UNIVERSITATEA ALEXANDRU IOAN CUZA FACULTATEA DE ECONOMIE SI ADMINISTRAREA AFACERILOR
PROGRAMUL DE MASTER DATA MINING

Explorarea Modelelor Predictive în Finanțe: Optimizarea Predicțiilor de Credit

Profesor coordonator:

Conf. Dr. Turturean Ciprian Ionel

Studenți:

Barsan Alexandru

Efanova Aurelian

Iași 2024

Cuprins

Introducere

Mulți oameni întâmpină dificultăți în obținerea împrumuturilor din cauza istoricelor de credit insuficiente sau inexistente. Și, din păcate, această populație este adesea exploatată de către împrumutători necinstiți.
Home Credit își propune să extindă incluziunea financiară pentru populația nebancarizată oferind o experiență de împrumut pozitivă și sigură. Pentru a asigura că această populație neglijată are o experiență pozitivă a împrumutului, Home Credit folosește o varietate de date alternative – inclusiv informații de la companiile de telecomunicații și tranzacționale – pentru a prezice capacitatea clienților lor de a rambursa.

Deși în prezent Home Credit folosește diverse metode statistice și de învățare automată pentru a face aceste predicții, ei îi provoacă pe participanții la Kaggle să-i ajute să deblocheze întregul potențial al datelor lor. Acest lucru va asigura că clienții capabili de rambursare nu sunt respinși și că împrumuturile sunt acordate cu un capital, o maturitate și un calendar de rambursare care îi vor împuternici pe clienți să aibă succes.
În cadrul acestui proiect, vom explora un set variat de caracteristici ale clienților, cum ar fi istoricul creditelor, comportamentul financiar și informațiile personale, pentru a identifica modele și tendințe care ne pot ajuta să prezicem riscul de neplată. Utilizând tehnici de prelucrare și analiză a datelor, împreună cu algoritmi de învățare automată, vom dezvolta un model capabil să evalueze riscul asociat fiecărui împrumut, permițând instituțiilor financiare să ia decizii mai informate și mai eficiente în gestionarea riscurilor de credit. Proiectul nu numai că abordează o problemă importantă din sectorul financiar, dar oferă și oportunitatea de a dezvolta și de a aplica competențe vitale în domeniul data science, cum ar fi prelucrarea datelor, analiza exploratorie, modelarea predictivă și evaluarea performanței modelului. Prin colaborare, experimentare și inovare, proiectul urmărește să stabilească un standard în predictibilitatea riscului de neplată și să deschidă calea către abordări mai sigure și mai sustenabile în acordarea creditelor.

În cadrul studiului nostru privind probabilitatea de rambursare a creditelor, explorăm diverse variabile ce pot avea un impact semnificativ asupra modelului predictiv. Unele dintre cele mai relevante caracteristici, conform importanței atribuite de modelele de învățare automată, includ:

EXT_SOURCE: Rezultatele din surse externe de evaluare a creditului sunt printre cele mai importante indicatoare. Acestea sunt scoruri, probabil provenind din alte agenții de credit, care măsoară solvabilitatea și istoricul financiar al unui individ, reflectând capacitatea acestuia de a-și onora angajamentele financiare.
DAYS_BIRTH: Vârsta solicitantului, calculată de la data nașterii până la momentul aplicării, poate fi un predictor al stabilității financiare, cu anumite grupuri de vârstă posibil având tendința de a fi mai fiabile în rambursarea creditelor.
AMT_CREDIT și **AMT_ANNUITY**: Suma totală a creditului și anuitatea plății, respectiv, oferă o perspectivă asupra nivelului de îndatorare și capacitatea de rambursare a clientului, elemente centrale în evaluarea riscului de credit.
DAYS_EMPLOYED: Perioada de timp de când solicitantul și-a început angajamentul de muncă curent, care poate indica stabilitatea profesională și, implicit, un flux constant de venituri.
Aceste variabile, împreună cu altele descrise în detaliu în fișierul cu descrierea coloanelor din setul de date, constituie piatra de temelie în construirea unui model predictiv robust. Înțelegerea lor profundă ne va permite să interpretăm corect importanța lor în contextul mai larg al creditării și să anticipăm momentele în care acestea vor reapărea în analizele ulterioare, subliniind rolul lor crucial în predicția riscului de neplată a creditelor.

1. Metodologia studiului empiric

1.1 Eșantionul de date

Datele sunt furnizate de Home Credit, un serviciu dedicat acordării de linii de credit (împrumuturi) populației nebancarizate. Predicția privind capacitatea unui client de a rambursa un împrumut sau dificultățile pe care le-ar putea întâmpina în acest sens reprezintă o necesitate critică pentru afacere, iar Home Credit găzduiește această competiție pe Kaggle pentru a vedea ce tipuri de modele poate dezvolta comunitatea de învățare automată pentru a-i ajuta în această sarcină.

Există 7 surse diferite de date:

- application_train/application_test: principalele date de antrenament și testare cu informații despre fiecare cerere de împrumut la Home Credit. Fiecare împrumut are propriul rând și este identificat prin caracteristica SK_ID_CURR. Datele de antrenament pentru aplicații vin cu TARGET indicând 0: împrumutul a fost rambursat sau 1: împrumutul nu a fost rambursat.
- bureau: date referitoare la creditele anterioare ale clientului de la alte instituții financiare. Fiecare credit anterior are propriul rând în bureau, dar un împrumut din datele aplicației poate avea multiple credite anterioare.
- bureau_balance: date lunare despre creditele anterioare în bureau. Fiecare rând reprezintă o lună dintr-un credit anterior, și un singur credit anterior poate avea multiple rânduri, câte unul pentru fiecare lună a duratei creditului.
- previous_application: cereri anterioare pentru împrumuturi la Home Credit ale clienților care au împrumuturi în datele aplicației. Fiecare împrumut curent din datele aplicației poate avea multiple împrumuturi anterioare. Fiecare cerere anterioară are un rând și este identificată prin caracteristica SK_ID_PREV.
- POS_CASH_BALANCE: date lunare despre împrumuturile anterioare la punctul de vânzare sau împrumuturile pe bază de numerar pe care clienții le-au avut cu Home Credit. Fiecare rând este o lună dintr-un împrumut la punctul de vânzare sau în numerar anterior, și un singur împrumut anterior poate avea multe rânduri.
- credit_card_balance: date lunare despre cardurile de credit anterioare pe care clienții le-au avut cu Home Credit. Fiecare rând este o lună de sold a cardului de credit, și un singur card de credit poate avea multe rânduri.
- installments_payment: istoricul plăților pentru împrumuturile anterioare la Home Credit. Există un rând pentru fiecare plată efectuată și un rând pentru fiecare plată ratată. Această diagramă arată cum sunt legate toate datele:

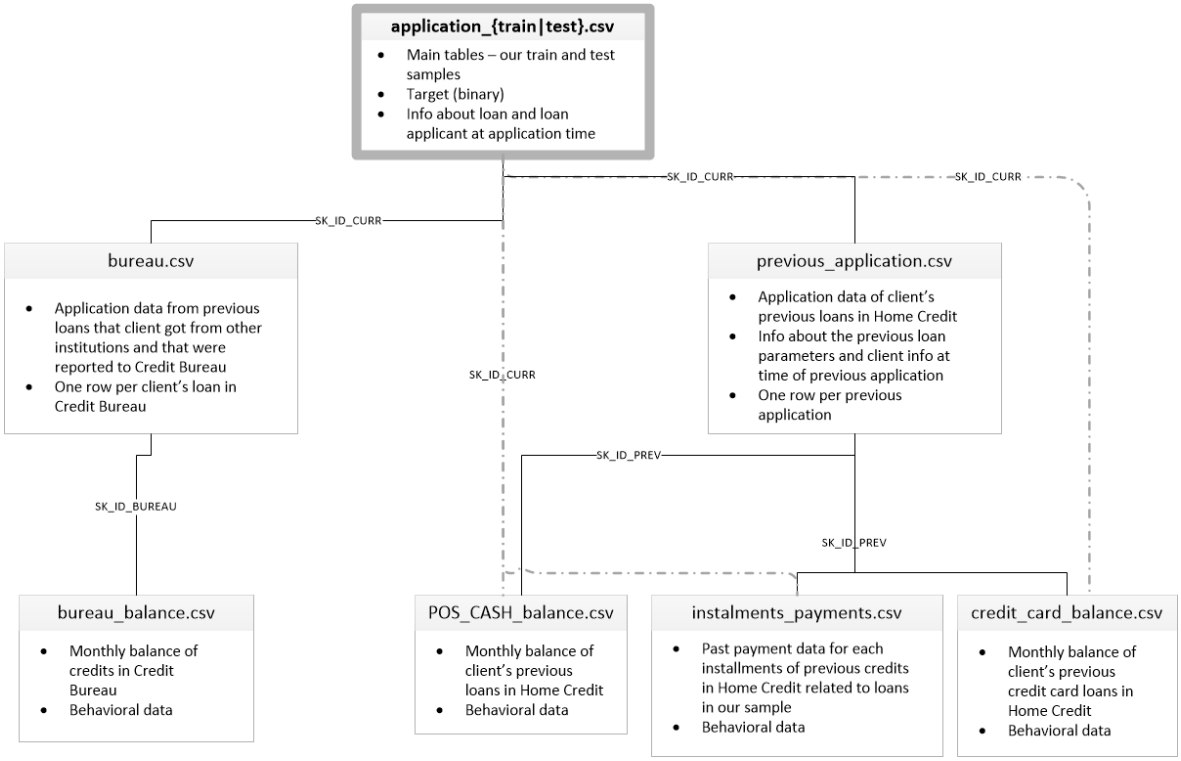

Fig. 1 Prezentarea relațiilor dintre seturile de date

1.2 Metrica utilizată

După ce ne familiarizăm cu datele (lectura descrierilor coloanelor este extrem de utilă), trebuie să înțelegem metrica prin care este evaluată contribuția noastră. În acest caz, se utilizează o metrică comună de clasificare cunoscută sub numele de *Receiver Operating Characteristic Area Under the Curve* (ROC AUC).

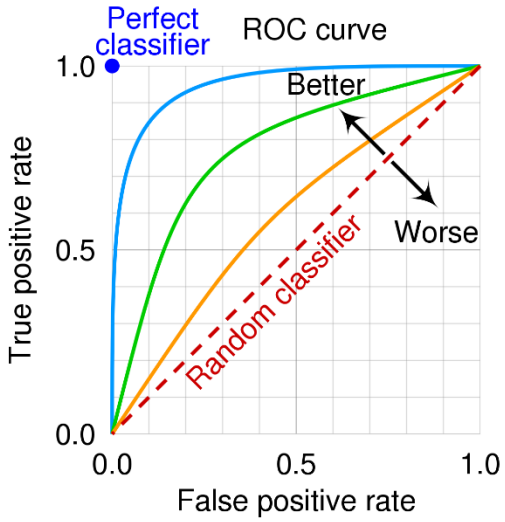

Fig. 2 ROC AUC

O singură linie pe grafic indică curba pentru un singur model, iar mișcarea de-a lungul unei linii indică schimbarea pragului folosit pentru clasificarea unei instanțe pozitive. Pragul începe de la 0 în colțul din stânga jos și merge până la 1 în colțul din dreapta sus. O curbă care este la stânga și deasupra unei alte curbe indică un model mai bun. De exemplu, modelul albastru este mai bun decât modelul verde, care este mai bun decât linia diagonală roșie, care indică un model naiv de ghicire aleatorie.

Aria de Sub Curba (AUC) e pur și simplu aria de sub curba ROC. (Aceasta este integrala curbei.) Această metrică este între 0 și 1, un model mai bun obținând un scor mai mare. Un model care ghicește pur și simplu la întâmplare va avea un ROC AUC de 0,5.

Când evaluăm un clasificator conform ROC AUC, nu generăm predicții de 0 sau 1, ci mai degrabă o probabilitate între 0 și 1. Acest lucru poate fi confuz, deoarece de obicei ne place să gândim în termeni de acuratețe, dar când ne confruntăm cu probleme cu clase dezechilibrate (cum vom vedea că este cazul), acuratețea nu este cea mai bună metrică. De exemplu, dacă am vrea să construim un model care să poată detecta teroriști cu o acuratețe de 99,9999%, am face pur și simplu un model care ar prezice că fiecare persoană nu este terorist. Clar, acest lucru nu ar fi eficace (rata de recall ar fi zero) și folosim metrici mai avansate, cum ar fi ROC AUC sau scorul F1, pentru a reflecta mai exact performanța unui clasificator. Un model cu un ROC AUC ridicat va avea, de asemenea, o acuratețe înaltă, dar ROC AUC este o reprezentare mai bună a performanței modelului.

2. Analiza empirică

2.1 Analiza exploratorie a datelor

2.1.1 Dimensiunea setului

Baza de date originală este compusă din 122 de variabile, inclusiv variabila *TARGET* pe care incercăm să o prezicem și 307511 împrumuturi. Setul de date *application_test* este folosit pentru a evalua modelele înscrise în competiția de pe Kaggle. Acesta nu va fi folosit în scopul acestui studiu.

```
train -  rows: 307511  columns: 122
application_test -  rows: 48744  columns: 121
bureau -  rows: 1716428  columns: 17
bureau_balance -  rows: 27299925  columns: 3
credit_card_balance -  rows: 3840312  columns: 23
installments_payments -  rows: 13605401  columns: 8
previous_application -  rows: 1670214  columns: 37
POS_CASH_balance -  rows: 10001358  columns: 8
```

Fig. 3 Numărul de instanțe si coloane pentru fiecare dataset

2.1.2 Distribuția variabilei TARGET

Setul de date este nebalansat, aproape 92% din instanțe fiind clienți fără probleme de plată, și 8% cu dificultăți de plată. Sunt mult mai multe împrumuturi care au fost plătite la timp decât cele care nu au fost plătite. Odată ce vom ajunge la partea de modelare, ne vom putea ocupa de acest dezechilibru prin setarea variabilelor arborilor de decizie.

```
1 = client with payment difficulties
0 = all other cases

0    0.919271
1    0.080729
```

Fig. 4 Distribuția variabilei TARGET

2.1.3 Valori lipsă

Funcția „*handle_missing_values”* este proiectată pentru a optimiza calitatea datelor în analiza predictivă prin gestionarea eficientă a valorilor lipsă din seturile de date. Aceasta abordează problema valorilor lipsă prin trei strategii principale:

Eliminarea coloanelor: Coloanele cu un procentaj mare de valori lipsă, peste un prag specificat (de exemplu, 50% sau 60%), sunt eliminate. Aceasta previne impactul negativ al datelor insuficiente asupra modelării și asigură că analiza se concentrează pe caracteristici bine reprezentate.

Imputarea valorilor numerice: Valorile lipsă din coloanele numerice sunt înlocuite cu mediana respectivei coloane. Acest pas ajută la păstrarea distribuției generale a datelor și minimizează influența outlier-ilor.

Imputarea valorilor categorice: Pentru coloanele categorice, valorile lipsă sunt înlocuite fie cu cel mai frecvent valoare observată în coloană, fie cu un placeholder prestabilit. Aceasta asigură consistență în categoriile de date și permite utilizarea completă a informațiilor disponibile.

Analiza acestor date arată că există multiple coloane cu valori lipsă, un aspect important în procesul de modelare, deoarece gestionarea acestora poate influența semnificativ performanța modelului.

Date de Antrenament si Testare:
Setul de date pentru antrenament conține 122 de coloane, din care 67 au valori lipsă. Cele mai afectate caracteristici includ COMMONAREA_MEDI, COMMONAREA_AVG, și COMMONAREA_MODE, fiecare cu 69.9% valori lipsă. Aceasta sugerează o lipsă semnificativă de informații despre zonele comune ale proprietăților solicitanților.

Datele Bureau:
Datele din Bureau, care conțin informații despre creditele anterioare ale solicitanților de la alte instituții financiare, includ 17 coloane, cu 7 având valori lipsă. Cea mai afectată caracteristică este AMT_ANNUITY, cu 71.5% valori lipsă, reflectând suma anuală a anuităților neînregistrate pentru creditele anterioare.

Datele Bureau Balance:
Aceste date, care oferă informații lunare despre creditele anterioare, nu prezintă coloane cu valori lipsă, ceea ce este un avantaj pentru analiza stabilității financiare lunare a solicitanților.

Datele despre Carduri de Credit:
Setul de date privind cardurile de credit are 23 de coloane, 9 dintre acestea având valori lipsă, inclusiv sumele trase de la ATM și cele legate de alte trageri, fiecare cu aproximativ 19.5% valori lipsă.
Datele despre Plăți în Rate:

Există 8 coloane, dintre care 2 prezintă valori lipsă, reflectând întârzierile în înregistrarea plăților efectuate și sumele plătite.

Datele POS Cash:
Similar cu datele despre plăți în rate, setul de date POS Cash are 8 coloane și prezintă deficiențe minore în înregistrarea viitoarelor și actualelor rate de plată.

Datele despre Aplicații Anterioare:
Cu 37 de coloane și 16 având valori lipsă, aceste date evidențiază o lipsă majoră de informații în ce privește condițiile de creditare, inclusiv ratele dobânzilor și sumele plătite inițial, ceea ce poate complica analiza istoricului de creditare al solicitanților.

Train Data
Your selected dataframe has 122 columns.
There are 67 columns that have missing values.

	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4

Fig. 5 Numărul de valori lipsă în train_data

Bureau Data
Your selected dataframe has 17 columns.
There are 7 columns that have missing values.

	Missing Values	% of Total Values
AMT_ANNUITY	1226791	71.5
AMT_CREDIT_MAX_OVERDUE	1124488	65.5
DAYS_ENDDATE_FACT	633653	36.9

Fig. 6 Numărul de valori lipsă în bureau

Credit Card Data
Your selected dataframe has 23 columns.
There are 9 columns that have missing values.

	Missing Values	% of Total Values
AMT_PAYMENT_CURRENT	767988	20.0
AMT_DRAWINGS_ATM_CURRENT	749816	19.5
AMT_DRAWINGS_OTHER_CURRENT	749816	19.5
AMT_DRAWINGS_POS_CURRENT	749816	19.5

Fig. 7 Numărul de valori lipsă în credit_data

Installments Data
Your selected dataframe has 8 columns.
There are 2 columns that have missing values.

	Missing Values	% of Total Values
DAYS_ENTRY_PAYMENT	2905	0.0
AMT_PAYMENT	2905	0.0

Fig. 8 Numărul de valori lipsă în installments_data

Previous Application Data
Your selected dataframe has 37 columns.
There are 16 columns that have missing values.

	Missing Values	% of Total Values
RATE_INTEREST_PRIMARY	1664263	99.6
RATE_INTEREST_PRIVILEGED	1664263	99.6
AMT_DOWN_PAYMENT	895844	53.6
RATE_DOWN_PAYMENT	895844	53.6

Fig. 9 Numărul de valori lipsă în previous_application

2.1.4 Vizualizarea datelor

Analiza distribuției unor variabile cheie în setul de date pentru competiția de predicție a riscului de neplată a creditelor dezvăluie aspecte importante despre profilul financiar al solicitanților de credite.

Distribuția venitului total (AMT_INCOME_TOTAL) este concentrată la valori mai mici, cu o scădere bruscă în frecvență pe măsură ce venitul crește, sugerând că majoritatea solicitanților au venituri relativ modeste.

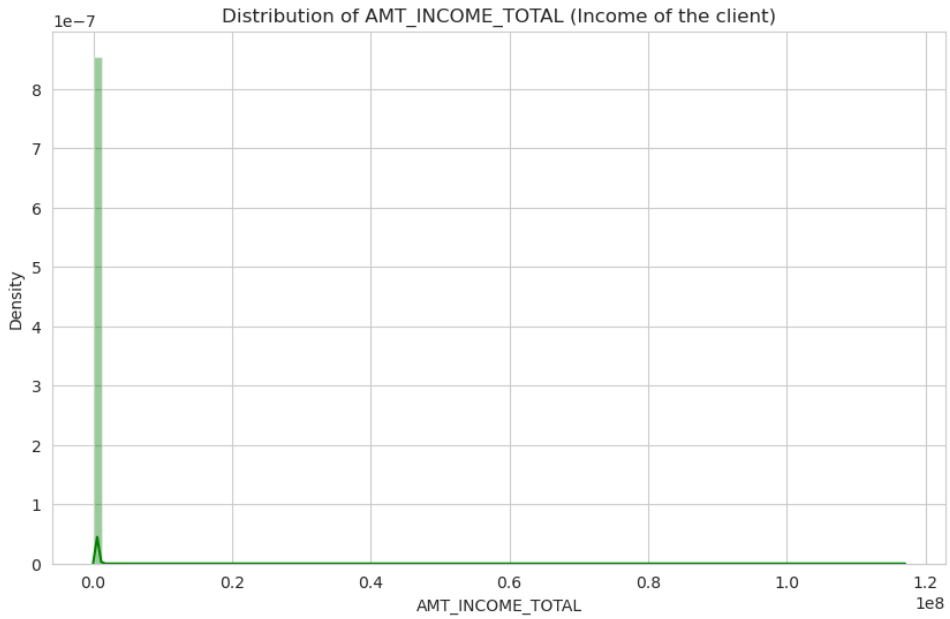

Fig. 10 Distribuția venitului clientului

Suma creditului (AMT_CREDIT) arată o varietate de sume împrumutate, cu vârfuri la diferite praguri, ceea ce poate indica diferite tipuri de produse de creditare.

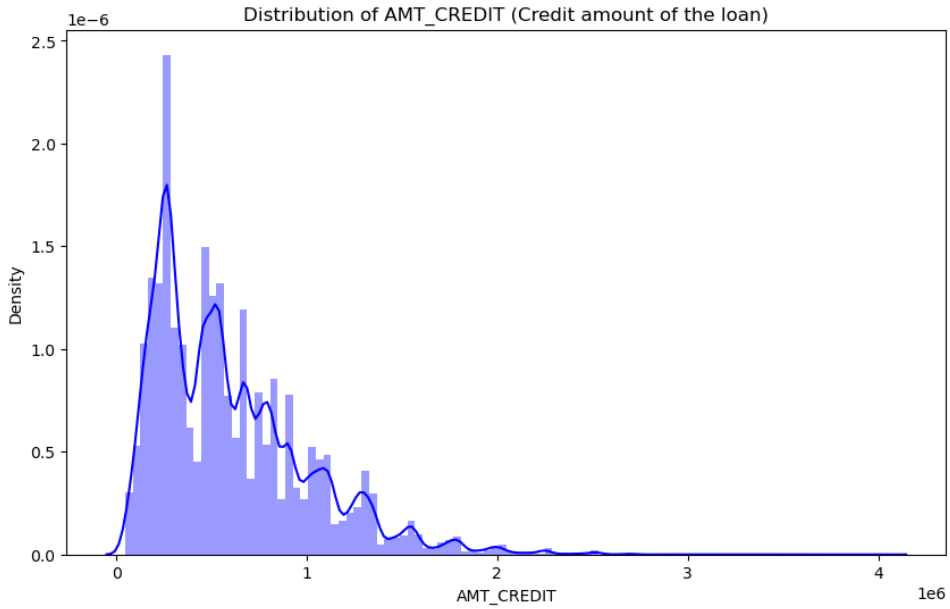

Fig. 11 Distribuția sumei creditului

Anuitatea împrumutului (AMT_ANNUITY) prezintă o distribuție cu vârfuri în zona valorilor mai mici, indicând că rambursările lunare sunt în general la un nivel accesibil pentru majoritatea solicitanților.

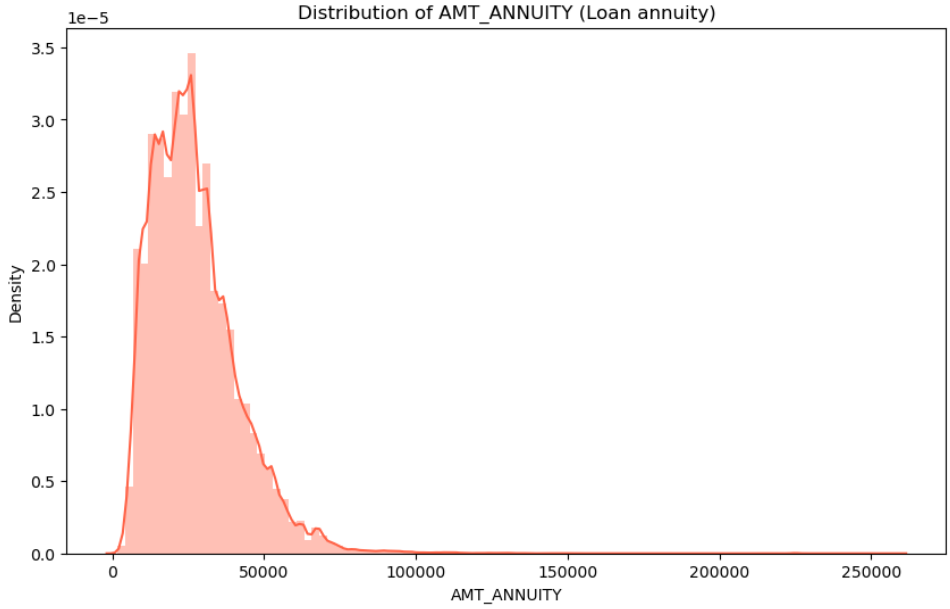

Fig. 12 Distribuția anuității împrumutului

Prețul bunurilor pentru care se solicită împrumutul (AMT_GOODS_PRICE) prezintă de asemenea variații semnificative, cu concentrări în jurul valorilor rotunde, posibil reprezentând prețurile standard pentru bunurile de consum finanțate.

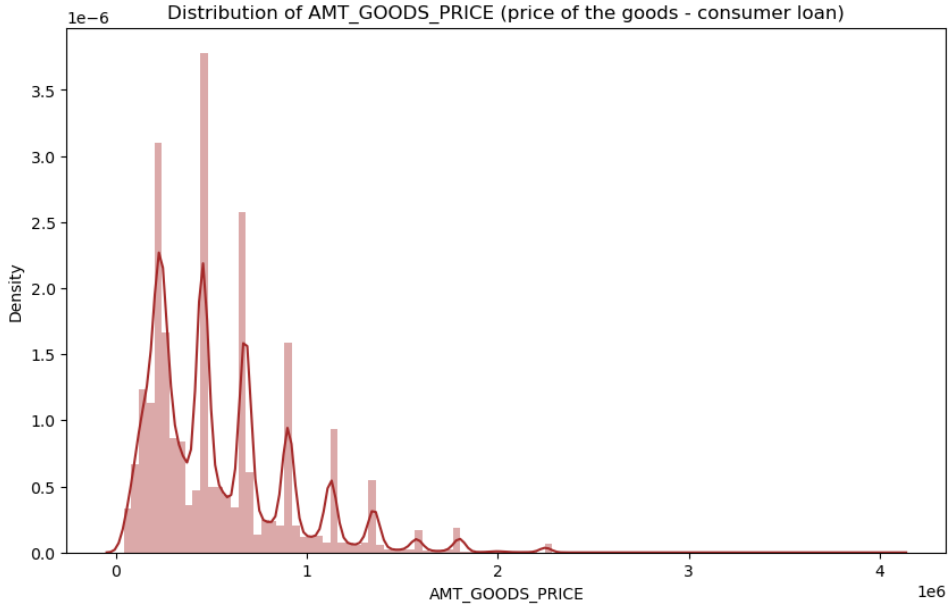

Fig. 13 Distribuția prețului bunurilor

Vârsta solicitanților (DAYS_BIRTH), reflectă o distribuție relativ uniformă peste diferite grupuri de vârstă, cu o ușoară inclinație spre grupurile mai tinere.

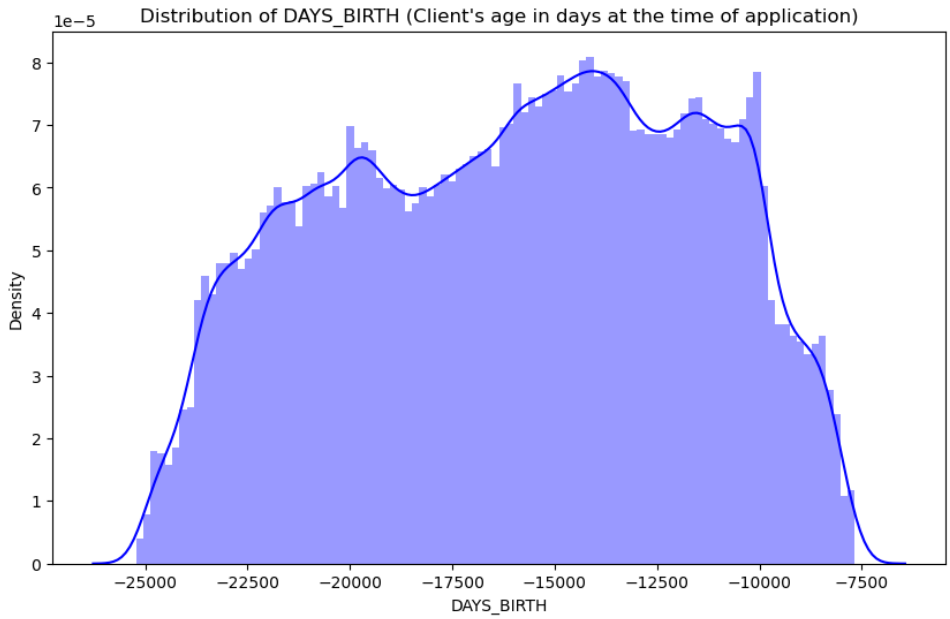

Fig. 14 Distribuția vârstei clientului (nr. de zile de la naștere)

Durata de angajare (DAYS_EMPLOYED) arată o anumită anomalie pentru o mică fracțiune de clienți cu valori extrem de mari, care ar putea necesita o investigație suplimentară, deoarece acestea pot fi erori sau valori speciale pentru tipuri diferite de ocupații.

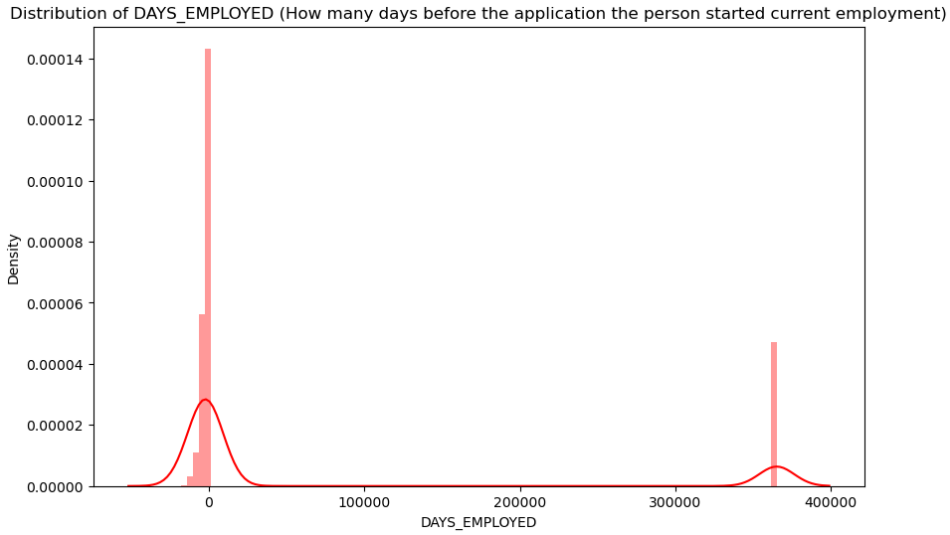

Fig. 15 Distribuția nr. de zile la locul actual de muncă

În final, comparația distribuțiilor variabilelor financiare între grupurile care au rambursat (TARGET = 0) și cele care nu au rambursat (TARGET = 1) creditul arată diferențe subtile care pot fi explorate pentru a construi modele predictiv mai precise. Observăm că în grupul care nu a rambursat există tendințe de a avea venituri, anuități și sume de împrumut mai mici, oferind indicii valoroase pentru profilul riscului de neplată.

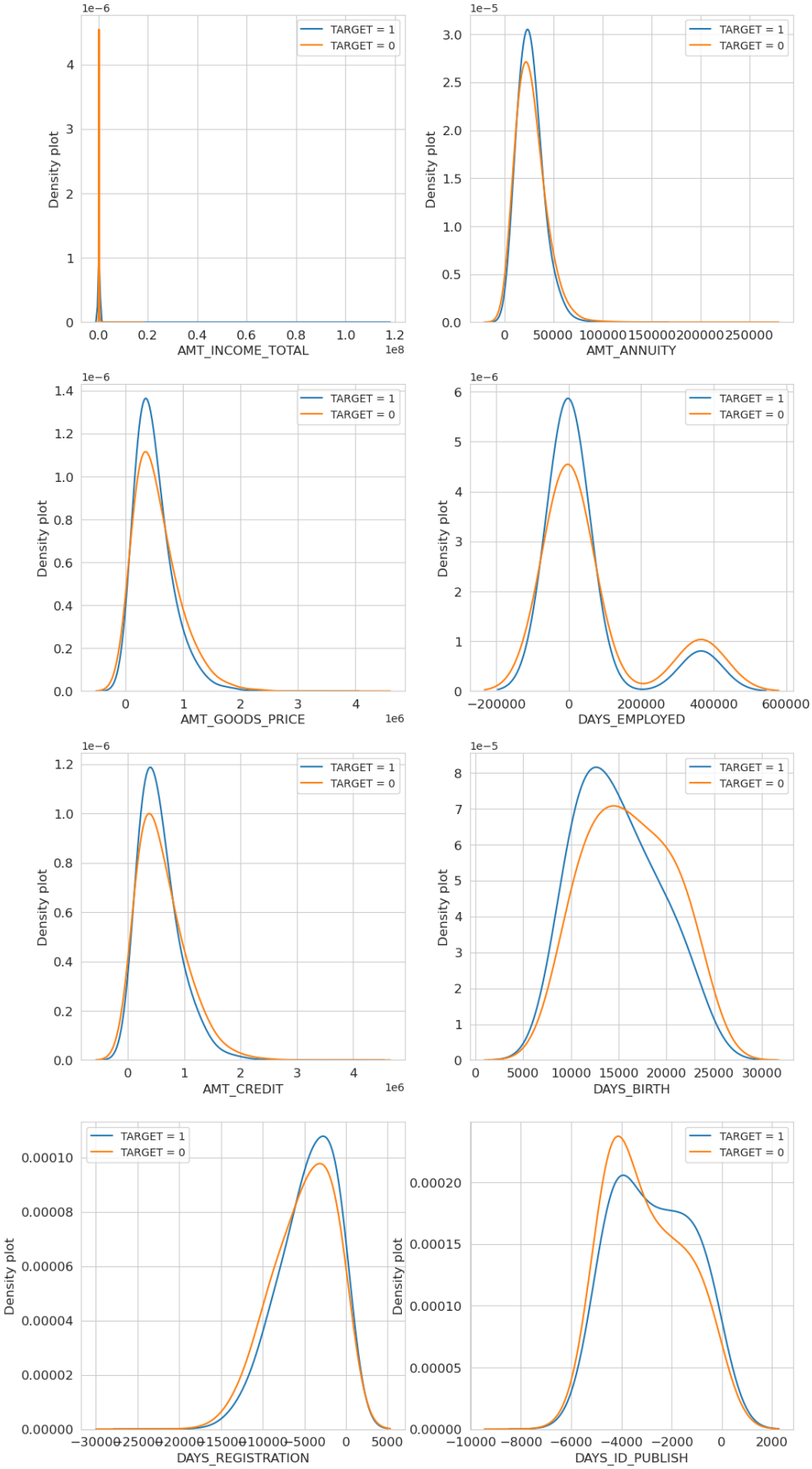

Fig. 16 Comparația distribuțiilor variabilelor financiare între cele două grupuri

2.1.4 Anomalii

În timpul analizei exploratorii a datelor (EDA), este esențial să fim vigilenți în privința anomaliilor care pot apărea în setul de date. Acestea pot proveni din greșeli de introducere a datelor, erori ale echipamentelor de măsurare, sau pot fi măsurători valide, deși extreme. Pentru a identifica anomalii într-o manieră cantitativă, putem folosi metoda "describe" pentru a analiza statisticile unei coloane. Valorile din coloana DAYS_BIRTH sunt negative deoarece sunt înregistrate relativ la data aplicării pentru credit. Pentru a interpreta aceste statistici în ani, putem înmulți cu -1 și împărți la numărul de zile dintr-un an.

Aceasta oferă următoarele statistici:

- Numărul de înregistrări: 307511
- Media: 43.94 ani
- Deviația standard: 11.96 ani
- Minim: 20.52 ani
- 25%: 34.01 ani
- Mediană: 43.15 ani
- 75%: 53.92 ani
- Maxim: 69.12 ani

Vârstele par a fi rezonabile și nu există valori extreme la niciun capăt al intervalului de vârstă. Ce se întâmplă însă cu zilele de angajare?

- Numărul de înregistrări: 307511
- Media: 63815 zile
- Deviația standard: 141275 zile
- Minim: -17912 zile
- 25%: -2760 zile
- Mediană: -1213 zile
- 75%: -289 zile
- Maxim: 365243 zile

Clienții fără anomalii au o rată de neplată de 8.66%.
Clienții cu anomalii au o rată de neplată de 5.40%.
Există 55374 de zile de angajare anomale.

Aceasta este cu adevărat interesant! Se pare că anomalii au o rată mai scăzută de neplată.
Gestionarea anomaliilor depinde de situația specifică și nu există reguli fixe. Una dintre cele mai sigure abordări este de a seta anomaliile la o valoare lipsă, care apoi să fie completată (prin imputare) înainte de modelarea pentru învățare automată. În acest caz, deoarece toate anomaliile au exact aceeași valoare, dorim să le umplem cu aceeași valoare în cazul în care toate aceste credite împărtășesc ceva comun. Valorile anomale par să aibă o oarecare importanță, așa că dorim să informăm modelul de învățare automată dacă

am completat într-adevăr aceste valori. Ca soluție, vom completa valorile anomale cu 'not a number' (np.nan) și apoi vom crea o nouă coloană booleană care indică dacă valoarea a fost sau nu anomală.

2.1.5 Corelații

După ce am abordat variabilele categorice și valorile extreme, ne continuăm analiza exploratorie a datelor (EDA) investigând potențialele corelații între caracteristici și variabila țintă. Unul dintre metodele prin care putem înțelege datele este calculul coeficientului de corelație Pearson între fiecare variabilă și țintă.

Deși coeficientul de corelație Pearson nu este cea mai bună metodă pentru a reprezenta "relevanța" unei caracteristici, acesta ne oferă o idee despre posibilele relații din date. Interpretările generale ale valorii absolute a coeficientului de corelație sunt:

- .00-.19 „foarte slabă”
- .20-.39 „slabă”
- .40-.59 „moderată”
- .60-.79 „puternică”
- .80-1.0 „foarte puternică”

Calculând corelațiile și ordonându-le, putem identifica cele mai pozitive și cele mai negative corelații cu variabila țintă.

Cele mai pozitive corelații sugerează că anumite variabile, cum ar fi vârsta la naștere (DAYS_BIRTH) sau zilele de angajare (DAYS_EMPLOYED), au o asociere directă cu probabilitatea de a rambursa un credit. De exemplu, DAYS_BIRTH prezintă cea mai puternică corelație pozitivă. Aceasta este de așteptat, deoarece vârsta este adesea asociată cu o stabilitate financiară crescută și cu un risc mai mic de neplată.

Pe de altă parte, cele mai negative corelații sunt asociate cu surse externe de scoruri de credit (EXT_SOURCE). Valorile mai ridicate ale acestor scoruri sunt corelate cu un risc scăzut de neplată, ceea ce indică faptul că aceste evaluări externe sunt indicatori semnificativi ai solvabilității unui client.

Aceste corelații ne oferă o înțelegere aprofundată a dinamicii și a interacțiunii dintre caracteristicile clienților și probabilitatea lor de a rambursa creditele. Este esențial să subliniem că corelația nu implică cauzalitate, și astfel, orice interpretare trebuie făcută cu prudență. Acest cadru de corelație ne va ghida în selectarea caracteristicilor pentru modelarea predictivă și în dezvoltarea ulterioară a strategiilor de creditare bazate pe date.

2.1.6 Efectul Vârstei asupra Rambursării Creditului

Investigarea datelor a relevat că vârsta clienților are o relație liniară negativă cu rambursarea creditelor, semnificând că, pe măsură ce clienții îmbătrânesc, ei tind să-și achite împrumuturile la timp mai frecvent. Această descoperire este susținută de coeficientul de corelație Pearson între vârsta clienților (exprimată în zile și convertită în ani) și variabila țintă, care indică o valoare de -0.078. Cu toate că aceasta nu reprezintă o corelație semnificativă, se observă că această variabilă ar putea fi utilă în modelarea predictivă, având în vedere influența sa asupra variabilei țintă.
Analiza distribuției vârstei, prin intermediul unei histograme, nu oferă multe informații izolate, în afara faptului că nu există valori extreme pentru vârsta clienților. Totuși, pentru a vizualiza efectul vârstei asupra rambursării creditului, am folosit un grafic de estimare a densității nucleului (KDE) colorat în funcție de

valoarea variabilei țintă. Acest tip de grafic ilustrează distribuția unei variabile unice și poate fi privit ca o histogramă, creat prin calcularea unui nucleu (de obicei, gaussian) la fiecare punct de date și apoi medierea tuturor nucleelor individuale pentru a dezvolta o curbă netedă unică.

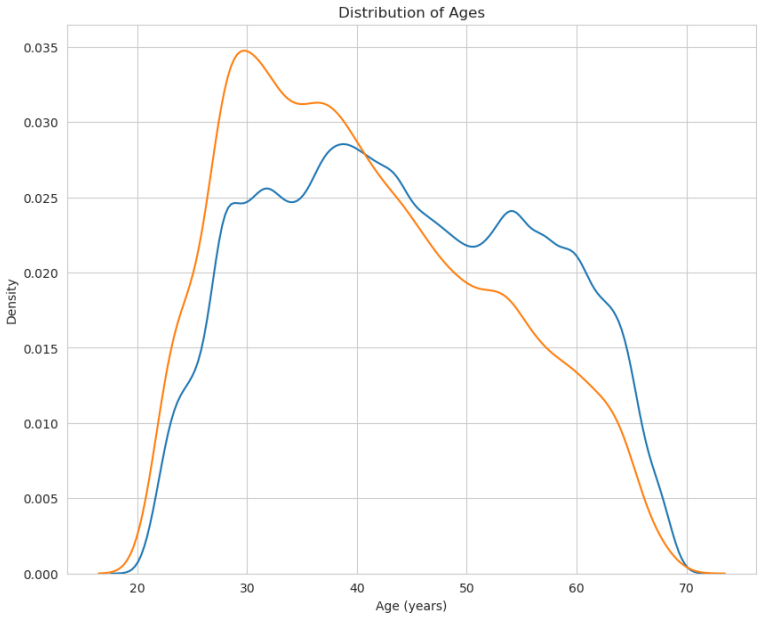

Fig. 17 Distribuția pe vârstă

O examinare suplimentară a vârstei clienților și a corelației acesteia cu incapacitatea de rambursare a creditelor a fost realizată prin segmentarea categoriei de vârstă în intervale de câte 5 ani. Pentru fiecare interval, am calculat valoarea medie a țintei, care ne informează asupra proporției împrumuturilor care nu au fost rambursate în fiecare categorie de vârstă.

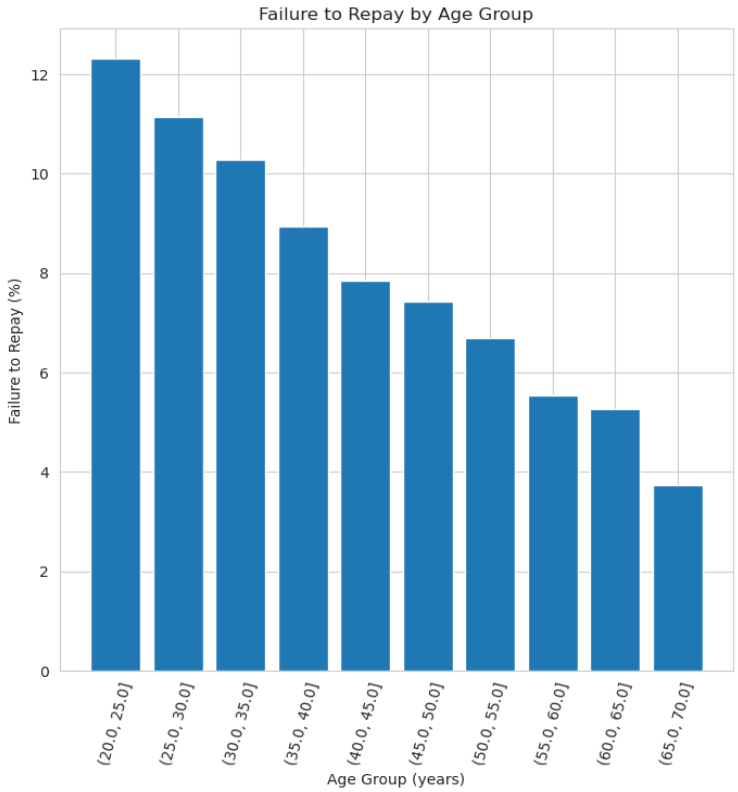

Fig. 18 Default în funcție de vârstă

Graficul rezultat a evidențiat o tendință clară: solicitanții mai tineri sunt mai predispuși să nu ramburseze creditul. Rata eșecului în rambursare este peste 10% pentru cele mai tinere trei grupe de vârstă și sub 5% pentru grupa de vârstă cea mai în vârstă.

Aceste informații pot fi utilizate direct de către bănci: având în vedere că clienții mai tineri sunt mai puțin predispuși să ramburseze creditul, poate ar fi prudent să li se ofere mai multe sfaturi de orientare sau planificare financiară. Acest lucru nu înseamnă că banca ar trebui să discrimineze clienții mai tineri, dar ar fi înțelept să se ia măsuri de precauție pentru a ajuta clienții mai tineri să plătească la timp.

2.1.7 Surse Externe

Cele trei variabile cu cele mai puternice corelații negative cu ținta sunt EXT_SOURCE_1, EXT_SOURCE_2 și EXT_SOURCE_3. Conform documentației, aceste caracteristici reprezintă un "scor normalizat dintr-o sursă de date externă". Nu este clar ce înseamnă acest lucru exact, dar se poate referi la un rating de credit cumulativ creat folosind numeroase surse de date.

Analizând corelațiile dintre caracteristicile EXT_SOURCE și variabila țintă, se observă că toate cele trei au corelații negative, indicând că pe măsură ce valoarea EXT_SOURCE crește, clienții sunt mai predispuși să își ramburseze împrumuturile. De asemenea, putem vedea că DAYS_BIRTH este corelat pozitiv cu EXT_SOURCE_1, sugerând că poate unul dintre factorii acestui scor este vârsta clientului.

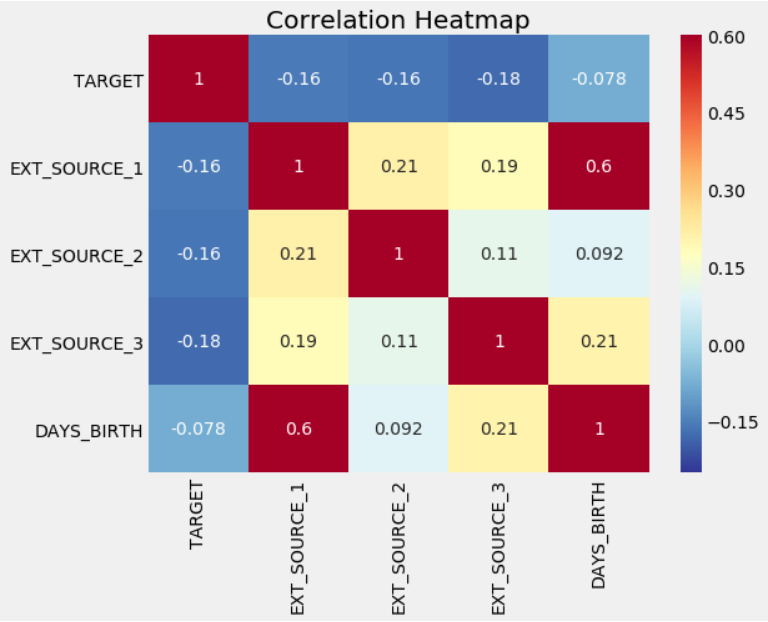

Fig. 19 Corelația intre Target și sursele externe

O examinare a distribuției fiecărei caracteristici EXT_SOURCE, colorată în funcție de valoarea țintei, ne permite să vizualizăm efectul acestei variabile asupra țintei. Se observă că EXT_SOURCE_3 prezintă cea mai mare diferență între valorile țintei. Este evident că această caracteristică are o relație cu probabilitatea ca un solicitant să ramburseze un împrumut. Deși relația nu este foarte puternică (de fapt, toate sunt considerate foarte slabe), aceste variabile vor fi totuși utile pentru un model de învățare automată în predicția dacă un solicitant va rambursa sau nu un împrumut la timp.

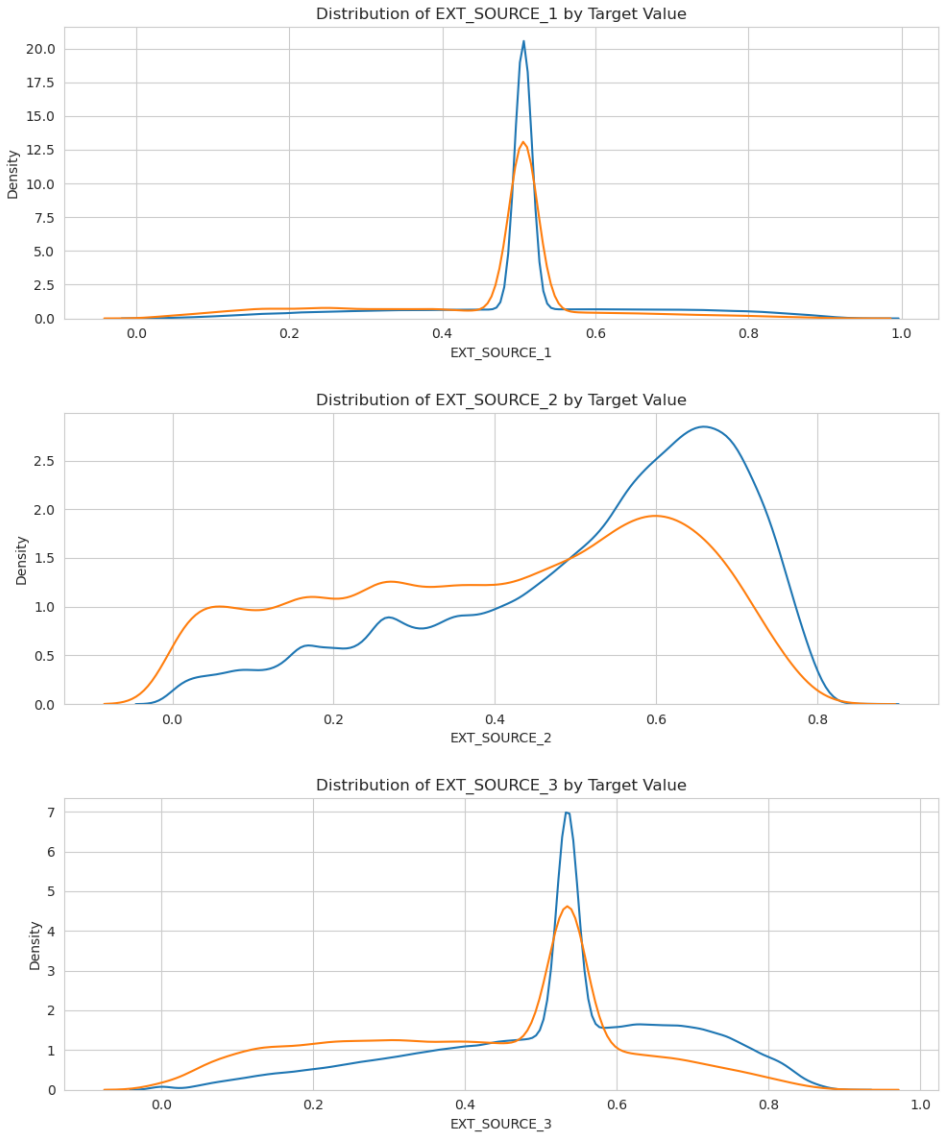

Fig. 20 Distribuția surselor externe

Această analiză a sursei externe subliniază un potențial instrument valoros pentru instituțiile financiare în evaluarea și managementul riscului de credit. Introducerea acestor scoruri în modelare ar putea îmbunătăți semnificativ capacitatea de a discerne între solicitanții de credit cu risc mai mic sau mai mare de neplată.

3. Bazele Ingineriei Automatizate a Caracteristicilor

În această secțiune, vom explora aplicarea ingineriei automatizate a caracteristicilor pe setul de date din competiția Home Credit Default Risk, folosind biblioteca Featuretools. Featuretools este un pachet Python open-source destinat creării automatizate de noi caracteristici (features) din multiple tabele de date structurate și interconectate. Este un instrument ideal pentru probleme precum competiția Home Credit Default Risk, unde există mai multe tabele corelate care trebuie combinate într-un singur dataframe pentru antrenare (și unul pentru testare).

Ingineria Caracteristicilor

Obiectivul ingineriei caracteristicilor este de a crea noi caracteristici (denumite și variabile explicative sau predictori) pentru a reprezenta cât mai multă informație din întregul set de date într-un singur tabel. De obicei, acest proces se realizează manual folosind operații pandas precum groupby, agg sau merge și poate fi foarte laborios. În plus, ingineria manuală a caracteristicilor este limitată atât de constrângerile de timp umane, cât și de imaginație: pur și simplu nu ne putem gândi la toate posibilele caracteristici care vor fi utile. Importanța creării caracteristicilor adecvate nu poate fi supraestimată deoarece un model de învățare automată poate învăța doar din datele pe care i le furnizăm. Extracția cât mai multor informații posibile din seturile de date disponibile este crucială pentru crearea unei soluții eficiente.

Ingineria automatizată a caracteristicilor urmărește să ne ajute cu problema creării caracteristicilor prin construirea automată a sute sau mii de noi caracteristici dintr-un set de date. Featuretools - singura bibliotecă pentru ingineria automatizată a caracteristicilor în momentul de față - nu va înlocui analistul de date, dar îi va permite acestuia să se concentreze pe părți mai valoroase ale fluxului de lucru al învățării automate, cum ar fi livrarea de modele robuste în producție.

Aici vom aborda conceptele ingineriei automatizate a caracteristicilor cu Featuretools și vom arăta cum să o implementăm pentru competiția Home Credit Default Risk. Ne vom limita la elementele de bază pentru a înțelege ideile și apoi vom construi pe această fundație în lucrările ulterioare când personalizăm Featuretools.

3.1 Featuretools

Featuretools este o bibliotecă Python open-source pentru crearea automată de caracteristici noi dintr-un set de tabele interconectate, folosind o tehnică numită sinteza profundă a caracteristicilor. Ingineria automată a caracteristicilor, ca multe subiecte din învățarea automată, este un domeniu complex bazat pe o fundație de idei mai simple. Trecând prin aceste idei pas cu pas, ne putem construi înțelegerea modului în care Featuretools funcționează, ceea ce ne va permite ulterior să îl folosim la maximum.

Vor fi câteva concepte pe care le vom acoperi pe parcurs:

- Entități și Seturi de Entități
- Relații între tabele
- Primitive de caracteristici: agregări și transformări
- Sinteza profundă a caracteristicilor

O entitate este pur și simplu un tabel sau, în Pandas, un dataframe. Observațiile sunt în rânduri și caracteristicile în coloane. O entitate în Featuretools trebuie să aibă un index unic, unde niciunul dintre elemente nu este duplicat. În prezent, doar app, bureau și previous au indici unici (SK_ID_CURR, SK_ID_BUREAU și SK_ID_PREV, respectiv). Pentru celelalte dataframe-uri, trebuie să trecem make_index = True și apoi să specificăm numele indexului. Entitățile pot avea, de asemenea, indici de timp unde fiecare intrare este identificată printr-un timp unic. (Nu există date și ore în niciunul dintre date, dar există timpuri relative, date în luni sau zile, pe care am putea lua în considerare tratamentul lor ca variabile de timp).

Un Set de Entități este o colecție de tabele și relațiile dintre ele. Acesta poate fi gândit ca o structură de date cu propriile sale metode și atribute. Utilizarea unui Set de Entități ne permite să grupăm mai multe tabele și să le manipulăm mult mai rapid decât tabelele individuale.

În primul rând, vom crea un set de entități gol numit clienți pentru a ține evidența tuturor datelor. Dupa, vom definim fiecare entitate sau tabel de date. Trebuie să trecem un index dacă datele au unul sau make_index = True dacă nu. Featuretools va deduce automat tipurile de variabile, dar le putem schimba și noi dacă este necesar. De exemplu, dacă avem o variabilă categorică care este reprezentată ca un întreg, s-ar putea să dorim să informăm Featuretools despre tipul corect.

3.2 Relațiile dintre date

Conceptul de relații este fundamental nu doar în Featuretools, ci în orice bază de date relațională. Cel mai bun mod de a gândi o relație de tipul unu-la-mulți este analogia părinte-copil. Un părinte este un individ unic, dar poate avea mai mulți copii. Copiii pot avea la rândul lor mai mulți copii. Într-un tabel părinte, fiecare individ are un singur rând. Fiecare individ din tabelul părinte poate avea mai multe rânduri în tabelul copil.

De exemplu, dataframe-ul app are un rând pentru fiecare client (SK_ID_CURR), în timp ce dataframe-ul bureau are mai multe împrumuturi anterioare (SK_ID_PREV) pentru fiecare părinte (SK_ID_CURR). Prin urmare, dataframe-ul bureau este copilul dataframe-ului app. La rândul său, dataframe-ul bureau este părintele bureau_balance deoarece fiecare împrumut are un rând în bureau, dar mai multe înregistrări lunare în bureau_balance.

Două tabele sunt legate printr-o variabilă comună. Dataframe-urile app și bureau sunt legate prin variabila SK_ID_CURR, în timp ce bureau și bureau_balance sunt legate cu SK_ID_BUREAU. Definirea relațiilor este relativ directă, iar diagrama oferită de competiție este utilă pentru a vedea relațiile. Pentru fiecare relație, trebuie să specificăm variabila părinte și variabila copil. În total, există șase relații între tabele.

```
Entityset: clients
  DataFrames:
    app [Rows: 307511, Columns: 134]
    bureau [Rows: 1716428, Columns: 17]
    previous [Rows: 1670214, Columns: 37]
    bureau_balance [Rows: 27299925, Columns: 4]
    cash [Rows: 10001358, Columns: 9]
    installments [Rows: 13605401, Columns: 10]
    credit [Rows: 3840312, Columns: 24]
  Relationships:
    bureau.SK_ID_CURR -> app.SK_ID_CURR
    bureau_balance.SK_ID_BUREAU -> bureau.SK_ID_BUREAU
    previous.SK_ID_CURR -> app.SK_ID_CURR
    cash.SK_ID_PREV -> previous.SK_ID_PREV
    installments.SK_ID_PREV -> previous.SK_ID_PREV
    credit.SK_ID_PREV -> previous.SK_ID_PREV
```

Fig. 21 Relațiile dintre date

Trebuie să fim precauți să nu creăm un grafic tip diamant, în care există mai multe căi de la un părinte la un copil. Dacă legăm direct aplicația (app) și numerarul (cash) prin SK_ID_CURR; aplicațiile anterioare (previous) și numerarul prin SK_ID_PREV; și aplicația și aplicațiile anterioare prin SK_ID_CURR, atunci am creat două căi de la aplicație la numerar. Aceasta duce la ambiguitate, deci abordarea pe care trebuie să o luăm este să legăm aplicația de numerar prin intermediul aplicațiilor anterioare. Stabilim o relație între aplicațiile anterioare (părintele) și numerar (copilul) folosind SK_ID_PREV. Apoi stabilim o relație între aplicația (părintele) și aplicațiile anterioare (acum copilul) folosind SK_ID_CURR. Apoi, Featuretools va putea crea caracteristici în aplicație derivate atât din aplicațiile anterioare cât și din numerar, prin stivuirea mai multor primitive.

Toate entitățile din setul de entități pot fi legate unele de altele. Teoretic, acest lucru ne permite să calculăm caracteristici pentru oricare dintre entități, dar în practică, vom calcula caracteristici doar pentru dataframe-ul aplicației, deoarece acesta va fi folosit pentru antrenare/testare.

3.3 Primitive de Caracteristici (Feature primitives)

Un primitiv de caracteristici este o operație aplicată unui tabel sau unui set de tabele pentru a crea o caracteristică. Acestea reprezintă calcule simple, multe dintre ele folosite deja în ingineria manuală a caracteristicilor, care pot fi suprapuse unele peste altele pentru a crea caracteristici complexe. Primitivele de caracteristici se împart în două categorii:

Agregare: funcție care grupează împreună punctele de date copil pentru fiecare părinte și apoi calculează o statistică precum media, minimul, maximul sau deviația standard. Un exemplu este calculul sumei maxime a împrumuturilor anterioare pentru fiecare client. O agregare funcționează pe mai multe tabele folosind relații între tabele.
Transformare: operațiune aplicată uneia sau mai multor coloane într-un singur tabel. Un exemplu ar fi luarea valorii absolute a unei coloane, sau găsirea diferenței între două coloane într-un singur tabel.
O listă a primitivelor de caracteristici disponibile în Featuretools poate fi vizualizată mai jos.

Primitive de Agregare:

max: Găsește valoarea maximă non-nulă a unei caracteristici numerice.
num_true: Găsește numărul de valori 'True' într-o variabilă booleană.
std: Găsește deviația standard a unei caracteristici numerice ignorând valorile nule.
min: Găsește valoarea minimă non-nulă a unei caracteristici numerice.
sum: Numără numărul de elemente ale unei caracteristici numerice sau booleană.

3.4 DFS

Sinteza Profundă a Caracteristicilor (Deep Feature Synthesis - DFS[1]) este procesul utilizat de Featuretools pentru a crea caracteristici noi. DFS suprapune primitivele de caracteristici pentru a forma caracteristici cu o "adâncime" egală cu numărul de primitive. De exemplu, dacă luăm valoarea maximă a sumelor împrumuturilor anterioare ale unui client (de exemplu, MAX(previous.loan_amount)), aceasta este o "caracteristică profundă" cu o adâncime de 1. Pentru a crea o caracteristică cu o adâncime de două, am putea suprapune primitive prin luarea valorii maxime a plăților medii lunare pe fiecare împrumut anterior al unui client (cum ar fi MAX(previous(MEAN(installments.payment)))). Articolul original despre ingineria automată a caracteristicilor utilizând sinteza profundă a caracteristicilor merită citit.

Pentru a efectua DFS în Featuretools, folosim funcția dfs, pasându-i un set de entități, entitatea țintă (unde dorim să creăm caracteristicile), primitivele de agregare de folosit, primitivele de transformare de folosit și adâncimea maximă a caracteristicilor. Aici vom utiliza primitivele de agregare și transformare implicite, o adâncime maximă de 2 și vom calcula primitive pentru entitatea app. Deoarece acest proces este costisitor din punct de vedere computațional, putem rula funcția folosind features_only = True pentru a returna doar o listă a caracteristicilor și nu pentru a calcula efectiv caracteristicile. Acest lucru poate fi util pentru a privi caracteristicile rezultate înainte de a începe un calcul extins.

3.5 Caracteristici specifice

Prin acest fragment cream caracteristici domeniu specifice, utilizând date din mai multe tabele relaționate, pentru a îmbogăți setul de date al aplicației de credit. În procesul de feature engineering, s-au definit și calculat următoarele caracteristici noi:

1. Raportul Datorie-Credit (debt_credit_ratio_None): Pentru fiecare client (identificat prin SK_ID_CURR), s-a calculat raportul dintre suma totală a datoriilor și suma totală a creditelor, reprezentând nivelul de îndatorare în raport cu creditul total obținut.
2. Raportul Credit-Anuitate (credit_annuity_ratio): Pentru fiecare aplicație, s-a calculat raportul dintre suma creditului (AMT_CREDIT) și anuitatea plății (AMT_ANNUITY), care poate reflecta capacitatea clientului de a gestiona plățile periodice față de mărimea creditului.
3. Ultima Combinație de Produse (prev_PRODUCT_COMBINATION): S-a selectat ultima combinație de produse financiare pentru fiecare client, bazată pe ordonarea cronologică inversă a deciziilor de credit (câmpul DAYS_DECISION).
4. Media Zilelor de Credit (DAYS_CREDIT_mean): S-a calculat media numărului de zile de la creditul anterior până la aplicația curentă, furnizând o estimare a frecvenței și recentității activităților de creditare anterioare.

[1] *https://dai.lids.mit.edu/wp-content/uploads/2017/10/DSAA_DSM_2015.pdf*

5. Raportul Credit-Pretul Bunurilor (credit_goods_price_ratio): Pentru fiecare aplicație, s-a determinat proporția dintre suma creditului și prețul bunurilor (împrumutul pentru achiziția de bunuri), sugerând nivelul finanțării în raport cu valoarea bunului achiziționat.
6. Ultimul Credit Activ (last_active_DAYS_CREDIT): S-a identificat data ultimului credit activ înregistrat, indicând cea mai recentă activitate de creditare.
7. Avansul Creditului (credit_downpayment): S-a calculat diferența dintre prețul bunurilor și suma creditului, ceea ce poate reprezenta avansul plătit.
8. Raportul Mediu al Plăților în Rate (installment_payment_ratio_1000_mean_mean): Pentru plățile în rate mai recente (mai puțin de 1000 de zile în trecut), s-a calculat media raportului dintre suma plătită și suma instalamentului, oferind o perspectivă asupra comportamentului de plată.
9. Raportul Anuitate la Instalmentul Maxim (annuity_to_max_installment_ratio): S-a comparat anuitatea plății cu cea mai mare plată în rate, furnizând o idee despre proporția anuității în raport cu cele mai mari obligații de plată ale clientului.
10. Media Surse Externe (EXT_SOURCES_MEAN) și Maximul Surse Externe (EXT_SOURCES_MAX): Pentru a evalua impactul general al scorurilor de la surse externe, s-au calculat media și maximul dintre cele trei scoruri externe disponibile.

Prin crearea acestor caracteristici domeniu specifice, setul de date este îmbogățit cu noi variabile care ar putea avea puterea de a îmbunătăți performanța modelului de învățare automată în predicția riscului de neplată a creditului.

4. Selectarea Caracteristicilor Folosind Permutarea Țintei (Null Importances)[2]

Această procedură este descrisă într-un articol care detaliază selectarea caracteristicilor folosind permutarea țintei. Această metodă testează importanța reală a caracteristicilor în comparație cu distribuția importanțelor caracteristicilor atunci când sunt ajustate la zgomot (ținta amestecată).

Sunt implementati următorii pași:

Crearea distribuțiilor de importanță nulă: acestea sunt create ajustând modelul de mai multe ori pe o versiune a țintei amestecată. Acest lucru arată cum modelul poate găsi un sens al unei caracteristici indiferent de țintă.

Ajustarea modelului pe ținta originală și colectarea importanțelor caracteristicilor. Acest lucru ne oferă un punct de referință al cărui semnificativitate poate fi testată împotriva Distribuției Importanțelor Nule.

Pentru fiecare caracteristică, testarea importanței actuale:

- Calcularea probabilității importanței actuale în raport cu distribuția nulă. Voi folosi o estimare foarte simplă folosind aparițiile, în timp ce articolul propune ajustarea distribuțiilor cunoscute la datele colectate. De fapt, aici vom calcula 1 - probabilitatea astfel încât lucrurile să fie în ordinea corectă.
- Simplu, compararea importanței actuale cu media și maximul importanțelor nule. Acest lucru va oferi un fel de importanță a caracteristicii care permite vizualizarea caracteristicilor majore din setul de date. Într-adevăr, metoda anterioară ne poate oferi multe rezultate unice.

Selectarea caracteristicilor în acest mod ne oferă o abordare robustă pentru a determina care caracteristici sunt cu adevărat importante pentru modelul nostru și care pot fi rezultatul zgomotului sau al coincidențelor aleatoare, contribuind astfel la îmbunătățirea generală a modelului și evitând supracomplexitatea.

[2] https://academic.oup.com/bioinformatics/article/26/10/1340/193348

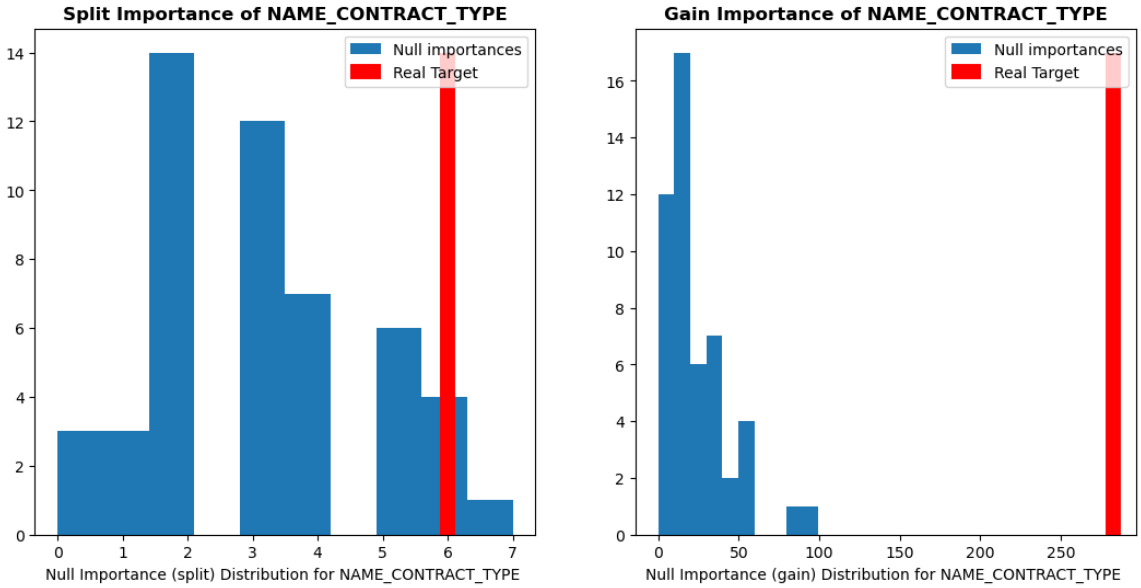

Fig. 22 Distribuția split și gain pentru NAME_CONTRACT_TYPE

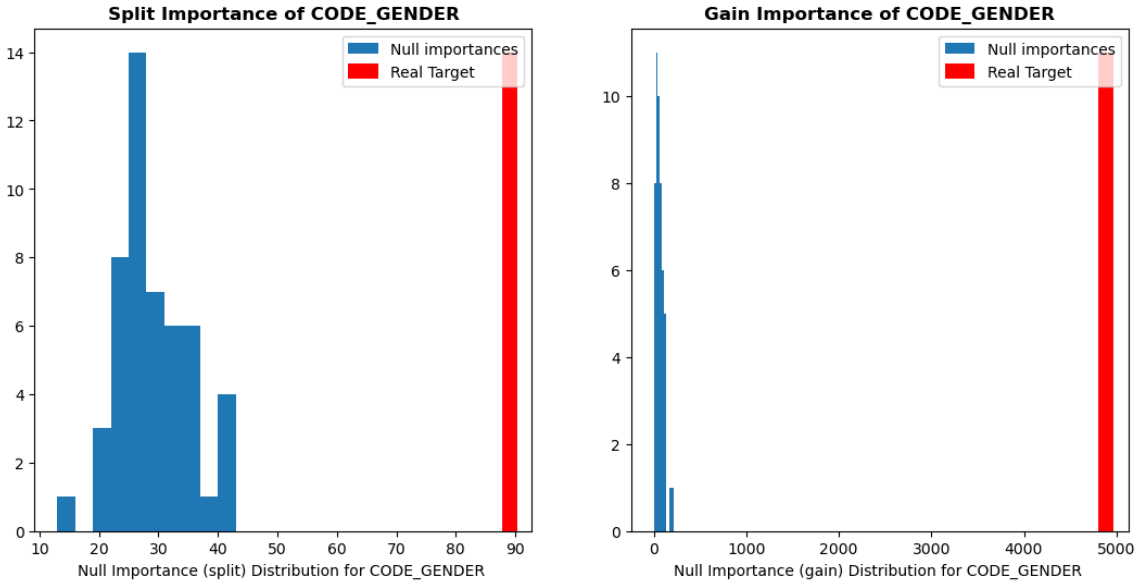

Fig. 23 Distribuția split și gain pentru CODE_GENDER

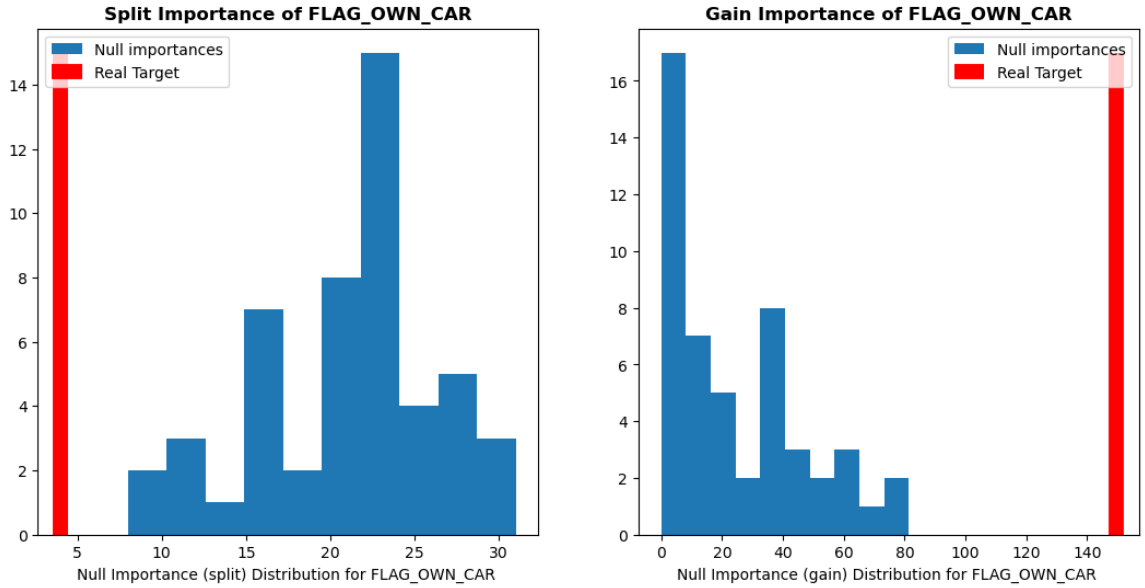

Fig. 24 Distribuția split și gain pentru FLAG_OWN_CAR

Graficele prezentate demonstrează eficacitatea metodei de selecție a caracteristicilor descrisă. În particular, trebuie subliniat faptul că:

- Orice caracteristică cu varianță adecvată poate fi utilizată și interpretată de modelele bazate pe arbori de decizie. Există întotdeauna posibilitatea de a găsi diviziuni care contribuie la îmbunătățirea acurateței.
- Importanțele caracteristicilor corelate tind să scadă odată ce una dintre ele este inclusă în model. Caracteristica utilizată inițial va avea o importanță majoră, iar cele corelate cu aceasta vor înregistra importanțe în scădere.

Metoda descrisă permite să:

- Eliminăm caracteristicile cu varianță înaltă, dar nerelevante pentru obiectivul analizei.
- Reducem impactul diminuării importanței caracteristicilor corelate, evidențiind astfel adevărata lor relevanță.

Graficele ilustrează două tipuri de importanțe: importanța diviziunii (Split Importance) și importanța câștigului (Gain Importance) pentru anumite caracteristici, prin compararea distribuției importanțelor nule (presupunând o permutare a obiectivului) cu importanța efectivă (obiectivul real). Bin-urile roșii din grafic reprezintă importanța caracteristicilor atunci când modelul este antrenat folosind obiectivul real, în timp ce cele albastre reflectă distribuția importanței bazată pe un obiectiv amestecat.

Din aceste grafice observăm că:

- Dacă importanța reală (linia roșie) depășește semnificativ distribuția importanțelor nule (bin-urile albastre), este probabil ca caracteristica să fie esențială pentru model și strâns legată de obiectiv.
- Dacă importanța reală este similară sau inferioară importanțelor nule, este posibil ca caracteristica să nu fie relevantă, fiind influențată de zgomotul din date.

Această metodologie de selecție a caracteristicilor ne ajută să ne focalizăm pe cele mai pertinente caracteristici pentru model, sporind astfel eficiența și performanța modelării.

Există mai multe metode prin care se pot evalua caracteristicile relevante:

- Se calculează numărul de probe în importanțele reale care se abat de la distribuția înregistrată a importanțelor nule.
- Se calculează rapoarte precum Importanța Actuală / Maximul Nul, Importanța Actuală / Media Nulă, Media Importanței Actuale / Maximul Nul.

Se va folosi logaritmul importanței caracteristicii actuale împărțit la percentila 75 a distribuției nule.

4.1 Cele mai bune caracteristici

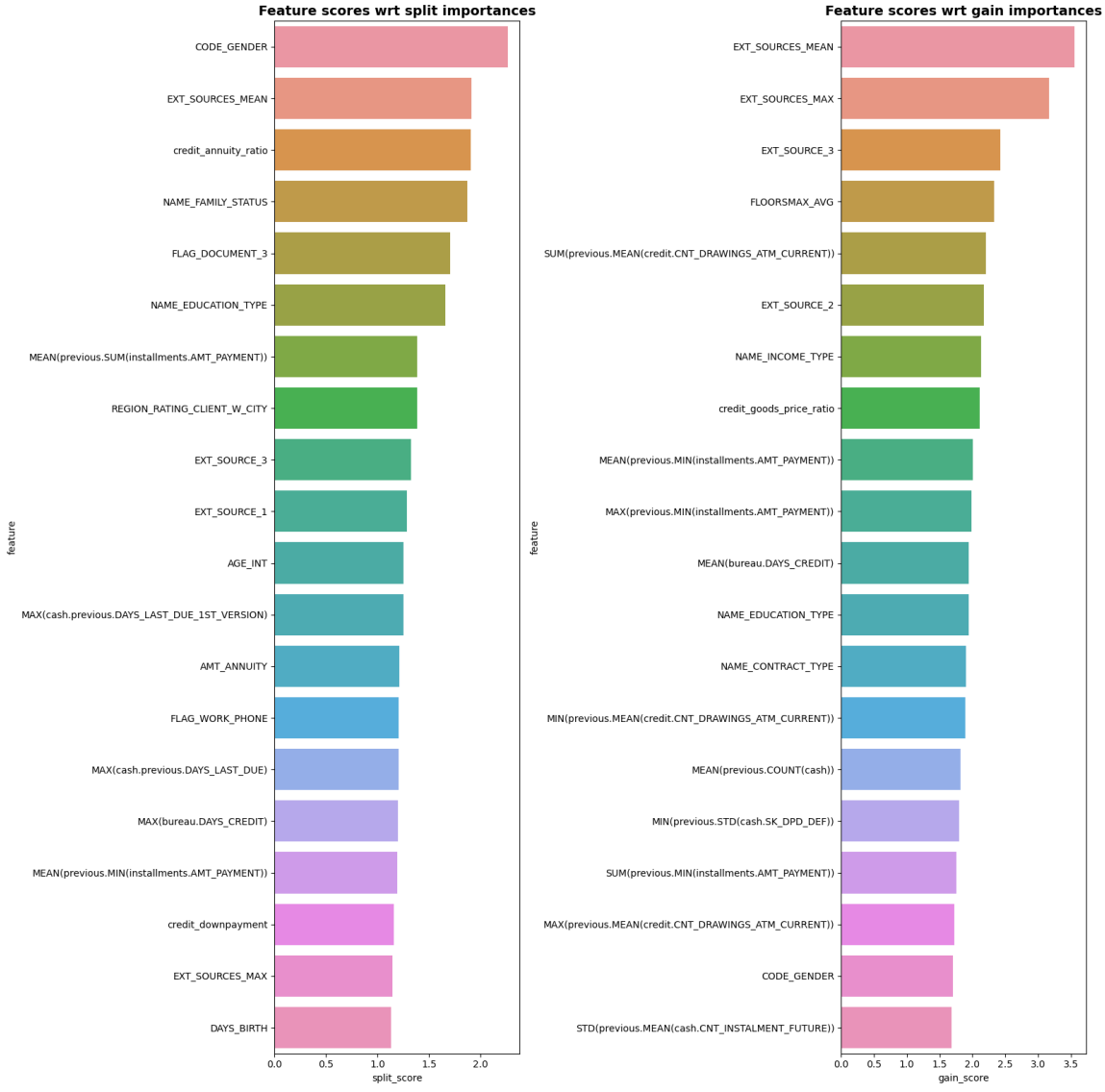

Fig. 25 Top 20 caracteristici

Graficul reprezintă o evaluare comparativă a scorurilor caracteristicilor în raport cu importanța diviziunilor și importanța câștigului, permițând o înțelegere profundă a relevanței fiecărei caracteristici în modelul predictiv.

Prin analiza scorurilor referitoare la importanța diviziunii, observăm cum diferite atribute contribuie la împărțirea datelor în cadrul modelului de învățare automată. Acest lucru este esențial pentru a înțelege modul în care deciziile sunt structurate în interiorul modelului și care caracteristici influențează cel mai mult rezultatele. Caracteristicile superioare din graficul respectiv sunt cele care ajută modelul să facă distincții clare între diferitele clase sau rezultate. De exemplu, atributul plasat cel mai sus sugerează o influență puternică asupra modului în care modelul segmentează datele.

Pe de altă parte, scorurile referitoare la importanța câștigului arată în ce măsură fiecare caracteristică îmbunătățește performanța modelului prin contribuția sa la precizia acestuia. Atributele cu cele mai înalte scoruri în această parte a graficului sunt cele care, atunci când sunt utilizate în model, conduc la cele mai mari creșteri ale acurateței. În mod particular, când o caracteristică prezintă un scor de câștig semnificativ, aceasta indică faptul că modelul devine mai bun la predicții atunci când ia în considerare acea caracteristică.

Analiza acestor două dimensiuni - importanța diviziunii și importanța câștigului - este vitală pentru optimizarea selecției caracteristicilor în modelele de învățare automată. Aceasta ne ajută să identificăm care atribute aduc o valoare reală modelului și să eliminăm zgomotul sau informațiile redundante. În final, interpretarea acestor grafice este crucială pentru dezvoltarea de modele predictive precise și robuste, permițându-ne să facem predicții bine informate și să extragem înțelegeri valoroase din seturile noastre de date.

4.2 Caracteristicile inutile

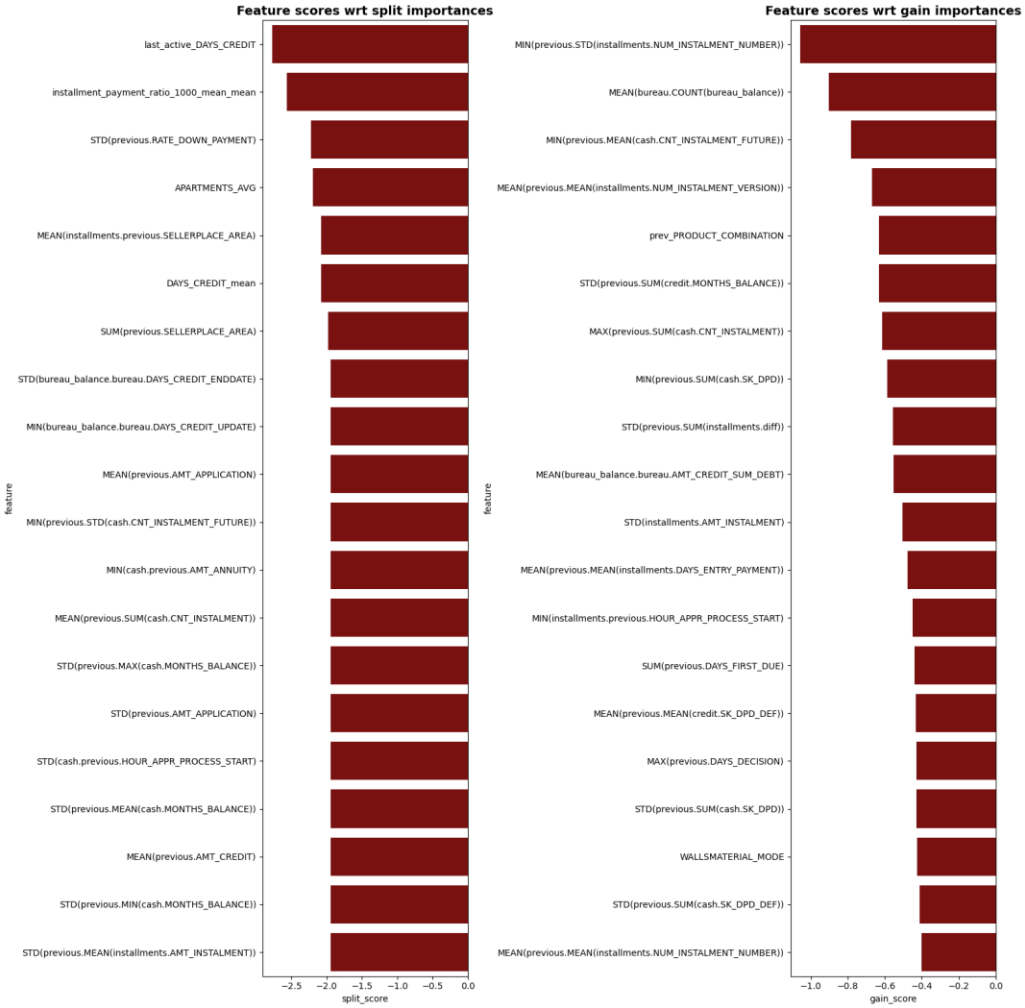

Fig. 26 Caracteristici inutile

Analiza grafică din Fig. 26 evidențiază scorurile caracteristicilor referitoare la importanțele diviziunii și ale câștigului, punând în lumină acele caracteristici care au o contribuție negativă sau nesemnificativă în modelul predictiv. Observăm că majoritatea caracteristicilor reprezentate aici derivă din procesul de inginerie a caracteristicilor folosind Featuretools, o metodă automatizată pentru generarea de noi caracteristici. Acest fapt accentuează o provocare comună în crearea de caracteristici noi: potențialul crescut de a introduce variabile care nu doar că nu adaugă valoare modelului, dar pot de fapt să-i scadă performanța.

Caracteristicile cu scorurile cele mai reduse, ilustrate în roșu închis, sugerează o asociere limitată cu variabila țintă sau chiar o influență dăunătoare asupra capacității modelului de a face predicții corecte. Este posibil ca aceste caracteristici să fie irelevante, să reflecte zgomotul din date, sau să fie prea complexe pentru a fi captate eficient de model în starea actuală. De exemplu, interacțiunile sau agregările complexe care nu sunt în strânsă corelație cu variabila țintă pot duce la overfitting sau la introducerea de confuzie în model.

Concluzia pe care o putem extrage din aceste observații este că, în timp ce Featuretools poate fi un instrument puternic pentru amplificarea spațiului caracteristicilor, este esențial să avem o strategie de selecție riguroasă și o evaluare critică a caracteristicilor generate automat. Aceasta asigură că doar cele mai informative și relevante caracteristici sunt păstrate, iar modelul rămâne robust și focalizat pe semnalele care aduc valoare predictivă.

4.3 Eliminarea caracteristicilor inutile

Selectarea pragului potrivit pentru scorurile caracteristicilor este un pas cheie în procesul de optimizare a unui model de învățare automată. Prin stabilirea unui prag adecvat, putem îmbunătăți semnificativ performanța modelului prin eliminarea caracteristicilor care nu aduc un beneficiu clar.

Analiza a arătat că, ajustând pragurile pentru scorurile de diviziune și de câștig, scorul AUC (Area Under the Curve) al modelului se modifică, evidențiind impactul direct pe care îl au aceste caracteristici asupra eficacității modelului. Scorul AUC este o măsură a capacității modelului de a clasifica corect cazurile pozitive și negative, iar modificările în scorul AUC reflectă îmbunătățiri sau scăderi în performanța predictivă a modelului.

S-a observat că, la un prag de 0 pentru ambele scoruri de diviziune și de câștig, modelul atinge un scor AUC optim. Acest rezultat sugerează că menținerea caracteristicilor cu scoruri pozitive contribuie la o mai bună distincție între clasificările pozitive și negative ale modelului. Pe măsură ce pragul este crescut sau scăzut, există o tendință de scădere a scorului AUC, ceea ce indică faptul că unele caracteristici cu scoruri mai mici pot conține informații valoroase sau că eliminarea excesivă a caracteristicilor poate reduce capacitatea modelului de a captura nuanțele din date.

Prin selectarea caracteristicilor utile, numărul total de caracteristici este redus, ceea ce duce la un model mai simplu și mai eficient, fără a sacrifica acuratețea. Eliminarea caracteristicilor inutile poate reduce, de asemenea, riscul de peste-ajustare (overfitting), creând un model mai generalizabil și mai robust.

În urma acestui proces, au fost eliminate 423 caracteristici. Selectarea atentă a caracteristicilor și stabilirea unui prag de scoruri sunt esențiale pentru eficientizarea modelului predictiv, asigurându-ne că ne concentrăm pe cele mai relevante semnale din date pentru a face predicții precise.

4.4 Eliminarea caracteristicilor puternic corelate

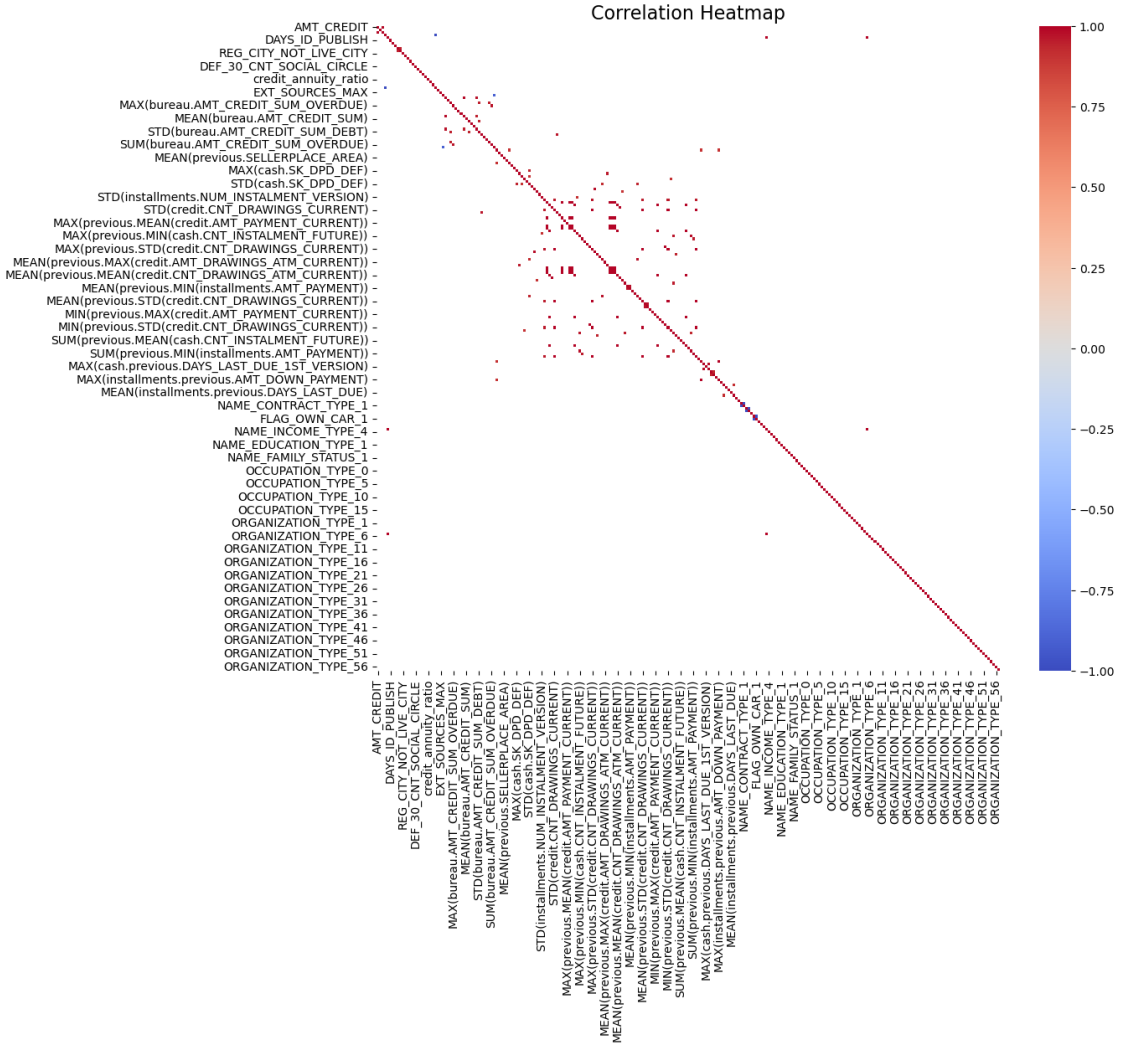

Fig. 27 Corelațiile mai mari de 0.95 între caracteristici

Harta de corelație prezentată ilustrează relațiile dintre variabilele modelului. Observând densitatea de puncte roșii, putem deduce că multe caracteristici afișează o corelație semnificativă. Acest lucru nu este neobișnuit în seturi de date complexe, unde caracteristicile derivate pot interacționa sau pot fi variantă de alte atribute în multiple moduri.

Esența acestei hărți este de a identifica și de a elimina caracteristici care au o corelație puternică între ele, cunoscută și sub denumirea de multicolinearitate. Aceasta poate duce la probleme în estimarea modelului, cum ar fi dificultăți în interpretarea importanței variabilelor individuale sau instabilitate în predicțiile modelului. În special, atunci când două caracteristici sunt foarte corelate, poate fi dificil pentru model să determine care dintre acestea oferă informații unice despre variabila țintă.

Pragul de corelație a fost setat să filtreze relațiile mai puternice decât 0.9, permițând o focalizare asupra caracteristicilor care oferă informații unice și relevante pentru predicții. Prin utilizarea acestui prag, putem reduce dimensiunea setului de date și potențialul de overfitting, crescând capacitatea generalizării modelului.

Eliminarea caracteristicilor cu corelație înaltă este, de asemenea, utilă pentru simplificarea modelului, ceea ce poate conduce la o îmbunătățire a timpilor de antrenare și o interpretare mai clară a rezultatelor. În final, acest proces ajută la asigurarea faptului că fiecare caracteristică inclusă în model aduce o contribuție distinctă și valoare predictivă, maximizând astfel potențialul modelului de a face predicții precise și robuste. În urma acestui proces, au fost eliminate 247 caracteristici.

4.5 Importanța SHAP

Bara de date SHAP (SHapley Additive exPlanations) prezentată oferă o interpretare vizuală a importanței caracteristicilor într-un model de clasificare. Acest grafic sintetizează modul în care fiecare caracteristică contribuie, în medie, la predicția modelului.

Caracteristica de top, cu cea mai mare lungime a barei, denotă cea mai mare influență medie asupra modelului. Aceasta sugerează că valoarea medie a efectelor SHAP pentru acea caracteristică este semnificativă și, prin urmare, are un rol important în modelul predictiv. Pe măsură ce ne mișcăm în jos pe grafic, lungimea barelor se scurtează, indicând o influență descrescătoare a caracteristicilor respective asupra predicțiilor modelului.

Valoarea medie a efectelor SHAP este un indicator al impactului caracteristicii asupra mișcării predicției de la valoarea de bază a modelului către predicția efectivă. Cu cât o caracteristică are o valoare SHAP mai mare, cu atât contribuția ei la predicțiile modelului este considerată mai esențială.

În contextul acestui model specific, variabilele legate de surse externe, raportul creditului la anuitate și suma anuității sunt printre cele mai influente. Alți predictori importanți includ genul codificat, vârsta mașinii proprietarului și vârsta solicitantului, între altele.

Rezultatele acestui grafic sunt cruciale pentru înțelegerea comportamentului modelului și pentru stabilirea priorităților în ajustările ulterioare ale modelului, fie prin rafinarea caracteristicilor existente, fie prin investigarea mai profundă a celor cu impact semnificativ.

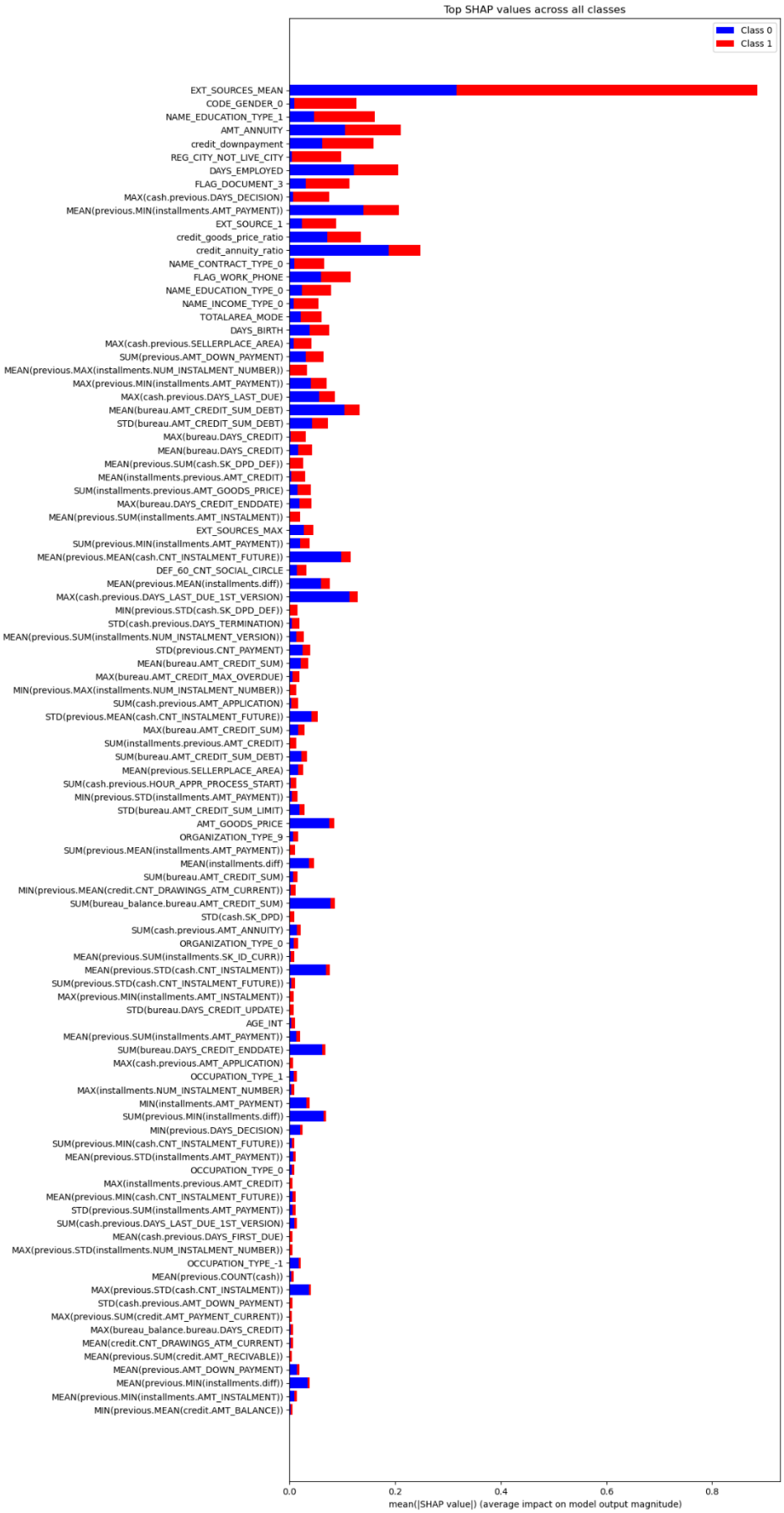

Fig. 28 Importanța SHAP în funcție de clasă

5. Modelare

5.1 Pregătirea datelor

În procesul de dezvoltare a unui model de clasificare, prima etapă a implicat pregătirea și curățarea setului de date. A fost creat un set de caracteristici dummy, un pas comun pentru a converti caracteristicile categorice într-un format numeric care poate fi interpretat de modelele de învățare automată. După această transformare, numărul total de caracteristici s-a determinat a fi semnificativ, indicând o largă varietate de posibili predictori.

În continuare, s-a efectuat o selecție doar a caracteristicilor numerice, deoarece sunt cele compatibile cu modelul de învățare automată utilizat și pot fi tratate direct pentru predicții. După extragerea etichetelor, a urmat procesul de separare a caracteristicilor (predictorilor) de variabila țintă, care este necesar pentru antrenarea oricărui model de învățare supervizată.

Ulterior, setul de date a fost împărțit în submulțimi de antrenare și testare. Acest pas este crucial pentru a evalua modelul pe date nevăzute anterior și pentru a verifica capacitatea sa de generalizare.

O etapă importantă a fost curățarea numelor caracteristicilor pentru a asigura compatibilitatea acestora cu modelele de învățare automată și eliminarea potențialelor probleme care ar putea apărea din cauza formatării necorespunzătoare a numelui caracteristicilor.

Valori infinite, care pot apărea ca rezultat al erorilor de calcul sau a unor anomalii în date, au fost înlocuite cu valori NaN pentru a preveni erorile la antrenarea modelului. Tratarea valorilor lipsă sau a celor extreme este esențială pentru menținerea performanței și stabilității modelului.

În final, dimensiunile seturilor de date de antrenare și testare au fost stabilite, pregătind terenul pentru etapele ulterioare de modelare. Numărul de caracteristici rămase sugerează că modelul final va fi bazat pe un set robust și complex de date, cu potențialul de a captura diferite aspecte și nuanțe ale problemei la îndemână. Această abordare sistematică și minuțioasă în pregătirea datelor asigură fundația pentru dezvoltarea unui model de predicție precis și de încredere.

```
Final Train shape:  (230633, 206)
Final Test shape:  (76878, 206)
```

Fig. 29 Dimensiunea setului de test si antrenare

5.2 Stabilirea unui benchmark

În procesul de dezvoltare a sistemelor predictive, stabilirea unui benchmark robust este esențială pentru a evalua și compara performanța diferitelor modele de învățare automată. Benchmarking-ul înseamnă antrenarea și testarea unei serii de modele folosind același set de date, cu scopul de a identifica care abordare oferă cea mai bună performanță pentru problema specifică de față.

Un benchmark eficient permite nu doar identificarea modelului cu cea mai înaltă acuratețe sau AUC, dar oferă și oportunitatea de a descoperi modele care prezintă un echilibru între sensibilitate (recall) și precizie.

Acest lucru este important deoarece un model cu o acuratețe generală ridicată poate încă să aibă deficiențe în clasificarea corectă a cazurilor individuale, în special în contextul seturilor de date neechilibrate.

Prin aplicarea unei game largi de modele — de la arbori de decizie simpli și modele de random forest până la algoritmi complexi de gradient boosting precum XGBoost, LightGBM, și CatBoost — putem obține o înțelegere cuprinzătoare a performanțelor posibile și a caracteristicilor diferitelor tehnici de clasificare. Acest proces comparativ este crucial pentru a asigura că modelul ales este cel mai potrivit pentru implementare, în funcție de cerințele specifice, cum ar fi interpretabilitatea modelului, timpul de antrenare și predicție, și capacitatea de generalizare.

	precision	recall	f1-score	support	model
0	0.923981	0.993521	0.957490	70687.000000	XGBClassifier
1	0.474168	0.066710	0.116964	6191.000000	XGBClassifier
accuracy	0.918884	0.918884	0.918884	0.918884	XGBClassifier
macro avg	0.699074	0.530115	0.537227	76878.000000	XGBClassifier
weighted avg	0.887757	0.918884	0.889802	76878.000000	XGBClassifier
Accuracy	0.918884	NaN	NaN	NaN	XGBClassifier
AUC	0.775051	NaN	NaN	NaN	XGBClassifier

	precision	recall	f1-score	support	model
0	0.919949	0.999519	0.958085	70687.000000	RandomForestClassifier
1	0.558442	0.006946	0.013720	6191.000000	RandomForestClassifier
accuracy	0.919587	0.919587	0.919587	0.919587	RandomForestClassifier
macro avg	0.739195	0.503232	0.485903	76878.000000	RandomForestClassifier
weighted avg	0.890837	0.919587	0.882035	76878.000000	RandomForestClassifier
Accuracy	0.919587	NaN	NaN	NaN	RandomForestClassifier
AUC	0.742690	NaN	NaN	NaN	RandomForestClassifier

	precision	recall	f1-score	support	model
0	0.922155	0.997298	0.958256	70687.000000	LGBMClassifier
1	0.556845	0.038766	0.072486	6191.000000	LGBMClassifier
accuracy	0.920107	0.920107	0.920107	0.920107	LGBMClassifier
macro avg	0.739500	0.518032	0.515371	76878.000000	LGBMClassifier
weighted avg	0.892737	0.920107	0.886925	76878.000000	LGBMClassifier
Accuracy	0.920107	NaN	NaN	NaN	LGBMClassifier
AUC	0.780845	NaN	NaN	NaN	LGBMClassifier

	precision	recall	f1-score	support	model
0	0.921945	0.997892	0.958416	70687.00000	CatBoostClassifier
1	0.595109	0.035374	0.066778	6191.00000	CatBoostClassifier
accuracy	0.920380	0.920380	0.920380	0.92038	CatBoostClassifier
macro avg	0.758527	0.516633	0.512597	76878.00000	CatBoostClassifier
weighted avg	0.895625	0.920380	0.886613	76878.00000	CatBoostClassifier
Accuracy	0.920380	NaN	NaN	NaN	CatBoostClassifier
AUC	0.782660	NaN	NaN	NaN	CatBoostClassifier

Fig. 30 XGB, RF, LGBM, CAT

	precision	recall	f1-score	support	model
0	0.919755	0.999816	0.958116	70687.000000	ExtraTreesClassifier
1	0.657895	0.004038	0.008027	6191.000000	ExtraTreesClassifier
accuracy	0.919626	0.919626	0.919626	0.919626	ExtraTreesClassifier
macro avg	0.788825	0.501927	0.483072	76878.000000	ExtraTreesClassifier
weighted avg	0.898668	0.919626	0.881605	76878.000000	ExtraTreesClassifier
Accuracy	0.919626	NaN	NaN	NaN	ExtraTreesClassifier
AUC	0.747302	NaN	NaN	NaN	ExtraTreesClassifier

	precision	recall	f1-score	support	model
0	0.926924	0.914624	0.920733	70687.000000	DecisionTreeClassifier
1	0.153458	0.176708	0.164264	6191.000000	DecisionTreeClassifier
accuracy	0.855199	0.855199	0.855199	0.855199	DecisionTreeClassifier
macro avg	0.540191	0.545666	0.542498	76878.000000	DecisionTreeClassifier
weighted avg	0.864636	0.855199	0.859814	76878.000000	DecisionTreeClassifier
Accuracy	0.855199	NaN	NaN	NaN	DecisionTreeClassifier
AUC	0.545666	NaN	NaN	NaN	DecisionTreeClassifier

	precision	recall	f1-score	support	model
0	0.921549	0.997411	0.957980	70687.000000	AdaBoostClassifier
1	0.508065	0.030528	0.057596	6191.000000	AdaBoostClassifier
accuracy	0.919548	0.919548	0.919548	0.919548	AdaBoostClassifier
macro avg	0.714807	0.513970	0.507788	76878.000000	AdaBoostClassifier
weighted avg	0.888251	0.919548	0.885472	76878.000000	AdaBoostClassifier
Accuracy	0.919548	NaN	NaN	NaN	AdaBoostClassifier
AUC	0.762426	NaN	NaN	NaN	AdaBoostClassifier

	precision	recall	f1-score	support	model
0	0.927092	0.918344	0.922698	70687.000000	DecisionTreeClassifier
1	0.158355	0.175416	0.166450	6191.000000	DecisionTreeClassifier
accuracy	0.858516	0.858516	0.858516	0.858516	DecisionTreeClassifier
macro avg	0.542724	0.546880	0.544574	76878.000000	DecisionTreeClassifier
weighted avg	0.865186	0.858516	0.861797	76878.000000	DecisionTreeClassifier
Accuracy	0.858516	NaN	NaN	NaN	DecisionTreeClassifier
AUC	0.546880	NaN	NaN	NaN	DecisionTreeClassifier

Fig. 31 ET, CART, ADA, ID3

Rezultatele din imagini indică performanța diferitelor modele de clasificare evaluate prin metrici standard precum precizia, recall-ul și scorul F1, împreună cu AUC (Area Under the Curve), care măsoară capacitatea unui clasificator de a distinge între clase. Metricile sunt desfășurate atât pentru clasificarea pozitivă (1) cât și negativă (0), precum și media lor globală (macro average) și media ponderată (weighted average), având în vedere suportul fiecărei clase, care reprezintă numărul de cazuri reale din datele de test.

Performanțele modelelor variază, cu unele modele având o acuratețe (precizia predicției generale) ridicată, în timp ce altele excellează în termenii valorii AUC, sugerând o capacitate mai bună de separare a claselor pozitive și negative. Modelele, cum ar fi XGBClassifier, CATBoost și LGBM, prezintă scoruri AUC relativ

înalte, ceea ce sugerează că acestea sunt bune la clasificarea probabilităților și la furnizarea unei predicții echilibrate.

Cu toate acestea, un aspect notabil este că multe modele au o capacitate limitată de a clasifica corect cazurile pozitive (clasificarea 1), după cum indică valorile scăzute pentru recall și scorul F1 pentru această clasă. Acest lucru ar putea sugera că setul de date este dezechilibrat sau că modelele sunt mai înclinate să prezică corect cazurile mai frecvente.

Prin examinarea acestor rezultate, putem identifica care modele sunt cele mai adecvate pentru problema la îndemână și putem efectua ajustări suplimentare pentru a îmbunătăți performanța lor, în special în clasificarea cazurilor mai puțin reprezentate. Fiecare model are caracteristici și puncte forte diferite, așa că selecția modelului potrivit va depinde de contextul specific și de obiectivele analizei. Această evaluare detaliată ne ajută să înțelegem care modele sunt mai potrivite pentru implementare într-un scenariu de producție și care necesită îmbunătățiri sau ajustări.

5.3 Evaluarea și Compararea Strategiilor de Îmbunătățire a Modelului cu LightGBM

LightGBM este o bibliotecă avansată de învățare automată care oferă o implementare eficientă a algoritmilor de gradient boosting. Un aspect cheie al acestei biblioteci este capacitatea de a experimenta cu diferite tipuri de boosting, fiecare având propriile strategii pentru a îmbunătăți performanța și a preveni peste-ajustarea. În cadrul acestei analize, vom evalua și compara trei variațiuni ale algoritmului LightGBM: Gradient Boosting Decision Tree (GBDT), Dropouts meet Multiple Additive Regression Trees (DART) și Gradient-based One-Side Sampling (GOSS).

GBDT este abordarea standard și cel mai des utilizată strategie de boosting în LightGBM, echilibrând bine viteza și acuratețea. DART adaugă un mecanism de dropout, permițând unele arbori să fie "omiseți" în timpul predicției, ceea ce poate ajuta în prevenirea peste-ajustării. GOSS se concentrează pe eșantionarea gradientelor pentru a oferi mai multă atenție observațiilor cu erori mari, sporind eficacitatea și precizia actualizărilor modelului.

Prin setarea parametrilor specifici fiecărei strategii, putem observa modul în care fiecare se comportă pe un set de date standard. Fiecare model a fost evaluat folosind AUC, precizie, recall și scorul F1 pentru ambele clase, oferind o imagine de ansamblu a performanței acestora. Importanța evaluării acestor metrici este dublă: pe de o parte, ne ajută să înțelegem capacitatea modelului de a clasifica corect observațiile din clasele pozitive și negative, iar pe de altă parte, AUC ne oferă o perspectivă asupra capacității modelului de a distinge între clasificările pozitive și negative pe tot intervalul de probabilități.

	precision	recall	f1-score	support	model
0	0.923092	0.996548	0.958415	70687.000000	LGBMClassifier
1	0.568905	0.052011	0.095309	6191.000000	LGBMClassifier
accuracy	0.920484	0.920484	0.920484	0.920484	LGBMClassifier
macro avg	0.745998	0.524280	0.526862	76878.000000	LGBMClassifier
weighted avg	0.894569	0.920484	0.888909	76878.000000	LGBMClassifier
Accuracy	0.920484	NaN	NaN	NaN	LGBMClassifier
AUC	0.790354	NaN	NaN	NaN	LGBMClassifier

	precision	recall	f1-score	support	model
0	0.921641	0.998359	0.958467	70687.000000	LGBMClassifier
1	0.622150	0.030851	0.058787	6191.000000	LGBMClassifier
accuracy	0.920445	0.920445	0.920445	0.920445	LGBMClassifier
macro avg	0.771896	0.514605	0.508627	76878.000000	LGBMClassifier
weighted avg	0.897523	0.920445	0.886016	76878.000000	LGBMClassifier
Accuracy	0.920445	NaN	NaN	NaN	LGBMClassifier
AUC	0.789556	NaN	NaN	NaN	LGBMClassifier

	precision	recall	f1-score	support	model
0	0.923111	0.996477	0.958392	70687.000000	LGBMClassifier
1	0.565445	0.052334	0.095801	6191.000000	LGBMClassifier
accuracy	0.920445	0.920445	0.920445	0.920445	LGBMClassifier
macro avg	0.744278	0.524406	0.527097	76878.000000	LGBMClassifier
weighted avg	0.894308	0.920445	0.888928	76878.000000	LGBMClassifier
Accuracy	0.920445	NaN	NaN	NaN	LGBMClassifier
AUC	0.789816	NaN	NaN	NaN	LGBMClassifier

Fig. 32 GBDT, DART, GOSS

Rezultatele evaluării indică o performanță comparabilă între cele trei tipuri de boosting, cu mici variații în scorurile AUC, ceea ce sugerează o consistență în calitatea modelării între strategii. Cu toate acestea, observăm că, în ciuda acurateței generale ridicate, modelele tind să aibă un recall și scoruri F1 scăzute pentru clasa minoritară (1), indicând o provocare în clasificarea eficientă a cazurilor mai rare sau dificile. Această tendință subliniază importanța echilibrării claselor sau ajustarea ponderii acestora în timpul antrenării pentru a îmbunătăți detectarea și clasificarea cazurilor pozitive.

În final, fiecare strategie de boosting oferă beneficii unice și poate fi mai potrivită pentru diferite tipuri de seturi de date sau probleme. Prin această analiză comparativă, putem ghida alegerea strategiei de boosting în funcție de nevoile specifice și obiectivele modelării, având ca scop îmbunătățirea generală a performanței și robusteții modelului predictiv.

În domeniul învățării automate, fine-tuning-ul hiperparametrilor este un pas critic în procesul de dezvoltare a unui model. Acesta constă în ajustarea precisă a parametrilor modelului pentru a maximiza performanța acestuia. Algoritmii de îmbunătățire a gradientului precum XGBoost, LightGBM și CatBoost au demonstrat performanțe remarcabile în numeroase probleme de clasificare și regresie, dar succesul lor depinde în mare măsură de alegerea corectă a setărilor hiperparametrilor.

Importanța Optimizării Hiperparametrilor

Optimizarea hiperparametrilor nu este doar o căutare arbitrară a unei combinații care funcționează; este un proces sistematic și riguros care explorează spațiul de parametri pentru a identifica cea mai bună configurație care conduce la cele mai precise predicții. Este deosebit de importantă pentru algoritmii de îmbunătățire a gradientului, unde numărul de hiperparametri este considerabil, iar sensibilitatea lor la modificări poate afecta în mod semnificativ atât acuratețea, cât și viteza de antrenare a modelului.

Procesul de Optimizare pentru XGBoost

Pentru XGBoost, un algoritm robust și flexibil, am aplicat o tehnică de optimizare Bayesiană, un pas avansat în explorarea spațiului de hiperparametri. Prin utilizarea unei funcții de pierdere (în cazul nostru, scorul AUC), și un algoritm care învață din iterații anterioare, am reușit să ajungem la un set de parametri care maximizează performanța modelului.

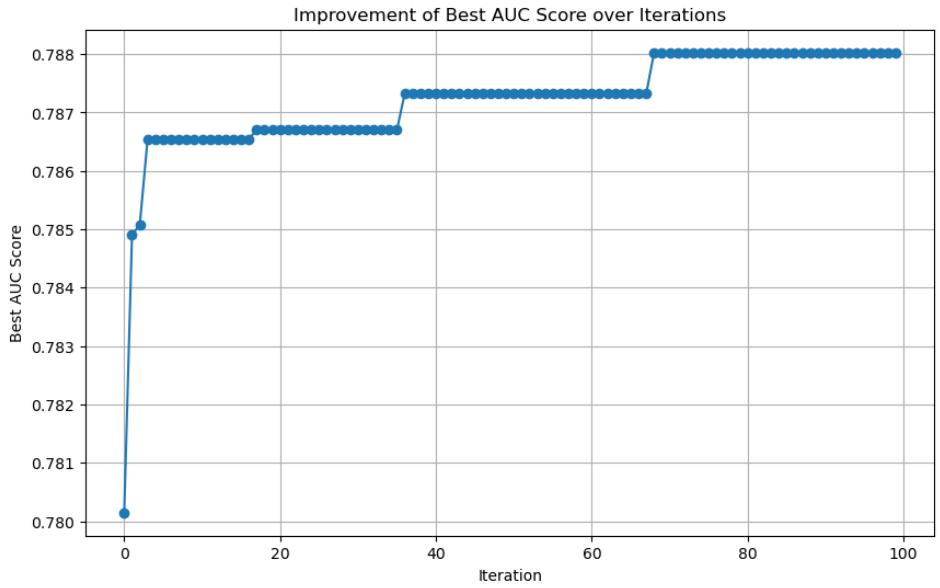

Fig. 33 Scorul AUC în cele 100 iterații pentru XGB

Graficul de îmbunătățire a scorului AUC peste iterații demonstrează progresul optimizării. Putem observa o ascensiune rapidă în calitatea modelului la început, urmată de îmbunătățiri incrementale, ceea ce sugerează că procesul de căutare converge spre un set optim de hiperparametri.

Importanța caracteristicilor, ilustrată în cel de-al doilea grafic, ne oferă o perspectivă asupra atributelor care contribuie cel mai mult la puterea predictivă a modelului. Prin înțelegerea acestor contribuții, putem lua decizii informate despre caracteristicile pe care să ne concentrăm în fazele ulterioare de modelare și despre cum putem structura prelucrarea datelor pentru a îmbunătăți performanța modelului.

Parametrii optimizați ai modelului XGBoost evidențiază o combinație bine echilibrată între complexitatea modelului și capacitatea de generalizare. O adâncime maximă (max_depth) de 4 împreună cu un număr substanțial de estimatori (n_estimators) de 813 indică o modelare prudentă, evitându-se supraajustarea. De asemenea, ratele de sub eșantionare (subsample și colsample_bytree) aproape de 1 sugerează că modelul beneficiază de aproape întregul set de date în timpul antrenării, în timp ce valorile reg_alpha și reg_lambda confirmă o regularizare blândă.

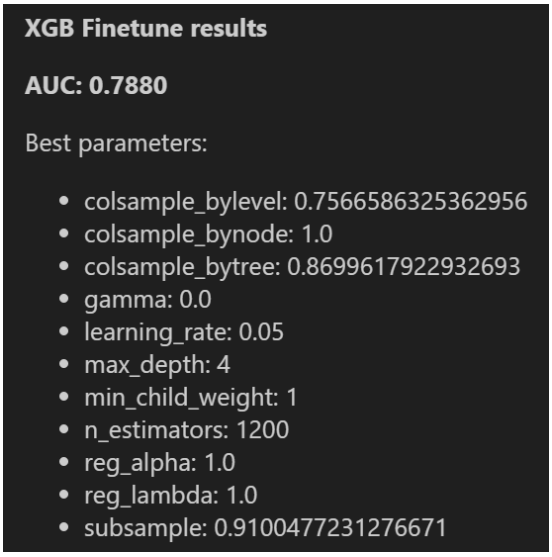

Fig. 34 Hiperparametrii XGBoost

Rezultatele optimizării, cu un scor AUC complet de 0.788870 și un scor AUC de test de 0.788893, demonstrează eficacitatea hiperparametrilor selectați. Aceste valori sunt competitive, arătând că modelul XGBoost este capabil să clasifice și să distingă eficient între evenimentele pozitive și negative din setul de date.

Procesul de optimizare Bayesiană aplicat pentru XGBoost ilustrează puterea de a folosi cunoștințe aprofundate despre algoritm pentru a îmbunătăți și a finisa modelele de învățare automată. Strategia de ajustare fină a hiperparametrilor ne ajută să navigăm eficient și metodic prin spațiul de căutare complex, conducând la modele predictive de înaltă calitate.

Importanța caracteristicilor după cross-validare

- Full AUC score 0.788870
- Test AUC score: 0.788893

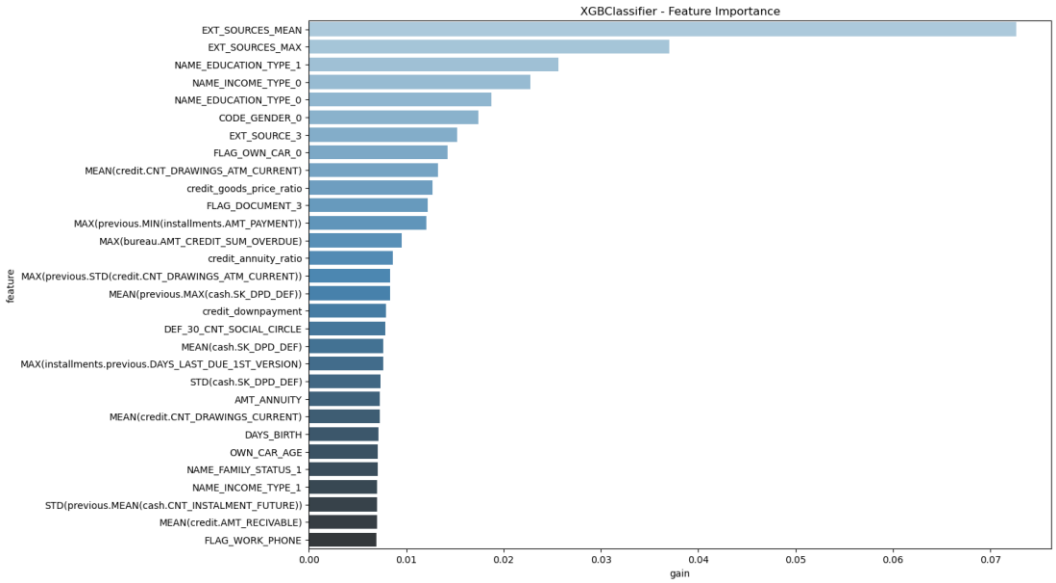

Fig. 35 Importanța caracteristicilor 10-fold XGBoost

Graficul ilustrează importanța caracteristicilor după un proces de optimizare pentru modelul XGBoost. Fiecare bară reprezintă o caracteristică individuală din setul de date, iar lungimea acesteia indică măsura în care fiecare atribut contribuie la predicțiile modelului, măsurată prin "gain".

În partea superioară a graficului, caracteristicile cu cea mai mare importanță – cum ar fi 'EXT_SOURCES_MEAN' și 'EXT_SOURCES_MAX' – se remarcă prin impactul lor considerabil asupra performanței modelului. Aceasta sugerează că sursele externe de date sunt predictori puternici și că modelele pot beneficia de pe urma includerii și analizei lor atentă.

Pe măsură ce privim spre partea inferioară a graficului, vedem caracteristici cu o importanță relativ mai mică, indicând că influența lor asupra modelului este mai restrânsă. Este vital să reținem că, deși unele caracteristici par să aibă o importanță mai mică, ele pot contribui încă la robustețea modelului, prin adăugarea diversității în date și prin îmbunătățirea capacității de generalizare.

Această vizualizare ne oferă o bază de date solidă pentru luarea deciziilor cu privire la caracteristicile care ar trebui să fie punctul focal al prelucrării ulterioare a datelor și ale analizei modelului, precum și pentru a informa potențialele direcții de inginerie a caracteristicilor pentru a îmbunătăți și mai mult performanța.

Procesul de Optimizare pentru LGBM

Procesul de optimizare pentru LightGBM a fost centrat pe ajustarea hiperparametrilor pentru a mări scorul AUC, similar cu abordarea pentru XGBoost, însă cu o atenție specifică distribuită asupra caracteristicilor unice ale LightGBM. Printre aceste caracteristici se numără 'num_leaves', care controlează numărul de frunze în fiecare arbore și 'min_split_gain', ce oferă o metodă de control asupra supracomplexității.

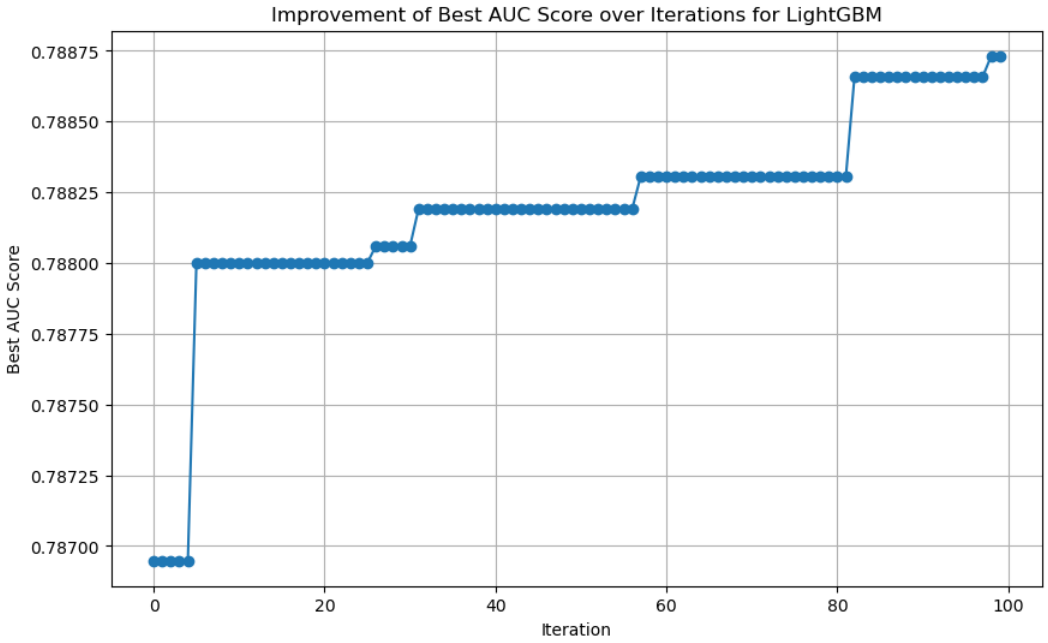

Fig. 36 Scorul AUC în cele 100 iterații LightGBM

LGBM

Best score: 0.7887

- colsample_bytree: 0.4
- learning_rate: 0.004955862989943362
- max_depth: 7
- min_split_gain: 0.04
- n_estimators: 5000
- num_leaves: 45
- reg_alpha: 1.0
- reg_lambda: 1.0

Fig. 37 Cei mai buni parametri LGBM

Importanța caracteristicilor după cross-validare

Scorul AUC optimizat de 0.789934 pe setul complet și de 0.791204 pe setul de test evidențiază o ușoară îmbunătățire comparativ cu XGBoost, semn că hiperparametrii au fost bine calibrați pentru specificitățile modelului LightGBM. Detaliile cheie ale optimizării, precum 'colsample_bytree' redus și 'learning_rate' delicat ajustat, au contribuit la această precizie îmbunătățită.

- Full AUC score 0.789934

- Test AUC score: 0.791204

Importanțele caracteristicilor în LightGBM ilustrează, de asemenea, diferitele priorități date predictorilor în modelare. Deși există unele suprapuneri cu XGBoost, cum ar fi importanța sursei externe, LightGBM acordă o atenție mai mare unor alte variabile, precum 'DAYS_BIRTH' și 'credit_annuity_ratio', care se dovedesc a fi influente în această configurație optimizată. Aceste diferențe subliniază adaptabilitatea LightGBM la diverse seturi de date și la diferite structuri de corelație dintre caracteristici.

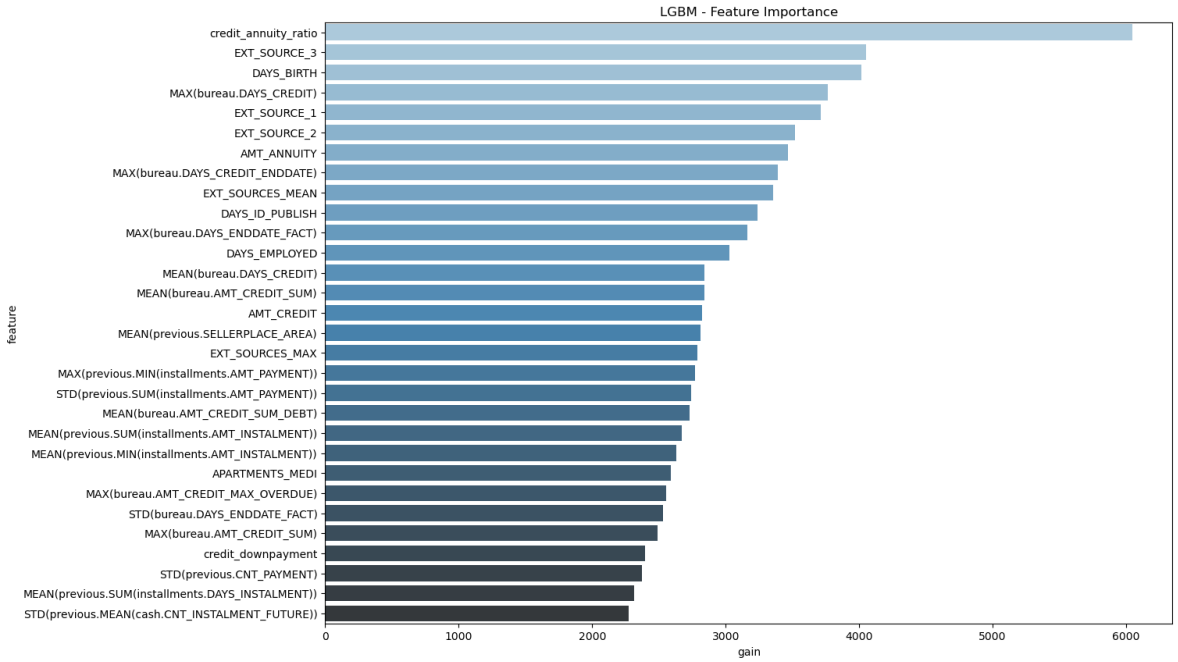

Fig. 38 Importanța caracteristicilor 10-fold LightGBM

În concluzie, procesul de optimizare detaliat pentru LightGBM și-a atins scopul de a descoperi o formulare eficientă a hiperparametrilor care maximizează AUC-ul, demonstrând capacitatea algoritmului de a gestiona și de a exploata cu succes datele pentru a oferi predicții de înaltă acuratețe.

Procesul de Optimizare pentru CATBoost

La fel ca în cazul XGBoost și LightGBM, optimizarea pentru CatBoost a implicat o strategie meticuloasă pentru a defini cel mai bun set de hiperparametri care maximizează performanța modelului, cu scopul de a îmbunătăți scorul AUC. Selecția hiperparametrilor a fost ghidată prin optimizare Bayesiană, ajungând la un set optim de valori care susțin un model predictiv robust.

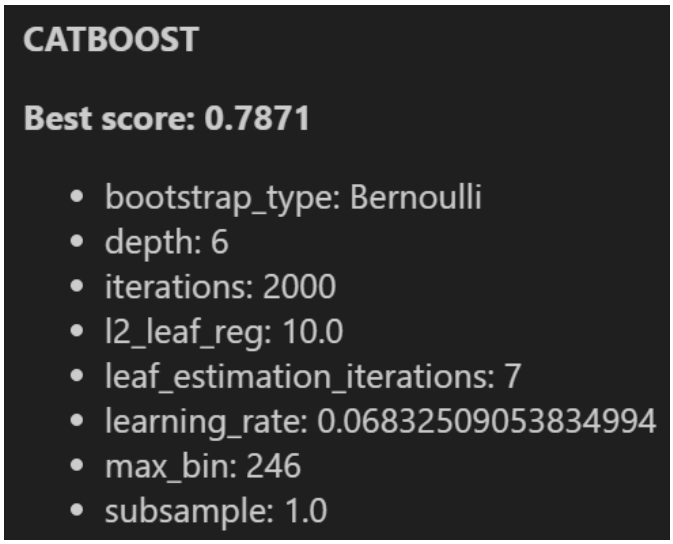

Fig. 39 Hiperparametrii CATBoost

- Full AUC score 0.788144
- Test AUC score: 0.790193

Graficul importanței caracteristicilor CATBoost evidențiază variabilele care au cea mai mare influență asupra puterii predictive a modelului. Observăm că 'EXT_SOURCES_MEAN', care s-a remarcat și în modelele XGBoost și LightGBM, își menține poziția dominantă, ceea ce confirmă importanța consistentă a acestei caracteristici peste diferite algoritmi de învățare automată.

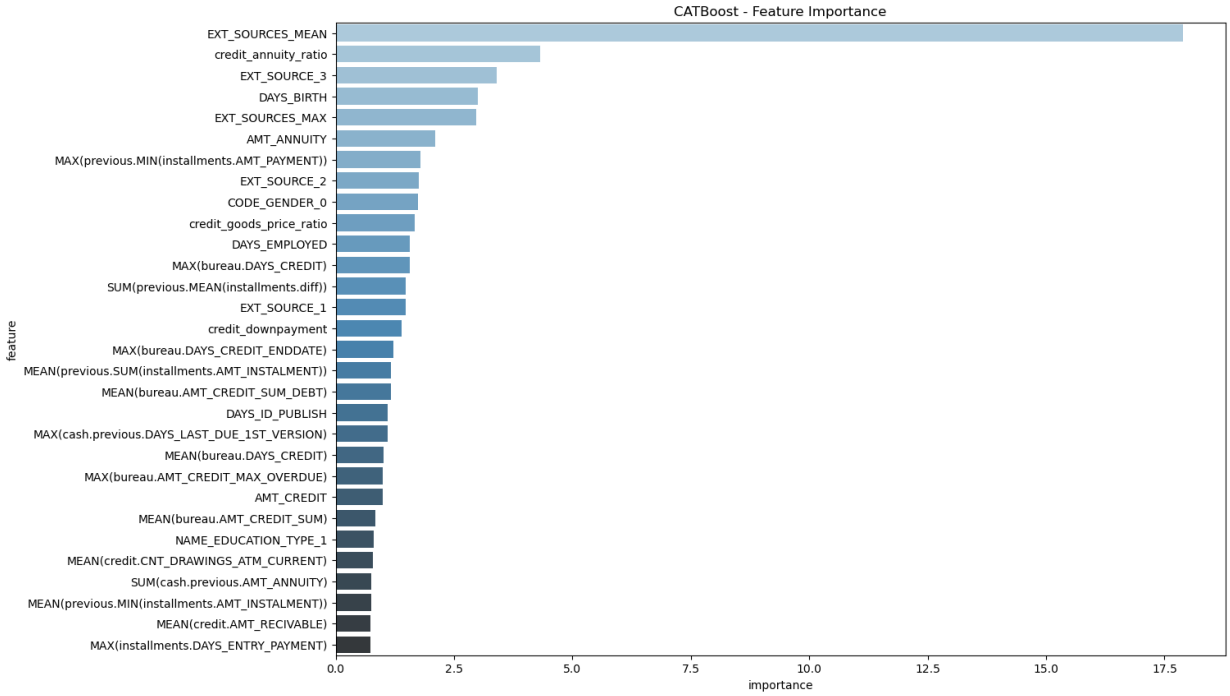

Fig. 40 Importanța caracteristicilor 10-fold CATBoost

Curba de optimizare a hiperparametrilor arată o îmbunătățire continuă și stabilă a scorului AUC până la o valoare maximă, indicând eficiența procesului de căutare a hiperparametrilor în identificarea unei soluții optime.

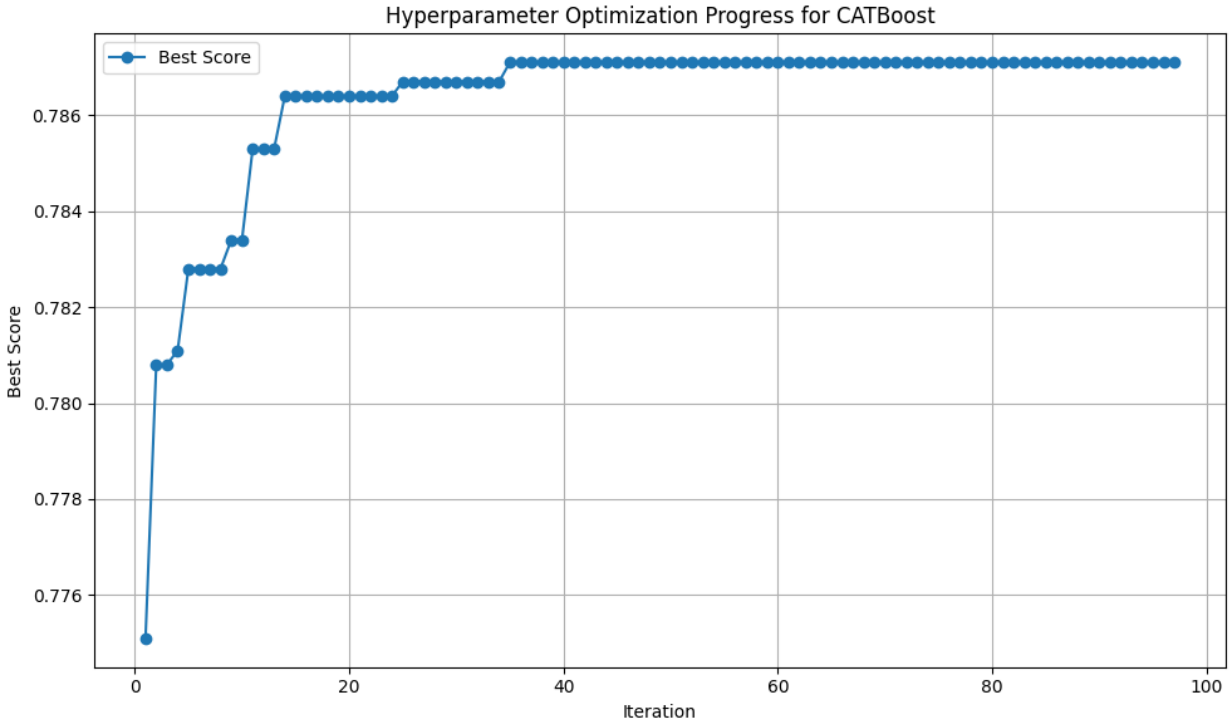

Fig. 41 Scorul AUC în cele 100 iterații CATBoost

Setul de hiperparametri optimizați pentru CatBoost, ilustrat mai sus, include o combinație strategică de adâncimea arborelui (depth), ratele de regularizare (l2_leaf_reg), și o rata de învățare (learning_rate) bine ajustată, precum și utilizarea întregii probe de antrenament (subsample). Această configurație subliniază un echilibru între capacitatea de învățare și evitarea peste-ajustării, esențial pentru performanța modelului în scenarii nevăzute.

Rezultatele finale ale scorului AUC de 0.788144 pentru setul de antrenament și de 0.790193 pentru setul de test, împreună cu optimizarea hiperparametrilor detaliată, confirmă capacitatea CatBoost de a gestiona complexitatea datelor și de a furniza predicții precise și fiabile.

În concluzie, procesul de optimizare pentru CatBoost a evidențiat importanța unei căutări riguroase și strategice a hiperparametrilor, culminând cu un model bine calibrat care se prevede a fi valoros în aplicarea practică a problemelor de clasificare în învățarea automată.

5.5 Matricele de confuzie

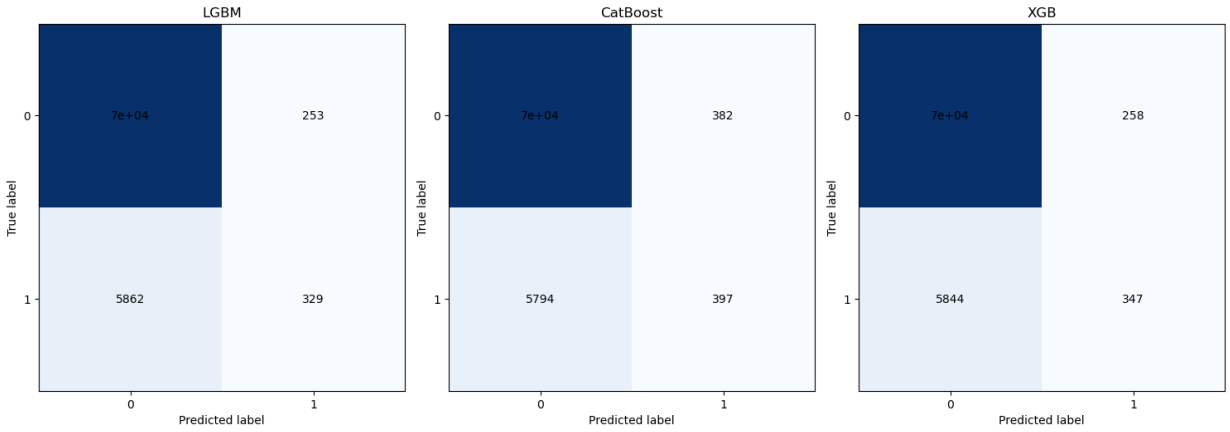

Fig. 42 Matricele de confuzie pentru modelele optimizate

Matricea de confuzie este un instrument vital în evaluarea performanței modelelor de clasificare, oferind o imagine clară a corectitudinii și greșelilor făcute de model. Pentru modelele fine-tunate LightGBM, CatBoost și XGBoost, matricea de confuzie arată o tendință generală a modelelor de a clasifica corect majoritatea cazurilor negative (clasa 0), dar cu o performanță variabilă în clasificarea cazurilor pozitive (clasa 1).

LightGBM prezintă un număr relativ mic de fals negativi și un număr modest de adevărați pozitivi, sugerând o sensibilitate decentă a modelului. CatBoost arată o ușoară îmbunătățire în detectarea cazurilor pozitive, în timp ce XGBoost are un echilibru similar cu LightGBM, indicând o performanță comparabilă între cele două modele.

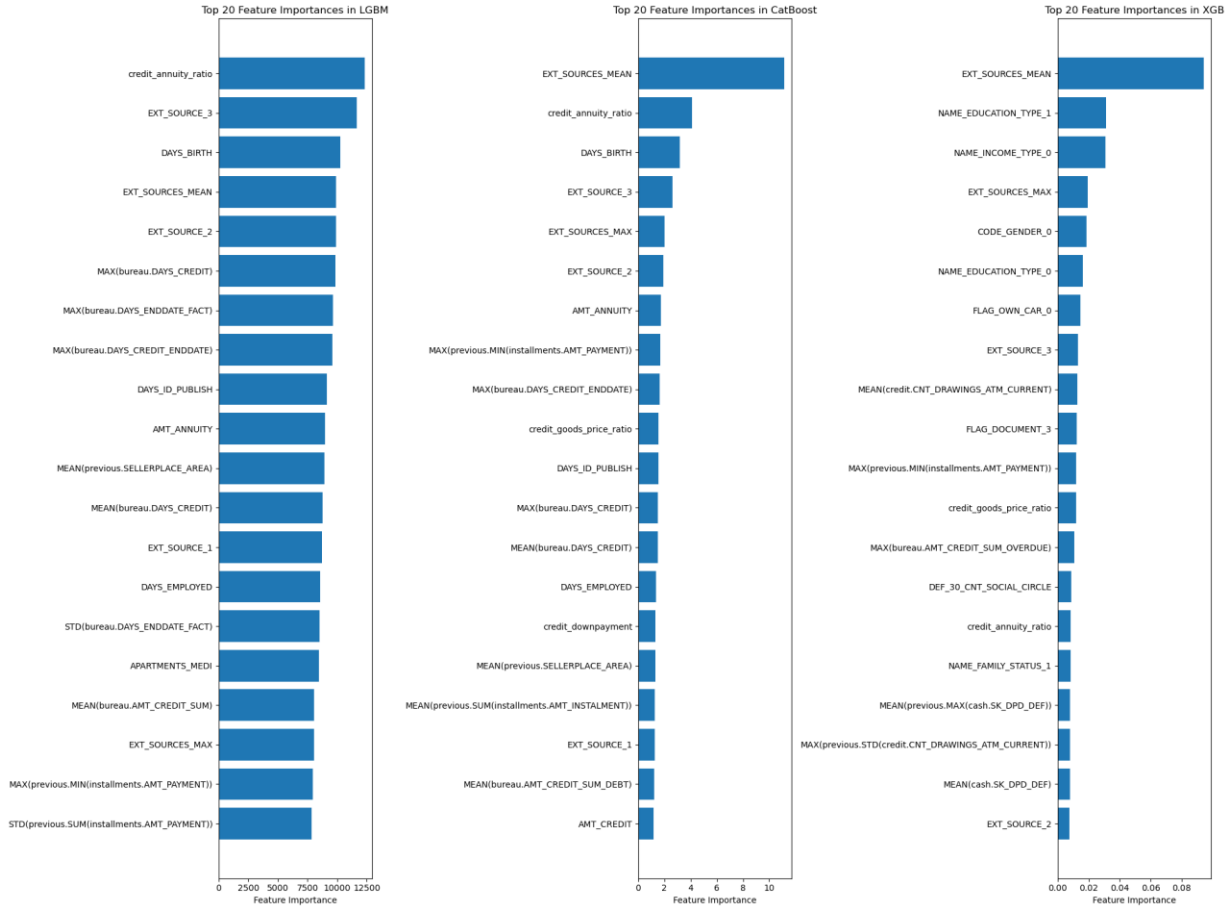

Fig. 43 Importanțele modelelor optimizate

Când privim la importanța caracteristicilor, observăm că 'EXT_SOURCES' și 'DAYS_BIRTH' sunt considerate importante în toate cele trei modele, subliniind impactul lor semnificativ în predicția riscului de credit. Acest lucru evidențiază o înțelegere coerentă a forțelor care conduc la decizia de creditare, indiferent de algoritmul specific utilizat.

Importanțele diferă însă în rest, cu CATBoost și XGBoost care acordă o importanță mai mare unor caracteristici legate de comportamentul de plată și istoricul creditului, în timp ce LightGBM pare să valorizeze mai mult caracteristicile legate de profilul clientului și istoricul tranzacțiilor.

Aceste matrice de confuzie și graficele de importanță ale caracteristicilor ne oferă indicii prețioase despre cum fiecare model procesează și interpretează datele. Înțelegerea acestor diferențe este esențială pentru îmbunătățirea continuă a modelelor și pentru adaptarea strategiilor de modelare la specificitățile problemei de creditare.

6. Concluzii

În concluzie, această teză subliniază rolul vital al analizei comprehensive a datelor și al modelării predictive avansate în îmbunătățirea evaluării riscului de credit. Prin investigarea diverselor seturi de date și utilizarea algoritmilor sofisticați de învățare automată, în special LightGBM, am reușit să împingem semnificativ limitele acurateței predicțiilor de credit.

Explorarea noastră a modelării predictive a evidențiat nu doar complexitatea și provocările în analiza riscului de credit, dar și potențialul transformării acestei domenii prin tehnici avansate de prelucrare și analiză a datelor. Modelul dezvoltat pe parcursul acestui studiu demonstrează eficiența în identificarea și interpretarea factorilor semnificativi care influențează probabilitatea de rambursare a creditelor, furnizând astfel instituțiilor financiare un instrument robust pentru gestionarea riscurilor.

Această cercetare nu doar că a adus clarificări importante în ceea ce privește variabilele critice în predicția riscului de credit, dar a și deschis calea pentru investigații viitoare care pot explora noi metode și tehnologii în îmbunătățirea modelării riscului de credit. Prin colaborare și inovație continuă, putem spera să atingem un nivel și mai înalt de precizie și eficiență în serviciile financiare, facilitând astfel decizii de creditare mai informate și mai echitabile.