UNIVERSITATEA BABEŞ-BOLYAI

Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor

Denumirea completă a programului de studii

Lucrare de licenţă / disertaţie

Absolvent,

Rareș-Andrei **Filip**

Coordonator ştiinţific,

Prof. univ. dr. Gheorghe-Cosmin **Silaghi**

2024

UNIVERSITATEA BABEŞ-BOLYAI

Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor

Denumirea completă a programului de studii

Lucrare de licenţă / disertaţie

Prezicerea tranzacțiilor frauduloase folosind machine learning

Absolvent,

Rareș-Andrei **Filip**

Coordonator ştiinţific,

Prof. univ. dr. Gheorghe-Cosmin **Silaghi**

2024

Rezumat

În ultimii ani, frauda cu carduri de credit a devenit o preocupare majoră datorită volumului tot mai mare de tranzacții online. Această lucrare explorează aplicarea algoritmilor de învățare automată pentru a îmbunătăți detectarea tranzacțiilor frauduloase în timp real. Prin analiza datelor provenite de la instituțiile financiare și utilizarea modelelor predictive, am urmărit să identific activitățile suspecte care ar putea indica fraude. Abordarea mea include implementarea modelelor Random Forest, XGBoost și rețele neuronale, care au demonstrat o acuratețe ridicată în identificarea tranzacțiilor frauduloase. Modelul Random Forest, cunoscut pentru acuratețea sa echilibrată, oferă capacități robuste de detectare cu o specificitate de 77%. Modelul XGBoost excelează în identificarea unui număr mai mare de tranzacții frauduloase, deși are tendința de a clasifica eronat unele tranzacții legitime ca fiind frauduloase. Rețelele neuronale s-au dovedit a fi cel mai performant model, având o specificitate de aproximativ 82%, fiind cel mai robust dintre toate modelele testate. Aplicația include, de asemenea, funcționalități pentru compararea modelelor și vizualizarea tiparelor tranzacțiilor, ajutând utilizatorii să înțeleagă și să atenueze riscurile de fraudă. Pe viitor, intenționez să extind funcționalitatea aplicației, să optimizez performanța modelelor și să integrez algoritmi avansați suplimentari. Descoperirile mele contribuie la domeniul securității financiare prin oferirea de instrumente practice pentru detectarea fraudei și evidențierea domeniilor de cercetare și dezvoltare ulterioară.

Cuprins

Introducere *1*

1. Identificarea și descrierea problemei *3*

1.1 Motivația **5**

1.2 Context **8**

1.2.1 Fațeta subiect 8

1.2.2 Fațeta utilizare 8

1.2.3 Fațeta IT 9

1.2.4 Fațeta dezvoltare 9

2. Revizuirea literaturii *10*

3. Metodologie *14*

3.1 Setul de date **14**

3.1.1 Colectarea datelor 14

3.1.2 Curățarea datelor 15

3.1.3 Ingineria atributelor 15

3.1.4 Analiza datelor 17

3.2 Metrici de evaluare **20**

3.2.1 Acuratețea (Accuracy) 21

3.2.2 Sensibilitatea (Sensitivity) 21

3.2.3 AUC-ROC (Area Under the Receiver Operating Characteristic Curve) 22

3.2.4 Precision-Recall AUC (PR AUC) 23

3.2.5 F1-Score 24

3.2.6 Specificitatea (Specificity) 25

3.3 Modele predictive **26**

3.3.1 Arbori de decizie 26

3.3.2 Random forest 29

3.3.3 Metode de boosting 30

3.3.4 Support Vector Machines 32

3.3.5 Rețele neuronale 34

3.4 Rezultate și discuții **38**

4. Aplicație *40*

4.1 Scopul și utlizarea 40

4.2 Elicitarea cerințelor 41

4.2.1 Cerințe funcționale 41

4.2.2 Cerințele de interfață cu utilizatorul 42

4.2.3 Cerințe de gestionare a datelor 42

4.3 Analiză 43

4.4 Proiectare 45

4.4.1 Arhitectura sistemului 45

4.4.2 Algoritmi și procese 46

4.5 Implementare 47

4.5.1 Configurarea conexiunii la baza de date 48

4.5.2 Backend 49

4.5.3 Frontend 52

4.5.4 Modele predictive 54

4.6 Testarea 56

5. Concluzii și dezvoltări viitoare ale aplicației 58

Bibliografie 60

Lista figuri si tabele

Figura 1 - Business Model Canvas 4
Figura 2 - Fishbone(cauză-efect) 6
Figura 3 - Archimate de motivație 7
Figura 4 - Diagrama de Flow 14
Figura 5 - Distribuție tranzacții pe categorii 18
Figura 6 – Distribuție sumă / fraudă 19
Figura 7 - Distribuția tranzacții fraudă/gen 19
Figura 8 - Diagrama fluxului de date 20
Figura 9 - Diagramă ROC curve 22
Figura 10 - Diagramă Precision Recall Curve 24
Figura 11 - Rpart Plot 27
Figura 12 - Misclassification rate 28
Figura 13 - Matricea de importanță XGB 31
Figura 14 - Diagramă use case 42
Figura 15 - UML de secvență 44
Figura 16 - Diagrama de componente 46

Tabel 1 - Atribute inițiale 15
Tabel 2 - Atribute modificate 16
Tabel 3 - Rezultate arbori de decizie 28
Tabel 4 - Rezultate random forest 30
Tabel 5 - Rezultate boosting 32
Tabel 6 - Rezultate SVM 33
Tabel 7 - Rezultate rețele neuronale 36
Tabel 8 - Rezultate finale 38
Tabel 9 - Rezultate XGBoost (testare) 39

Introducere

În epoca digitală în care trăim, frauda cu carduri de credit este o problemă des întâlnită și foarte periculoasă pentru siguranța financiară a consumatorilor și a instituțiilor financiare. Volumul tranzacțiilor efectuate pe internet a crescut enorm în ultimii ani, odată cu aceasta au crescut și oportunitățile pentru activități sofisticate de fraudă, după cum semnalează și Mastercard(2024), o corporație multinațională de servicii financiare. Detectarea acestor fraude la timp poate salva bani si proteja datele personale ale utilizatorilor.

Numeroasele forme de fraudă includ phising, clonarea cardurilor și atacurile de tip malware, care afectează datele personale și financiare ale utilizatorilor. Instituțiile financiare sunt sub obligate să investească resurse substanțiale în detectarea și prevenirea fraudei. Browne (2024) a prezentat aplicația de tip ChatGPT lansată de Mastercard pentru a ajuta băncile să detecteze fraudele. Detectarea în timp util a acestor fraude poate salva bani și proteja datele personale ale utilizatorilor. Învățarea automată și inteligența artificială sunt din ce în ce mai folosite pentru a analiza tranzacțiile și a identifica tipare suspecte.

Algoritmii de învățare automată sunt capabili să identifice comportamente anormale care pot indica activitate fraudulentă prin analiza unui volum mare de date în timp real. Descoperirea fraudelor este o sarcină foarte complexă de realizat. Așadar, selecția de instanțe adecvate, clasificarea instanțelor și metodele de clustering vor reuși să determine tranzacțiile fraudulente într-un set de date. Există multe activități care determină probabilitatea unei tranzacții fraudulente, cum ar fi comportamentul clienților, cheltuielile clienților și cercetarea anterioară a fraudelor și tiparele acestora. În consecință, orice metodă de detectare a fraudei ar trebui să aibă caracteristicile următoare:

- capacitatea să identifice cu precizie un număr cât mai mare de tranzacții fraudulente
- capacitatea de a identifica frauda în timp real
- nu ar trebui să caracterizeze tranzacțiile reale ca fraudă

După cum semnalează și Sardineni(2021), tranzacțiile fraudulente sunt tranzacții ilegale efectuate de altcineva în numele tău, fie prin furtul detaliilor cardului de credit sau ale contului

pentru folosirea acestora în scopuri personale. Voi prezenta câteva metode prin care tranzacțiile ilegale sunt realizate:

- Skimming - Această tehnică este folosită pentru a obține detaliile de pe banda magnetică a cardului de credit prin injectarea acestuia într-un dispozitiv electronic
- Phishing - Această tehnică este folosită pentru a obține detaliile cardului prin capcane de e-mail, unde se trimit mail-uri asemănătoare celor de la bănci și solicită detaliile private ale cardului
- Card Not Present Fraud – Această tehnică folosește numărul cardului și data de expirare al acestuia pentru a face tranzactii fie prin e-mail, fie prin telefon, fără a utiliza efectiv cardul.
- Preluarea contului – Aceasta are loc atunci când informații confidențiale sunt împărtășite unui străin și acesta le folosește în beneficiul personal.
- Carduri capturate în timpul expedierii – Aceasta are loc atunci când un nou card de credit este emis si ajunge în mâinile greșite.
- Pierderea cardului – Atunci cand un card este pierdut și ajunge în mâinile greșite, există șansa ca informațiile confidențiale să fie furate.
- Site-uri false – Acestea atrag atenția clienților făcându-i să creadă ca site-ul este autentic și îi determină să cumpere produse online, furnizând detaliile cardului de credit.

Această lucrare prezintă procesul de dezovltare a modelelor predictive pentru detectarea tranzacțiilor fraudulente, urmat de integrarea celor mai bune modele într-o aplicație web folosind ShinyR. Scopul principal este de a demonstra eficiența și utilitatea acestora atât pentru o instituție financiară, cât și pentru un utilizator extern.

Metodologia pe care o voi utiliza pune accent pe implementarea modelelor predictive. Inițial, modelele vor fi create și testate in RStudio folosind diverși algoritmi de machine learning. Aceste modele vor fi selectate bazat pe performanța lor de a detecta tranzacții suspecte.

Pentru a facilita utilizarea practică a acestor modele, unele din ele vor fi integrate într-o aplicație web folosind framework-ul ShinyR. Acesta facilitează construcția de interfețe web care permit utilizatorilor sa interacționeze cu datele si modelele predictive într-un mod vizual. Baza de date a aplicației este gestionată folosind SQLite.

1. Identificarea și descrierea problemei

Pentru instituțiile financiare, fraudele financiare au devenit o problemă semnificativă ca urmare a creșterii majore a utilizării cardurilor de credit și a tranzacțiilor online. Aceste tranzacții pot avea repercusiuni grave, cum ar fi pierderi financiare mari, prejudicii pentru imaginea companiei și pierderea încrederii clienților. În timp ce tehnicile de fraudă devin din ce în ce mai complexe, infractorii continuă să descopere noi moduri de a exploata vulnerabilitățile sistemelor financiare, ceea ce exacerbează această problemă.

În aceste circumstanțe, instituțiile financiare trebuie să dezvolte și să implementeze metode eficiente de detectare și prevenire a fraudei. Tradițional, detectarea fraudei a folosit reguli predefinite și alerte manuale, dar aceste tehnici nu sunt suficiente pentru a gestiona complexitatea și cantitatea mare de tranzacții care au loc în prezent. Prin urmare, modelele predictive și a algoritmilor de machine learning a devenit o soluție promițătoare pentru a îmbunătăți detectarea fraudei.

Principalul scop al acestei lucrări este de a crea si evalua diverse modele predictive pentru detectarea tranzacțiilor frauduloase. Utilizarea acestor modele ar facilita identificarea anomaliilor și comportamentelor suspecte, oferind protecție atât instituțiilor bancare cât si utilizatorilor.

Ca răspuns pentru aceste probleme, am propus dezvoltarea unei aplicații care detectează aceste fraude. În prealabil, voi avea o fază de creare, analizare și comparare a modelelor predictive în funcție de diferite rezultate pentru a alege cele mai bune modele în detectarea acestor fraude. În alegerea acestor modele, voi analiza mai multi indicatori de performanță cum ar fi: AUC-ROC, specificitate, sensibilitate și precizie.

Întrebările la care mi-am propus să raspund în această lucrare sunt:

1. Ce metode de învățare automată sunt cele mai eficiente în detectarea fraudelor cu carduri de credit?
2. Care sunt caracteristicile tranzacțiilor ce contribuie cel mai mult la clasificarea lor ca fiind frauduloase?
3. Cum poate fi îmbunătățită acuratetea predicțiilor folosind datele tranzacționale disponibile?

O aplicație care să integreze aceste modele analizate ar fi ideală pentru rezolvarea problemei de fraudă.

1. Key Partners	2. Key activities	3. Value Proposition	4. Customer Relationship	5. Customer Segments
Instituții financiare (bănci, companii de carduri de credit) Companii de securitate cibernetică Furnizori de tehnologie (companii care oferă infrastructură de cloud) Universități și institute de cercetare (pentru colaborări în dezvoltarea algoritmilor de machine learning) Consultanți de securitate	Dezvoltarea și testarea algoritmilor de machine learning pentru detectarea fraudei Monitorizarea și actualizarea continuă a modelelor predictive Implementarea și întreținerea aplicației web Shiny Furnizarea de suport tehnic și servicii de consultanță pentru instituțiile financiare	Detectare rapidă și precisă a tranzacțiilor frauduloase Protejarea resurselor financiare și a reputației instituțiilor financiare Interfață web intuitivă și ușor de utilizat Reducerea pierderilor financiare și a timpului de răspuns la fraude Îmbunătățirea securității și încrederii clienților	Suport tehnic dedicat (telefon, email, chat) Servicii de consultanță pentru implementare și optimizare Training și workshop-uri pentru utilizatori Feedback constant și actualizări de software	Bănci și instituții financiare Procesatori de plăți online Companii de carduri de credit Comercianți care gestionează un volum mare de tranzacții Clienți finali (consumatori) care doresc protecție împotriva fraudei
	6. Key Resources		**7. Channels**	
	Echipa de dezvoltare software și data scientists Infrastructura IT (servere, baze de date, servicii de cloud) Algoritmi și modele de machine learning Baze de date cu tranzacții istorice pentru antrenarea modelelor Parteneriate strategice cu instituții financiare și companii de securitate		Aplicație web (Shiny) disponibilă online Integrare API cu sistemele instituțiilor financiare Platforme de parteneri și marketplace-uri de software Evenimente și conferințe de securitate și tehnologie Website și campanii de marketing digital	

8. Cost	9. Revenue Streams
Costuri de dezvoltare software și întreținere: investiția personală în timpul dedicat dezvoltării și întreținerii platformei Costuri de marketing: Resurse alocate pentru promovarea aplicației pentru companii Suport si customer service: Inițial suportul va fi întreținut de mine, dar în viitor vor fi nevoie de resurse financiare pentru acest domeniu	Licențierea software-ului: vânzarea licențelor pentru utilizarea aplicației. Parteneriate și colaborări cu alte companii: venituri din parteneriate strategice și colaborări cu alte companii din domeniul securității și tehnologiei financiare.

Figura 1 - Business Model Canvas

1.1 Motivația

Principala sursă de motivație este reprezentată de o experiență personală neplăcută legată de acest domeniu. Anul trecut mi-am uitat cardul personal într-o benzinărie. Acesta a fost copiat și folosit pe site-uri necunoscute, fără 3D-Secure, pentru a efectua plăți cu el. Din păcate, sistemele băncii au depistat această fraudă abia aproximativ după trei luni, perioadă în care au avut loc multiple tranzacții fraudulente. Această întâmplare mi-a subliniat vulnerabilitatea sistemelor financiare actuale și necesitatea unor metode mai eficiente pentru detectarea și prevenirea fraudei.

Pe lângă motivații personale, există și motivații economice și de securitate care arată importanța creării unei astfel de aplicații. Instituțiile bancare se confruntă cu pierderi majore datorate fraudei. Spre exemplu Statele Unite ale Americii se confruntă cu sume uriașe pierdute din cauza fraudei, aproximativ 10 miliarde de dolari in 2023, dupa cum a semnalat și protecția consumatorilor din SUA (Federal Trade Comission). Prevenirea acestor pierderi este esențială pentru păstrarea stabilității financiare și a încrederii clienților. Un sistem eficient de detectare a fraudei poate reduce costurile operaționale prin automatizarea proceselor de monitorizare. Prin acest sistem se elimină necesitatea de intervenție manuală. În plus, detectarea acestor fraude în timp real duce la protejarea datelor personale a clienților, și automat la creșterea numărului de clienți mulțumiți.

Un alt aspect al motivației pentru crearea acestui proiect provine din dorința de a contribui la domeniul cercetării în machine learning și securitate financiară. Explorarea de noi metode și algoritmi pentru detectarea fraudelor poate aduce un plus în această industrie. Prin testarea diferitelor metode si modele predictive acest proiect poate sublinia soluții eficiente care încă nu au fost explorate pe deplin. Rezultatele acestei lucrări pot fi publicate, iar alți cercetători pot să le utilizeze, aducând un plus acestei industrii.

Prin urmare, dezvoltarea unor modele predictive și integrarea acestora într-o aplicație ar putea aduce mari beneficii pe termen lung pentru domeniul securității financiare.

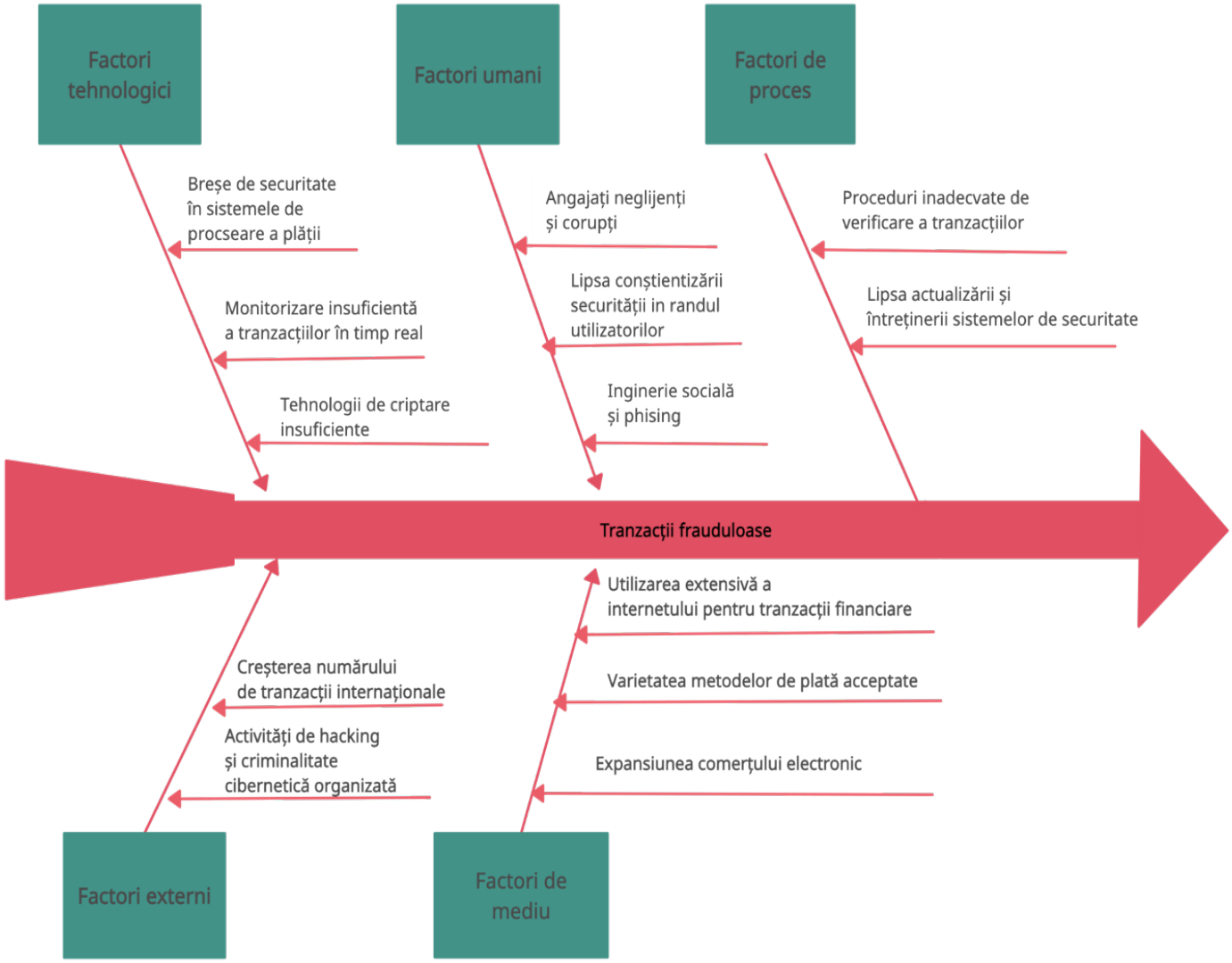


Figura 2 - Fishbone(cauză-efect)

Cu ajutorul unei diagrame de tipul Fishbone (cauză-efect) am identificat principalele probleme care contribuie la necesitatea dezvoltării unei aplicații care să detecteze eficient tranzacțiile frauduloase. Din figura aceasta se poate desprinde cauza generală care a dus la nevoia unor modele predictive și a unei aplicații. Aceste modele trebuie sa fie cât mai eficiente în detectarea fraudelor în mod corect.

Prin urmare, prin implementarea acestei aplicație se urmărește rezolvarea a câtor mai multe probleme din cele prezentate mai sus.

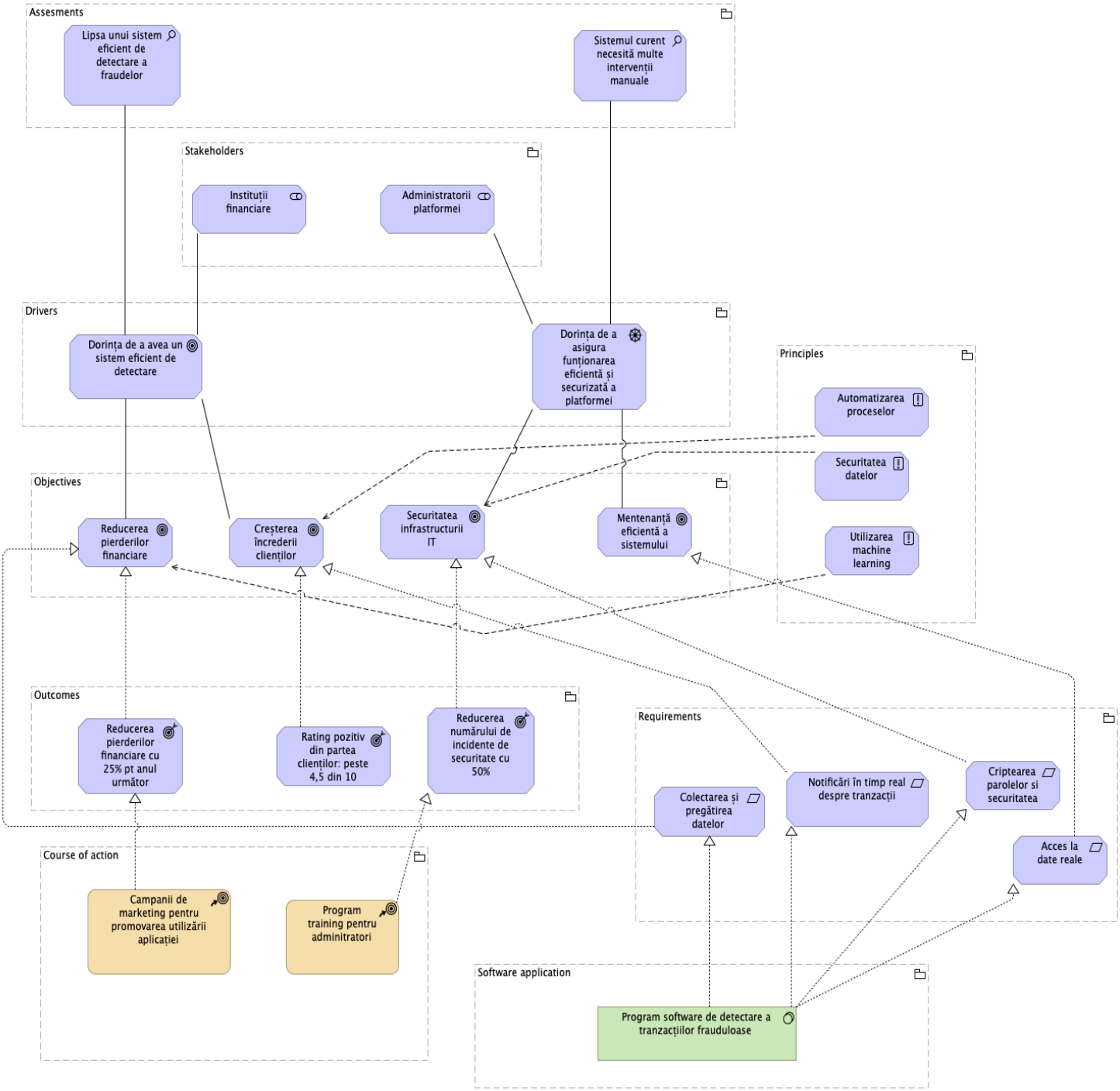


Figura 3 - Archimate de motivație

După etapa în care am clarificat situația de business și problemele cu care ne confruntăm, următorul pas este stabilirea motivației și a obiectivelor. Pentru acest lucru am creat această diagrama Archimate de motivație care include stakeholderii, obiectivele, principiile și planul de acțiune pentru proiectul meu.

1.2 Context

Aplicația mea este o aplicație destinată instituțiilor financiare, în special bancare, care au de suferit de pe urma unui număr în creștere al fraudelor în tranzacțiile cu carduri de credit și alte tipuri de plăți electronice. Într-un mediu financiar extrem de digitalizat, siguranța tranzacțiilor financiare este esențială pentru menținerea încrederii clienților și pentru protejarea resurselor financiare ale băncilor. Prin urmare, implementarea unui sistem eficient de detectare a fraudei este vitală pentru prevenirea unei astfel de pierderi financiare și pentru a oferi utilizatorilor o experiență mai bună prin reducerea fraudelor. Aplicația mea folosește modele predictive pentru a identifica tranzacțiile frauduloase în timp real.

Astfel, sistemul dezvoltat va fi structurat în patru dimensiuni esențiale, care tratează situațiile descrise anterior:

1. Fațeta subiect
2. Fațeta utilizare
3. Fațeta IT
4. Fațeta dezvoltare

1.2.1 Fațeta subiect

Fațeta subiect se referă la algoritmii și modelele care sunt utilizate pentru a detecta tranzacțiile frauduloase. Se utilizează diferiți algoritmi de machine learning și modele predictive diferite, cum ar fi: Naive-Bayes, Random Forest, Gradient Boosting Machines, Rețele Neuronale și Support Vector Machines. Aceste modele sunt antrenate pe seturi de date care conțin tranzacții etichetate ca frauduloase și nefrauduloase. Modelele sunt testate folosind metrici specifice de performanță, cum ar fi AUC-ROC [1], PRAUC, dar și specificitatea [2].

De asemenea, sistemul trebuie să poată procesa și analiza volume mari de date în timp real, pentru a putea emite avertizări prompte și precise legate de tranzacțiile suspecte.

1.2.2 Fațeta utilizare

Utilizatorii aplicației de detectare a tranzacțiilor frauduloase sunt profesioniști și activează în diferite departamente ale instituțiilor financiare. Fiecare dintre acești utilizatori are cerințe și sarcini diferite privind utilizarea sistemului. Utilizatorii acestei aplicații se împart în mai multe categorii: analiști de fraudă, personalul IT și manageri executivi. Se dorește ca utilizatorii acestei aplicații să poată reuși să distingă rapid si eficient tranzacțiile frauduloase de cele legitime, îmbunătățind astfel securitatea financiară a instituțiilor.

Analiștii de fraudă sunt cei care interacționează cel mai frecvent cu aplicația. Aceștia monitorizează constant stadiul tranzacțiilor. Ei folosesc aplicația pentru a accesa date detaliate și alerte în timp real pentru a identifica și analiza tranzacțiile frauduloase.

Personalul IT este responsabil pentru integrarea, configurarea și mentenanța tehnică a aplicației. Aceștia folosesc interfața tehnică a sistemului pentru a monitoriza securitatea. De asemenea, ei se ocupă de actualizarea periodică a algoritmilor si modelelor predictive, precum și de implementarea noilor funcționalități.

Managerii executivi accesează rapoartele și analizele pentru a-și face o imagine de ansamblu despre situație.

1.2.3 Fațeta IT

Fațeta IT a aplicației web pe care am conceput-o cuprinde totalitatea tehnologiilor implicate în dezvoltarea aplicației. Această componentă a fost esențială pentru a asigura funcționarea eficientă a sistemului. Platforma este o aplicație web care folosește framework-ul Shiny, scrisă în limbajul de programare R, hosting-ul fiind asigurat de un server extern, iar codul aplicației fiind găzduit pe GitHub. Baza de date utilizată pentru stocarea și gestionarea datelor aplicației este SQLite. SQLite este cel mai utilizat motor de baza de date din lume.

1.2.4 Fațeta dezvoltare

Aplicația web pentru detectarea tranzacțiilor frauduloase a fost realizată pentru a satisface nevoile instituțiillor financiare în prevenirea și detecția activităților frauduloase într-un mod eficient și precis. Acest software furnizează un sistem intuitiv și prietenos, atât pentru utilizatorii finali, cât și pentru administrator, care permite controlul pe trasele tranzacțiilor și analiza acestora în timp real respectând astfel aspectele financiare ale instituției. Astfel, selectarea unui model de dezvoltare potrivit arhitecturii și obiectivului aplicației reprezintă un aspect important, care asigură structura, fragmentarea și organizarea eficientă a metodelor de lucru. Pentru aplicația de detectare a tranzacțiilor frauduloase, metodologia care corespunde cel mai bine nevoilor și obiectivelor este Agile. Agile implementează anumite practici care reduc timpul de dezvoltare, facilitează implementarea de noi funcționalități și permit rezolvarea problemelor aplicației în timp real.

Din această categorie, Scrumban este framework-ul care promovează adaptări permanente pentru sistemul implementat, ajutând echipa să îmbunătățească modul în care se face munca. Combinând structura Scrum și flexibilitatea Kanban, am creat un cadru hibrid care permite creșterea versatilității și agilității în gestionarea fluxurilor de lucru

2. Revizuirea literaturii

După cum prezintă și Awoyemi JO (2017), abordările de machine learning joacă un rol crucial în multe domenii eficiente de procesare datelor, unul dintre acestea fiind identificarea fraudei cu carduri. Prin cercetările anterioare, s-au sugerat mai multe metode pentru detectarea fraudei, inclusiv strategii supervizate și nesupervizate, precum și strategii hibride. Identificarea fraudei se concentrează pe interpretarea acțiunilor cardului în timpul achizițiilor. Cele mai implementate strategii includ rețele neuronale artificiale (ANN), algoritmi genetici (GA), mașini de suport vectorial (SVM) și procese pentru Naive Bayes (NB).

În cadrul acestor metode, regresia logistică și analiza Naive Bayes sunt utilizate frecvent pentru detectarea fraudei cu carduri de credit. Arborii de decizie, machine learning și regresia logistică sunt evaluate în contextul detectării fraudei. Fiecare dintre aceste abordări oferă perspective unice și contribuții semnificative în îmbunătățirea identificării tranzacțiilor frauduloase.

Regresia logistică este una dintre metodele statistice cele mai des utilizate în detectarea fraudei. Aceasta implică modelarea probabilității ca o tranzacție să fie frauduloasă pe baza unor variabile explicative. Cheng(2023) demonstrează eficiența regresiei logistice în identificarea fraudelor în tranzacțiile cu carduri de credit. Acesta reușeste sa obțină un acuratete de 86,31%. Studiul utilizează un set de date de un milion de tranzacții și utilizează regresia logistică pentru a detecta fraude. Cheng subliniază importanța transformării logaritmice a anumitor variabile și utilizarea unui model de regresie logistică generalizată pentru a îmbunătăți acuratețea predicțiilor. De asemenea, este evidențiat faptul că utilizarea unui PIN și a unui cip reduce semnificativ probabilitatea de fraudă, în timp ce comenzile online și prețurile ridicate sunt corelate pozitiv cu apariția fraude.

Random Forest este o metodă populară în detectarea fraudei datorită capacității sale de a gestiona seturi mari de date și de a trata variabile complexe. Shiyang Xuan (2018) a demonstrat eficiența utilizării unui clasificador Random Forest pentru detectarea tranzacțiilor frauduloase. În acest studiu, modelul a fost antrenat pe un set de date mare și a reușit să obțină o acuratețe de 96,77%, evidențiind robustețea metodei în identificarea activităților suspecte. Studiul a utilizat două tipuri de păduri aleatoare, una bazată pe Random Trees și alta pe CART, și a concluzionat că modelul bazat pe CART a avut performanțe superioare. Lucrarea detaliază procesul de antrenare a modelului folosind datele de tranzacții furnizate de o companie de e-commerce din China. Datele au fost împărțite în seturi de antrenament și de testare, iar performanța celor două modele Random Forest a fost comparată.

Modelul bazat pe CART a demonstrat o acuratețe mai mare, precum și o rată de recall superioară, ceea ce indică o capacitate mai bună de a detecta tranzacțiile frauduloase. Studiul a evidențiat, de asemenea, importanța abordării problemelor de dezechilibru al datelor, care sunt frecvente în seturile de date de fraudă, utilizând metode de sub-eșantionare pentru a echilibra proporția de tranzacții legale și frauduloase.

Support Vector Machines (SVM) sunt utilizate frecvent în detectarea fraudelor cu carduri de credit datorită capacității lor de a gestiona date dezechilibrate și de a oferi rezultate precise. Xia Jianglin(2022) a folosit algoritmul SVM pentru a construi modele capabile să detecteze fraudele în tranzacțiile cu carduri de credit. Studiul a utilizat un set de date de pe Kaggle și a arătat că modelul optimizat a obținut un AUC de 0.90 și un F1-score de 0.26. Parametrii optimizați au inclus un factor de regularizare C=10 și un kernel rbf cu gamma=0.01. Support Vector Machines (SVM) sunt utilizate frecvent în detectarea fraudelor cu carduri de credit datorită capacității lor de a gestiona date dezechilibrate și de a oferi rezultate precise. Xia Jianglin a folosit algoritmul SVM pentru a construi modele capabile să detecteze fraudele în tranzacțiile cu carduri de credit. Studiul a utilizat un set de date de pe Kaggle și a arătat că modelul optimizat a obținut un AUC de 0.90 și un F1-score de 0.26. Parametrii optimizați au inclus un factor de regularizare C=10 și un kernel rbf cu gamma=0.01.

Acest studiu subliniază importanța ajustării hiperparametrilor pentru a obține performanțe optime în detectarea fraudelor. Autorii au reușit să găsească combinația ideală de parametri pentru a maximiza scorul AUC și pentru a reduce fenomenul de overfitting. De asemenea, cercetarea a demonstrat că, deși SVM a obținut rezultate bune, există întotdeauna posibilități de îmbunătățire, cum ar fi utilizarea unor metode suplimentare de reechilibrare a datelor, cum ar fi Synthetic Minority Oversampling Technique (SMOTE) (Nitesh V Chawla, 2002). Aceste ajustări și îmbunătățiri constante sunt esențiale pentru menținerea relevanței și eficienței algoritmilor de detectare a fraudelor în fața metodelor din ce în ce mai sofisticate utilizate de fraudatori.

Trivedi, Simaiya și Lilhore (2020) prezintă un model eficient pentru detectarea fraudei cu carduri de credit folosind diverse metode de machine learning. Autorii propun un sistem de detectare a fraudei care include un mecanism de feedback pentru a îmbunătăți performanța clasificatorului. Studiul utilizează seturi de date reale pentru antrenarea și testarea modelului, evidențiind eficiența algoritmilor de machine learning precum Random Forest și Gradient Boosting. În plus, lucrarea discută importanța preprocesării datelor și echilibrării setului de date pentru a îmbunătăți acuratețea detectării. Rezultatele obținute arată că modelul propus oferă o performanță superioară în comparație cu alte metode tradiționale de detectare a fraudei.

Un alt aspect interesant abordat în document este utilizarea tehnicilor de undersampling și oversampling pentru a gestiona problema dezechilibrului datelor. Prin echilibrarea setului de date, modelul devine mai capabil să detecteze atât tranzacțiile frauduloase, cât și pe cele legitime. Lucrarea include și o analiză detaliată a performanței modelului folosind metrici precum acuratețea, precizia, recall-ul și scorul F1. Modelul propus este testat pe un set de date de tranzacții reale, iar rezultatele arată că acesta poate detecta fraudele cu o rată de succes semnificativ mai mare comparativ cu metodele existente. În plus, autorii discută implementarea practică a modelului într-un sistem de monitorizare a tranzacțiilor în timp real, subliniind beneficiile unui astfel de sistem pentru instituțiile financiare.

Baabdullah (2024) introduce un sistem inovator de detectare a fraudei cu carduri de credit, integrând învățarea federată (FL) cu tehnologia blockchain. Această integrare asigură păstrarea confidențialității și protecția datelor, esențiale în tranzacțiile financiare. În cadrul învățării federate, modelul global de învățare este stabilit pe serverul cloud și transmite parametrii inițiali către modele locale de învățare aflate pe noduri de tip fog (bănci). Fiecare bancă își antrenează local modelul de învățare, asigurând confidențialitatea datelor, și trimite înapoi parametrii actualizați către modelul global. Blockchain-ul este utilizat pentru a oferi stocare descentralizată, rezistență la manipulare și trasabilitate, sporind securitatea și confidențialitatea datelor procesate în detectarea fraudei. Sistemul utilizează trei algoritmi de machine learning și rețele neuronale profunde: Random Forest (RF), rețele neuronale convoluționale (CNN) și memorie pe termen lung și scurt (LSTM). Aceste algoritme, împreună cu tehnici de optimizare profundă precum ADAM, SGD și MSGD, sunt utilizate pentru a îmbunătăți performanța modelului. Sistemul abordează problema seturilor de date dezechilibrate folosind tehnica Synthetic Minority Oversampling Technique (SMOTE) pentru a echilibra setul de date înainte de antrenarea modelului. Această tehnică contribuie la îmbunătățirea performanței clasificării și a acurateței predicțiilor. Rezultatele experimentale demonstrează eficiența și eficacitatea cadrului propus. Integrarea învățării federate și a blockchain-ului îmbunătățește performanța clasificării și acuratețea predicțiilor, păstrând în același timp confidențialitatea datelor. Sistemul este testat pe seturi de date reale de tranzacții cu carduri de credit, evidențiind aplicabilitatea sa practică și robustețea în detectarea tranzacțiilor frauduloase. Metodologia propusă include un model global de învățare pe serverul cloud care distribuie parametrii către modelele locale de învățare de pe nodurile fog. Aceste actualizări sunt trimise înapoi la modelul global, asigurând o îmbunătățire continuă a performanței. Lucrarea subliniază importanța unui sistem robust de detectare a fraudei în contextul creșterii continue a numărului de tranzacții online și a complexității atacurilor

frauduloase. Gregorius Airlangga (2024) evaluează eficiența diferitelor modele de machine learning în detectarea fraudei cu carduri de credit, folosind un set de date de 555,719 tranzacții. Studiul compară modele tradiționale și avansate, precum regresia logistică, mașinile de suport vectorial (SVM), Random Forest, Gradient Boosting, k-Nearest Neighbors (k-NN), Naive Bayes, AdaBoost, LightGBM, XGBoost și perceptroni multiclayer (MLP). Rezultatele arată că XGBoost este cel mai eficient model, cu o acuratețe medie de 0.9990 și o variabilitate minimă, demonstrând capacitatea sa de a gestiona relații complexe non-lineare din date. Random Forest a avut, de asemenea, performanțe excelente, cu o acuratețe medie de 0.9982, fiind robust împotriva overfitting-ului și oferind o stabilitate ridicată în diverse scenarii operaționale. Gradient Boosting și SVM au arătat performanțe remarcabile, cu acurateți de 0.9969 și, respectiv, 0.9961, beneficiind de abordările lor secvențiale și capacitatea de a gestiona spațiile de înaltă dimensionalitate.

Modelele k-NN și MLP au demonstrat utilitatea învățării bazate pe instanțe și a rețelelor neuronale în detectarea tiparelor complexe, fiecare cu o acuratețe de 0.9962. Naive Bayes, deși a avut cea mai mică acuratețe de 0.9922, este apreciat pentru eficiența și viteza sa, fiind util pentru evaluările inițiale și aplicațiile cu resurse computaționale limitate. Studiul subliniază importanța preprocesării datelor și ingineriei caracteristicilor în optimizarea performanței modelelor, evidențiind succesul etapelor implementate anterior antrenării modelului. Standardizarea caracteristicilor numerice a fost crucială pentru algoritmi precum SVM și rețelele neuronale, asigurând o influență uniformă a fiecărui atribut asupra rezultatelor modelului. Performanța ridicată a tuturor modelelor sugerează că metodologiile de preprocesare și selecție a caracteristicilor au fost extrem de eficiente, pregătind bine setul de date pentru analiza ulterioară.

3. Metodologie

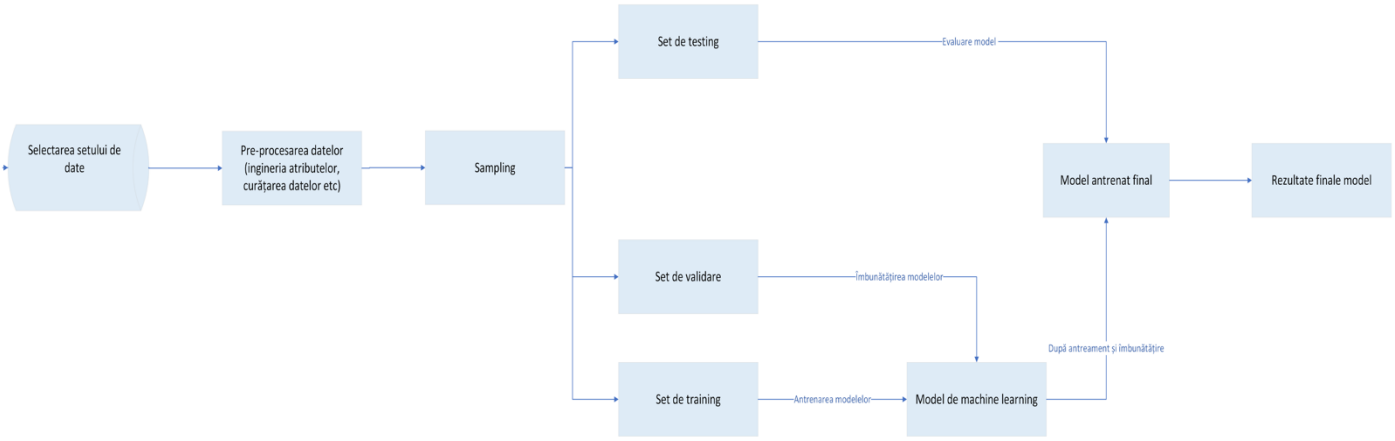


Figura 4 - Diagrama de Flow

3.1 Setul de date

Un set de date de calitate superioară este esențial pentru succesul oricărui proiect de machine learning deoarece permite modelelor să învețe și să generalizeze eficient comportamentele. Scopul meu este să adopt o abordare sistematică și riguroasă pentru a asigura că datele sunt curate, reprezentative și relevante. Astfel, sper sa obțin rezultate fiabile și precise, esențiale pentru detecția corectă a tranzacțiilor frauduloase. Voi prezenta metodele și tehnicile utilizate pentru colectarea, prelucrarea și analiza datelor esențiale în detectarea fraudei.

3.1.1 Colectarea datelor

Setul de date colectat provine de pe platforma Kaggle și contine aproxiamativ 1,3 milioane de instanțe. Autorul nu a furnizat detalii referitoare la instituția bancara de unde provin aceste date. Cu privire la structura datelor, acestea vor fi prezentate mai jos sub formă de tabel pentru o mai bună vizualizare.

Tabelul de mai jos prezintă 22 de atribute ale unei tranzacții financiare. Fiecare atribut înregistrează informații specifice, cum ar fi data și ora tranzacției, numărul cardului de credit, comerciantul implicat, suma tranzacționată, detalii despre titularul cardului (prenume, nume, gen, loc de muncă, data nașterii), locația tranzacției (strada, orașul, statul/județul, codul poștal, coordonatele geografice), și un indicator care specifică dacă o tranzacție este fraudă sau nu, fiind variabila țintă.

Nr. crt. atribut	Nume atribut
1	trans_date_trans_time (data și ora tranzacției)
2	cc_num (numărul cardului de credit)
3	merchant (comerciantul)
4	category (categoria tranzacției)
5	amt (suma tranzacției)
6	first (prenume)
7	last (nume)
8	gender (genul)
9	street (strada)
10	city (orașul)
11	state (statul/județul)
12	zip (codul postal)
13	lat (latitudinea adresei persoanei)
14	long (longitudinea adresei persoanei)
15	city_pop (populația orașului)
16	job (locul de muncă)
17	dob (data nașterii)
18	trans_num (ID-ul tranzacției)
19	unix_time (timpul calculat de la 1970 până la data tranzacției în secunde)
20	merch_lat (latitudinea comerciantului)
21	merch_long (longitudinea comerciantului)
22	is_fraud (dacă tranzacția este fraudă sau nu: 1/0)

Tabel 1 - Atribute inițiale

3.1.2 Curățarea datelor

Curățarea datelor este un pas crucial în pregătirea și prelucrarea datelor pentru detectarea fraudei. Acest proces implică identificarea și corectarea erorilor din setul de date, eliminarea valorilor lipsă sau incorecte și asigurarea coerenței și acurateței datelor. Curățarea datelor îmbunătățește calitatea acestora, ceea ce conduce la rezultate mai fiabile și relevante în analiza și modelarea datelor. În proiectul meu, curățarea datelor este crucială pentru a elimina zgomotul și a preveni anomaliile care ar putea afecta performanța modelului de detectare a fraudei.

3.1.3 Ingineria atributelor

Ingineria atributelor reprezintă un pas esențial în procesul de pregătire a datelor pentru modelele de învățare automată, deoarece calitatea și relevanța caracteristicilor extrase din date influențează direct performanța modelului. În cadrul acestui proiect, am realizat mai multe transformări și prelucrări ale setului de date pentru a extrage caracteristici semnificative și a îmbunătăți distribuția variabilelor, facilitând astfel detectarea eficientă a fraudei.

Tabelul rezultat cuprinde nouă variabile care au fost considerate relevante pentru detectarea fraudei în tranzacțiile cu carduri de credit. Aceste variabile includ caracteristici

precum categoria tranzacției, valoarea tranzacției, ziua săptămânii, genul utilizatorului, distanța până la comerciant, dacă tranzacția este frauduloasă sau nu, grupul de vârstă, grupul de ore și categoria populației orașului.

Nr. crt. atribut	Nume atribut
1	category (categoria tranzacției)
2	amt (suma tranzacției)
3	weekday (ziua săptămânii)
4	gender (genul)
5	distance_to_merchant (distanța până la comerciant)
6	age_group (grupul de vârstă 1-4)
7	hour_group (grupul de oră 1-4)
8	city_pop_category (grupul de populație 1-4)
9	is_fraud(0/1)

Tabel 2 - Atribute modificate

Suma tranzacției (amt) a fost logaritmată pentru o mai bună distribuție a datelor. De asemenea, pe baza latitudinii și a longitudinii a adresei persoanei și a latitudinii și longitudinii a comerciantului va fi calculată a utilizând formula Haversine (Azdy, 2020), care estimează distanța pe baza coordonatelor geografice.

Pentru a identifica eventualele modele de fraudă bazate pe ora la care au fost efectuate tranzacțiile, am împărțit ziua în patru intervale de câte șase ore fiecare. Fiecare tranzacție a fost astfel alocată unui grup orar specific:

- **1** pentru tranzacțiile efectuate între 00:00 și 05:59
- **2** pentru tranzacțiile efectuate între 06:00 și 11:59
- **3** pentru tranzacțiile efectuate între 12:00 și 17:59
- **4** pentru tranzacțiile efectuate între 18:00 și 23:59

Am clasificat utilizatorii în cinci grupe de vârstă pentru a examina cum variază riscul de fraudă în funcție de vârsta utilizatorilor:

- **1** pentru utilizatori sub 18 ani
- **2** pentru utilizatori între 18 și 34 de ani
- **3** pentru utilizatori între 35 și 49 de ani
- **4** pentru utilizatori între 50 și 64 de ani
- **5** pentru utlizatori între 65 de ani

Pentru a evalua impactul mărimii orașului asupra riscului de fraudă, am grupat orașele în patru categorii de populație:

- **1** pentru orașe cu populație sub 10,000 de locuitori
- **2** pentru orașe cu populație între 10,000 și 49,999 de locuitori
- **3** pentru orașe cu populație între 50,000 și 99,999 de locuitori
- **4** pentru orașe cu populație de 100,000 sau mai mulți locuitori

3.1.4 Analiza datelor

Voi examina în detaliu datele procesate, explorând diverse tehnici statistice și vizualizări pentru a înțelege mai bine caracteristicile acestora și relațiile dintre variabile. Analiza datelor este esențială pentru a identifica tiparele și anomaliile care pot indica tranzacții frauduloase și pentru a informa selectarea și configurarea modelelor de învățare automată. Scopul analizei datelor este de a obține o înțelegere profundă a setului de date și de a identifica caracteristicile cheie care contribuie la detectarea fraudei.

În analiza datelor, am observat că am la dispoziție 1.296.675 de tranzacții pe parcursul a 12.900,22 ore (sau 46.440.792 secunde), ceea ce corespunde unei perioade de 534 de zile. Această cantitate mare de date îmi permite să extrag informații valoroase despre tiparele tranzacțiilor și comportamentele asociate cu fraudele. Dintre acestea, doar 7.506 tranzacții, reprezentând aproximativ 0,579%, au fost identificate ca fiind frauduloase. Acest lucru arată un dezechilibru semnificativ între tranzacțiile frauduloase și cele ne-frauduloase în cadrul setului de date.

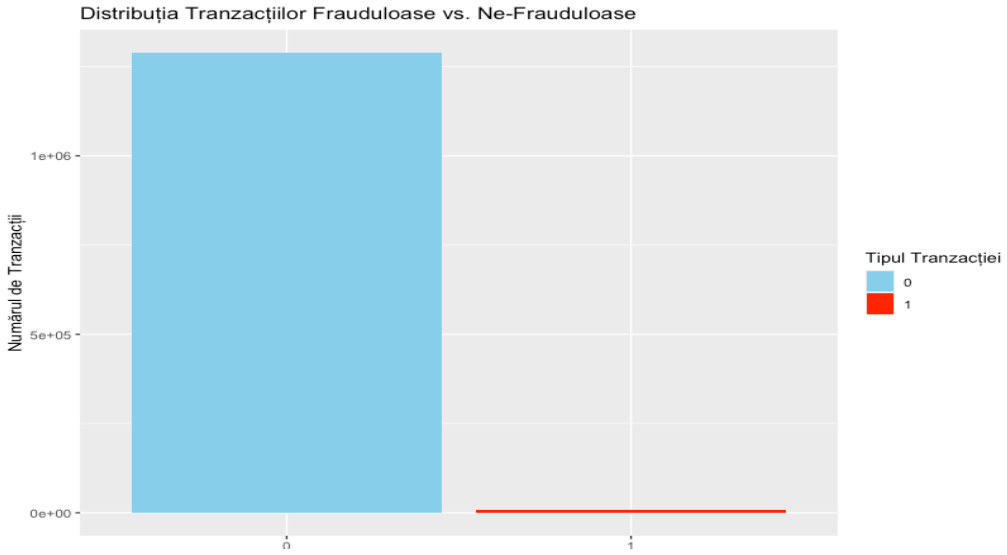


Figura 5 – Distribuție tranzacții

În imaginea de mai sus, se poate observa distribuția acestor tranzacții, unde tranzacțiile ne-frauduloase sunt reprezentate de o bară albastră foarte mare, iar tranzacțiile frauduloase sunt reprezentate de o bară roșie mult mai mică. Această neuniformitate subliniază necesitatea unor metode robuste de detectare a fraudelor, capabile să distingă eficient tranzacțiile legitime de cele suspecte.

Am realizat si o analiză a diferitelor tranzacții pe categorii de tranzacție.

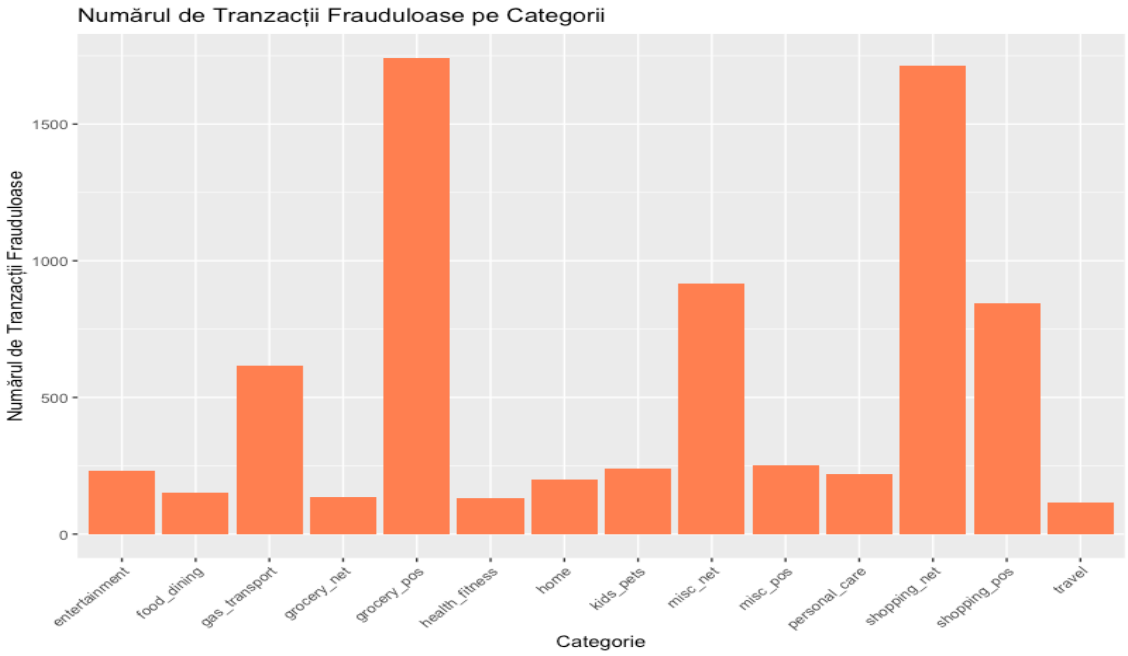


Figura 5 - Distribuție tranzacții pe categorii

Graficul prezentat indică distribuția tranzacțiilor frauduloase pe diferite categorii. Observăm o concentrare semnificativă a acestor tranzacții în categoriile grocery_pos, shopping_net, shopping_pos, și misc_net (diverse tranzacții pe internet). Aceste categorii par să fie mai susceptibile la fraude, probabil din cauza naturii lor online și a volumului mare de tranzacții care le face mai greu de monitorizat individual. De asemenea, categoriile health_fitness și travel prezintă un număr notabil de tranzacții frauduloase, sugerând că aceste domenii ar putea necesita măsuri suplimentare de securitate.

În continuare, voi analiza corelația dintre sumele tranzacțiilor si fraudă. În analizarea corelației dintre sumele tranzacțiilor și fraudă, este esențial să observăm cum se distribuie aceste valori în funcție de tranzacțiile frauduloase și cele nefrauduloase. Din graficul prezentat, se poate observa că tranzacțiile frauduloase tind să aibă sume mai mici comparativ cu tranzacțiile nefrauduloase. Aceasta ar putea indica faptul că tranzacțiile frauduloase sunt realizate cu sume mai mici pentru a trece neobservate. În graficul atașat mai jos se poate observa faptul ca majoritatea tranzacțiilor frauduloase tind să fie sub sumele de 1000 dolari.

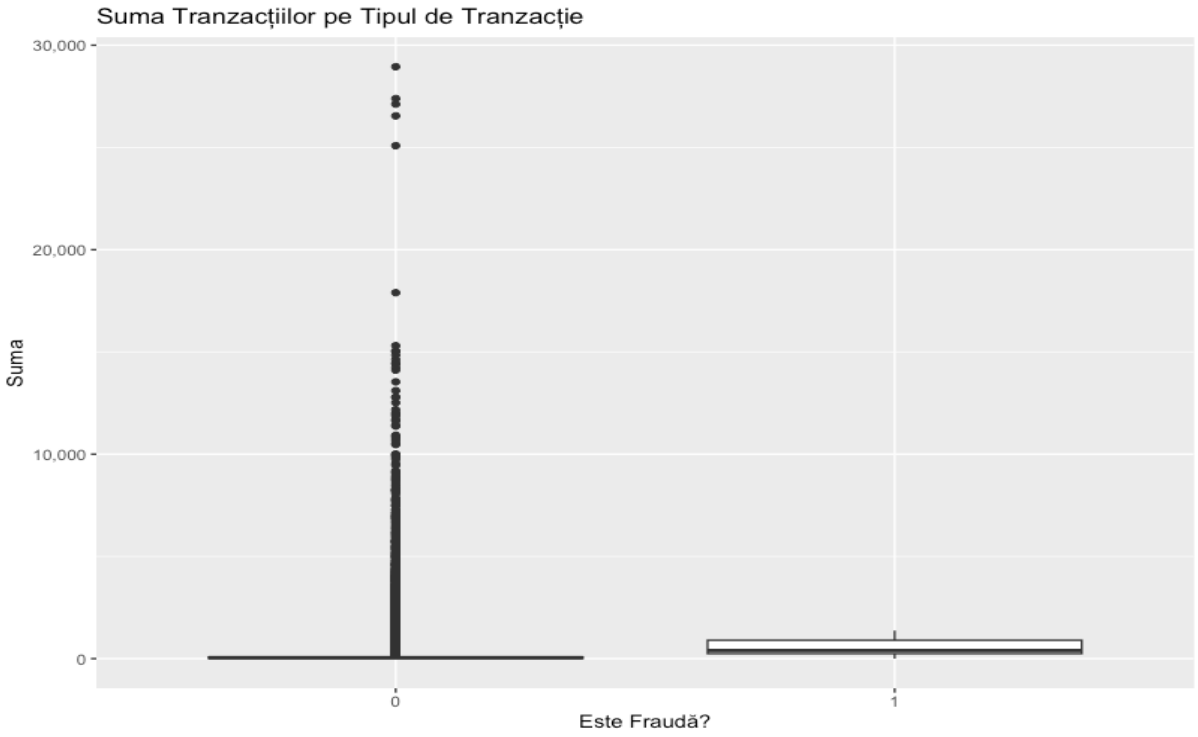


Figura 6 – Distribuție sumă / fraudă

Am analizat tranzacțiile frauduloase în funcție de genul persoanelor implicate. Rezultatele arată că genul nu influențează în mod semnificativ probabilitatea unei tranzacții frauduloase. În total, avem 3735 tranzacții frauduloase realizate de persoane de gen feminin și 3771 tranzacții frauduloase realizate de persoane de gen masculin, dintr-un total de 7506 tranzacții frauduloase. Dupa cum arată si graficul de mai jos, fraudele sunt distribuite aproape uniform între genuri, indicând faptul că alte variabile sunt mai relevante în identificarea tranzacțiilor frauduloase.

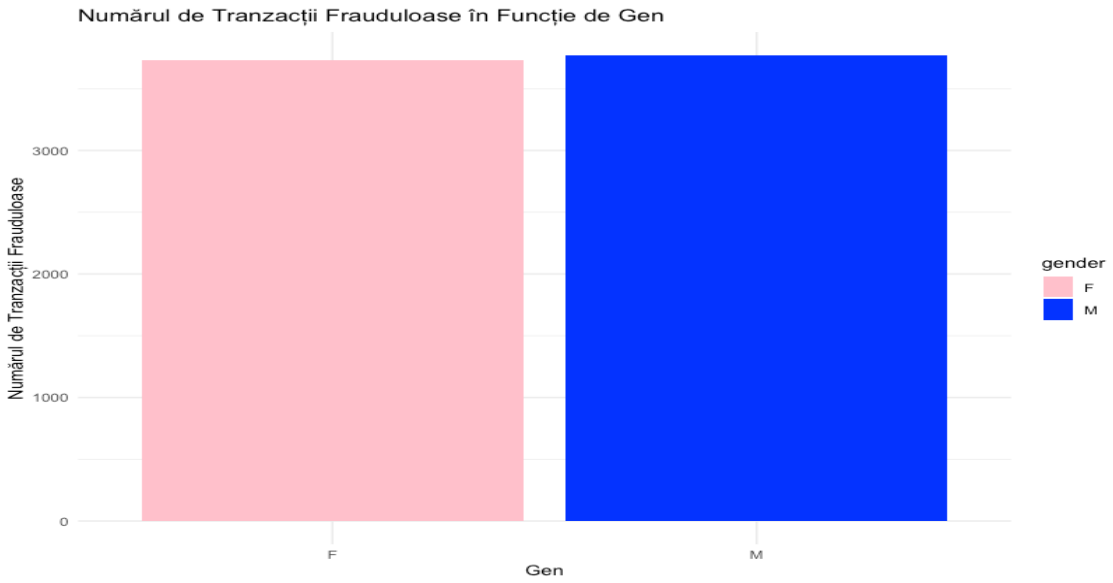


Figura 7 - Distribuția tranzacții fraudă/gen

Analiza datelor oferă o înțelegere profundă a caracteristicilor tranzacțiilor și a modului în care acestea influențează detectarea fraudelor. Am observat că setul de date este foarte dezechilibrat, cu un procent mic de tranzacții frauduloase. Corelațiile dintre variabile au evidențiat relații importante care pot ghida dezvoltarea modelelor predictive.

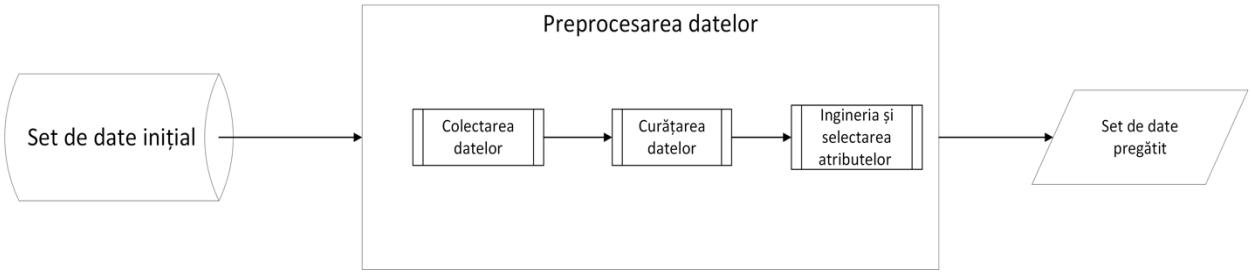


Figura 8 - Diagrama fluxului de date

Diagrama oferă o vizualizare clară și concisă a fluxului de preprocesare, subliniind importanța fiecărei etape în pregătirea datelor pentru analize ulterioare.

3.2 Metrici de evaluare

Evaluarea performanței modelelor de învățare automată este esențială pentru a înțelege cât de bine acestea funcționează în contextul specific al problemei analizate. În cazul detectării fraudei cu carduri de credit, este deosebit de important să utilizăm metrici adecvate, deoarece nu toate erorile au aceeași consecință. De menționat este faptul că toate definițiile acestor metrici sunt conform cărții “An Introduction to Statistical Learning: With Application in R”.

Utilizarea unui set variat de metrici ajută la obținerea unei imagini complete a performanței modelului. Fiecare metrică oferă o perspectivă diferită și este relevantă în evaluarea unor aspecte specifice ale modelului. Fiecare metrică oferă o perspectivă diferită și este relevantă în evaluarea unor aspecte specifice ale modelului. De exemplu, acuratețea globală a modelului poate părea ridicată într-un set de date dezechilibrat, însă aceasta nu reflectă capacitatea modelului de a identifica corect tranzacțiile frauduloase, care sunt de obicei rare. Pe lângă metricile clasice ale unui model, voi folosi metrici relevante pentru evaluarea seturilor de date dezechilibrate. De notat este faptul că positive class este ‘1’, adica tranzacții frauduloase. Înainte de a explora fiecare metrică, este important să înțelegem câțiva termeni fundamentali:

- **True Positives (TP):** Tranzacțiile frauduloase corect identificate ca fiind frauduloase.

- **True Negatives (TN):** Tranzacțiile ne-frauduloase corect identificate ca fiind ne-frauduloase
- **False Positives (FP)**: Tranzacțiile ne-frauduloase identificate incorect ca fiind frauduloase.
- **False Negatives (FN)**: Tranzacțiile frauduloase identificate incorect ca fiind ne-frauduloase.

3.2.1 Acuratețea (Accuracy)

Acuratețea este metrica de bază și reprezintă procentul de predicții corecte realizate de model în raport cu totalul predicțiilor. Deși este utilă pentru a oferi o primă impresie asupra performanței generale, nu este suficientă în sine, mai ales în seturi de date dezechilibrate cum este în cazul proiectului meu. Acuratețea poate fi exprimată astfel:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Deși acuratețea este o metrică importantă, ea poate fi înșelătoare în seturi de date dezechilibrate, unde o clasă este mult mai frecventă decât cealaltă. Totuși, este important Accuracy să fie mai mare ca NIR (No information rate). NIR (No Information Rate) reprezintă rata de predicție corectă a clasei majoritare din setul de date. În cazul unui set de date dezechilibrat, cum este cel utilizat în acest proiect, este esențial ca acuratețea modelului să fie mai mare decât NIR pentru a demonstra că modelul are performanțe mai bune decât o simplă presupunere a clasei majoritare. Valoarea p (p-value) se utilizează pentru a testa semnificația statistică a diferenței dintre acuratețea modelului și NIR. Aceasta indică probabilitatea ca diferența observată să fie întâmplătoare. O valoare p mică (de obicei < 0.05) sugerează că modelul meu are o performanță semnificativ mai bună decât presupunerea aleatorie.

Astfel, deși acuratețea este un indicator important, evaluarea performanței modelului trebuie să țină cont și de NIR și de valoarea p pentru a oferi o imagine completă a eficacității sale.

3.2.2 Sensibilitatea (Sensitivity)

Sensitivitatea, cunoscută și sub denumirea de rata de adevărat pozitive, este o metrică fundamentală în evaluarea performanței unui model de clasificare, în special în contextul detectării fraudei. Aceasta măsoară proporția tranzacțiilor frauduloase care sunt corect identificate ca fiind frauduloase de către model. În esență, sensibilitatea ne spune cât de bine este capabil modelul să identifice cazurile de fraudă din totalul tranzacțiilor frauduloase reale.

Este crucială pentru a minimiza numărul de tranzacții frauduloase care trec neobservate, asigurând astfel o detectare eficientă a fraudei. Sensitivitatea poate fi exprimată astfel:

$$Sensitivity = \frac{TP}{TP + FN}$$

O sensibilitate ridicată indică faptul că modelul este foarte eficient în detectarea tranzacțiilor frauduloase, în timp ce o sensibilitate scăzută sugerează că multe tranzacții frauduloase sunt clasificate greșit ca fiind nefrauduloase, ceea ce poate reprezenta un risc major pentru securitatea financiară. În contextul detectării fraudei, sensibilitatea este deosebit de importantă, deoarece scopul principal este de a minimiza pierderile și riscurile asociate cu tranzacțiile frauduloase. Pe parcursul acestei lucrări, voi analiza îndeaproape această metrică, deoarece este esențială pentru obiectivele mele de detectare a fraudei.

3.2.3 AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

AUC-ROC este o măsură care evaluează performanța unui model de clasificare, reprezentând relația dintre sensibilitate și specificitate. Sensibilitatea a fost prezentată mai sus. Specificitatea reprezintă proporția tranzacțiilor nefrauduloase corect identificate:

$$Specificity = \frac{TN}{TN + FP}$$

AUC este valoarea de sub curba ROC și variază de la 0 la 1, unde o valoare mai mare indică o performanță mai bună a modelului. Un model perfect are un AUC de 1, iar un model aleatoriu are un AUC de 0.5. AUC ROC este deosebit de adecvată pentru seturi de date dezechilibrate, cum este cazul setului meu, deoarece oferă o evaluare cuprinzătoare a

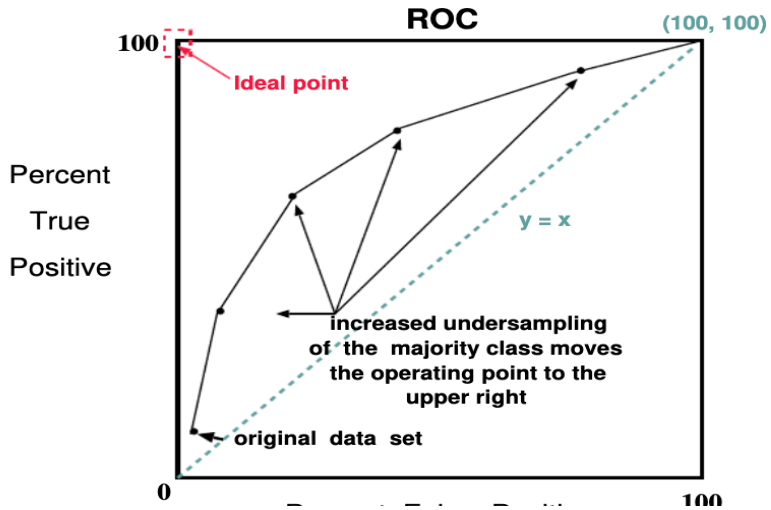


Figura 9 - Diagramă ROC curve

performanței modelului. Voi prezenta această metrica atât ca scor numeric, cât și sub formă de grafic pentru a ilustra capacitatea modelului de a discrimina între tranzacțiile frauduloase și cele nefrauduloase.

Curba ROC (Receiver Operating Characteristic) este un grafic ce ilustrează performanța unui model de clasificare la diferite praguri de decizie. În diagrama ROC, pe axa Y se află Sensibilitatea (True Positive Rate), iar pe axa X se află 1 – Specificitate (False Positive Rate).

Astfel, diagrama ROC și AUC sunt instrumente esențiale pentru evaluarea performanței modelelor în cazul seturilor de date dezechilibrate, cum este proiectul nostru de detectare a tranzacțiilor frauduloase. Pe parcursul lucrării, voi prezenta atât valorile AUC cât și graficele ROC pentru a ilustra eficacitatea modelelor dezvoltate.

3.2.4 Precision-Recall AUC (PR AUC)

Curba Precision-Recall (PR AUC) este esențială pentru evaluarea performanței modelelor de clasificare, mai ales în seturi de date dezechilibrate. Aceasta oferă o imagine detaliată a relației dintre precizie și recall pentru diferite valori de praguri, fiind deosebit de utilă pentru identificarea performanțelor modelelor în clasificarea exemplelor pozitive. Precision si Recall sunt reprezentate astfel:

- **Precizia (Precision)**: Este definită ca proporția de predicții pozitive corecte din totalul predicțiilor pozitive. Formula este: $Precizie = \frac{TP}{TP+FP}$
- **Recall (Sensibilitatea)**: Este definită ca proporția de predicții pozitive corecte din totalul cazurilor pozitive reale. Formula este: $Recall = \frac{TP}{TP+FN}$

Metrica PR AUC este deosebit de relevantă în contextul seturilor de date dezechilibrate, unde clasa pozitivă este mult mai rară decât clasa negativă. În astfel de cazuri, ROC AUC poate oferi o impresie exagerată a performanței modelului, deoarece include și proporția de negative corecte, care este predominantă. PR AUC se concentrează pe performanța modelului în identificarea corectă a exemplelor pozitive, ceea ce este esențial în aplicațiile practice unde identificarea corectă a exemplelor positive (fraudă) este mai critică decât a celor negative.

Această diagramă evidențiază importanța selectării unui model care maximizează atât precizia cât și recall-ul, având în vedere pragurile de decizie adecvate pentru a echilibra rata fals pozitive și fals negative. În cadrul proiectului meu de detectare a fraudelor, PR AUC oferă

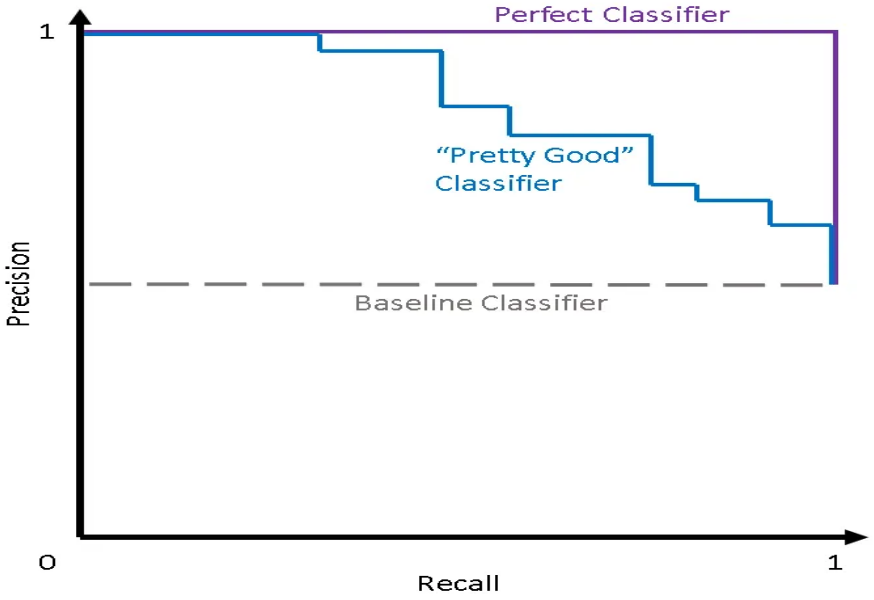


Figura 10 - Diagramă Precision Recall Curve

o măsură robustă pentru evaluarea și compararea diferitelor modele, având în vedere dezechilibrul puternic al setului de date.

3.2.5 F1-Score

F1 Score este o metrică utilizată pentru a evalua performanța unui model de clasificare, mai ales în contextul datelor dezechilibrate. Aceasta este definită ca media armonică a preciziei (precision) și a sensibilității (recall), oferind un echilibru între aceste două măsuri:

$$F1 = 2 * \frac{Precission * Recall}{Precission + Recall}$$

F1 Score este deosebit de util în contexte în care avem date dezechilibrate, cum ar fi detectarea fraudelor, deoarece oferă o imagine mai clară asupra performanței modelului în comparație cu simpla acuratețe. În cazul seturilor de date dezechilibrate, acuratețea poate fi înșelătoare, deoarece modelul poate avea performanțe bune pe clasa majoritară, ignorând complet clasa minoritară. F1 Score ajută la evidențierea eficienței modelului în detectarea corectă a tranzacțiilor frauduloase. În contextul proiectului meu, voi analiza și utiliza F1 Score alături de alte metrici pentru a evalua și optimiza modelele de clasificare.

3.2.6 Specificitatea (Specificity)

Specificitatea, cunoscută și ca rata de adevărate negative (True Negative Rate - TNR), este o metrică importantă în evaluarea performanței unui model de clasificare, în special în contextul detectării fraudelor. Specificitatea măsoară proporția tranzacțiilor nefrauduloase corect identificate de model ca nefrauduloase. Formula pentru specificitate este:

$$Specificitatea = \frac{TN}{TN + FP}$$

În contextul detectării fraudelor, specificitatea este importantă deoarece ne ajută să înțelegem cât de bine modelul poate să reducă numărul de alarme false (tranzacții nefrauduloase identificate ca fiind frauduloase). Un număr mare de alarme false poate duce la nemulțumirea clienților și la pierderea resurselor pentru investigarea acestor tranzacții.

O specificitate ridicată indică faptul că modelul este foarte eficient în detectarea tranzacțiilor frauduloase, în timp ce o scăzută sugerează că multe tranzacții frauduloase sunt clasificate greșit ca fiind nefrauduloase, ceea ce poate reprezenta un risc major pentru securitatea financiară. În contextul detectării fraudei, sensibilitatea este deosebit de importantă, deoarece scopul principal este de a minimiza pierderile și riscurile asociate cu tranzacțiile frauduloase. Pe parcursul acestei lucrări, voi analiza îndeaproape această metrică, deoarece este esențială pentru obiectivele mele de detectare a fraudei.

3.3 Modele predictive

În această secțiune, voi explora diferite modele predictive utilizate pentru detectarea fraudei în tranzacțiile cu carduri de credit. Scopul este de a compara performanța mai multor algoritmi de machine learning și deep learning pentru a determina care metodă oferă cele mai bune rezultate în ceea ce privește identificarea tranzacțiilor frauduloase. Voi analiza atât modele clasice, cum ar fi arborii de decizie, cât și modele avansate, cum ar fi rețelele neuronale.

Fiecare model va fi evaluat pe baza unor metrici specifice, inclusiv acuratețea, sensibilitatea, specificitatea, scorul F1, AUC-ROC și PR AUC. De asemenea, voi discuta despre avantajele și dezavantajele fiecărui model, precum și despre posibilele îmbunătățiri și ajustări ale hiperparametrilor care pot conduce la o performanță mai bună.

În cele ce urmează, voi prezenta pe rând fiecare model implementat, metodele de tuning utilizate pentru optimizare, și voi compara rezultatele obținute.

3.3.1 Arbori de decizie

Arborii de decizie sunt modele predictive puternice utilizate pe scară largă atât pentru sarcini de clasificare, cât și de regresie.

Avantajele arborilor de decizie:

- **Interpretabilitate ușoară**: Arborii de decizie sunt ușor de înțeles și interpretat. Structura lor grafică permite utilizatorilor să vadă clar deciziile luate și regulile aplicate în fiecare ramură a arborelui.
- **Capacitatea de a gestiona orice date**: Pot gestiona atât date numerice, cât și categorice, fără a necesita transformări complexe sau normalizări.

Știind cât de costisitor ar putea fi procesul din punct de vedere al resurselor, am decis să folosesc doar 50% din datele disponibile pentru antrenare și testare. Am început prin a împărți acest eșantion în două părți: 70% pentru antrenament și 30% pentru restul datelor. Odată ce am avut aceste date, am împărțit cele 30% rămase în două seturi egale: unul pentru validare și unul pentru testare. Astfel, am reușit să creez un set robust de date care să mă ajute să îmi construiesc și să îmi verific modelele.

Modelul cu Rpart

Primul model pe care l-am abordat a fost Rpart, un model de clasificare bazat pe arbori de decizie. Am creat o grilă de hiperparametri pentru cost-complexity pruning (cp) și am definit o funcție care să evalueze specificitatea modelului pe setul de validare.

Am rulat mai multe iterații pentru a găsi valoarea optimă a lui cp, iar după ce am găsit cel mai bun model, am combinat seturile de antrenament și validare pentru a antrena modelul final. Modelul a fost evaluat pe setul de testare.

Arborele de decizie arată cum suma tranzacției și tipul acesteia sunt critice în detectarea fraudelor. De exemplu, tranzacțiile mici și frecvente în anumite categorii sunt mai susceptibile de a fi frauduloase. În schimb, tranzacțiile mari efectuate în anumite intervale orare și categorii prezintă un risc mai mic de fraudă.

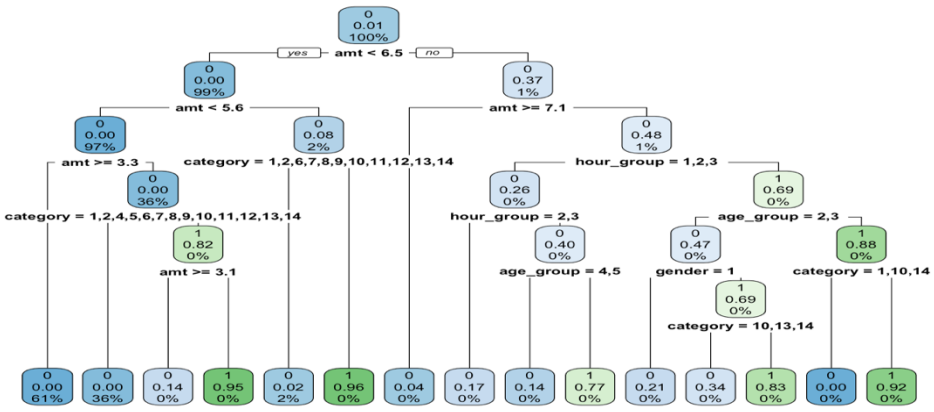


Figura 11 - Rpart Plot

Modelul Tree

Următorul pas a fost să încerc modelul Tree, similar cu Rpart, dar cu propriile sale subtilități. Am creat o nouă grilă de hiperparametri, de această dată pentru mincut și minsize, și am eliminat combinațiile invalide. Am folosit o funcție de evaluare pentru a găsi cei mai buni parametri și am antrenat modelul final pe setul combinat de antrenament și validare. Evaluând din nou pe setul de testare, am obținut rezultate promițătoare. Următorul pas a fost să încerc modelul Tree, similar cu Rpart, dar cu propriile sale subtilități. Am creat o nouă grilă de hiperparametri, de această dată pentru mincut și minsize, și am eliminat combinațiile invalide. Am folosit o funcție de evaluare pentru a găsi cei mai buni parametri și am antrenat modelul final pe setul combinat de antrenament și validare. Specificitatea cea mai bună a fost obținută cu mincut = 5 și minsize = 10, cu o specificitate de 0.6831.

Modelul bagging

Am decis să încerc și un model de bagging pentru a vedea dacă aș putea obține performanțe mai bune. Am antrenat inițial un model pe setul de antrenament și l-am evaluat pe cel de validare.

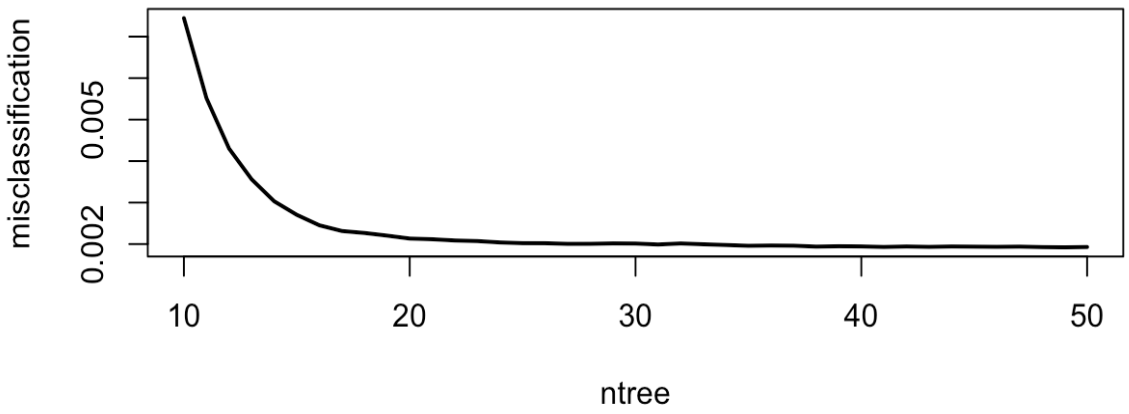


Figura 12 - Misclassification rate

Graficul de mai sus arată că rata de clasificare greșită scade rapid pe măsură ce numărul de arbori crește, stabilizându-se în jurul valorii de 20-30 de arbori. După acest punct, adăugarea de arbori suplimentari nu mai îmbunătățește semnificativ performanța modelului. Astfel, un număr optim de arbori pentru modelul de bagging este în jur de 30-34, unde rata de clasificare greșită este minimă și stabilă. Acest punct de echilibru asigură performanța optimă a modelului fără a consuma resurse suplimentare în mod inutil. Am ales sa fac modelul de bagging cu 34 de arbori pentru o performanță optimă. Combinând din nou seturile de antrenament și validare, am antrenat modelul final de bagging și l-am evaluat pe setul de validare. Rezultatele au fost impresionante, iar modelul gestioneaze foarte bine detectarea fraudelor.

Metrică	Modelul cu Rpart	Modelul cu Tree	Modelul cu Bagging
Accuratețe	0.9977	0.9971	**0.9982**
Sensibilitate	0.9995	0.9989	**0.9996**
Specificitate	0.7038	0.6831	**0.7757**
F1	0.9988	0.9985	**0.9991**
ROC-AUC	0.9452	**0.9698**	0.9679
PR-AUC	0.7702	0.7181	**0.8770**

Tabel 3 - Rezultate arbori de decizie

Rezultatele obținute pentru modelele de arbori de decizie arată diferențe semnificative în ceea ce privește specificitatea și alte metrici relevante. Modelul Rpart prezintă o specificitate de 0.7038, ceea ce indică o capacitate moderată de a identifica corect cazurile pozitive (frauduloase), în timp ce modelul Tree are o specificitate ușor mai mică, de 0.6798. Modelul de Bagging depășește celelalte două modele cu o specificitate de 0.7757, un rezultat bun. Modelul de Bagging excelează din nou cu cele mai bune valori PR-AUC (0.8770) și ROC-AUC (0.9679), indicând o performanță generală superioară în detectarea fraudelor. Aceste rezultate sugerează că modelul de Bagging nu doar că îmbunătățește specificitatea comparativ cu modelele individuale de arbori de decizie, dar oferă și o performanță generală mai robustă, fiind capabil să echilibreze bine între identificarea corectă a fraudelor și reducerea alarmelor.

3.3.2 Random forest

Random Forest combină rezultatele mai multor arbori de decizie pentru a obține o predicție finală mai robustă și mai precisă. Fiecare arbore din pădure este construit folosind un subset aleatoriu de date și un subset aleatoriu de caracteristici, ceea ce introduce diversitate și reduce varianța modelului, contribuind astfel la prevenirea suprapotrivirii (overfitting).

Pentru a gestiona eficient resursele disponibile și a reduce costurile computaționale, am eșantionat un subset de 50% din datele originale. Acest subset a fost împărțit ulterior în seturi de antrenament, validare și testare, asigurându-mă că fiecare subset este reprezentativ pentru distribuția claselor. Am antrenat un model de bază Random Forest pe setul de antrenament și l-am optimizat folosind setul de validare pentru a găsi cei mai buni hiperparametri efectuând o căutare în domeniul mtry in c(2, 3, 4, 5), nodesize in c(1, 5, 10). După o analiză detaliată, am identificat **mtry = 4** și **nodesize = 1** ca fiind valorile optime, obținând un AUC de 0.9873 pe setul de validare.

Pentru a îmbunătăți și mai mult performanța modelului și a gestiona dezechilibrul dintre clase, am aplicat ponderi de clasă (class weights) în modelul final, acordând o pondere de 1 pentru clasa 0 (non-fraudă) și o pondere de 10 pentru clasa 1 (fraudă).

Metrică	Random forest de bază	Random forest cu hiperparametri si class weights
Acuratețe	0.9982	**0.9983**
Sensibilitate	**0.9999**	0.9996
Specificitate	0.7182	**0.7760**
F1	0.9991	0.9991
ROC-AUC	0.9834	**0.9915**

PR-AUC	0.8835	**0.8884**

Tabel 4 - Rezultate random forest

Rezultatele obținute pentru modelul Random Forest arată îmbunătățiri semnificative după optimizarea hiperparametrilor și aplicarea ponderilor de clasă. Specificitatea a crescut de la 0.7182 la 0.7760, indicând o capacitate mai bună de a identifica corect cazurile negative și reducând astfel rata fals-pozitivelor. De asemenea, ROC-AUC a crescut de la 0.9834 la 0.9915, demonstrând o capacitate generală îmbunătățită de discriminare între clase. PR-AUC a avut, de asemenea, o creștere ușoară, de la 0.8835 la 0.8884, sugerând un echilibru mai bun între precizie și recall în contextul claselor dezechilibrate. Aceste îmbunătățiri indică faptul că modelul optimizat este mai robust și mai fiabil în detectarea fraudelor.

3.3.3 Metode de boosting

Boosting este o tehnică puternică de învățare automată care combină performanța mai multor clasificatori slabi pentru a construi un clasificator puternic. Clasificatorii slabi sunt modele care au performanțe ușor mai bune decât o clasificare aleatorie. Prin antrenarea secvențială a acestor clasificatori și ajustarea ponderilor în funcție de erorile comise la fiecare pas, boosting-ul reușește să îmbunătățească semnificativ acuratețea modelului final. Printre cele mai populare metode de boosting se numără **XGBoost** și **Gradient Boosting Machine (GBM)**, pe care le voi folosi in proiectul meu.

XGBoost

Am eșantionat 10% din setul de date original pentru a gestiona eficient resursele computaționale și pentru a reduce timpul de antrenare, păstrând totuși o reprezentativitate adecvată a datelor. Datele eșantionate au fost împărțite în seturi de antrenament, validare și testare pentru a permite evaluarea performanței modelului într-un mod robust și pentru a preveni suprapotrivirea. Eliminarea valorilor lipsă este importantă pentru a asigura integritatea seturilor de date și pentru a preveni problemele în timpul antrenării modelului. Am antrenat un model inițial XGBoost pe setul de antrenament și am evaluat performanța acestuia pe setul de validare, utilizând metrica AUC, care este foarte potrivită pentru seturile de date dezechilibrate.

Pentru a îmbunătăți performanța modelului, am folosit un grid search în domeniul eta = c(.01, .05, .1), max_depth = c(3, 5, 7), min_child_weight = c(1, 3, 5), subsample = c(.7, .8, .9), colsample_bytree = c(.7, .8, .9) pentru tuning-ul hiperparametrilor. De asemenea, am crescut numărul de **arbori** la **5000**. Evaluarea fiecărei configurații pe setul de validare ne-a permis să selectăm cei mai buni hiperparametri pe baza valorii maxime a AUC. Cei mai buni

hiperparametri au fost: eta 0.01, max depth 7, min_child_weight 5, subsample 0.8, colsample_bytree 0.8, scale_pos_weight 168.7353.

Antrenarea modelului final XGBoost cu acești parametri optimi a asigurat că modelul este bine ajustat și performant.

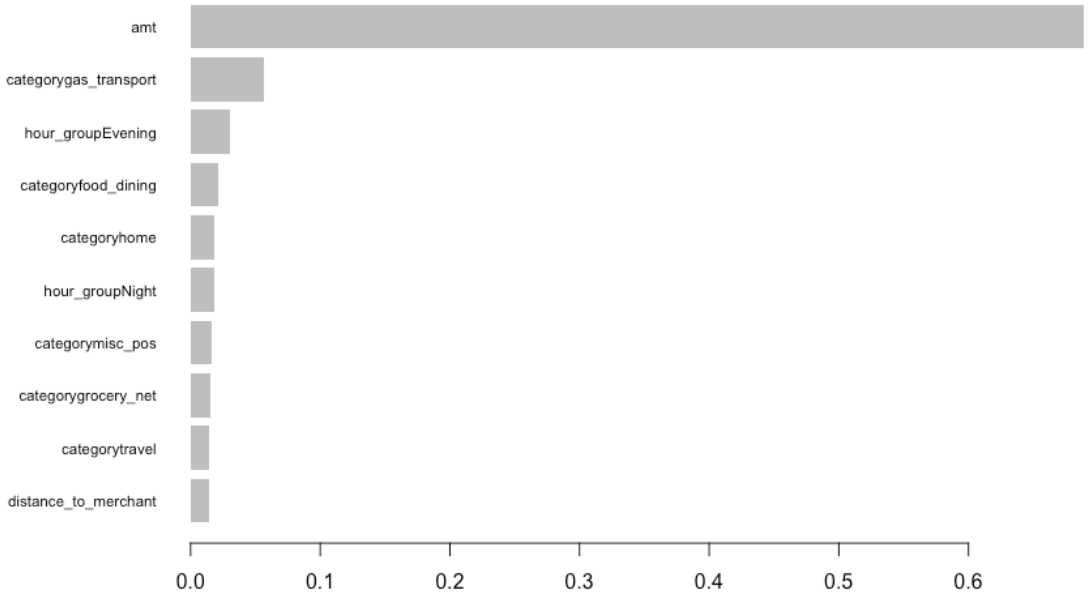


Figura 13 - Matricea de importanță XGB

Graficul arată că variabila amt (suma tranzacției) este de departe cea mai importantă în modelul XGBoost pentru detectarea fraudelor. Următoarele variabile importante includ category_gas_transport și intervalele orare (hour_groupEvening și hour_groupNight), sugerând că tranzacțiile din aceste categorii și perioade de timp sunt indicatori semnificativi ai comportamentului fraudulos. Acest lucru evidențiază importanța sumelor tranzacțiilor și contextul temporal în detectarea fraudelor.

În cele din urmă, am evaluat modelul final pe setul de validare și am obținut rezultate foarte bune.

GBM

Gradient Boosting Machine (GBM) este un algoritm puternic de învățare automată folosit pentru probleme de clasificare și regresie. GBM este un exemplu de metodă de boosting, care combină mai mulți estimatori slabi pentru a forma un model puternic. Principiul fundamental

al boosting-ului este de a construi modele secvențiale, fiecare model nou fiind antrenat pentru a corecta erorile comise de modelele anterioare.

Am construit un model de bază GBM (Gradient Boosting Machine) și l-am evaluat folosind diverse metrici, inclusiv AUC-ROC, AUC-PR și F1 Score. Am efectuat tuning-ul hiperparametrilor folosind un grid search în domeniul n.trees =c(50,100,200), interaction.depth = c(1, 3, 5), shrinkage = c(0.01, 0.1), n.minobsinnode = c(10, 20) . În cele din urmă, am antrenat modelul final GBM cu cei mai buni parametri și l-am evaluat pe setul de validare pentru a determina performanța sa.

Metrică	GBM	XGB basic	XGB cu search grid
Accuratețe	0.9978	**0.998**	0.9961
Sensibilitate	**0.9996**	0.9991	0.9967
Specificitate	0.6969	0.8182	**0.8836**
F1	0.9988	**0.9990**	0.9980
ROC-AUC	0.9800	0.9946	**0.9965**
PR-AUC	0.7984	0.8742	**0.8898**

Tabel 5 - Rezultate boosting

Rezultatele arată că toate modelele - GBM, XGBoost de bază și XGBoost optimizat cu grid search - au obținut performanțe ridicate, dar cu diferențe semnificative în anumite metrici. Modelul XGBoost optimizat a excelat în specificitate (0.8836), ROC-AUC (0.9965) și PR-AUC (0.8898), indicând o capacitate superioară de a separa corect tranzacțiile frauduloase de cele nefrauduloase și o mai bună precizie-recall.

3.3.4 Support Vector Machines

Support Vector Machines (SVM) sunt algoritmi de învățare automată folosiți pentru problemele de clasificare și regresie. SVM funcționează prin găsirea unui hiperplan optim care separă clasele într-un spațiu de caracteristici multidimensional. În clasificarea binară, obiectivul SVM este de a maximiza marginea, adică distanța dintre hiperplan și cele mai apropiate puncte din fiecare clasă, cunoscute ca vectori de suport. SVM poate utiliza diverse kernel-uri (ex. linear, radial, polynomial) pentru a transforma datele și a găsi un hiperplan într-un spațiu de dimensiuni mai mari, făcându-l astfel potrivit pentru seturi de date care nu sunt liniar separabile.

Din cauză ca setul meu de date este foarte mare, iar SVM este un algoritm care necesită foarte mule resurse, pentru acest caz am selectat doar 5% din setul de date inițial. Am antrenat un model SVM cu kernel radial pe setul de antrenare și l-am evaluat pe setul de validare, calculând metrici precum AUC-ROC, PR AUC și scorul F1 pentru a determina performanța în detectarea tranzacțiilor frauduloase. Acesta a avut performanțe destul de slabe.

Metrică	SVM
Accuratețe	0.9961
Sensibilitate	0.9995
Specificitate	0.4590
F1	0.9980
ROC-AUC	0.5127
PR-AUC	0.8826

Tabel 6 - Rezultate SVM

Rezultatele obținute pentru modelul SVM indică performanțe slabe în detectarea tranzacțiilor frauduloase. Specificitatea modelului este de doar 0.4590, ceea ce arată o capacitate foarte redusă de a identifica corect tranzacțiile non-frauduloase și, implicit, un număr ridicat de alarme false. De asemenea, scorul ROC-AUC de 0.5127 sugerează că modelul SVM nu reușește să diferențieze eficient între tranzacțiile frauduloase și cele legitime. Aceste valori subliniază limitările semnificative ale modelului SVM în contextul setului de date dezechilibrat, făcându-l inadecvat pentru detectarea precisă a fraudelor.

3.3.5 Rețele neuronale

Rețelele neuronale sunt modele predictive puternice inspirate din structura și funcționarea creierului uman. Acestea sunt compuse din straturi de neuroni artificiali interconectați, care procesează datele într-un mod similar cu modul în care neuronii biologici procesează informațiile. În contextul detectării fraudelor, rețelele neuronale sunt folosite pentru a învăța și a identifica modele complexe în datele tranzacțiilor, care pot indica activități frauduloase.

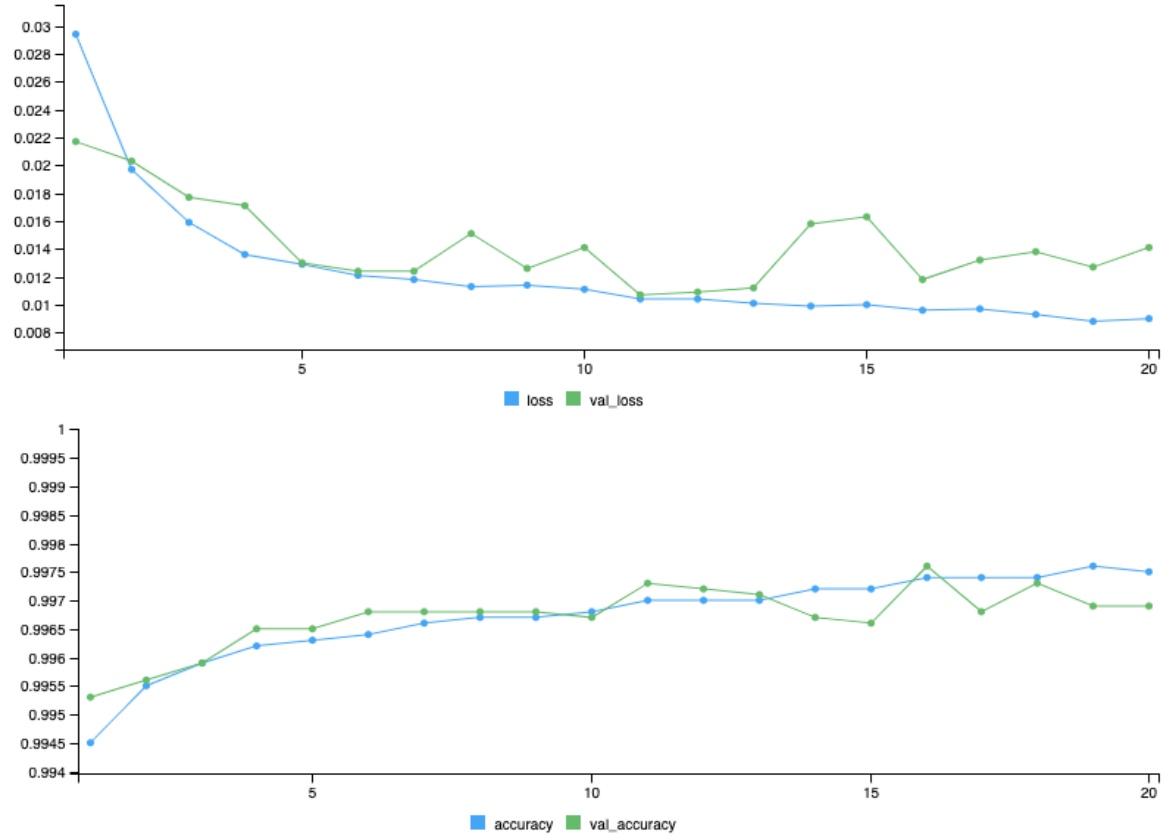
Rețelele neuronale prezintă câteva avantaje cruciale:

- **Capacitatea de a învăța relații complexe**: Pot modela relații intricate între variabilele de intrare și ieșire, fiind foarte eficiente în detectarea tiparelor subtile care indică fraude.
- **Adaptabilitate**: Pot fi ajustate și îmbunătățite prin adăugarea de noi date și recalibrarea ponderilor.
- **Generalizare**: Pot generaliza bine pe seturi de date mari și variate, ceea ce le face potrivite pentru aplicații practice în detectarea fraudelor.

Pentru început, variabila țintă, is_fraud a fost transformată în format numeric pentru a putea folosi funcția de pierdere binary_crossentropy în timpul antrenării. Am împărțit setul de date în trei subseturi: antrenare (70%), validare (15%) și testare (15%). Această împărțire mi-a permis să evaluez modelul în mod corespunzător și să previn suprapotrivirea(overfitting). În ceea ce privește arhitectura modelului, am definit o rețea neuronală cu două straturi ascunse. Primul strat a avut 64 de unități, iar al doilea strat a avut 32 de unități. Am adăugat și un strat dropout pentru a preveni suprapotrivirea. Modelul a fost compilat folosind optimizatorul Adam și funcția de pierdere binary_crossentropy și funcția de activare sigmoid la stratul de ieșire, cu accuracy ca metrică de performanță. Antrenarea modelului s-a realizat pe setul de date de antrenare, iar validarea s-a efectuat pe setul de validare. Am observat unde începe suprapotrivirea modelului și am antrenat modelul final pe numărul ideal de epochs pentru a preveni acest fenomen.

```
> summary(model)
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_11 (Dense)            (None, 64)                4352
 dense_10 (Dense)            (None, 32)                2080
 dropout_3 (Dropout)         (None, 32)                0
 dense_9 (Dense)             (None, 1)                 33
=================================================================
Total params: 6,465
Trainable params: 6,465
Non-trainable params: 0
_________________________________________________________________
```

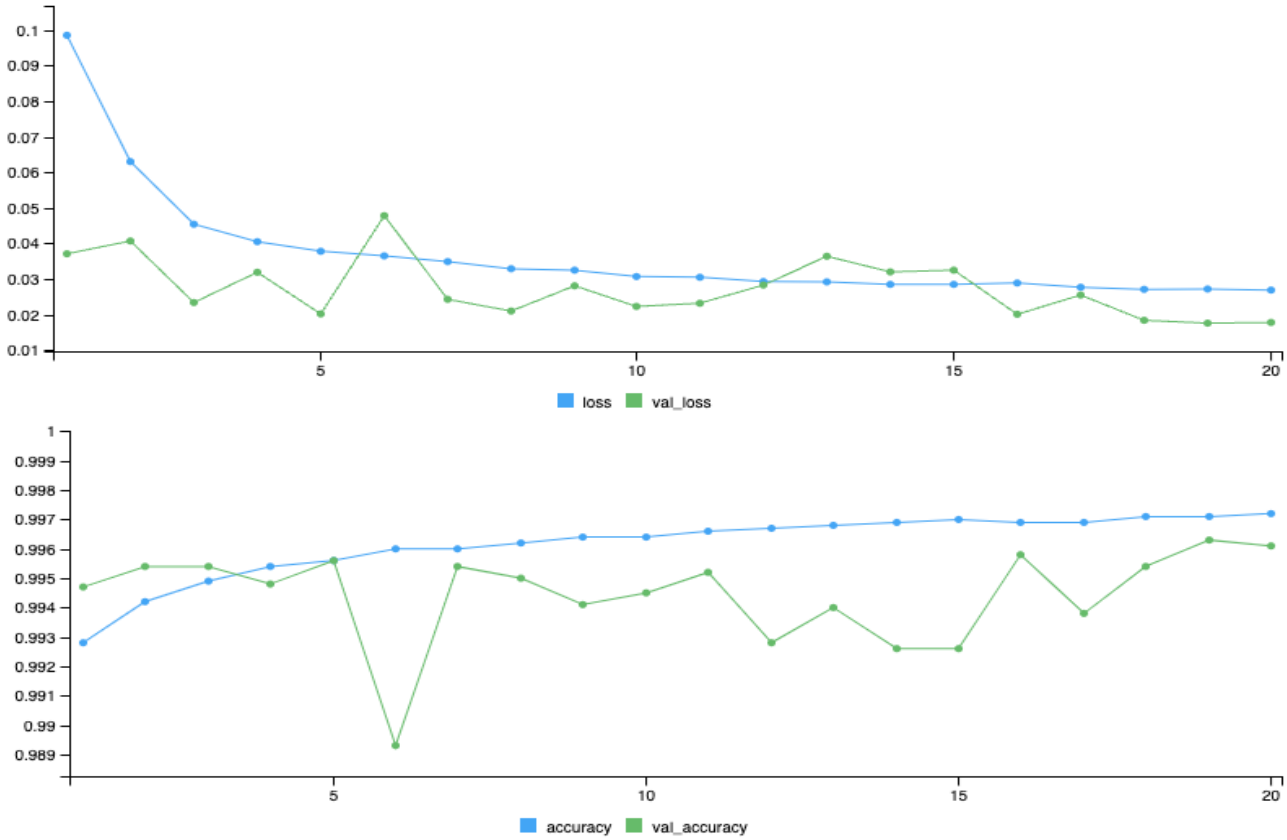
Ultimul strat (dense_9) este un strat dense cu funcția de activare sigmoid, folosit pentru clasificarea binară.



După 11 epochs se poate observa ca apare suprapotrivirea. Astfel, ma voi opri la 11 epochs în crearea modelului pentru a obține rezultate ideale.

Pentru a optimiza hiperparametrii modelului, am folosit grid search în domeniul: units1 = c(32, 64, 128), units2 = c(16, 32, 64), dropout_rate = c(0.2, 0.3, 0.4), epochs = c(5, 10, 20) . Grid search a permis explorarea sistematică a diferitelor combinații de hiperparametri, identificând setările optime care au dus la cea mai bună performanță pe setul de validare. De asemenea, am aplicat ponderi de clasă pentru a aborda dezechilibrul din setul de date, atribuind o greutate mai mare tranzacțiilor frauduloase (“0” = 1, “1” = 5) pentru a le face mai relevante

în procesul de antrenare. Am evaluat apoi cel mai bun model pe setul de testare, calculând aceleași metrici de performanță pentru a verifica îmbunătățirile obținute. Parametrii care ofera



ofera cea mai bună acuratețe sunt: units1 = 128 , units2 = 64 , dropout_rate = 0.3. Am construit un model cu acești parametri.

Dupa epoch 5 apare in mod evident suprapotrivirea. Voi antrena modelul optimizat final cu grid search pe doar 5 epochs.

Metrică	Model de bază (11 epochs)	Model optimizat cu grid search
Acuratețe	**0.998**	0.9967
Sensibilitate	0.9995	**0.9979**
Specificitate	0.7204	**0.8303**
F1	**0.8026**	0.7534
ROC-AUC	**0.9935**	0.9934
PR-AUC	0.8497	**0.8524**

Tabel 7 - Rezultate rețele neuronale

Deși acuratețea modelului optimizat este ușor mai mică decât cea a modelului de bază, diferența este nesemnificativă. Ambele modele performează aproape la fel de bine, sugerând că optimizarea și reducerea numărului de epochs nu au compromis în mod semnificativ acuratețea generală. Sensibilitatea a scăzut ușor în modelul optimizat, dar rămâne foarte

ridicată. Acest lucru indică faptul că modelul optimizat continuă să identifice majoritatea cazurilor pozitive, cu o ușoară reducere în comparație cu modelul de bază. Specificitatea a crescut semnificativ în modelul optimizat. Acest lucru este un rezultat pozitiv, deoarece înseamnă că modelul optimizat este mai bun la identificarea corectă a cazurilor negative, reducând astfel rata fals-pozitivelor. Scorul F1 a scăzut în modelul optimizat. Acest scor ia în considerare atât sensibilitatea, cât și precizia. Scăderea indică un echilibru mai puțin favorabil între sensibilitate și precizie în modelul optimizat. Valorile ROC-AUC sunt aproape identice pentru ambele modele, ceea ce sugerează că capacitatea de discriminare a celor două modele este similară. Aceasta arată că optimizarea nu a afectat negativ performanța modelului în a diferenția între clasele pozitive și negative. PR-AUC este ușor mai mare pentru modelul optimizat. Acest lucru indică o îmbunătățire ușoară în echilibrul dintre precizie și sensibilitate în contextul claselor dezechilibrate, sugerând că modelul optimizat gestionează mai bine astfel de situații.

Aceasta analiză arată că modelul optimizat este o alegere bună, oferind un echilibru mai bun în performanțele metrice, în special în ceea ce privește specificitatea.

3.4 Rezultate și discuții

Așa cum am menționat anterior, am împărțit setul de date în trei părți distincte: setul de antrenament, setul de validare și setul de testare (70% antrenare, 15% testare, 15% validare). Până acum, am folosit primele două pentru a antrena modelele și pentru a ajusta hiperparametrii lor.

Metrică	Bagging	Random Forest	GBM	XGBoost	SVM	NN
Accuratețe	0.9982	**0.9983**	0.9978	0.9961	0.9961	0.9967
Sensibilitate	**0.9996**	**0.9996**	**0.9996**	0.9967	0.9995	0.9979
Specificitate	0.7757	0.7760	0.6969	**0.8836**	0.4590	0.8303
F1	**0.9991**	**0.9991**	0.9988	0.9980	0.9980	0.7534
ROC-AUC	0.9679	0.9915	0.9800	**0.9965**	0.5127	0.9934
PR-AUC	0.8770	**0.8884**	0.7984	0.8898	0.8826	0.8524

Tabel 8 - Rezultate finale

Rezultatele obținute din diferitele modele de machine learning pentru detectarea fraudelor arată performanțe remarcabile, dar cu diferențe notabile între metode. Random Forest, XGBoost și Rețele Neuronale (NN) au demonstrat cele mai bune performanțe generale.

Random Forest a obținut o acuratețe de 0.9983 și o specificitate de 0.7760, indicând o capacitate robustă de a identifica corect tranzacțiile non-frauduloase. Sensibilitatea sa ridicată, de 0.9996, sugerează că detectează aproape toate tranzacțiile frauduloase. F1-score-ul de 0.9991 și ROC-AUC-ul de 0.9915 confirmă echilibrul acestui model în detectarea atât a tranzacțiilor frauduloase, cât și a celor non-frauduloase.

XGBoost s-a remarcat printr-o specificitate de 0.8871 și un ROC-AUC de 0.9970, indicând o capacitate excepțională de a diferenția între tranzacțiile frauduloase și cele legitime. Cu aceste performanțe, XGBoost este cel mai robust model pentru detectarea fraudelor, oferind un echilibru optim între identificarea corectă a tranzacțiilor frauduloase și reducerea alarmelor false. Specificitatea ridicată subliniază abilitatea modelului de a minimiza ratele de fals pozitive, în timp ce ROC-AUC-ul superior indică o excelentă capacitate de discriminare între clase.

Rețelele Neuronale (NN) au demonstrat un echilibru general foarte bun, având o specificitate de 0.8303 și o sensibilitate de 0.9979. ROC-AUC-ul de 0.9934 și PR-AUC-ul de 0.8524 subliniază performanța excelentă a acestui model în identificarea tranzacțiilor frauduloase. Rețelele neuronale sunt capabile să învețe și să identifice modele complexe în datele tranzacțiilor, făcându-le extrem de eficiente în detectarea comportamentelor frauduloase subtile. Deși specificitatea lor este ușor mai scăzută decât cea a modelului XGBoost, rețelele

neuronale oferă o performanță robustă și sunt deosebit de utile în situațiile în care relațiile dintre variabile sunt complexe și nelineare.

Aceste rezultate sugerează că, în contextul detectării fraudelor, modelul XGBoost este cel mai performant, oferind o specificitate și o capacitate de discriminare între clase superioare.

Metrică	**XGBoost**
Acuratețe	0.9961
Sensibilitate	0.9967
Specificitate	**0.8871**
F1	0.9980
ROC-AUC	**0.9970**
PR-AUC	0.8894

Tabel 9 - Rezultate XGBoost (testare)

Am folosit setul de testare pentru a obține aceste rezultate, asigurând astfel o evaluare robustă și realistă a performanței modelului. Modelul XGBoost excelează în ceea ce privește specificitatea, cu o valoare de 0.8871, și demonstrează o capacitate excepțională de a diferenția între tranzacțiile frauduloase și cele legitime, având un ROC-AUC de 0.9970.

În contrast, modelele precum Bagging și GBM au arătat performanțe mai slabe, în special în ceea ce privește specificitatea și PR-AUC, subliniind importanța alegerii unor modele mai sofisticate și bine calibrate pentru detectarea fraudelor.

Este de menționat faptul că metricile **ROC-AUC** și **PR-AUC** sunt extrem de importante pe seturi de date dezechilibrate, cum este cazul detectării fraudelor, deoarece acestea oferă o măsură a performanței modelului indiferent de distribuția claselor, fiind mai reprezentative pentru capacitatea modelului de a diferenția între clasele minoritare și majoritare.

Având în vedere aceste rezultate, voi implementa în aplicația finală de detectare a tranzacțiilor frauduloase aceste două modele de predicție: Random Forest și XGBoost. Această alegere va permite utilizatorilor să beneficieze de cele mai bune tehnologii disponibile pentru a identifica tranzacțiile frauduloase, asigurând astfel un nivel ridicat de securitate și acuratețe în analiza tranzacțiilor financiare. Integrând aceste modele, aplicația va oferi flexibilitate și eficiență în detectarea și gestionarea fraudelor, maximizând astfel valoarea adăugată pentru utilizatori.

4. Aplicație

În acest capitol, explorez designul și implementarea unei aplicații menite să îmbunătățească detectarea tranzacțiilor frauduloase, utilizând modelele mele de învățare automată descrise anterior. Pe parcursul acestui capitol, mă voi concentra pe principiile de design și detaliile de implementare ale aplicației, discutând modul în care utilizatorii din instituții financiare pot interacționa cu aplicația. Voi evidenția cum datele sunt colectate și utilizate de modelele mele predictive pentru a identifica tranzacțiile suspecte.

În dezvoltarea aplicației de detectare a tranzacțiilor frauduloase, am folosit metodologia Scrum. Această metodologie a fost introdusă pentru prima dată de Hirotaka Takeuchi și Ikujiro Nonaka în articolul lor „The New Product Development Game" (1986, p. 137). Ideea centrală pe care au subliniat-o autorii este că echipele mici, care pot transfera rapid sarcinile între membri, obțin cele mai bune rezultate.

Prin diverse metode de elicitație, menționate în capitolul anterior, am reușit să identific și să grupez cerințele sistemului. Interacțiunea cu utilizatorii finali a fost crucială pentru a înțelege nevoile specifice ale aplicației.

Metodologia Scrum a fost potrivită pentru acest proiect datorită perioadei scurte de dezvoltare. Fiind singurul dezvoltator, mi-a fost mai ușor să planific sprint-uri de una sau două săptămâni pentru a implementa funcționalitățile necesare.

Astfel, sprint-urile au constat în identificarea entităților și proiectarea bazei de date. Mai apoi, am dezvoltat funcționalitățile de login și logout pentru utilizatori. Tot în acest pas, a trebuit să diferențiez nivelurile de acces ale aplicației, prin atribuirea de roluri diferite pentru persoanele care vor administra platforma și utilizatorii obișnuiți.

În continuare, m-am ocupat de principala funcționalitate a aplicației, mai exact încărcarea fișierelor CSV cu tranzacții financiare și analiza acestora pentru detectarea fraudelor. Utilizatorii pot selecta fișierul dorit, iar aplicația va analiza datele pentru a identifica tranzacțiile suspecte. Crearea modelelor de machine learning a fost o parte esențială și a durat mult timp, implicând cercetare și testare pentru a asigura acuratețea predicțiilor.

4.1 Scopul și utlizarea

Doresc să ajut utilizatorii, în special pe cei din instituțiile financiare, să detecteze tranzacțiile frauduloase într-un mod eficient și precis.

Având în vedere acest lucru, audiența mea țintă pentru aplicație sunt profesioniștii din domeniul financiar care doresc să îmbunătățească securitatea tranzacțiilor și să minimizeze riscurile asociate cu fraudele. Utilizarea așteptată a aplicației este ca acești profesioniști să încarce fișiere CSV cu tranzacții financiare și să utilizeze modelele mele predictive pentru a identifica tranzacțiile suspecte de fraudă.

Pentru a oferi o experiență mai bună utilizatorului, am încorporat predictorii mei într-o aplicație ușor de utilizat, care nu necesită cunoștințe despre modelele de învățare automată. Aplicația este accesibilă printr-o interfață web intuitivă, permițând utilizatorilor să încarce datele și să vizualizeze rezultatele fără efort suplimentar. De asemenea, pentru a facilita utilizarea acesteia, aplicația necesită un minim de configurare inițială.

Prin aceste caracteristici, îmi propun să ofer un instrument robust și eficient pentru detectarea fraudelor financiare, care să se integreze ușor în fluxurile de lucru existente ale instituțiilor financiare.

4.2 Elicitarea cerințelor

Am împărțit cerințele aplicației mele în trei categorii: cerințele funcționale (referitoare la ce ar trebui să facă aplicația), cerințele legate de interfața cu utilizatorul (descriind experiența utilizatorului când interacționează cu sistemul meu) și cerințele de gestionare a datelor.

4.2.1 Cerințe funcționale

Cerințele funcționale se referă la specificațiile de bază ale aplicației și la modul în care aceasta ar trebui să funcționeze. Acestea includ:

- **Detectarea Fraudei**: Aplicația trebuie să poată analiza tranzacțiile financiare încărcate de utilizatori și să identifice tranzacțiile suspecte de fraudă folosind modelele predictive.
- **Încărcarea Fișierelor CSV**: Utilizatorii trebuie să aibă posibilitatea de a încărca fișiere CSV care conțin tranzacțiile financiare ce urmează să fie analizate.
- **Raportare**: Aplicația trebuie să genereze rapoarte detaliate pentru tranzacțiile suspecte, oferind utilizatorilor informații clare și concise despre tranzacțiile identificate ca fiind frauduloase.
- **Actualizarea Modelului**: Aplicația trebuie să permită actualizarea modelelor predictive pe baza datelor noi, asigurând astfel acuratețea și relevanța continuă a detectării fraudei.

4.2.2 Cerințele de interfață cu utilizatorul

Cerințele legate de interfața cu utilizatorul se concentrează pe asigurarea unei experiențe intuitive și eficiente pentru utilizatori:

- **Interfață Intuitivă**: Aplicația trebuie să aibă o interfață ușor de utilizat, care să permită utilizatorilor să navigheze și să îndeplinească sarcinile necesare fără dificultăți.
- **Autentificare Securizată**: Utilizatorii trebuie să se poată autentifica într-un mod securizat pentru a accesa funcționalitățile aplicației.
- **Feedback Imediat**: Aplicația trebuie să ofere feedback imediat utilizatorilor după încărcarea fișierelor și analiza tranzacțiilor, pentru a asigura un flux de lucru eficient.

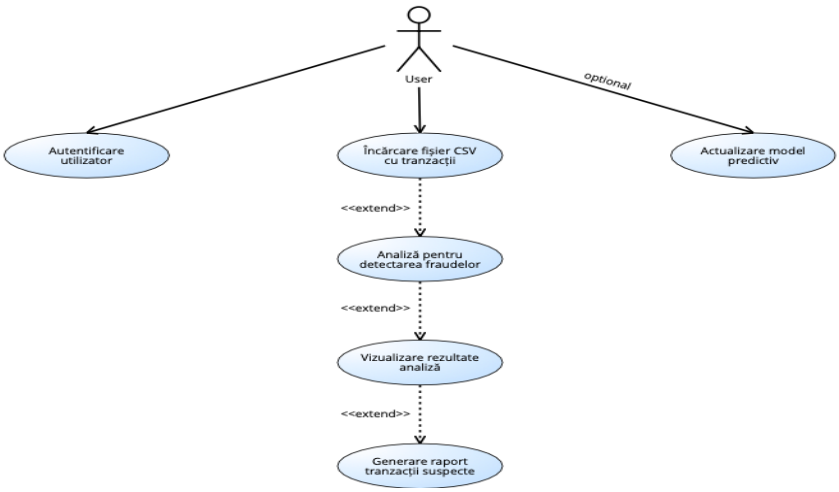


Figura 14 - Diagramă use case

Diagrama ilustrează principalele funcționalități ale aplicației mele de detectare a tranzacțiilor frauduloase, evidențiind interacțiunile utilizatorului cu sistemul. Utilizatorul se poate autentifica, încărca fișiere CSV cu tranzacții, analiza tranzacțiile pentru detectarea fraudelor, vizualiza rezultatele analizei, genera rapoarte detaliate și, opțional, actualiza modelul predictiv pentru a menține acuratețea acestuia.

4.2.3 Cerințe de gestionare a datelor

Cerințele de gestionare a datelor se referă la modul în care datele sunt colectate, stocate și procesate în cadrul aplicației:

- **Stocare Securizată a Datelor**: Datele financiare încărcate de utilizatori trebuie stocate într-un mod securizat, pentru a proteja confidențialitatea și integritatea acestora.

- **Preprocesarea Datelor**: Aplicația trebuie să 43orrect mecanisme de preprocesare a datelor, asigurându-se că datele sunt curate și formatate 43orrect înainte de a fi analizate de modelele predictive.
- **Scalabilitate**: Sistemul de gestionare a datelor trebuie să fie scalabil pentru a putea gestiona volume mari de date tranzacționale, asigurând performanța și eficiența aplicației.

4.3 Analiză

În secțiunea anterioară, am prezentat cerințele aplicației mele în limbaj natural. Acum voi încerca să reduc ambiguitatea acestora oferind mai multe detalii despre fluxurile fiecărui caz de utilizare identificat anterior din perspectiva utilizatorului.

Încărcare Fișier CSV cu Tranzacții

Utilizatorul se autentifică în aplicație și accesează pagina principală, unde este disponibilă opțiunea de încărcare a unui fișier CSV. În această pagină, există o secțiune intitulată "Încărcare Tranzacții", situată vizibil pe ecran. Utilizatorul selectează fișierul CSV de pe dispozitivul său și apasă butonul "Încărcare".

Odată apăsat butonul, fișierul este trimis către Serviciul de Preprocesare, iar butonul este înlocuit cu un indicator de încărcare. După finalizarea procesării, utilizatorul primește o notificare de succes sau de eroare, în funcție de validitatea și integritatea datelor încărcate. În cazul unei încărcări reușite, utilizatorul poate continua cu analiza tranzacțiilor.

Analiză Tranzacții pentru Detectarea Fraudelor

Acest caz de utilizare extinde "Încărcare Fișier CSV cu Tranzacții". După încărcarea fișierului, utilizatorul poate accesa secțiunea "Analiză Tranzacții" pentru a iniția procesul de detectare a fraudelor. Utilizatorul apasă butonul "Predict fraud", care trimite datele către Serviciul Predictiv.

Pe parcursul analizei, un indicator de încărcare este afișat, iar utilizatorul așteaptă rezultatele. În cazul unui succes, rezultatele analizei sunt afișate sub formă de listă a tranzacțiilor suspecte. De asemenea, este afisat si numarul tranzactiilor totale detectate ca fiind suspecte. Utilizatorul primește si un id la fiecare tranzacție suspectă pentru a putea să raporteze aceste rezultate mai departe.

Vizualizare Rezultate Analiză

Acest caz de utilizare extinde "Analiză Tranzacții pentru Detectarea Fraudelor". După ce analiza este completă, utilizatorul poate vizualiza rezultatele direct în interfața aplicației.

Rezultatele includ o listă de tranzacții identificate ca fiind suspecte, împreună cu detalii relevante pentru fiecare tranzacție.

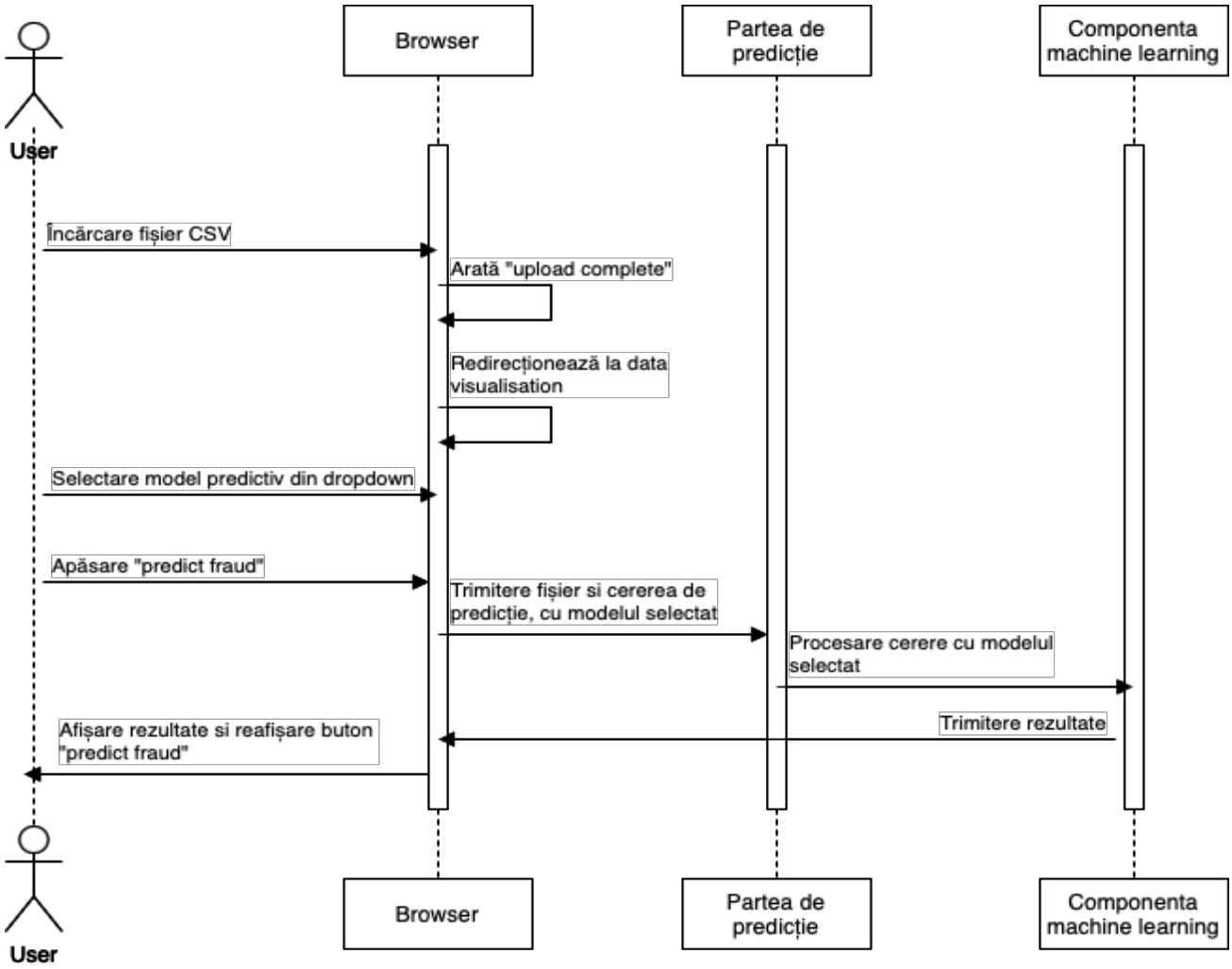


Figura 15 - UML de secvență

Diagrama de secvență prezentată ilustrează procesul complet prin care un utilizator încarcă un fișier CSV cu tranzacții, selectează un model predictiv dintr-un dropdown și inițiază predicția fraudelor. Browserul trimite cererea de predicție către serviciul de predicție, care la rândul său trimite datele către componenta de machine learning pentru procesare. Rezultatele analizei sunt trimise înapoi către browser, care afișează utilizatorului rezultatele și reafișează butonul "Predict Fraud", asigurând astfel un flux de utilizare clar și eficient în aplicație.

4.4 Proiectare

În această secțiune, voi descrie arhitectura generală a aplicației de detectare a tranzacțiilor frauduloase, incluzând componentele principale și modul în care acestea interacționează. Voi prezenta, de asemenea, o diagramă arhitecturală, modelele de date și algoritmii principali utilizați în detectarea fraudelor.

4.4.1 Arhitectura sistemului

Arhitectura aplicației de detectare a tranzacțiilor frauduloase este de tip multi-strat (multi-tier) și include următoarele componente principale:

- **Interfața Utilizatorului (UI):** Aceasta este implementată folosind R și Shiny pentru a oferi o interfață web interactivă. Utilizatorii pot încărca fișiere CSV cu tranzacții, selecta modele predictive și vizualiza rezultatele analizei. Interfața asigură o experiență ușor de utilizat și intuitivă, oferind feedback imediat utilizatorilor și facilitând navigarea prin funcționalitățile aplicației.
- **Serviciul de Preprocesare:** Această componentă se ocupă de validarea și preprocesarea datelor încărcate de utilizatori. Preprocesarea include verificarea integrității datelor, eliminarea valorilor lipsă sau anormale și transformarea datelor în formatul necesar pentru analiză. Acest pas este esențial pentru asigurarea calității datelor înainte de a fi analizate de modelele predictive.
- **Serviciul Predictiv:** Aceasta este componenta centrală care gestionează cererile de predicție. Serviciul Predictiv primește datele preprocesate de la Serviciul de Preprocesare și le trimite către componentele de machine learning pentru analiză. Această componentă este responsabilă pentru orchestrarea fluxului de date și asigurarea că fiecare cerere este procesată corect și eficient.
- **Componentele de Machine Learning (ML):** Acestea includ modelele predictive, cum ar fi XGBoost și rețele neuronale, care analizează tranzacțiile și identifică cele suspecte. Modelele predictive sunt antrenate pe date istorice pentru a învăța tiparele de fraudă și a putea detecta tranzacțiile anormale în datele noi. Componentele ML sunt concepute să fie scalabile și eficiente, pentru a putea gestiona volume mari de date tranzacționale.

- **Baza de Date:** (opțional) Aceasta componentă stochează datele tranzacționale și rezultatele analizelor, dacă aplicația necesită păstrarea unei evidențe detaliate a tranzacțiilor și rezultatelor analizelor. Baza de date este utilizată pentru a permite utilizatorilor să acceseze istoricul analizelor și să genereze rapoarte detaliate. Stocarea securizată a datelor este esențială pentru protejarea confidențialității și integrității informațiilor financiare.

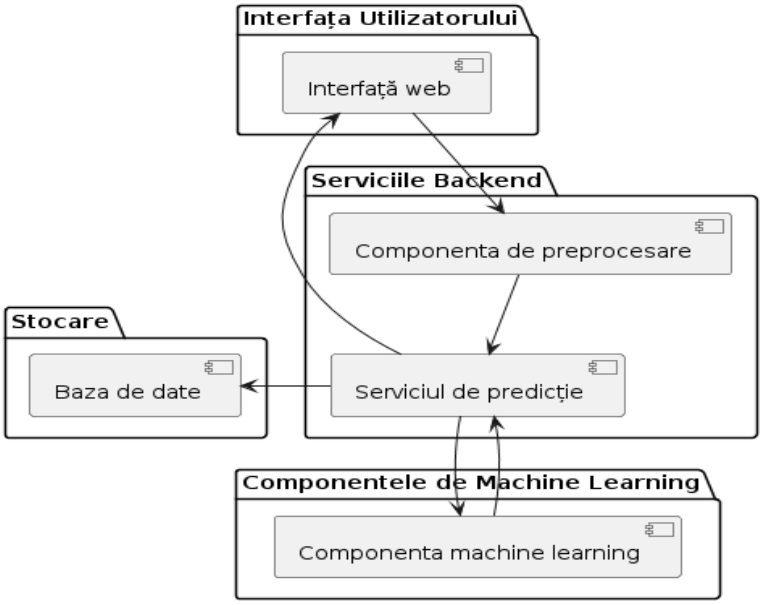


Figura 16 - Diagrama de componente

Diagrama de componente ilustrează arhitectura aplicației de detectare a tranzacțiilor frauduloase. Aplicația este organizată în patru pachete principale: interfața utilizatorului, serviciile backend, componentele de machine learning și stocare. Interfața web permite utilizatorilor să interacționeze cu aplicația, trimițând date către componenta de preprocesare pentru validare și curățare. Datele procesate sunt apoi transmise serviciului de predicție, care utilizează componentele de machine learning pentru a detecta tranzacțiile frauduloase. Rezultatele analizei sunt stocate în baza de date și afișate utilizatorilor prin interfața web.

4.4.2 Algoritmi și procese

Aplicația de detectare a tranzacțiilor frauduloase utilizează trei algoritmi principali de machine learning: Random forest, XGBoost și rețele neuronale. Acești algoritmi au fost selectați și optimizați pentru a oferi performanță maximă în identificarea tranzacțiilor suspecte de fraudă, fiecare având propriile avantaje și caracteristici unice.

Random Forest

Am dezvoltat un model Random Forest echilibrat, care oferă rezultate foarte bune în detectarea fraudelor. Random Forest este cunoscut pentru robustetea sa și pentru capacitatea de a gestiona variații și zgomot în date. Modelul nostru a fost optimizat pentru a atinge o specificitate de 77%, ceea ce înseamnă că este foarte eficient în identificarea tranzacțiilor non-frauduloase, reducând astfel numărul de alarme false. Alegerea acestui model a fost motivată de nevoia de a avea un model robust și fiabil, capabil să gestioneze date diverse și complexe.

XGBoost

Modelul XGBoost a fost ales pentru specificitatea sa foarte bună. XGBoost este cunoscut pentru performanțele sale superioare și eficiența în manipularea datelor mari și complexe. Totuși, acest model tinde să genereze mai multe fals pozitive, ceea ce înseamnă că poate eticheta în mod eronat tranzacții legitime ca fiind suspecte. Pentru a aborda problema tranzacțiilor frauduloase minoritare, au fost folosite class weights, punând accent mai mare pe aceste tranzacții în timpul antrenării modelului. Această caracteristică face din XGBoost un model excelent pentru a fi utilizat în combinație cu alte modele, pentru a asigura un echilibru între detectarea fraudei și minimizarea alarmelor false. Acest model a fost introdus pentru utlizatorii care acceptă un numar mare de fals positive în schimbul a găsirii mai multor tranzacții frauduloase.

Rețele Neuronale

Rețelele neuronale reprezintă cel mai bun și echilibrat model dintre toate, cu performanțe excelente, atingând o specificitate de aproximativ 82%. Acest model este cel mai robust dintre toate, fiind capabil să învețe și să identifice tipare complexe și nelineare în datele tranzacționale. Pentru a aborda problema tranzacțiilor frauduloase minoritare, au fost folosite class weights, punând accent mai mare pe aceste tranzacții în timpul antrenării modelului. Rețelele neuronale sunt deosebit de eficiente în probleme de clasificare și detecție a anomaliilor, motiv pentru care au fost alese pentru a asigura acuratețea și eficiența maximă a sistemului de detectare a fraudelor.

4.5 Implementare

Capitolul de implementare al lucrării descrie procesul prin care aplicația de detectare a tranzacțiilor frauduloase a fost transpusă din faza de proiectare tehnică în cod executabil. În acest capitol, voi detalia modul în care am realizat componentele esențiale ale aplicației, atât pe partea de back-end cât și pe front-end. De asemenea, voi include fragmente de cod elocvente care ilustrează funcționalitățile cheie. Implementarea a fost realizată cu atenție la detalii pentru

a asigura funcționarea corectă și eficientă a sistemului, respectând bunele practici în dezvoltarea software.

Pentru început, voi prezenta structura generală a proiectului, urmată de detalii despre părțile componente. Apoi, voi evidenția utilizarea tehnologiilor și bibliotecilor specifice, care au contribuit la funcționalitatea generală a aplicației. De asemenea, includerea fragmentelor de cod relevante va exemplifica modul în care funcționează sistemul. În plus, voi prezenta codul pentru cele două modele de predicție utilizate în aplicație: Random Forest și XGBoost.

Proiectul este structurat în două părți principale: back-end și front-end.

4.5.1 Configurarea conexiunii la baza de date

Pentru a stoca și gestiona informațiile utilizatorilor, am utilizat o bază de date SQLite. Conexiunea cu baza de date și crearea tabelei de utilizatori au fost realizate astfel:

```
library(DBI)
library(RSQLite)

con <- dbConnect(SQLite(), dbname = "users_db.sqlite")
dbExecute(con, "CREATE TABLE IF NOT EXISTS users (
                  id INTEGER PRIMARY KEY,
                  username TEXT UNIQUE,
                  password TEXT,
                  email TEXT)")
```

Acest cod inițializează conexiunea la baza de date SQLite și creează tabela de utilizatori. Tabela users conține patru coloane: id, username, password și email. Coloana id este definită ca cheie primară și se auto-incrementează. Coloana username este unică pentru a preveni înregistrarea multiplă a aceluiași utilizator.

Gestionarea Utilizatorilor

Aplicația gestionează autentificarea utilizatorilor prin verificarea credențialelor stocate în baza de date. La înregistrarea unui nou utilizator, informațiile acestuia sunt adăugate în tabela users. Dacă un utilizator încearcă să se autentifice, aplicația verifică dacă datele introduse coincid cu cele stocate în baza de date:

```
observeEvent(input$login_button, {
  req(input$login_user, input$login_password)
  query <- sprintf("SELECT * FROM users WHERE username = '%s' AND
password = '%s'",
                   input$login_user, input$login_password)
  user <- dbGetQuery(con, query)
  if (nrow(user) == 1) {
    userLogged(TRUE)
    session$userData$username <- input$login_user
    updateTabsetPanel(session,    "main_tabs",    selected    =
"upload_csv")
  } else {
```

```
    showModal(modalDialog(
      title = "Login Failed",
      "Invalid username or password",
      easyClose = TRUE,
      footer = NULL
    ))
  }
})
```

În cazul în care autentificarea eșuează, utilizatorul primește un mesaj de eroare. Înregistrarea unui nou utilizator se face printr-un formular simplu, iar datele acestuia sunt inserate în baza de date:

```
observeEvent(input$register_button, {
  req(input$register_user,              input$register_password,
input$register_email)
  query <- sprintf("INSERT INTO users (username, password, email)
VALUES ('%s', '%s', '%s')",
                   input$register_user, input$register_password,
input$register_email)
  tryCatch({
    dbExecute(con, query)
    showModal(modalDialog(
      title = "Registration Successful",
      "You can now log in with your credentials.",
      easyClose = TRUE,
      footer = NULL
    ))
    updateTabsetPanel(session, "tabs", selected = "Login")
  }, error = function(e) {
    showModal(modalDialog(
      title = "Registration Failed",
      "Username already exists. Please choose another username.",
      easyClose = TRUE,
      footer = NULL
    ))
  })
})
```

4.5.2 Backend

Backend-ul aplicației este responsabil pentru gestionarea autentificării utilizatorilor, încărcarea fișierelor CSV și efectuarea predicțiilor folosind modelele de machine learning. Pentru aceasta, am utilizat limbajul R și mai multe biblioteci, printre care shiny, DBI, caret, randomForest, xgboost și dplyr.

Încărcarea Fișierelor CSV

Utilizatorii pot încărca fișiere CSV care conțin tranzacțiile financiare pentru analiză:

```
uploaded_data <- reactive({
  req(input$file1)
  df <- read.csv(input$file1$datapath)
  df$id <- 1:nrow(df)


  df <- df[, c("id", setdiff(names(df), "id"))]

  return(df)
})

output$contents <- renderTable({
  df <- uploaded_data()
  return(head(df, 6))
})
```

Acest cod gestionează încărcarea fișierului CSV de către utilizator și adaugă o coloană ID pentru a numerota fiecare rând. După încărcare, primele șase rânduri ale fișierului sunt afișate în aplicația Shiny.

Efectuarea Predicțiilor

Serviciul de predicție se ocupă de realizarea predicțiilor utilizând modelele pre-antrenate și de prezentarea rezultatelor. Setul de date încărcat de utilizator este preprocesat pentru a se asigura că toate coloanele necesare pentru predicții sunt prezente și că datele sunt în formatul corect. Acest proces include verificarea și convertirea variabilelor categorice în factori, precum și crearea matricii de date pentru modelele utilizate.

```
observeEvent(input$predict_button, {
  df <- uploaded_data()
  req(df)

  required_columns <- c("category", "amt", "weekday", "gender",
"distance_to_merchant",        "age_group",        "hour_group",
"city_pop_category")
  missing_columns <- setdiff(required_columns, colnames(df))

  if (length(missing_columns) > 0) {
    showModal(modalDialog(
      title = "Error",
      paste("The uploaded file is missing the following required
columns:", paste(missing_columns, collapse = ", ")),
      easyClose = TRUE,
      footer = NULL
    ))
    return(NULL)
  }
  df <- df %>%
    mutate(across(where(is.character), as.factor))
```

```
  model <- switch(input$model_select,
                  rf = rf_model_weighted,
                  xgb = xgb.fit_final)

  if (input$model_select == "rf") {
    predictions    <-    predict(model,    newdata    =    df[,
required_columns], type = "response")
    df$predicted_fraud    <-    ifelse(predictions    ==    "X1",
"Fraudulent", "Non-Fraudulent")
  } else if (input$model_select == "xgb") {

    model_matrix  <-  model.matrix(~  .  -  1,  data  =  df[,
required_columns])
    dmatrix <- xgb.DMatrix(data = model_matrix)
    predictions <- predict(model, dmatrix)
    df$predicted_fraud <- ifelse(predictions > 0.5, "Fraudulent",
"Non-Fraudulent")
  }

  fraud_count <- sum(df$predicted_fraud == "Fraudulent")
  total_count <- nrow(df)

  output$fraud_count <- renderText({
    paste(fraud_count, "out of", total_count, "transactions are
fraudulent.")
  })

  output$fraudulent_transactions <- renderTable({
    df[df$predicted_fraud == "Fraudulent", ]
  })
})
```

Această secțiune detaliază procesul de preprocesare a datelor și de realizare a predicțiilor utilizând modelele Random Forest și XGBoost. Preprocesarea include verificarea existenței coloanelor necesare, convertirea variabilelor categorice în factori și crearea matricii de date pentru modelul XGBoost.

Vizualizarea Ploturilor

Funcționalitatea de vizualizare a ploturilor permite utilizatorilor să selecteze și să vizualizeze graficele relevante pentru analiza tranzacțiilor frauduloase. Utilizatorii pot alege tipul de grafic pe care doresc să-l vizualizeze, iar aplicația generează automat plotul corespunzător pe baza rezultatelor predicțiilor. După realizarea predicțiilor, utilizatorii au posibilitatea de a selecta tipul de grafic dorit dintr-un meniu drop-down. Aplicația generează și afișează graficul selectat, oferind o vizualizare clară a distribuției tranzacțiilor frauduloase și nefrauduloase.

```
  output$fraud_plot <- renderPlot({
    df <- reactive_predictions()
    req(df)
```

```
    if (input$plot_select == "fraud_weekday") {
      ggplot(df,    aes(x    =    factor(weekday),    fill    =
predicted_fraud)) +
        geom_bar(position = "dodge") +
        labs(title = "Distribuția tranzacțiilor frauduloase și
nefrauduloase pe zilele săptămânii",
             x = "Ziua săptămânii",
             y = "Numărul de tranzacții") +
        theme_minimal() +
        geom_text(stat = "count", aes(label = ..count..), vjust
= -0.5)
    } else if (input$plot_select == "fraud_category") {
      ggplot(df,    aes(x    =    factor(category),    fill    =
predicted_fraud)) +
        geom_bar(position = "dodge") +
        labs(title = "Distribuția tranzacțiilor frauduloase și
nefrauduloase pe categorii",
             x = "Categorie",
             y = "Numărul de tranzacții") +
        theme_minimal() +
        geom_text(stat = "count", aes(label = ..count..), vjust
= -0.5)
    }
  })
})
```

Implementarea backend-ului pentru aplicația de detectare a tranzacțiilor frauduloase a fost realizată cu scopul de a oferi o funcționalitate robustă și eficientă. Prin utilizarea unui stack de tehnologii bine definit și a unor bune practici în dezvoltarea software, am reușit să creăm un sistem care să gestioneze eficient datele încărcate de utilizatori și să efectueze predicții precise asupra tranzacțiilor. Componentele backend includ servicii esențiale precum încărcarea și preprocesarea fișierelor CSV, autentificarea utilizatorilor și realizarea predicțiilor folosind modele de machine learning pre-antrenate. Utilizarea modelului Random Forest și a modelului XGBoost ne-a permis să obținem un echilibru optim între acuratețe și performanță în detectarea fraudelor. Structura modulară și scalabilă a backend-ului asigură flexibilitate în adăugarea de noi funcționalități și îmbunătățirea continuă a performanței aplicației. În ansamblu, soluția implementată demonstrează o abordare eficientă în gestionarea și analizarea tranzacțiilor financiare pentru detectarea fraudelor, oferind utilizatorilor rezultate relevante și de încredere.

4.5.3 Frontend

Partea de front-end a aplicației de detectare a tranzacțiilor frauduloase este esențială pentru a asigura o interacțiune intuitivă și eficientă cu utilizatorul. Aceasta a fost realizată folosind biblioteca Shiny pentru R, care permite dezvoltarea de interfețe web interactive direct din cod R. Interfața utilizatorului este organizată în mai multe tab-uri, fiecare având funcționalități

specifice: autentificare, încărcare fișier CSV, vizualizare date și predicții, și detalii despre modele.

```
library(shiny)

ui <- fluidPage(
  titlePanel("Fraud Detection"),
  sidebarLayout(
    sidebarPanel(
      tabsetPanel(
        id = "tabs",
        tabPanel("Login",
                 textInput("login_user", "Username"),
                 passwordInput("login_password", "Password"),
                 actionButton("login_button", "Login"),
                 actionButton("show_register", "Register")
        ),
        tabPanel("Register",
                 textInput("register_user", "Username"),
                 passwordInput("register_password", "Password"),
                 textInput("register_email", "Email"),
                 actionButton("register_button", "Register"),
                 actionButton("back_to_login", "Back to Login")
        )
      )
    ),
    mainPanel(
      tabsetPanel(
        id = "main_tabs",
        tabPanel("Home", value = "home",
                 h2("Welcome to the Fraud Detection App")),
        tabPanel("Upload CSV", value = "upload_csv",
                 fileInput("file1", "Upload CSV File"),
                 tableOutput("contents")
        ),
        tabPanel("Data Visualization", value = "data_vis",
                 h3("Data Summary"),
                 textOutput("fraud_count"),
                 tableOutput("fraudulent_transactions"),
                 selectInput("model_select", "Select Model",
                             choices = list("Random Forest" =
"rf",
                                            "XGBoost" = "xgb")),
                 actionButton("predict_button",         "Predict
Fraud"),
                 conditionalPanel(
                   condition = "output.showPlotOptions",
                   selectInput("plot_select", "Select Plot",
                               choices = list("Fraud by Weekday"
= "fraud_weekday",
                                              "Fraud          by
Category" = "fraud_category")),
                   plotOutput("fraud_plot")
                 )
        ),
```

```
        tabPanel("Model Details", value = "model_details",
                 uiOutput("model_details")
        )
      )
    )
  )
)
```

Front-end-ul aplicației de detectare a tranzacțiilor frauduloase oferă o interfață intuitivă și ușor de utilizat, asigurând o experiență plăcută pentru utilizatori. Implementarea cu Shiny pentru R permite dezvoltarea rapidă și eficientă a interfețelor web interactive, integrând perfect partea de back-end cu partea de front-end pentru a crea o aplicație robustă și fiabilă. Interfața utilizatorului organizează funcționalitățile în mod clar și accesibil, facilitând utilizarea aplicației pentru detectarea tranzacțiilor frauduloase.

4.5.4 Modele predictive

Random Forest

```
# Antrenează modelul final cu hiperparametrii optimi
class_weights <- c("X0" = 1, "X1" = 10)
set.seed(123)
rf_model_weighted <- randomForest(
  is_fraud ~ .,
  data = bind_rows(fraud_train, fraud_valid),
  mtry = best_mtry,
  ntree = 500,
  nodesize = best_nodesize,
  classwt = class_weights
)


# Realizează predicțiile pe noul set de date
pred_probs_new    <-    predict(rf_model_weighted,    newdata    =
fraud_test, type = "prob")

# Aplică un prag pentru a determina fraudele (ajustabil)
threshold <- 0.5
predicted_fraud <- ifelse(pred_probs_new[, "X1"] > threshold, 1,
0)

# Afișează rezumatul fraudelor prezise
print("Rezumatul fraudelor prezise:")
print(table(predicted_fraud))

# Evaluează modelul final pe setul de testare
pred_probs_rf <- predict(rf_model_weighted, newdata = fraud_test,
type = "prob")
```

```
metrics_rf_final     <-     compute_metrics(fraud_test$is_fraud,
pred_probs_rf[, "X1"])
print(metrics_rf_final$confusion)
print(paste("AUC-ROC:", metrics_rf_final$auc_roc))
print(paste("PR AUC:", metrics_rf_final$pr_auc))
print(paste("F1 Score:", metrics_rf_final$f1_score))
```

În această etapă, finală, modelul este antrenat folosind parametrii mtry, ntree și nodesize, care au fost stabiliți în urma procesului de tuning. De asemenea, ponderile claselor sunt ajustate pentru a acorda o importanță mai mare tranzacțiilor frauduloase (class_weights). Această metodă mi-a adus cele mai bune performanțe.

XGBoost

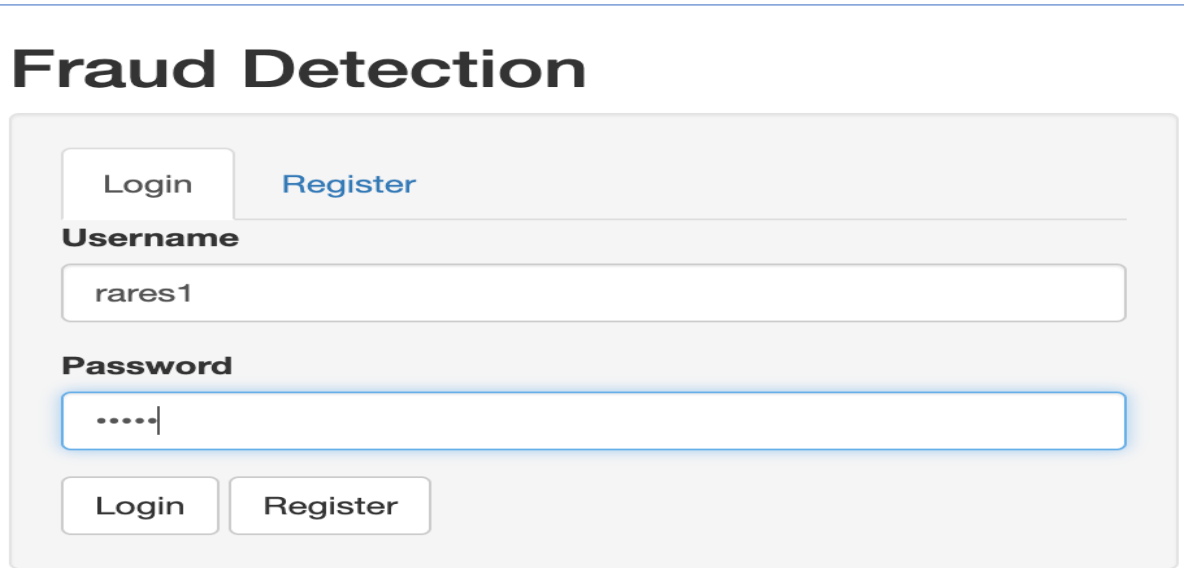
```
# Definește parametrii optimi
best_params <- list(
  eta = 0.01,
  max_depth = 7,
  min_child_weight = 5,
  subsample = 0.8,
  colsample_bytree = 0.8,
  scale_pos_weight = 168.7353,
  objective = "binary:logistic"
)

# Antrenează modelul final cu hiperparametrii optimi
set.seed(123)
xgb.fit_final <- xgboost(
  params = best_params,
  data = dtrain,
  nrounds = 1000,
  verbose = 1
)

# Realizează predicțiile pe setul de testare
pred_test <- predict(xgb.fit_final, dtest)
response_test <- as.factor(fraud_test$is_fraud)
metrics_xgb_test <- compute_metrics(response_test, pred_test)
print(metrics_xgb_test$confusion)
print(paste("AUC-ROC:", metrics_xgb_test$auc_roc))
print(paste("PR AUC:", metrics_xgb_test$pr_auc))
print(paste("F1 Score:", metrics_xgb_test$f1_score))
```

Modelul final XGBoost este antrenat utilizând hiperparametrii definiți anterior. Setul de date de antrenament (dtrain) este folosit pentru antrenare, iar numărul de runde (nrounds) este setat la 1000 pentru a permite modelului să învețe eficient din date. Acest cod reprezintă tunningul final pentru modelul XGBoost, optimizat pentru a oferi rezultate precise și robuste în detectarea fraudelor financiare.

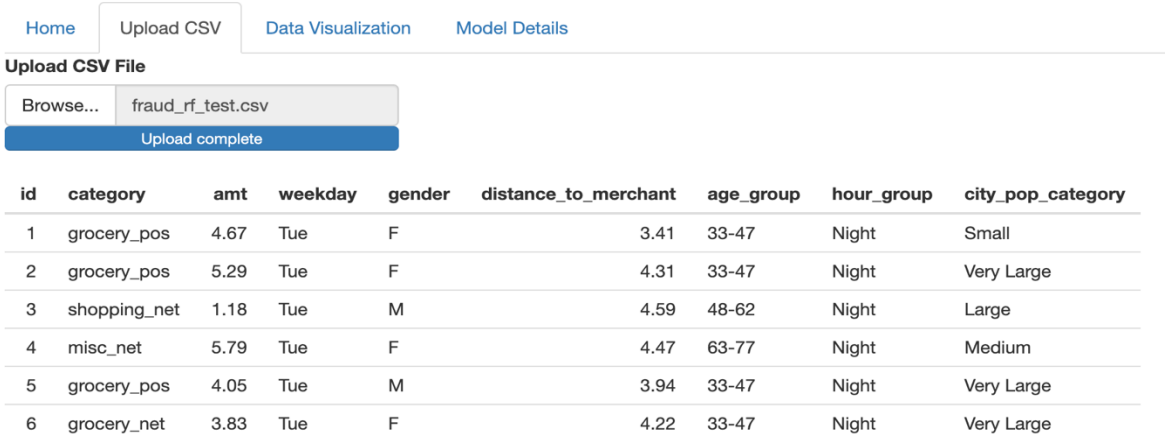
4.6 Testarea

Testarea software reprezintă o etapă esențială în procesul de dezvoltare a aplicațiilor, având rolul de a verifica și valida că toate funcționalitățile implementate corespund cerințelor inițiale și că aplicația funcționează conform așteptărilor. În cadrul aplicației de detectare a tranzacțiilor frauduloase, testarea a fost realizată riguros, utilizând diverse metode și instrumente pentru a asigura acoperirea completă a aspectelor critice legate de funcționalitate, performanță și utilizabilitate.

Am testat autentificarea corectă a utilizatorilor pe baza credențialelor introduse. Utilizatorii autentificați corect sunt redirecționați la pagina principală, iar utilizatorii cu credențiale incorecte primesc un mesaj de eroare.

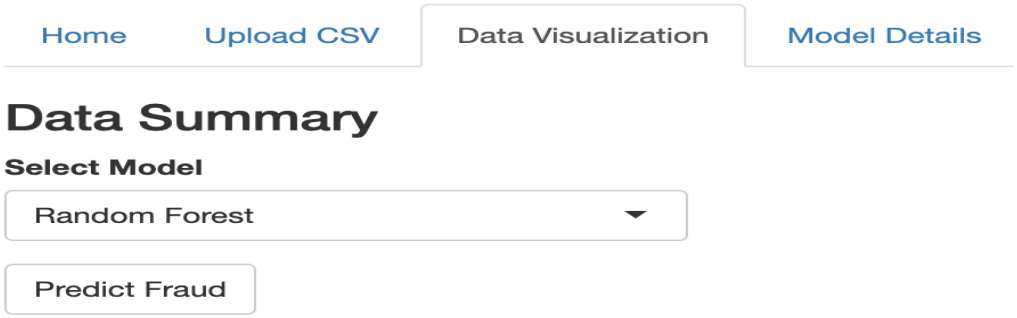


Am testat încărcarea corectă a fișierelor CSV. După autentificare, utilizatorul este redirecționat la pagina de încărcare a fișierelor CSV. Fișierele CSV sunt încărcate și preprocesate corect, iar datele sunt afișate corect.



id	category	amt	weekday	gender	distance_to_merchant	age_group	hour_group	city_pop_category
1	grocery_pos	4.67	Tue	F	3.41	33-47	Night	Small
2	grocery_pos	5.29	Tue	F	4.31	33-47	Night	Very Large
3	shopping_net	1.18	Tue	M	4.59	48-62	Night	Large
4	misc_net	5.79	Tue	F	4.47	63-77	Night	Medium
5	grocery_pos	4.05	Tue	M	3.94	33-47	Night	Very Large
6	grocery_net	3.83	Tue	F	4.22	33-47	Night	Very Large

Fișierul a fost încărcat cu succes de către utilizator, acesta urmeaza sa fie redirectionat pe pagina “Data Visualisation”.



La acest pas, user-ul alege modelul cu care dorește să facă predicția și apasă butonul “Predict Fraud”.

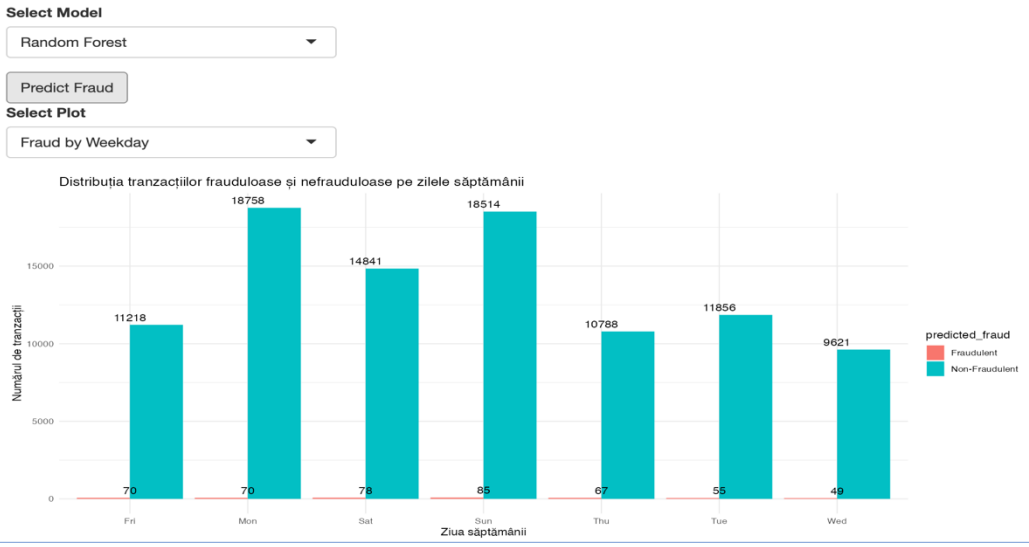
Predicția a fost făcuta, fiind afișat un tabel cu toate tranzacțiile fraudulente și ID-ul lor. De asemenea, înaintea tabelului a fost facută o nunmărare a tranzacțiilor fraudululente.

Data Summary

474 out of 96070 transactions are fraudulent.

id	category	amt	weekday	gender	distance_to_merchant	age_group	hour_group	city_pop_category	predicted_fraud
326	grocery_pos	5.76	Fri	F	4.63	63-77	Night	Medium	Fraudulent
533	shopping_pos	6.59	Sat	M	4.21	78-92	Evening	Small	Fraudulent
538	grocery_pos	5.74	Sat	M	1.48	78-92	Evening	Small	Fraudulent
871	gas_transport	2.76	Tue	F	4.26	18-32	Night	Medium	Fraudulent
1042	grocery_pos	5.72	Wed	F	4.52	63-77	Night	Very Large	Fraudulent

Apoi la finalul tabelului, utilizatorul poate să selecteze un alt model de predicție. De asemenea, acesta poate sa realizeze grafice cu actualele date din tabel.



5. Concluzii și dezvoltări viitoare ale aplicației

Această lucrare a explorat aplicarea algoritmilor de machine learning pentru detectarea tranzacțiilor frauduloase în seturi de date financiare. Am dezvoltat o aplicație web care permite utilizatorilor să încarce fișiere CSV cu tranzacții și să utilizeze modele predictive pentru a identifica tranzacțiile suspecte. Aplicația utilizează două modele principale: Random Forest și XGBoost, fiecare cu propriile sale avantaje și limitări.

Abordarea mea se diferențiază prin accentul pus pe preprocesarea datelor și echilibrarea clasei de date, asigurând astfel performanțe mai bune ale modelelor predictive. După o analiză detaliată a datelor, am implementat modele care au demonstrat o capacitate ridicată de a detecta fraudele, cu specificități notabile. Modelul Random Forest a avut o specificitate de 77%, fiind unul dintre cele mai robuste modele dezvoltate. Modelul XGBoost, deși a avut tendința de a clasifica greșit tranzacțiile nefrauduloase ca fiind frauduloase, a reușit să detecteze un număr mai mare de tranzacții frauduloase.

Cu toate acestea, există mai multe direcții de îmbunătățire și extindere a aplicației:

Extinderea Funcționalităților:

- Integrări suplimentare cu alte platforme financiare pentru a oferi utilizatorilor un set divers de date.
- Adăugarea unor module suplimentare de preprocesare pentru a trata mai eficient datele lipsă și pentru a îmbunătăți acuratețea modelelor.

Îmbunătățirea Performanței:

- Optimizarea hiperparametrilor modelelor pentru a obține rezultate și mai precise.
- Explorarea unor algoritmi avansați precum rețele neuronale și modele ensemble pentru a compara performanțele.

Automatizarea și Ușurința în Utilizare:

- Implementarea unor pipeline-uri CI/CD automate pentru a asigura o experiență de dezvoltare îmbunătățită și pentru a garanta calitatea software-ului.
- Extinderea interfeței utilizatorului pentru a include vizualizări interactive și rapoarte detaliate pentru utilizatori.

Evaluarea Performanței:

- Realizarea unor teste de performanță mai detaliate pentru a măsura impactul în timp real asupra seturilor de date mari.
- Investigarea tehnicilor de reducere a timpului de procesare și optimizarea resurselor pentru a asigura scalabilitatea aplicației.

În concluzie, am reușit să dezvolt o aplicație eficientă și robustă pentru detectarea tranzacțiilor frauduloase, utilizând sisteme inteligente. Implementarea modelelor de machine learning, precum Random Forest și XGBoost, a demonstrat eficiența acestor tehnici în identificarea tranzacțiilor suspecte, îmbunătățind astfel securitatea și integritatea datelor financiare. Aplicația mea oferă un cadru solid pentru analiza și prevenirea fraudelor, iar rezultatele obținute evidențiază potențialul acestor metode în domeniul financiar.

În final, doresc să adresez mulțumiri speciale profesorului coordonator, prof. univ. dr. Gheorghe Cosmin Silaghi, pentru îndrumarea si suportul oferit pe parcursul acestei lucrări.

Bibliografie

Awoyemi, J. O., A. A. (2017). Credit card fraud detection using machine learning techniques: a cooperative analysis. 1-9.

Baabdullah, T., Alzahrani, A., Rawat, D.B., & Liu, C. (2024). Efficiency of Federated Learning and Blockchain in Preserving Privacy and Enhancing the Performance of Credit Card Fraud Detection (CCFD) Systems. *Future Internet*, 16(6), 196.

Browne, R. (2024, February 1). From CNBC: https://www.cnbc.com/2024/02/01/mastercard-launches-gpt-like-ai-model-to-help-banks-detect-fraud.html

Cheng, H. (2023). Credit Card Fraud Detection Using Logistic Regression and Machine Learning Algorithms.

Federal Trade Commission. (2024, February 9). Federal Trade Commission. From https://www.ftc.gov/news-events/news/press-releases/2024/02/nationwide-fraud-losses-top-10-billion-2023-ftc-steps-efforts-protect-public

[1] Fawcett, T. (2006). Introduction to ROC analysis. Pattern Recognition Letters.

García, S., & Fdez, A. (2008, September 20). A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Retrieved April 20, 2024, from https://link.springer.com/article/10.1007/s00500-008-0392-y

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.

Hearst, M. A., & T., S. (1998). Support vector machines. IEEE Intelligent Systems and their Applications, 18-28.

Mastercard. (2024). B2B Mastercard. Preluat de pe: https://b2b.mastercard.com/news-and-insights/blog/navigating-sophisticated-transaction-fraud-trends-in-2024/

Sadineni, P. K. (2021). Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC).

Takeuchi, H., & Nonaka, I. (1986). The new product development game. *Harvard Business Review*, 64 (ediția 1), p. 137-146. Preluat de pe: https://hbr.org/1986/01/the-new-new-product-development-game

[2] Ting, K. M. (2017). Sensitivity and Specificity. Encyclopedia of Machine Learning and Data Mining, 1052-1052.

Xia, J. (2022). Credit Card Fraud Detection Based on Support Vector Machines. Highlights in Science Engineering and Technology.

Xuan, S. L. (2018). Random Forest for Credit Card Fraud Detection. IEEE.

Trivedi, N. K., Simaiya, S., & Lilhore, U. K. (2020). An efficient credit card fraud detection model based on machine learning methods. *International Journal of Advanced Engineering Research and Science*, 7(6), 182-190.

***Archi, Site oficial: https://www.archimatetool.com/

***Bee-Up, Site oficial: https://bee-up.omilab.org/activities/bee-up/download-details/

***Creately, Site oficial: https://www.creately.com

***Dataset: https://www.kaggle.com/datasets/dermisfit/fraud-transaction-dataset/data

***Draw.io, Site oficial: https://app.diagrams.net/

***Luchidchart, Site oficial: https://www.lucidchart.com/

***Metrici, Site: https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc

***PlantUML, Site oficial: https://www.plantuml.com

***RStudio, Site oficial: https://www.rstudio.com/tags/website/

***Shiny, Site oficial: https://www.rstudio.com/products/shiny/

***SQLite, Site oficial: https://www.sqlite.org