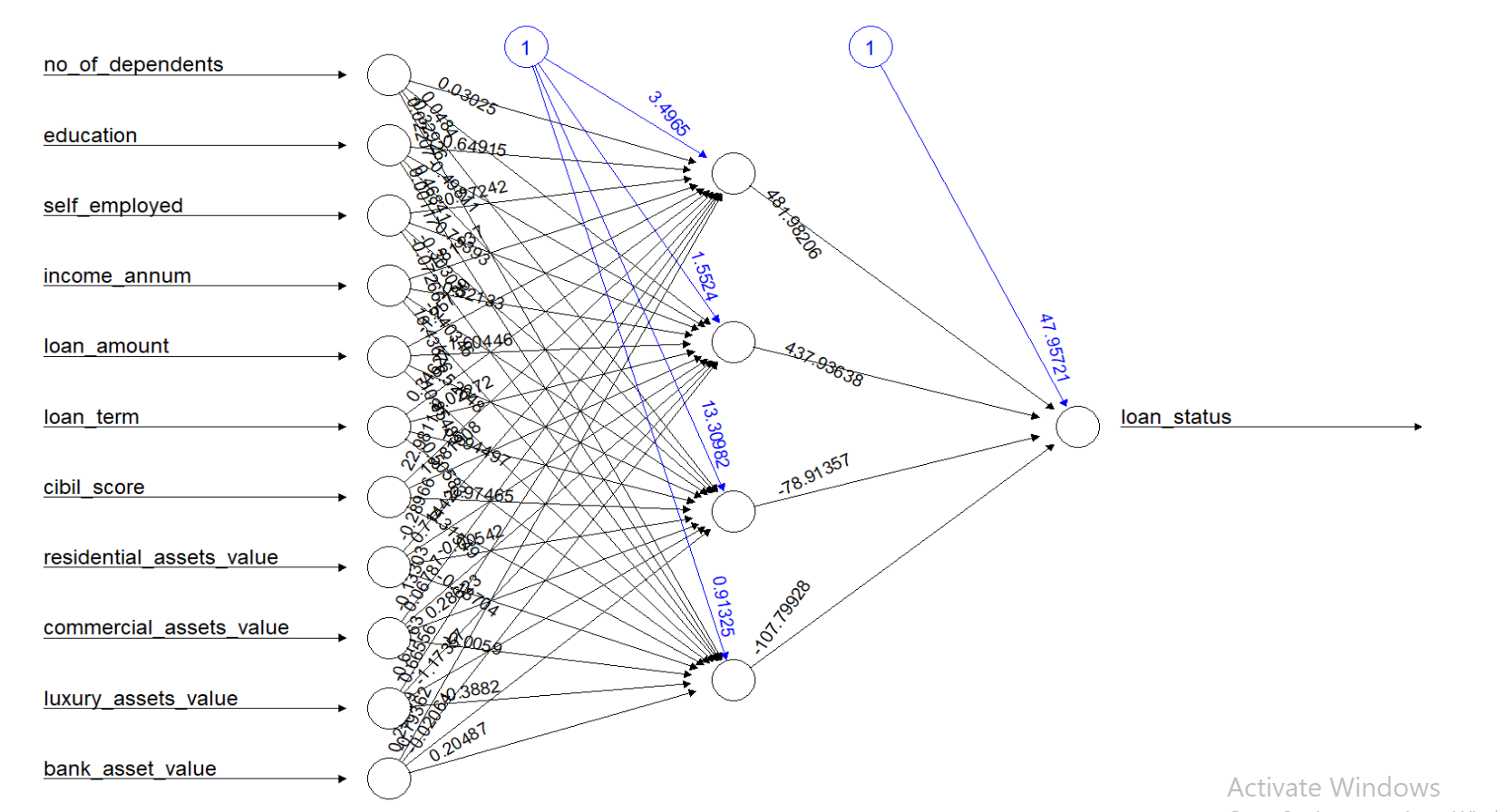
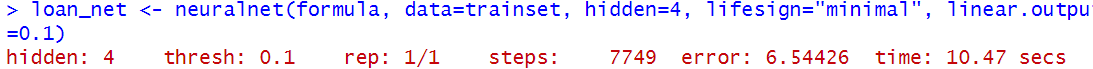


Figure 60:Output rețea



* In cele 7749 iteratii, s a obtinut o eroare de 6.54426
* În partea dreaptă, avem nodul de ieșire care reprezintă predicția rețelei pentru loan\_status.
* Rețeaua combină informațiile de la nodurile ascunse pentru a genera un rezultat final care este comparat cu valoarea reală a loan\_status pentru a evalua acuratețea modelului.
* Liniile care leagă nodurile de intrare de nodurile ascunse și nodurile ascunse de nodul de ieșire reprezintă greutățile sinaptice.
* Valorile de pe aceste linii indică importanța și influența pe care fiecare caracteristică de intrare o are asupra nodurilor ascunse și, în final, asupra predicției.
* De exemplu, dacă greutatea dintre income\_annum și un nod ascuns este mare, înseamnă că venitul anual al solicitantului are o influență semnificativă asupra acelui nod ascuns și asupra predicției finale.

Dupa ce am antrenat reteaua, o vom testa utilizand setul de testare

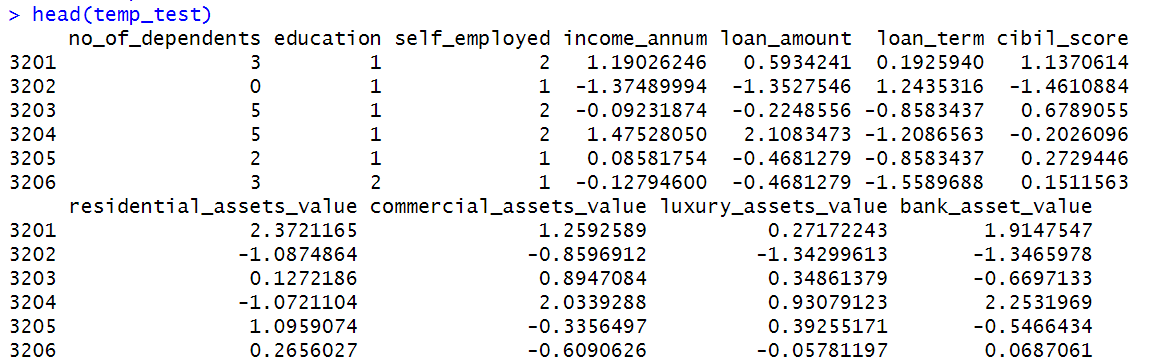


Figure 61: testare retelei cu ajutorul temp\_set

Am folosit functia compute pentru a determina outputul, avand deja o retea antrenata

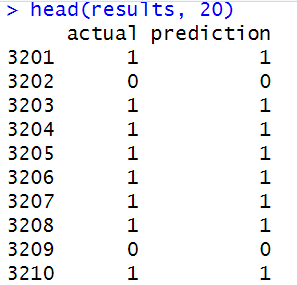
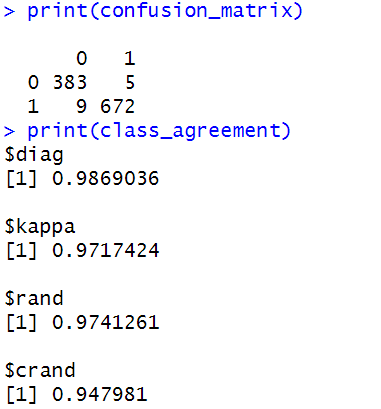


Figure 62: Valori reale vs Valori previzionate

Rezultatele afișate de funcția classAgreement reflectă performanța rețelei neuronale în clasificarea corectă a loan\_status

diag: Procentul Valorilor Corecte pe Diagonala Principală

Valoare: 0.9869036

Interpretare: Acest indicator reprezintă proporția predicțiilor corecte din totalul predicțiilor realizate de model. Cu alte cuvinte, este o măsură a acurateței brute a modelului. O valoare de 0.9869036 (sau 98.69%) indică faptul că aproape 99% din predicțiile modelului au fost corecte.

Figure 63: Matricea de confuzie si Funcția classAgreement

kappa: Valoarea Ajustată a Indicatorului Kappa

Valoare: 0.9717424

Interpretare: Kappa este o măsură a acordului dintre clasificările prevăzute de model și clasificările reale, ajustată pentru probabilitatea de acord întâmplător. O valoare de 0.9717424 sugerează un acord foarte bun, aproape perfect, între predicțiile modelului și valorile reale, ajustate pentru șansa de clasificare întâmplătoare.

rand: Indexul Rand

Valoare: 0.9741261

Interpretare: Indexul Rand măsoară similaritatea dintre seturile de clasificări, luând în considerare atât perechile corect clasificate cât și cele incorect clasificate. O valoare de 0.9741261 indică un nivel foarte înalt de similaritate între predicțiile modelului și clasificările reale.

crand: Indexul Rand Ajustat

Valoare: 0.947981

Interpretare: Indexul Rand ajustat corectează indexul Rand pentru probabilitatea de clasificare întâmplătoare, similar cu kappa. O valoare de 0.947981 sugerează că modelul are un nivel foarte bun de performanță în clasificarea corectă, luând în considerare șansa de clasificare întâmplătoare.

Concluzii

În cadrul acestui proiect, am explorat utilizarea tehnicilor de machine learning pentru a prezice aprobarea împrumuturilor (loan\_status) folosind un set de date complex. Scopul principal a fost de a dezvolta și evalua modele predictive care pot ajuta instituțiile financiare să ia decizii informate privind aprobarea cererilor de împrumut.

Preprocesarea Datelor

Am început prin preprocesarea datelor, care a inclus transformarea variabilelor categorice în factori și scalarea variabilelor numerice. Aceasta a asigurat uniformitatea datelor și a îmbunătățit performanța modelelor. Datele au fost împărțite în seturi de antrenare și testare într-o proporție de 80-20%, permițând evaluarea adecvată a modelelor dezvoltate.

Regresie Logistică Binomială

Am construit un model de regresie logistică binomială pentru a prezice loan\_status pe baza scorului CIBIL (cibil\_score) și a termenului împrumutului (loan\_term). Modelul a demonstrat o capacitate ridicată de predicție, cu o acuratețe de 91.18%. Curba ROC a indicat o performanță excelentă, confirmată de valoarea AUC (Area Under the Curve).

Regresie Logistică Multinomială

Pentru a evalua impactul variabilelor asupra categoriilor de venit, am aplicat regresia logistică multinomială. Variabila income\_annum a fost transformată într-o variabilă categorială cu patru niveluri (low, medium-low, medium-high, high). Modelul a arătat că educația, suma împrumutului și scorul CIBIL sunt factori determinanți pentru diferitele categorii de venit. Acuratețea modelului a fost de 83.7%.

Arbori de Decizie

Am utilizat arborii de decizie pentru a prezice atât venitul anual (income\_annum), cât și loan\_status. Arborii de decizie au evidențiat importanța variabilelor precum valoarea activelor de lux și suma împrumutului în predicția venitului anual. Arborii de clasificare au indicat că scorul CIBIL și termenul împrumutului sunt factori esențiali în decizia de aprobare a împrumutului. Acuratețea modelului curățat a fost de 95.8%, demonstrând robusteză și generalizabilitate.

Algoritmul K-Nearest Neighbors (KNN)

Am antrenat un model KNN pentru a clasifica loan\_status, determinând valoarea optimă a lui k folosind validarea încrucișată. Modelul optimizat a avut o acuratețe de 91.4% și o valoare AUC de 0.906, indicând o performanță foarte bună în distingerea între clasele pozitive și negative.

Rețele Neuronale

Am construit și antrenat o rețea neuronală pentru a prezice loan\_status. Modelul a fost optimizat folosind o grilă de parametri și validare încrucișată. Rețeaua neuronală optimizată a avut o acuratețe ridicată și un coeficient Kappa ridicat, sugerând că modelul este foarte eficient în clasificarea corectă a statutului de împrumut.

**LISTA FIGURI**

[Figure 1: Setul de date cu primele 12 randuri 3](#_Toc168000257)

[Figure 2:Data Encoding 3](#_Toc168000258)

[Figura 3: Summary 4](#_Toc168000259)

[Figura 4: Describe 5](#_Toc168000260)

[Figura 5: Histograme 6](#_Toc168000261)

[Figura 6: Boxplot-uri 7](#_Toc168000262)

[Figura 7: Barplot-uri 8](#_Toc168000263)

[Figure 8: Corelatii 9](#_Toc168000264)

[Figure 9: Rezultatele clusterizarii 10](#_Toc168000265)

[Figure 10: A partenenta obiectelor la clustere 10](#_Toc168000266)

[Figure 11: Graficul clusterizarii in functie de scorul cibil si venit 11](#_Toc168000267)

[Figure 12: Data frame clustere 11](#_Toc168000268)

[Figure 13: Graficul apartenentelor la clustere dupa PCA 12](#_Toc168000269)

[Figure 14: Determinare coeficientilor silueta 12](#_Toc168000270)

[Figure 15: Silhouette plot 12](#_Toc168000271)

[Figure 16: Determinam silueta individuala a fiecarui obiect 13](#_Toc168000272)

[Figure 17: Plot cibil\_scor si loan\_term 14](#_Toc168000273)

[Figure 18: Split 14](#_Toc168000274)

[Figure 19: Plot cibil\_score si loan\_term 15](#_Toc168000275)

[Figure 20: Regresia logistica cu glm 15](#_Toc168000276)

[Figure 21: Coeficenti model 16](#_Toc168000277)

[Figure 22: Functia prob 16](#_Toc168000278)

[Figure 23: Matricea de confuzie-set de antrenare 17](#_Toc168000279)

[Figure 24: Acuratetea modelului 17](#_Toc168000280)

[Figure 25: Curba ROC 18](#_Toc168000281)

[Figure 26: transformarea unei variabile continue într-una categorică pe baza distribuției sale. 19](#_Toc168000282)

[Figure 27: tabela nivelurilor de venit 19](#_Toc168000283)

[Figure 28: Summary 20](#_Toc168000284)

[Figure 29: Coeficienți model 21](#_Toc168000285)

[Figure 30: Predictie 22](#_Toc168000286)

[Figure 31: Probabilitati 22](#_Toc168000287)

[Figure 32: Matricea de confuzie 22](#_Toc168000288)

[Figure 33: Accuracy 23](#_Toc168000289)

[Figure 34: Functia split 23](#_Toc168000290)

[Figure 35: Set de antrenare - arbori de regresie 24](#_Toc168000291)

[Figure 36: Set de testare - arbori de regresie 24](#_Toc168000292)

[Figure 37: Arbore de regresie 24](#_Toc168000293)

[Figure 38: funcția cv.trees$size 25](#_Toc168000294)

[Figure 39: funcția cv.trees$dev 25](#_Toc168000295)

[Figure 40: Reprezentăm grafic eroarea în funcție de marimea arborelui 25](#_Toc168000296)

[Figure 41: Devianța reprezentată grafic 26](#_Toc168000297)

[Figure 42: Predictie 26](#_Toc168000298)

[Figure 43: Plot predicții 27](#_Toc168000299)

[Figure 44: Eroarea de predicție 27](#_Toc168000300)

[Figure 45: Arborele multinomial initial 28](#_Toc168000301)

[Figure 46: Predicție pentru setul de testare 29](#_Toc168000302)

[Figure 47: Matricea de confuzie a arborelui de clasificare initial 29](#_Toc168000303)

[Figure 48: Validare incrucisata 29](#_Toc168000304)

[Figure 49: Arborele multinomial curatat 30](#_Toc168000305)

[Figure 50: Matricea de confuzie a arborelui multinomial curatat 31](#_Toc168000306)

[Figure 51: Acuratetea arborelui multinomial curatat 31](#_Toc168000307)

[Figure 52: Matrice confuzie 32](#_Toc168000308)

[Figure 53: Accuracy model KNN 32](#_Toc168000309)

[Figure 54: k-Nearest Neighbors 33](#_Toc168000310)

[Figure 55: Variația Acurateței Modelului în Funcție de Valorile lui k pentru K-Nearest Neighbors 33](#_Toc168000311)

[Figure 56: Matricea de confuzie pt modelul cu k optim 34](#_Toc168000312)

[Figure 57: Acuratetea 34](#_Toc168000313)

[Figure 58: Curba ROC pentru Modelul KNN Optimizat 34](#_Toc168000314)

[Figure 59: Dimensiunea seturilor de date de antrenare și testare 35](#_Toc168000315)

[Figure 60:Output rețea 35](#_Toc168000316)

[Figure 61: testare retelei cu ajutorul temp\_set 36](#_Toc168000317)

[Figure 62: Valori reale vs Valori previzionate 36](#_Toc168000318)

[Figure 63: Matricea de confuzie si Funcția classAgreement 37](#_Toc168000319)

1. ANEXE

library(dplyr)

library(rpart)

date <- read.csv("loan\_approval\_dataset.csv")

date <- as.data.frame(date)

date <- date[, -1]

date %>% head()

table(date$loan\_status)

str(date)

date <- date %>%

mutate(

education = ifelse(education == " Graduate", 1, 0),

self\_employed = ifelse(self\_employed == " Yes", 1, 0),

loan\_status = ifelse(loan\_status == " Approved", 1, 0)

)

date$education <- as.integer(date$education)

date$self\_employed <- as.integer(date$self\_employed)

date$loan\_status <- as.integer(date$loan\_status)

#### Statistici descriptive ####

summary(date)

library(psych)

describe(date)

#Histograme

# Generare histograme pentru variabilele numerice

par(mfrow = c(3, 3))

hist(

date$income\_annum,

main = "Histogram of Annual Income",

xlab = "Annual Income",

col = "blue",

breaks = 50

)

hist(

date$loan\_amount,

main = "Histogram of Loan Amount",

xlab = "Loan Amount",

col = "green",

breaks = 50

)

hist(

date$loan\_term,

main = "Histogram of Loan Term",

xlab = "Loan Term (months)",

col = "red",

breaks = 20

)

hist(

date$cibil\_score,

main = "Histogram of CIBIL Score",

xlab = "CIBIL Score",

col = "purple",

breaks = 50

)

hist(

date$residential\_assets\_value,

main = "Histogram of Residential Assets Value",

xlab = "Residential Assets Value",

col = "orange",

breaks = 50

)

hist(

date$commercial\_assets\_value,

main = "Histogram of Commercial Assets Value",

xlab = "Commercial Assets Value",

col = "cyan",

breaks = 50

)

hist(

date$luxury\_assets\_value,

main = "Histogram of Luxury Assets Value",

xlab = "Luxury Assets Value",

col = "pink",

breaks = 50

)

# Generare boxplot-uri pentru variabilele numerice

par(mfrow = c(3, 3))

boxplot(

date$income\_annum,

main = "Boxplot of Annual Income",

ylab = "Annual Income",

col = "blue"

)

boxplot(

date$loan\_amount,

main = "Boxplot of Loan Amount",

ylab = "Loan Amount",

col = "green"

)

boxplot(date$loan\_term,

main = "Boxplot of Loan Term",

ylab = "Loan Term (months)",

col = "red")

boxplot(

date$cibil\_score,

main = "Boxplot of CIBIL Score",

ylab = "CIBIL Score",

col = "purple"

)

boxplot(

date$residential\_assets\_value,

main = "Boxplot of Residential Assets Value",

ylab = "Residential Assets Value",

col = "orange"

)

boxplot(

date$commercial\_assets\_value,

main = "Boxplot of Commercial Assets Value",

ylab = "Commercial Assets Value",

col = "cyan"

)

boxplot(

date$luxury\_assets\_value,

main = "Boxplot of Luxury Assets Value",

ylab = "Luxury Assets Value",

col = "pink"

)

boxplot(

date$bank\_assets\_value,

main = "Boxplot of Bank Assets Value",

ylab = "Bank Assets Value",

col = "brown"

)

# Generare diagramă de bare pentru variabilele categoriale

barplot(

table(date$no\_of\_dependents),

main = "Number of Dependents",

xlab = "Number of Dependents",

ylab = "Frequency",

col = "lightblue"

)

barplot(

table(date$education),

main = "Education Level",

xlab = "Education Level",

ylab = "Frequency",

col = c("lightgreen", "lightcoral"),

names.arg = c("Not Graduate", "Graduate")

)

barplot(

table(date$self\_employed),

main = "Self Employed Status",

xlab = "Self Employed",

ylab = "Frequency",

col = c("lightgrey", "lightyellow"),

names.arg = c("No", "Yes")

)

barplot(

table(date$loan\_status),

main = "Loan Status Distribution",

xlab = "Loan Status",

ylab = "Frequency",

col = c("red", "green"),

names.arg = c("Rejected", "Approved")

)

library(corrplot)

corrplot(cor(date), method = "number")

install.packages('caTools')

library(caTools)

################REGRESIE LOGISTICA#####################

#1#

date\_scale <- scale(date)

date\_scale %>% head()

date\_scale <- as.data.frame(date\_scale)

date\_scale$loan\_status <- date$loan\_status

date\_scale$loan\_status <- as.factor(date\_scale$loan\_status)

head(date\_scale)

View(date\_scale)

set.seed(123)

split <- sample.split(date$loan\_status, SplitRatio = 0.8)

split

library(ggplot2)

plot <-

ggplot(data = date,

aes(

x = date$cibil\_score,

y = date$loan\_term,

col = loan\_status

))

plot <- plot + geom\_point(aes(size = 5))

plot

training <- subset(date\_scale, split == TRUE)

testing <- subset(date\_scale, split == FALSE)

plot <-

ggplot(data = date\_scale,

aes(

x = date$cibil\_score,

y = date$loan\_term,

col = loan\_status

))

plot <- plot + geom\_point(aes(size = 5))

plot

View(testing)

# Antrenarea modelului de regresie logistică

logistic\_model <-

glm(loan\_status ~ cibil\_score + loan\_term,

data = training,

family = binomial)

summary(logistic\_model)

# Exponențierea coeficienților

exp(coef(logistic\_model))

# Previzionarea probabilităților pe setul de testare

prob <- predict(logistic\_model, testing, type = "response")

prob

# Construirea vectorului de predicții

pred <- rep("0", dim(testing)[1])

pred[prob > 0.5] <- "1"

pred

# Matricea de confuzie

conf\_matrix <- table(pred, testing$loan\_status)

print(conf\_matrix)

# Calcularea acurateței modelului

accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)

accuracy

print(paste(

"Acuratețea regresiei logistice este:",

round(accuracy \* 100, 2),

"%"

))

#Curba ROC

prob <- predict(logistic\_model, testing, type = "response")

# Construirea obiectului de predicție

pred <- prediction(prob, testing$loan\_status)

# Performanța modelului pentru curba ROC

perf <- performance(pred, measure = "tpr", x.measure = "fpr")

# Plotarea curbei ROC

plot(perf, col = "blue", main = "Curba ROC pentru Modelul Logistic")

# Linie de referință pentru clasificare aleatorie

abline(

a = 0,

b = 1,

lty = 2,

col = "red"

)

# Calcularea AUC

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

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

print(paste("AUC:", round(auc\_value, 2)))

########REGRESIE MULTINOMIALA###########################

library(nnet)

library(MASS)

library(caTools)

set.seed(88)

date\_multinom <- date\_scale

date\_multinom$income\_category <- cut(

date\_multinom$income\_annum,

breaks = quantile(date\_multinom$income\_annum, probs = seq(0, 1, by = 0.25)),

include.lowest = TRUE,

labels = c("low", "medium-low", "medium-high", "high")

)

date\_multinom <- date\_multinom[, -4]

date\_multinom %>% head()

date\_multinom$income\_category <-

as.factor(date\_multinom$income\_category)

table(date\_multinom$income\_category)

training <- subset(date\_multinom, split == TRUE)

testing <- subset(date\_multinom, split == FALSE)

model <- multinom(income\_category ~ ., data = training, trace = FALSE)

summary(model)

exp(coef(model))

predictions <- predict(model, testing)

predictions

predict(model, testing, type = "prob")

conf\_matrix <-

confusionMatrix(table(predictions, testing$income\_category))

conf\_matrix

accuracy <- mean(predictions == testing$income\_category)

accuracy

##########ARBORE DE REGRESIE#####################################################

library(tree)

library(ISLR)

# Împărțim setul de date în antrenare și testare

set.seed(123)

split <-

createDataPartition(y = date$income\_annum,

p = 0.5,

list = FALSE)

split

# Definim setul de antrenare și setul de testare

train <- date\_scale[split,]

test <- date\_scale[-split,]

train

test

# Construim arborele de regresie

arbore\_regresie <- rpart(income\_annum ~ ., data = train)

arbore\_regresie2 <- tree(income\_annum ~ ., data = train)

# Reprezentăm grafic arborele

rpart.plot(arbore\_regresie)

# Validare încrucișată

cv\_regresie <- cv.tree(arbore\_regresie2)

cv\_regresie$size

cv\_regresie$dev

# Reprezentăm grafic eroarea în funcție de marimea arborelui

plot(cv\_regresie$size, cv\_regresie$dev, type = 'b')

# Construim arborele curățat

arbore\_regresie\_curatat <- prune.tree(arbore\_regresie2, best = 5)

plot(arbore\_regresie\_curatat)

text(arbore\_regresie\_curatat, pretty = 0)

# Predictie pe setul de testare

pred <- predict(arbore\_regresie\_curatat, test)

pred

# Calculăm eroarea de predicție

mean((pred - test$income\_annum) ^ 2)

plot(x = pred , y = test$income\_annum)

abline(0, 1)

###################ARBORE DE CLASIFICARE#############

# Transformăm loan\_status într-un factor

date$loan\_status <- as.factor(date$loan\_status)

date %>% head()

# Împărțirea setului de date în antrenare (60%) și testare (40%)

set.seed(111)

split <-

createDataPartition(y = date$loan\_status,

p = 0.6,

list = FALSE)

# Setul de antrenare

train <- date\_scale[split,]

# Setul de testare

test <- date\_scale[-split,]

# Construim arborele de clasificare

arbore <- tree(loan\_status ~ ., data = train, method = "class")

arbore2 <- rpart(loan\_status ~ ., data = train, method = "class")

rpart.plot(arbore2)

# Predicție pentru setul de testare

pred <- predict(arbore, test, type = 'class')

pred

# Calculăm acuratețea

accuracy <- mean(pred == test$loan\_status)

accuracy

print(paste(

"Acuratețea inițială a modelului este:",

round(accuracy \* 100, 2),

"%"

))

# Matricea de confuzie

confusionMatrix <- table(pred, test$loan\_status)

print(confusionMatrix)

# Îmbunătățirea acurateței predictiei utilizând validarea încrucișată

set.seed(12)

cv.tree1 <- cv.tree(arbore, FUN = prune.misclass)

# Reprezentăm grafic eroarea în funcție de mărimea arborelui

plot(

cv.tree1$size,

cv.tree1$dev,

type = 'b',

xlab = "Număr de noduri terminale",

ylab = "Eroare de validare",

main = "Validare încrucișată"

)

# Curățăm arborele de clasificare

arbore1 <-

prune.misclass(arbore, best = cv.tree1$size[which.min(cv.tree1$dev)])

plot(arbore1)

text(arbore1)

# Predicție pentru setul de testare cu arborele curățat

prune.predictie <- predict(arbore1, test, type = 'class')

# Calculăm acuratețea pentru arborele curățat

prune\_accuracy <- mean(prune.predictie == test$loan\_status)

prune\_accuracy

print(paste(

"Acuratețea modelului curățat este:",

round(prune\_accuracy \* 100, 2),

"%"

))

#Matricea de confuzie pentru arborele curățat

confusionMatrixCV <- table(prune.predictie, test$loan\_status)

print(confusionMatrixCV)

#################### KNN CLASIFICARE###############

library(FNN)

library(caret)

library(caTools)

library(tidyverse)

date$loan\_status <- as.factor(date$loan\_status)

data\_subset <- date\_scale

# Împărțirea setului de date în antrenare (80%) și testare (20%)

set.seed(123)

split <-createDataPartition(data\_subset$loan\_status, p = 0.8, list = FALSE)

train\_data <- data\_subset[split,]

test\_data <- data\_subset[-split,]

# Normalizarea datelor

train\_data %>% head()

# Antrenarea modelului KNN cu k = 3

knn\_model <-

knn(

train = train\_data[, -12],

test = test\_data[, -12],

cl = train\_data$loan\_status,

k = 3

)

# Evaluarea performanței modelului

conf\_matrix <- table(Predicted = knn\_model, Actual = test\_data$loan\_status)

print(conf\_matrix)

accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)

print(paste("Acuratețea modelului KNN este:", round(accuracy \* 100, 2), "%"))

# Determinarea valorii optime a lui k

train\_control <- trainControl(method = "cv", number = 5)

knn\_tuned <-

train(

loan\_status ~ .,

data = train\_data,

method = "knn",

tuneLength = 20,

trControl = train\_control,

)

sum(is.na(train\_data))

sum(is.na(test\_data))

# Afișarea modelului optim

print(knn\_tuned)

plot(knn\_tuned)

# Predictii cu modelul optim

best\_k <- knn\_tuned$bestTune$k

knn\_best\_model <-

knn(

train = train\_data[, -12],

test = test\_data[, -12],

cl = train\_data$loan\_status,

k = best\_k

)

# Evaluarea performanței modelului optim

conf\_matrix\_best <-table(Predicted = knn\_best\_model, Actual = test\_data$loan\_status)

print(conf\_matrix\_best)

accuracy\_best <- sum(diag(conf\_matrix\_best)) / sum(conf\_matrix\_best)

print(paste(

"Acuratețea modelului KNN optim este:",

round(accuracy\_best \* 100, 2),

"%"

))

roc\_curve <- roc(as.numeric(test\_data$loan\_status), as.numeric(as.character(knn\_best\_model)))

plot(roc\_curve)

auc(roc\_curve)

###################################RETELE NEURONALE###############

install.packages("neuralnet")

install.packages("e1071")

library(neuralnet)

library(e1071)

loan\_data <- date

loan\_data$education <- as.factor(loan\_data$education)

loan\_data$self\_employed <- as.factor(loan\_data$self\_employed)

loan\_data$loan\_status <- as.factor(loan\_data$loan\_status)

loan\_data$education <- as.numeric(loan\_data$education)

loan\_data$self\_employed <- as.numeric(loan\_data$self\_employed)

loan\_data$loan\_status <- as.numeric(loan\_data$loan\_status) - 1 # Ajustare pentru a avea valori 0 și 1

# Scalare date numerice

scale\_columns <- c('income\_annum', 'loan\_amount', 'loan\_term', 'cibil\_score',

'residential\_assets\_value', 'commercial\_assets\_value',

'luxury\_assets\_value', 'bank\_asset\_value')

loan\_data[scale\_columns] <- scale(loan\_data[scale\_columns])

# Împărțirea datelor în seturi de antrenare și testare

set.seed(1234567890)

trainset <- loan\_data[1:3200, ]

testset <- loan\_data[3201:nrow(loan\_data), ]

# Construirea rețelei neuronale

formula <- as.formula("loan\_status ~ no\_of\_dependents + education + self\_employed + income\_annum + loan\_amount + loan\_term + cibil\_score + residential\_assets\_value + commercial\_assets\_value + luxury\_assets\_value + bank\_asset\_value")

loan\_net <- neuralnet(formula, data=trainset, hidden=4, lifesign="minimal", linear.output=FALSE, threshold=0.1)

# Vizualizarea rețelei neuronale

plot(loan\_net, rep="best")

temp\_test <- subset(testset, select=c('no\_of\_dependents', 'education', 'self\_employed', 'income\_annum', 'loan\_amount', 'loan\_term', 'cibil\_score', 'residential\_assets\_value', 'commercial\_assets\_value', 'luxury\_assets\_value', 'bank\_asset\_value'))

compute\_result <- compute(loan\_net, temp\_test)

predictions <- compute\_result$net.result

results <- data.frame(actual=testset$loan\_status, prediction=predictions)

# Rotunjirea valorilor pentru predicții

results$prediction <- round(results$prediction)

# Convertirea în factor pentru comparație

results$actual <- as.factor(results$actual)

results$prediction <- as.factor(results$prediction)

# Matricea de confuzie

confusion\_matrix <- table(results$actual, results$prediction)

# Evaluarea performanței modelului

class\_agreement <- classAgreement(confusion\_matrix)

print(confusion\_matrix)

print(class\_agreement)

head(results, 20)

########################################CLUSTERIZARE FUZZY

date\_numeric <- date[, sapply(date, is.numeric)]

result <-

cmeans(

date\_numeric,

centers = 3,

iter.max = 100,

m = 2,

method = "cmeans"

)

print(result)

plot(date$income\_annum, date$cibil\_score, col = result$cluster)

points(result$centers[, c(4, 7)],

col = 1:3,

pch = 8,

cex = 2)

o <- order(result$cluster)

data.frame(date[o, ], Cluster = result$cluster[o])

clustered\_date <- data.frame(date, Cluster = result$cluster)

res.fanny <- fanny(date[1:500, ], 3)

data.frame(Cluster = res.fanny$clustering, date[1:500, ])

fviz\_cluster(

res.fanny,

ellipse.type = "norm",

repel = TRUE,

palette = "jco",

ggtheme = theme\_minimal(),

legend = "right"

)

fviz\_silhouette(res.fanny, palette = "jco", ggtheme = theme\_minimal())

print(res.fanny$silinfo)