

National University of Science and Technology  
POLITEHNICA Bucharest  
Faculty of Automatic Control and Computers  
Computer Science and Engineering Department



## DIPLOMA PROJECT

Utilizarea tehnicilor de explicabilitate pentru rețele neurale de  
clasificare a cancerului

Your name

**Thesis advisors:**

Prof./Conf./S.I./As. dr./drd. ing. X

Prof./Conf./S.I./As. dr./drd. ing. Y

**BUCHAREST**

202x

# CONTENTS

<b>List of figures</b>	<b>iii</b>
<b>List of tables</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>1 Introducere</b>	<b>1</b>
1.1 Motivație . . . . .	1
1.2 Obiective . . . . .	1
1.3 Context . . . . .	2
<b>2 Soluții existente</b>	<b>3</b>
2.1 Class Activation Maps (CAM) . . . . .	4
2.1.1 Principiul de functionare al cam-urilor . . . . .	4
2.1.2 Variante si extinderi ale CAM-urilor . . . . .	5
2.2 LIME – Local Interpretable Model-agnostic Explanations . . . . .	6
2.3 SHAP – SHapley Additive exPlanations . . . . .	7
2.4 Guided Backpropagation . . . . .	8
2.5 Integrated Gradients . . . . .	8
<b>3 Abordare propusă</b>	<b>10</b>
3.1 Explicabilitatea modelelor pentru date tabelare . . . . .	10
3.1.1 Analiza exploratorie a datelor (EDA) . . . . .	10
3.2 Descrierea setului de date pentru diabet . . . . .	16
3.2.1 Structura datelor . . . . .	16
3.2.2 Exemplu de instanțe . . . . .	17
3.2.3 Observații . . . . .	17

3.2.4	Analiză exploratorie . . . . .	17
3.2.5	Optimizarea hiperparametrilor pentru Random Forest . . . . .	19
3.2.6	Compararea mai multor modele de clasificare . . . . .	21
3.2.7	Explicabilitatea modelului folosind LIME . . . . .	24
3.2.8	Exemplu de interpretare LIME pentru o instanță . . . . .	25
3.2.9	Explicabilitatea modelului folosind SHAP . . . . .	27
3.2.10	Vizualizarea detaliată a contribuțiilor variabilelor folosind SHAP Waterfall	28
3.3	Corpus . . . . .	30
3.3.1	Descriptives . . . . .	30
3.4	(Neural) Architecture . . . . .	30
3.5	Performance Metrics . . . . .	30
<b>4</b>	<b>Rezultate experimentale</b>	<b>31</b>
<b>5</b>	<b>Discuții</b>	<b>32</b>
5.1	Performance Comparison . . . . .	32
5.2	Limitations . . . . .	32
<b>6</b>	<b>Concluzii și dezvoltări ulterioare</b>	<b>33</b>
	<b>References</b>	<b>34</b>

## LIST OF FIGURES

1	Grafic care prezintă distribuția claselor . . . . .	12
2	Grafic care prezintă valoarea IQR pentru fiecare caracteristică numerică din setul de date . . . . .	13
3	Grafic care prezintă valorile pentru corelația Pearson . . . . .	13
4	Histograma pentru atributul worst concave points . . . . .	14
5	Histograma pentru atributul worst perimetre . . . . .	14
6	Heatmap de corelații pentru primele 12 caracteristici cu cea mai mare corelație absolută cu target . . . . .	15
7	Scatter plot cu rezultatele aplicării PCA . . . . .	15
8	Distribuția claselor din setul de date cu informații despre diabet . . . . .	17
9	Histograme pentru variabilele <i>ma0.5</i> , <i>ma1.0</i> , <i>ex0.5</i> , <i>ex1.0</i> și <i>eucmac</i> . . . . .	18
10	Matricea de corelații dintre variabilele setului de date, reprezentată sub formă de heatmap. . . . .	19
11	Explicația locală a predicției modelului Random Forest pentru instanța 328 utilizând LIME. . . . .	27
12	Vizualizarea contribuțiilor variabilelor la predicția modelului Random Forest pentru o instanță specifică, folosind metoda SHAP. . . . .	28
13	Contribuția fiecărei variabile la predicția modelului Random Forest pentru instanța 9, vizualizată cu SHAP Waterfall. . . . .	29

## LIST OF TABLES

1	Exemplu de înregistrări din setul de date <i>Diabetino</i> . . . . .	17
2	Performanța clasificatoarelor pe setul de test. . . . .	23
3	Valorile principale ale variabilelor pentru două instanțe de test. . . . .	25

4	Valorile variabilelor pentru instanța 328. . . . .	26
---	--	----

## **ABSTRACT**

The abstract should contain the following information (1/2 sentences per item):

- Problem statement / context
- Aim of the thesis/research objective(s)
- Brief method details
- Main results
- Concluding remarks

## **REZUMAT**

Translation to be made at the end

**Keywords:** max 5, e.g., Language Models, Natural Language Processing

# 1 INTRODUCERE

Cancerul reprezintă una dintre principalele cauze de mortalitate la nivel global, iar diagnosticul precoce este esențial pentru creșterea șanselor de supraviețuire și pentru îmbunătățirea calității vieții pacienților. În ultimii ani, progresele în domeniul inteligenței artificiale, în special în rețelele neurale profunde (deep learning), au adus îmbunătățiri semnificative în analiza și interpretarea imaginilor medicale utilizate în diagnosticarea cancerului. Aceste modele au demonstrat performanțe comparabile sau chiar superioare specialiștilor umani în diverse sarcini de clasificare.

Totuși, adoptarea pe scară largă a acestor metode în practica medicală este adesea limitată de natura lor de „cutie neagră” (black box), care face dificilă înțelegerea procesului decizional intern. Lipsa transparenței și a interpretabilității ridică probleme etice, legale și de încredere, în special în domeniul medical, unde deciziile pot influența în mod direct viața pacienților.

În acest context, tehnicile de explicabilitate a inteligenței artificiale (Explainable Artificial Intelligence – XAI) au un rol crucial. Aceste metode permit interpretarea rezultatelor furnizate de modelele de învățare automată, evidențiind factorii și caracteristicile relevante care contribuie la clasificare. În cazul rețelelor neurale aplicate la detectarea și clasificarea cancerului, explicabilitatea poate facilita înțelegerea modului în care modelul analizează datele, sprijinind medicii în luarea deciziilor și crescând nivelul de încredere în tehnologie.

## 1.1 Motivație

Cancerul de sân este o problemă semnificativă de sănătate la nivel mondial, reprezentând 11,7% din cauzele de deces. Pacienții cu cancer în stadiu incipient au avut o rată de supraviețuire mult mai mare și un risc semnificativ mai mic de a muri din cauza malignității. Diverse metode de imagistică, cum ar fi razele X, mamografia, RMN, ultrasunetele și imagistica foto-acustică au fost utilizate pentru screening-ul și diagnosticarea cancerului de sân. Cu toate acestea, au limitări, cum ar fi sensibilitate redusă, rate fals pozitive și disponibilitate limitată pentru terapia chirurgicală.

## 1.2 Obiective

Aceasta lucrare își propune să investigheze și să aplice tehnici de explicabilitate pentru rețele neurale destinate clasificării cancerului, cu scopul de a evidenția atât performanța modelelor, cât și transparența acestora. Se vor analiza metode de tip post-hoc, precum Grad-CAM, și se va evalua utilitatea lor în context clinic, urmărind crearea unui echilibru între acuratețea

predicțiilor și interpretabilitate.

## 1.3 Context

În ultima perioadă au apărut din ce în ce mai multe cazuri de cancer, iar numărul medicilor experți este foarte mic raportat la numărul total de pacienți. Astfel, este utilă utilizarea unor metode automate de diagnostic însă este esențial ca acestea să poată fi interpretabile. Cancerul de sân a devenit o cauză principală de mortalitate, incidența sa crescând continuu, ridicând probleme de sănătate la nivel mondial. Diagnosticul precis și tratamentul în timp util sunt esențiale în scăderea ratei mortalității asociate cu cancerul de sân. Cu toate acestea, practicile medicale se confruntă cu provocări precum resursele insuficiente de asistență medicală și eroarea umană în interpretare, care împiedică detectarea și diagnosticarea în timp util și precisă a cancerului de sân. În acest context, tehnologiile inteligente de asistență diagnostică au apărut ca soluții inovatoare în lupta împotriva cancerului de sân. Progresele recente în învățarea profundă au revoluționat diagnosticul cancerului de sân, depășind limitările metodelor tradiționale.

## 2 SOLUȚII EXISTENTE

Khater et al. Khater et al. (2023) subliniază că, în detecția cancerului, existența rezultatelor false pozitive sau false negative poate avea consecințe grave asupra pacienților. Autorii arată că de la un algoritm de inteligență artificială nu se dorește doar o predicție binară legată de prezența sau absența cancerului, ci și o justificare a rezultatului obținut. O astfel de abordare crește transparența și gradul de încredere în utilizarea acestor tehnologii. Pentru atingerea acestui obiectiv, ei propun folosirea tehnicilor de *Explainable Artificial Intelligence* (XAI), menționând, printre altele, utilitatea reprezentărilor de tip *Partial Dependence Plot* (PDP), care evidențiază modul în care o caracteristică influențează predicțiile modelului.

Analizând literatura de specialitate, Khater et al. Khater et al. (2023) arată că numeroase studii privind clasificarea cancerului de sân au raportat rate ridicate de diagnostic corect. Totuși, autorii remarcă faptul că dimensiunea explicativă este adesea insuficient dezvoltată. În multe lucrări sunt identificate caracteristici importante pentru predicție, dar nu se pune accent pe modul concret în care acestea contribuie la decizia finală a algoritmului. În acest context, studiul lor aduce o contribuție prin integrarea unor metode explicative care clarifică relația dintre caracteristici și rezultatele modelului.

Pentru validarea abordării, Khater et al. Khater et al. (2023) utilizează două seturi de date consacrate:

- *Wisconsin Breast Cancer (WBC)*, care conține 10 caracteristici pentru fiecare instanță, ultima indicând natura malignă sau benignă a tumorii;
- *Wisconsin Diagnostic Breast Cancer (WDBC)*, care conține 30 de caracteristici pentru fiecare instanță și o etichetă ce diferențiază tumorile maligne de cele benigne.

Autorii propun un flux metodologic structurat, care include: colectarea și curățarea datelor, identificarea caracteristicilor relevante, alegerea algoritmilor de învățare automată, antrenarea și validarea modelelor pe subseturi de date și, ulterior, evaluarea performanțelor. În plus, pentru a asigura transparența rezultatelor, predicțiile sunt analizate cu ajutorul tehnicilor XAI, precum PDP, care ilustrează impactul fiecărei variabile asupra deciziilor modelului.

Khater et al. Khater et al. (2023) arată că anumite caracteristici morfologice, precum forma, dimensiunea sau suprafața celulelor, au un impact semnificativ asupra preciziei predicțiilor. De asemenea, autorii compară performanțele mai multor algoritmi, identificând metodele cu acuratețea cea mai ridicată pe fiecare set de date.

Conform studiului lui Khater et al. Khater et al. (2023), în cadrul setului WBC, cel mai performant algoritm a fost *K-Nearest Neighbors* (KNN), iar pentru setul WDBC cel mai bun

rezultat a fost obținut cu un model de tip *Artificial Neural Network* (ANN). Ei evidențiază, de asemenea, importanța unor caracteristici specifice: "bare nucleii" pentru WBC și "area worst" pentru WDBC. În final, autorii sugerează că, pe viitor, integrarea datelor genetice ar putea oferi o analiză mai avansată și cu relevanță clinică mai mare.

## 2.1 Class Activation Maps (CAM)

În ultimii ani rețelele neurale adânci au devenit standardul de facto în sarcinile de clasificare a imaginilor medicale, datorita capacității lor de a învăța automat caracteristici complexe și relevante. Totuși, abordările bazate pe aceste modele sunt percepute drept "cutii negre", deoarece procesul prin care ele ajung la o decizie finală nu este ușor de interpretat. În domeniul medical, unde deciziile pot influența direct diagnosticul și tratamentul pacienților, accesata lipsa de transparență reprezintă o problemă majoră.

Pentru a răspunde acestei provocări au fost dezvoltate diverse metode de explicabilitate care încearcă să evidențieze regiunile dintr-o imagine ce au contribuit cel mai mult la predicția realizată de model. Printre aceste tehnici Class Activation Maps ocupă un loc important deoarece oferă o modalitate intuitivă de a vizualiza ceea ce putem numi atenția modelului. Practic, cam-urile generează o hartă de importanță ce poate fi suprapusă peste imaginea inițială, arătând zonele care au influențat decizia de clasificare.

În contextul clasificării cancerului, cam-urile sunt extrem de utile. Ele nu doar că permit verificarea faptului că rețeaua neurală se uită la zonele patologice relevante, dar contribuie și la creșterea încrederii medicilor în utilizarea sistemelor automate de analiză a datelor medicale. De exemplu, o metodă ce vine la pachet cu o tehnică de explicabilitate îi poate evidenția medicului o tumoră într-o radiografie sau o regiune suspectă într-o imagine histopatologică. Astfel, este mult mai ușor pentru medic să decidă cât de relevant este diagnosticul pus de sistemul automat deoarece totul se poate rezuma la analizarea porțiunii evidențiate de metoda de explicabilitate.

### 2.1.1 Principiul de funcționare al cam-urilor

Idea de bază a metodelor de tip cam este de a identifica regiunile dintr-o imagine care au contribuit cel mai mult la predicția unei anumite clase pe baza calculului de gradienti. Într-o rețea neurală convoluțională, care până nu de mult era principala variantă utilizată pentru recunoașterea cancerului, straturile convoluționale învață hărți de caracteristici care codifică informații spațiale și semantice despre imagine. În mod normal, aceste hărți sunt aplatizate și trecute printr-un modul de clasificare bazat de cele mai multe ori pe straturi complet conectate pentru a putea obține predicția finală. Acest lucru ducea la pierderea informației legate de localizarea exactă a trăsăturilor relevante.

Metoda CAM rezolvă această problemă prin utilizarea unui strat de tip Global Average Pooling

(GAP) înainte stratului de clasificare. GAP are rolul de a reduce fiecare harta de caracteristici la o singura valoare prin calcularea mediei activarilor pe intreaga suprafata. In acest fel, fiecare harta convolutionala este direct legata de o pondere din stratul final de clasificare, pastrand insa informatia despre importanta sa globala pentru predictia unei clase.

Formal, harta de activare pentru o clasă  $c$  poate fi calculată astfel:

$$M_c(x, y) = \sum_k w_k^c \cdot f_k(x, y)$$

unde:

- $f_k(x, y)$  reprezintă activarea neuronului din harta de caracteristici  $k$  la poziția  $(x, y)$ ;
- $w_k^c$  este ponderea asociată clasei  $c$  pentru harta  $k$ ;
- $M_c(x, y)$  este scorul CAM pentru poziția  $(x, y)$ .

Rezultatul este o harta de activare (Heatmap) care evidentiaza pozitiile din imagine cu contributie ridicata la predictia finala. Prin suprapunerea acestei harti peste imaginea initiala, se poate observa intuitiv ce regiuni au influenta predictia clasificatorului. Astfel, cam-urile ofera o legatura directa intre predictia numerica a modelului si informatia vizuala perceputa de acesta. In cazul clasificarii cancerului, acest mecanism permite verificarea faptului ca modelul se concentreaza pe zonele patologice relevante si nu pe artefacte sau regiuni irelevante ale imaginii.

## 2.1.2 Variante si extinderi ale CAM-urilor

De la introducerea initiala a metodei cam, au aparut mai multe variante si extinderi menite sa rezolve diverse limitari identificate pentru metoda de baza. In continuare, vor fi prezentate cele mai importante si utilizate variante sau extinderi ale metodei CAM.

### Grad-CAM

**Grad-CAM (Gradient-weighted Class Activation Mapping)** este o tehnică ce utilizează gradientul funcției de pierdere față de hărțile de caracteristici pentru a calcula importanța fiecărei hărți în predicția finală. Astfel, metoda nu mai necesită existența unui strat de *Global Average Pooling*, putând fi aplicată pe o gamă mai largă de arhitecturi CNN.

Formula generalizată a hărții de activare este:

$$M_c(x, y) = \text{ReLU} \left( \sum_k \alpha_k^c \cdot f_k(x, y) \right)$$

unde:

- $\alpha_k^c$  reprezintă importanța calculată pe baza gradientului pentru clasa  $c$ ,
- $f_k(x, y)$  este activarea hărții de caracteristici  $k$ ,
- $M_c(x, y)$  este harta Grad-CAM pentru clasa  $c$ .

## DT Grad-CAM

**DT Grad-CAM** Metoda DT Grad-CAM îmbunătățește abordarea tradițională Grad-CAM prin mai mulți pași. Fiecare pas îmbunătățește vizualizarea regiunilor specifice clasei, făcând rezultatele mai interpretabile și mai relevante în scopuri de diagnosticare.

## Grad-CAM++

**Grad-CAM++** adresează o limitare a metodei Grad-CAM, și anume dificultatea de a localiza mai multe regiuni de interes simultan. Aceasta folosește o formulă de ponderare mai complexă, bazată pe gradient de ordin superior, pentru a oferi hărți mai detaliate și o localizare mai precisă.

## Score-CAM

**Score-CAM** elimină complet utilizarea gradientului. În schimb, măsoară contribuția fiecărei hărți de activare prin mascarea imaginii inițiale cu acea hartă și observarea variației scorului de clasificare. Această abordare tinde să fie mai stabilă și să reducă zgomotul introdus de gradient.

## Ablation-CAM

**Ablation-CAM** calculează importanța fiecărei hărți de caracteristici prin "oprirea" acestora și observarea impactului asupra predicției finale. Deși poate fi mai costisitoare computațional, metoda are avantajul de a oferi o măsură directă a relevanței fiecărei hărți.

## 2.2 LIME – Local Interpretable Model-agnostic Explanations

LIME (*Local Interpretable Model-agnostic Explanations*) este o metodă de explicabilitate propusă de Ribeiro et al. care permite interpretarea deciziilor unui model complex prin construirea unui model local, simplu și interpretabil, în jurul unei predicții individuale Ribeiro et al. (2016). Ideea centrală este că, deși modelul global poate fi foarte complex și greu de interpretat, în jurul unei observații specifice comportamentul său poate fi aproximat de un model liniar sau de alt tip simplu, a cărui structură este ușor de înțeles.

În contextul clasificării cancerului, LIME poate fi aplicat atât pe date tabulare, cât și pe imagini medicale. Pentru imaginile radiologice, metoda generează perturbări ale imaginii de intrare (de exemplu, prin segmentarea în superpixeli) și evaluează cum modificarea sau eliminarea fiecărui segment afectează predicția modelului. Astfel, se obține o hartă de importanță care evidențiază regiunile relevante pentru decizia algoritmului, oferind radiologilor informații intuitive despre caracteristicile tumorii sau ale țesutului suspect.

Avantajul principal al LIME este că este agnostic față de model, adică poate fi folosit pentru orice tip de algoritm de învățare automată, inclusiv rețele neuronale profunde sau ensemble-uri. Totuși, LIME are și limitări: explicabilitatea este locală, adică explicațiile se aplică unei singure predicții și nu reflectă comportamentul global al modelului, și rezultatele pot fi sensibile la modul în care se generează perturbările. Prin urmare, LIME oferă o modalitate practică de a crește transparența și încrederea în predicțiile AI pentru detectarea cancerului, complementând metode vizuale precum CAM sau Grad-CAM.

## 2.3 SHAP – SHapley Additive exPlanations

SHAP (*SHapley Additive exPlanations*) este o metodă de explicabilitate inspirată din teoria jocurilor, care atribuie fiecărei caracteristici o "valoare Shapley" ce reprezintă contribuția acesteia la predicția finală a modelului Lundberg & Lee (2017). Această abordare oferă o interpretare matematică riguroasă, cuantificând cât de mult a influențat fiecare caracteristică decizia modelului.

În contextul clasificării cancerului, SHAP poate fi aplicat atât pe date tabulare, cum ar fi caracteristicile extrase din tumori (dimensiune, forma, textură), cât și pe date imagistice, prin maparea valorilor Shapley pe regiuni sau pixeli ai imaginii. Astfel, se obține o evaluare detaliată a factorilor care determină predicția modelului, oferind medicilor informații interpretabile despre relevanța fiecărei caracteristici.

Un avantaj major al SHAP este consistența și posibilitatea de a furniza explicații globale și locale. Valorile Shapley pot fi agregate pentru a analiza comportamentul modelului la nivelul întregului set de date (explicații globale) sau pentru fiecare predicție individuală (explicații locale). Totuși, SHAP poate fi mai costisitor din punct de vedere computațional, mai ales pentru modele complexe sau seturi de date mari.

Prin aplicarea SHAP, explicabilitatea modelelor AI în detecția cancerului este îmbunătățită, oferind un echilibru între rigurozitate matematică și interpretabilitate practică, fiind complementară altor metode vizuale precum CAM sau Grad-CAM.

## 2.4 Guided Backpropagation

*Guided Backpropagation* este o tehnică de vizualizare utilizată pentru interpretarea rețelelor neuronale convoluționale (CNN), care permite evidențierea caracteristicilor din imagine ce contribuie cel mai mult la predicția modelului Springenberg et al. (2014). Metoda se bazează pe propagarea inversă a gradientului, dar cu o modificare: doar semnalele pozitive sunt propagate înapoi prin rețea, ceea ce generează hărți de activare mai clare și mai ușor de interpretat decât backpropagation-ul standard.

În contextul clasificării cancerului, Guided Backpropagation poate fi aplicat pe imagini medicale, cum ar fi mamografii sau scanări histopatologice, pentru a evidenția detalii fine ale celulelor sau ale structurilor tumorale care influențează decizia modelului. Această metodă ajută radiologii și cercetătorii să înțeleagă ce regiuni ale imaginii au fost decisive pentru clasificarea ca benignă sau malignă, oferind astfel un nivel ridicat de interpretabilitate vizuală.

Avantajul principal al Guided Backpropagation constă în capacitatea sa de a evidenția detalii subtile, ceea ce îl face util pentru analiza imaginilor complexe. Totuși, metoda este limitată la modelele de tip CNN și nu oferă explicații cuantitative pentru importanța caracteristicilor, fiind o tehnică pur vizuală. Pentru a obține o interpretabilitate mai completă, Guided Backpropagation este adesea combinat cu alte metode, precum Grad-CAM sau LIME.

## 2.5 Integrated Gradients

*Integrated Gradients* este o metodă de explicabilitate propusă pentru a cuantifica contribuția fiecărei caracteristici sau a fiecărui pixel la predicția unui model de învățare automată, respectând principiile de sensibilitate și consistență Sundararajan et al. (2017). Metoda se bazează pe calcularea integralului gradientului predicției față de o referință (*baseline*), de la starea de referință la exemplul curent, ceea ce permite estimarea contribuției cumulative a fiecărei intrări.

În contextul detecției cancerului, Integrated Gradients poate fi aplicat atât pe date tabulare, cât și pe imagini medicale, cum ar fi mamografii sau scanări histopatologice. Rezultatul este o hartă de importanță care evidențiază regiunile sau caracteristicile critice pentru predicția modelului, oferind radiologilor și cercetătorilor informații detaliate despre factorii care influențează decizia AI.

Avantajele metodei constau în faptul că oferă explicații cuantitative și respectă proprietăți teoretice riguroase, precum faptul că suma contribuțiilor caracteristicilor este egală cu diferența dintre predicția modelului pentru exemplul curent și cea pentru baseline. Totuși, Integrated Gradients necesită alegerea unei referințe corespunzătoare și poate fi mai costisitor computațional pentru imagini de mari dimensiuni sau modele complexe.

Prin aplicarea Integrated Gradients, interpretabilitatea modelelor AI pentru clasificarea can-

cerului este îmbunătățită semnificativ, permițând o analiză detaliată a factorilor care contribuie la deciziile modelului și complementând alte metode vizuale, cum ar fi Grad-CAM sau Guided Backpropagation.

## 3 ABORDARE PROPUȘĂ

### 3.1 Explicabilitatea modelelor pentru date tabelare

În cadrul acestei lucrări am utilizat Breast Cancer Wisconsin Diagnostic Dataset, disponibil în biblioteca `scikit-learn`. Acest set de date este unul dintre cele mai folosite benchmark-uri în problemele de clasificare binară din domeniul medical, fiind construit pe baza unor măsurători obținute din imagini digitale ale unor biopsii mamare. Scopul său este de a sprijini dezvoltarea și evaluarea metodelor de învățare automată pentru diagnosticarea diferențială între tumori maligne și benigne.

Setul de date conține un număr de 569 de observații, fiecare reprezentând un caz clinic. Pentru fiecare observație sunt disponibile 30 de variabile numerice ce descriu caracteristici morfologice ale nucleilor celulari (de exemplu: raza medie, textura, perimetrul, aria, concavitatea sau simetria). Aceste caracteristici sunt derivate prin prelucrarea imaginilor și oferă o descriere cantitativă a aspectului celular.

Variabila țintă este denumită `target` și are două valori posibile:

- **0 – malignant** (cazuri de cancer de sân cu caracter malign);
- **1 – benign** (cazuri de cancer de sân cu caracter benign).

Distribuția claselor este ușor dezechilibrată, cu 212 cazuri maligne și 357 cazuri benigne, ceea ce subliniază importanța folosirii unor metode de evaluare corespunzătoare, precum ROC-AUC și matricea de confuzie, în completarea simplei acurateți.

Alegerea acestui set de date este motivată de:

1. relevanța practică în domeniul medical, unde interpretabilitatea modelelor este esențială pentru susținerea deciziilor clinice;
2. prelucrarea preexistentă și lipsa valorilor lipsă, ceea ce permite concentrare pe aspectele de analiză exploratorie și explicabilitate;
3. utilizarea frecventă în literatură, ceea ce facilitează comparabilitatea rezultatelor obținute.

#### 3.1.1 Analiza exploratorie a datelor (EDA)

Analiza exploratorie a datelor (EDA) reprezintă un pas esențial în înțelegerea setului de date și în identificarea caracteristicilor relevante pentru modelare. Scopul EDA este de a investiga

structura, distribuțiile și corelațiile dintre variabile, precum și de a detecta eventuale probleme de calitate a datelor.

Primul pas în analiza setului de date a constat în evaluarea integrității și calității acestuia. S-a verificat prezența valorilor lipsă, tipurile de date și numărul de instanțe unice pentru fiecare caracteristică. Setul de date Breast Cancer Wisconsin Diagnostic este complet, fără valori lipsă, iar toate variabilele numerice sunt de tip float, ceea ce simplifică preprocesarea. Distribuția claselor pentru variabila țintă target a fost, de asemenea, analizată și rezultatul este disponibil în Figura 1: 212 observații aparțin clasei maligne și 357 clasei benigne, indicând un dezechilibru moderat care trebuie luat în considerare în alegerea metricilor de evaluare și în împărțirea train/test prin stratificare. Această etapă preliminară asigură că datele sunt curate și consistente, evitând astfel erori în analizele ulterioare.

Codul folosit pentru această etapă este următorul:

```
data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target
df.head()
print(df.info())
```

Rezultatele obținute sunt prezentate în continuare:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   mean radius                           569 non-null    float64
1   mean texture                           569 non-null    float64
2   mean perimeter                         569 non-null    float64
3   mean area                             569 non-null    float64
4   mean smoothness                       569 non-null    float64
5   mean compactness                      569 non-null    float64
6   mean concavity                         569 non-null    float64
7   mean concave points                   569 non-null    float64
8   mean symmetry                         569 non-null    float64
9   mean fractal dimension                569 non-null    float64
10  radius error                          569 non-null    float64
11  texture error                         569 non-null    float64
12  perimeter error                       569 non-null    float64
13  area error                           569 non-null    float64
14  smoothness error                      569 non-null    float64
```

15	compactness error	569 non-null	float64
16	concavity error	569 non-null	float64
17	concave points error	569 non-null	float64
18	symmetry error	569 non-null	float64
19	fractal dimension error	569 non-null	float64
20	worst radius	569 non-null	float64
21	worst texture	569 non-null	float64
22	worst perimeter	569 non-null	float64
23	worst area	569 non-null	float64
24	worst smoothness	569 non-null	float64
25	worst compactness	569 non-null	float64
26	worst concavity	569 non-null	float64
27	worst concave points	569 non-null	float64
28	worst symmetry	569 non-null	float64
29	worst fractal dimension	569 non-null	float64
30	target	569 non-null	int64

dtypes: float64(30), int64(1)  
memory usage: 137.9 KB

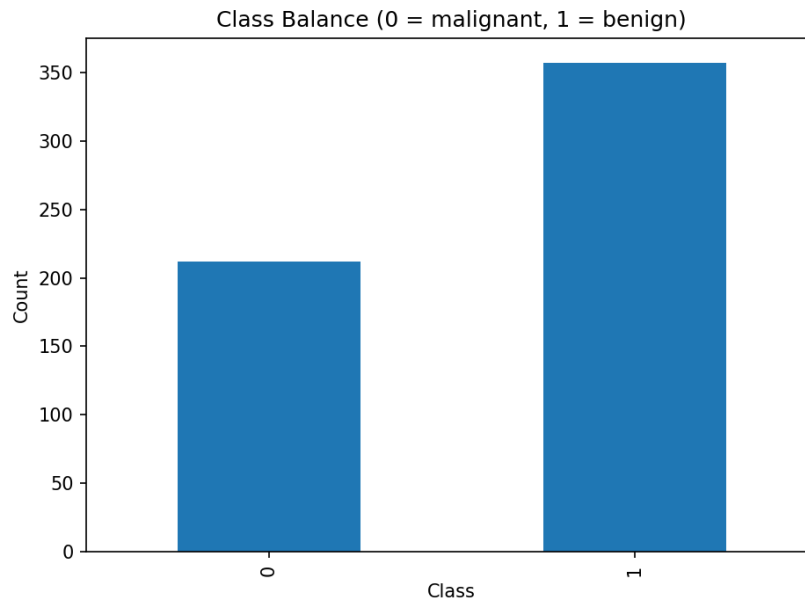


Figure 1: Grafic care prezintă distribuția claselor

Următorul pas a constat în calcularea statisticilor descriptive pentru fiecare caracteristică numerică, incluzând media, deviația standard, valorile minime și maxime, precum și intervalul interquartil (IQR). Aceste măsuri oferă o imagine asupra distribuției și dispersiei datelor, precum și a variabilității între instanțe. De exemplu, caracteristici precum raza și aria nucleului celular prezintă variații semnificative între clase, sugerând că acestea pot fi utile în discriminarea tumorilor maligne de cele benigne. IQR-ul, în special, ajută la identificarea valorilor

extreme și a posibilelor outliere, care ar putea influența modelele de clasificare. În Figura 2 se pot observa aceste valori.

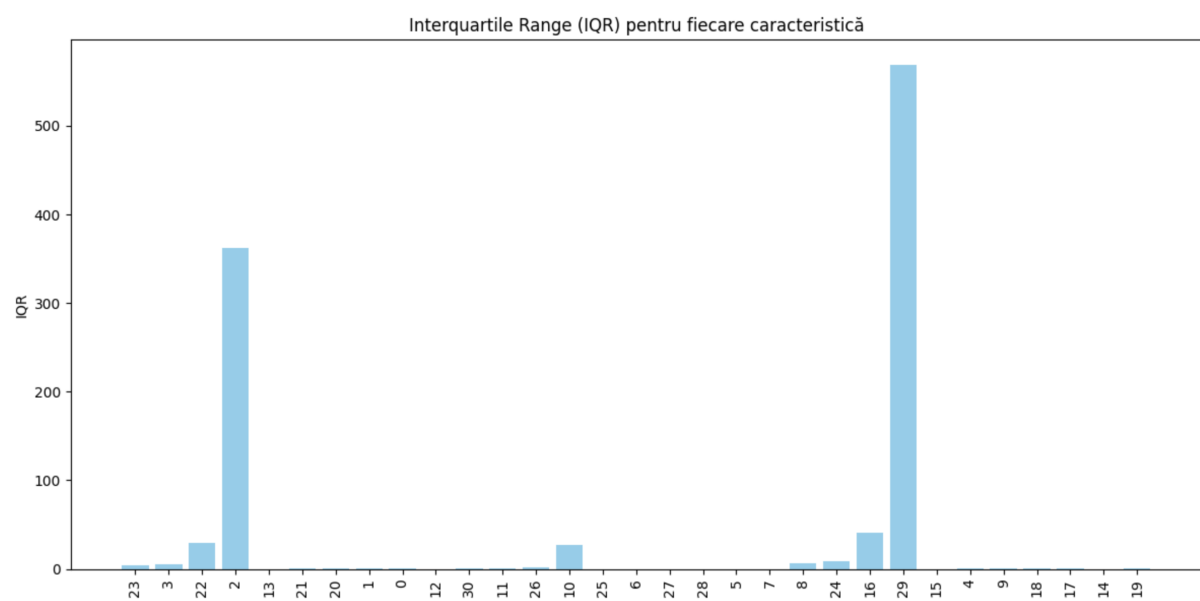


Figure 2: Grafic care prezintă valoarea IQR pentru fiecare caracteristică numerică din setul de date

Analiza corelațiilor între caracteristicile numerice și variabila țintă a permis identificarea atributelor cu putere discriminativă ridicată. Rezultatele obținute sunt prezentate în Figura 3. S-a folosit corelația Pearson pentru a măsura relația liniară dintre fiecare caracteristică și target. Caracteristici cu corelații mari, pozitive sau negative, indică faptul că variațiile lor sunt strâns legate de tipul tumorii. Aceste informații nu doar că facilitează selecția caracteristicilor relevante, dar oferă și o perspectivă interpretabilă asupra factorilor biologici determinanți, ceea ce este esențial în contextul aplicării medicale.

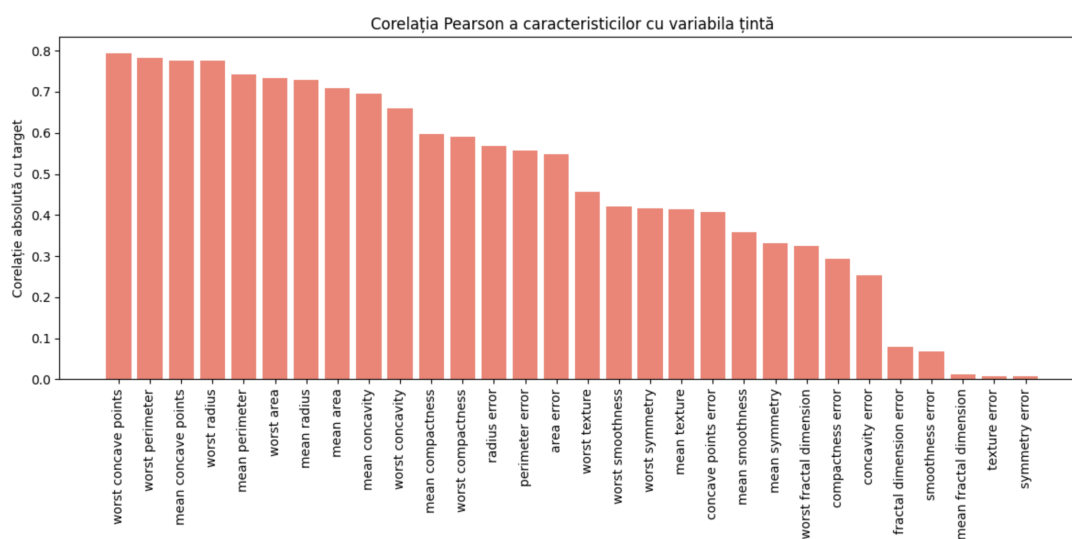


Figure 3: Grafic care prezintă valorile pentru corelația Pearson

Vizualizările sunt un instrument crucial în EDA, deoarece permit o interpretare intuitivă a

datelor. Pentru cele mai relevante caracteristici, am generat histograme, care arată distribuția frecvenței valorilor și diferențele dintre clase, și boxplot-uri, care permit identificarea valorilor extreme și compararea distribuțiilor mediane între clase. În plus, un heatmap de corelații pentru primele 12 caracteristici cu cea mai mare corelație absolută cu target a evidențiat relațiile liniar-coliniare dintre variabile, utile pentru detectarea redundanțelor care ar putea afecta performanța modelului. Această vizualizare facilitează, de asemenea, interpretarea intuitivă a modului în care caracteristicile interacționează între ele.

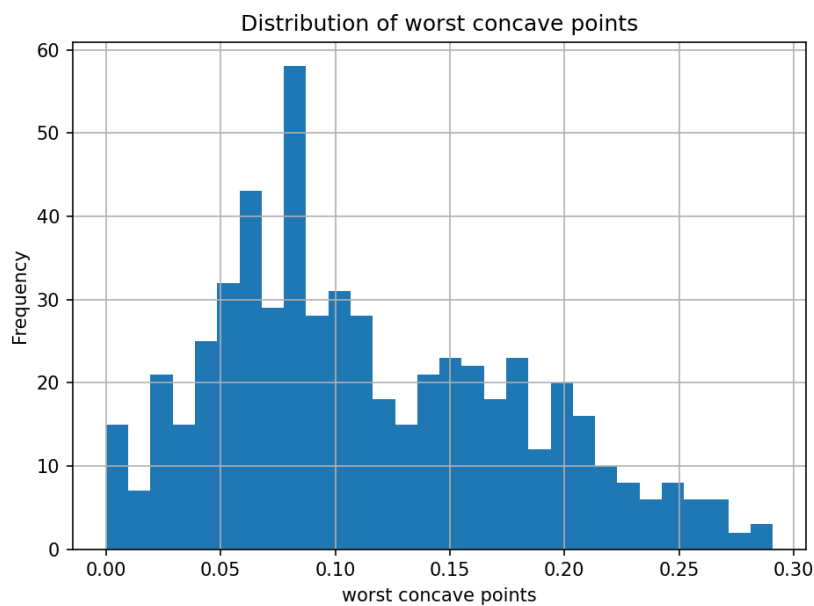


Figure 4: Histograma pentru atributul worst concave points

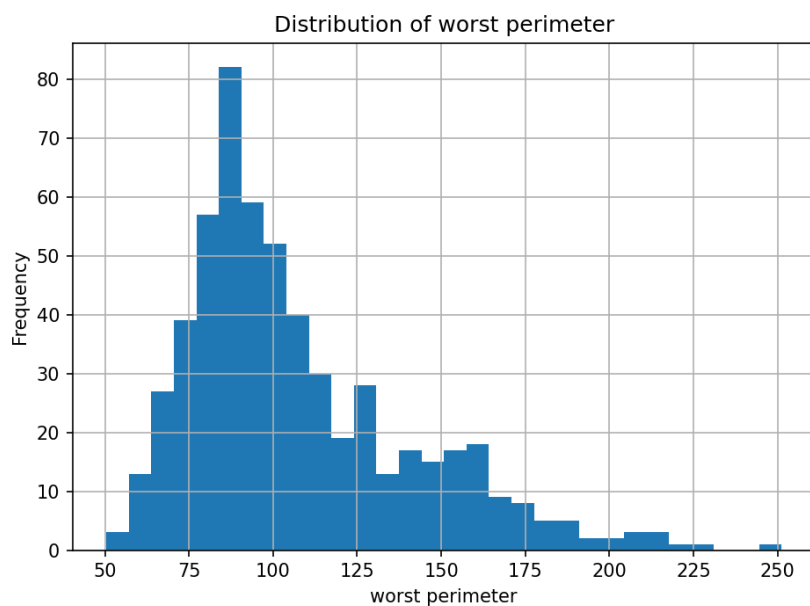


Figure 5: Histograma pentru atributul worst perimeter

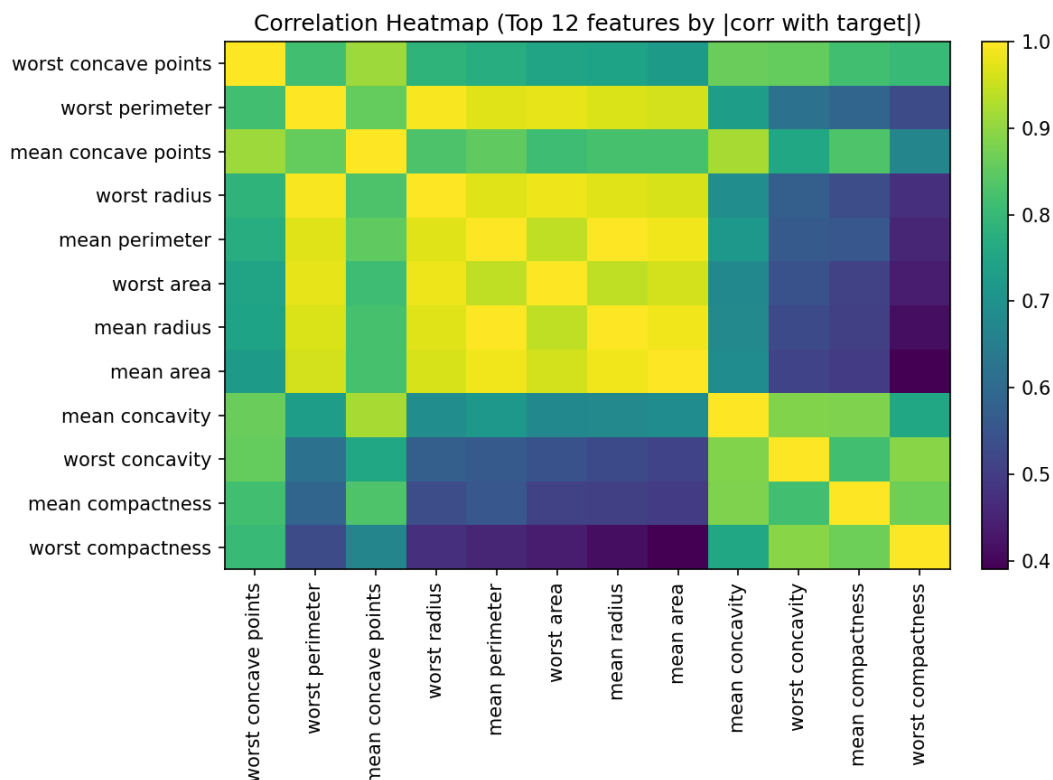


Figure 6: Heatmap de corelații pentru primele 12 caracteristici cu cea mai mare corelație absolută cu target

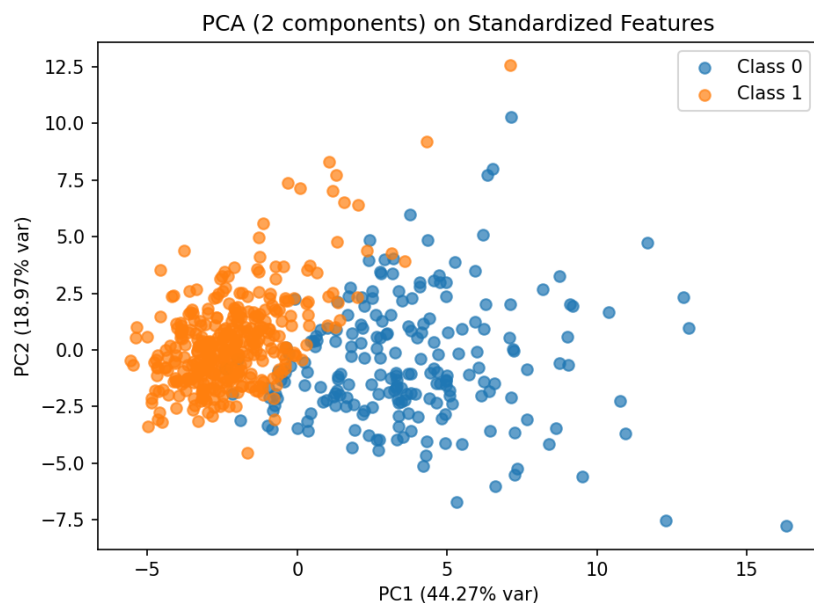


Figure 7: Scatter plot cu rezultatele aplicării PCA

Pentru a explora separabilitatea globală a claselor și pentru a vizualiza relațiile complexe dintre caracteristici, s-a aplicat Analiza Componentelor Principale (PCA) pe datele standardizate. PCA reduce dimensiunea setului de date într-un spațiu bidimensional, păstrând cât mai mult

din variația originală a datelor. Scatter plot-ul din Figura 7 a evidențiat o separare parțială între tumorile maligne și cele benigne, confirmând că setul de date conține informații suficiente pentru antrenarea unui model de clasificare. În plus, analiza componentelor principale oferă un cadru vizual intuitiv pentru detectarea grupărilor neobișnuite și a instanțelor atipice, contribuind la o înțelegere mai profundă a structurii datasetului.

Pentru a identifica caracteristicile care diferă cel mai mult între clase, s-au calculat mijloacele per clasă și diferențele absolute. Această analiză arată clar care atribute contribuie cel mai mult la separarea tumorilor maligne de cele benigne. Caracteristici precum *worst radius*, *mean concavity* și *worst perimeter* prezintă diferențe medii absolute semnificative, sugerând că acestea vor fi probabil cele mai informative în modelul de clasificare. Această etapă întărește concluziile preliminare obținute prin corelație și vizualizări.

## 3.2 Descrierea setului de date pentru diabet

Pentru dezvoltarea și evaluarea modelelor de predicție a diabetului, am utilizat un set de date denumit *Diabetino*. Acesta conține un număr de exemple etichetate, fiecare descris printr-un ansamblu de caracteristici numerice și binare. În total, setul de date include  $n$  instanțe și  $p$  variabile.

### 3.2.1 Structura datelor

Fiecare înregistrare reprezintă un pacient și este caracterizată de următoarele câmpuri:

- **qa, ps** – variabile binare asociate metadatelor inițiale (control de calitate, preprocesare).
- **ma0.5, ma0.6, ma0.7, ma0.8, ma0.9, ma1.0** – măsurători cantitative la praguri diferite ale parametrului *ma* (media mobilă cu fereastră variabilă).
- **ex0.5, ex0.6, ex0.7, ex0.8, ex0.9, ex1.0** – măsurători cantitative la praguri diferite pentru parametrul *ex*.
- **exm1, exm2** – valori derivate asociate parametrului *ex*.
- **eucmac** – o măsură numerică bazată pe distanța euclidiană între indicatori metabolici.
- **diaopt** – variabilă de tip scor, calculată ca o combinație ponderată a indicatorilor diagnostici.
- **amfm** – indicator numeric rezultat din analiza frecvenței și amplitudinii semnalelor metabolice.
- **dr** – eticheta asociată fiecărui exemplu; valoarea 1 indică prezența diabetului, iar valoarea 0 indică absența acestuia.

### 3.2.2 Exemplu de instanțe

În Tabelul 1 este prezentat un extras din primele cinci înregistrări din setul de date *Diabetino*.

qa	ps	ma0.5	ma0.6	ma0.7	ma0.8	ma0.9	ma1.0	ex0.5	ex0.6	eucmac	dr
1	1	22	22	22	19	18	14	49.89	17.77	0.4869	0
1	1	24	24	22	18	16	13	57.71	23.79	0.5209	0
1	1	62	60	59	54	47	33	55.83	27.99	0.5309	1
1	1	55	53	53	50	43	31	40.47	18.45	0.4833	0
1	1	44	44	44	41	39	27	18.03	8.570709	0.410381	0

Table 1: Exemplu de înregistrări din setul de date *Diabetino*.

### 3.2.3 Observații

Setul de date conține atât variabile numerice continue, cât și variabile binare, ceea ce permite testarea mai multor tipuri de modele de învățare automată. Eticheta **dr** este variabila țintă, utilizată pentru clasificarea pacienților în două clase: cu diabet și fără diabet.

Un aspect important este interpretabilitatea rezultatelor: prin analizarea coeficienților modelelor și a metodelor de explicabilitate (precum *SHAP* sau *LIME*), putem identifica variabilele cu impact major asupra diagnosticului.

### 3.2.4 Analiză exploratorie

În Figura 8 este prezentată distribuția variabilei țintă **dr**. Se observă un ușor dezechilibru între clase: **611 instanțe** sunt etichetate cu valoarea 1 (pacienți diagnosticați cu diabet), în timp ce **540 instanțe** sunt etichetate cu valoarea 0 (pacienți fără diabet).

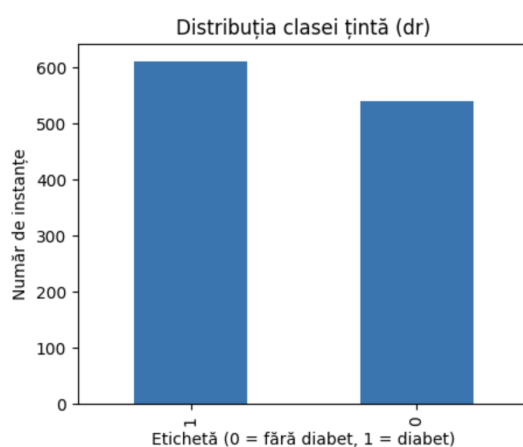


Figure 8: Distribuția claselor din setul de date cu informații despre diabet

Acest raport, de aproximativ 53% la 47%, indică un set de date relativ echilibrat, ceea ce reduce riscul ca modelele de clasificare să fie părtinitoare față de o anumită clasă.

În Figura 9 sunt reprezentate histogrammele pentru un subset de variabile numerice (*ma0.5*, *ma1.0*, *ex0.5*, *ex1.0* și *eucmac*).

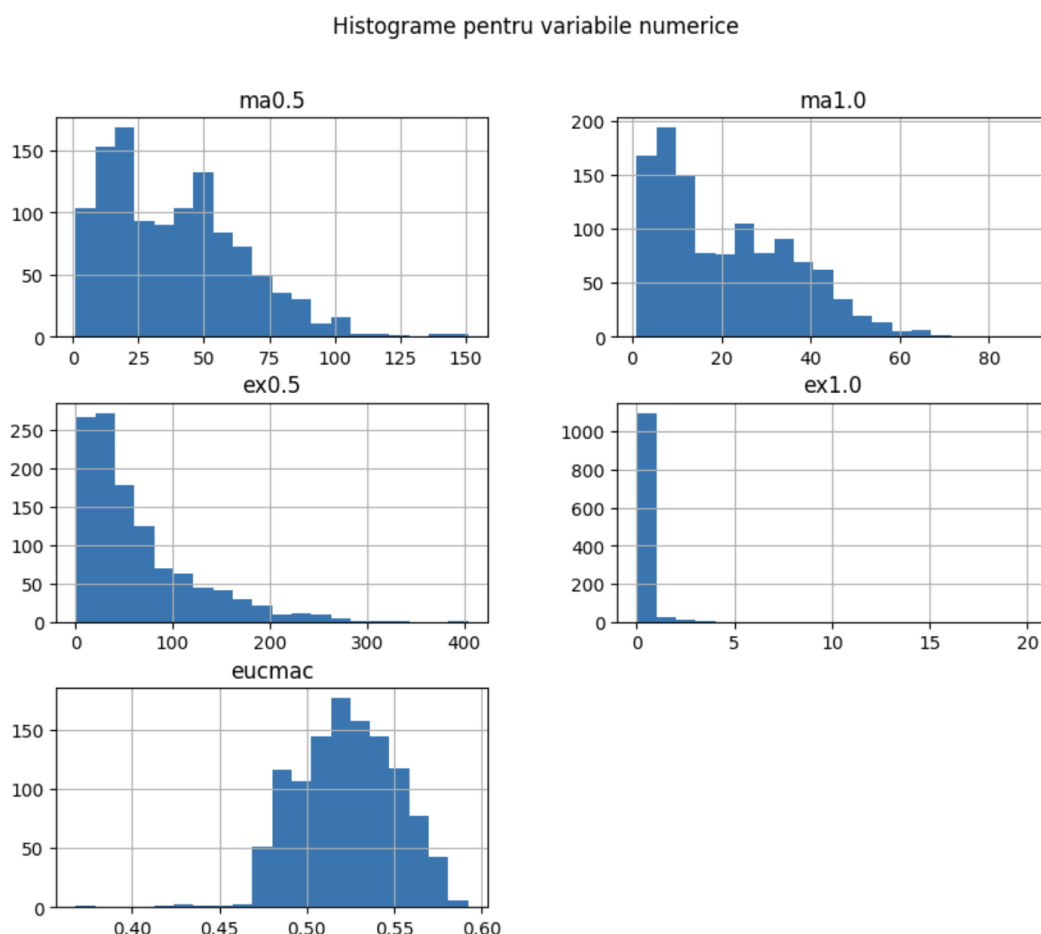


Figure 9: Histogramme pentru variabilele *ma0.5*, *ma1.0*, *ex0.5*, *ex1.0* și *eucmac*.

Analiza statistică evidențiază următoarele aspecte:

- Variabila **ma0.5** are valori cuprinse între **1** și **151**, cu o medie de **38.42** și o deviație standard de **25.62**. Distribuția este ușor dispersată, dar majoritatea valorilor se află între 16 și 55.
- Variabila **ma1.0** prezintă o medie de **21.15**, deviația standard fiind **15.10**. Valorile sunt între **1** și **89**, cu un interval intercuartilic cuprins între 8 și 32. Aceasta sugerează o concentrare a valorilor în jurul mediei, cu câteva valori extreme spre dreapta.
- Variabila **ex0.5** are o dispersie foarte mare, având un minim de **0.35**, un maxim de **403.93**, o medie de **64.09** și o deviație standard de **58.48**. Acest lucru indică o variabilitate ridicată între pacienți și prezența unor valori extreme.
- Variabila **ex1.0** este puternic asimetrică, cu valori între **0** și **20.09**. Deși media este doar **0.21**, deviația standard (**1.05**) arată că există un număr mic de valori mari care

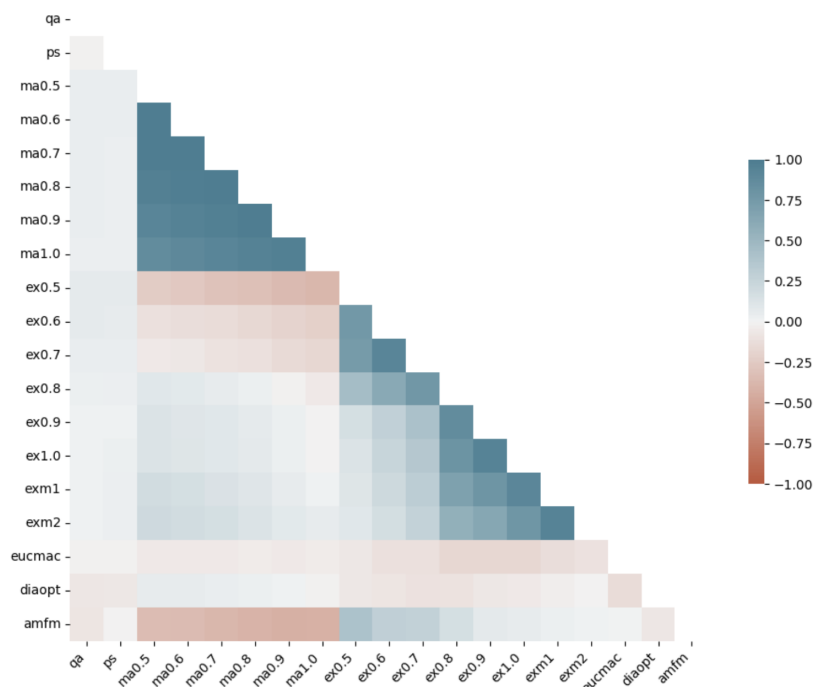


Figure 10: Matricea de corelații dintre variabilele setului de date, reprezentată sub formă de heatmap.

influențează distribuția.

- Variabila **eucmac** are un interval restrâns, cu valori între **0.36** și **0.59**, media fiind **0.52**. Distribuția este concentrată, ceea ce sugerează un indicator relativ stabil și robust.

Pentru a înțelege mai bine relațiile dintre variabilele din setul de date, am calculat matricea de corelații folosind coeficientul Pearson și am reprezentat rezultatele sub forma unei hărți de căldură (*heatmap*), prezentată în Figura 10.

Am ales să vizualizăm doar triunghiul inferior al matricei pentru a elimina redundanța și a facilita interpretarea. Reprezentarea grafică evidențiază gradul de asociere liniară dintre variabile: valorile apropiate de  $+1$  indică o corelație pozitivă puternică, cele apropiate de  $-1$  o corelație negativă, iar valorile din jurul lui 0 lipsa unei relații liniare semnificative.

Am realizat această analiză deoarece corelațiile dintre variabile pot influența performanța modelelor de învățare automată. Identificarea variabilelor puternic corelate este utilă atât pentru reducerea dimensionalității (prin eliminarea redundanțelor), cât și pentru explicabilitatea modelelor, permițând înțelegerea mai clară a rolului fiecărei variabile în predicție.

### 3.2.5 Optimizarea hiperparametrilor pentru Random Forest

Pentru a obține un model performant și robust, am utilizat o căutare exhaustivă în spațiul de hiperparametri prin intermediul metodei `GridSearchCV`. Această tehnică permite evaluarea sistematică a mai multor combinații de parametri pentru clasificatorul `RandomForest`,

selectând configurația care maximizează scorul mediu pe seturile de validare.

```
1 from sklearn.model_selection import GridSearchCV
2 from scipy.stats import randint
3 from sklearn.ensemble import RandomForestClassifier
4
5 # Initialize the RandomForestClassifier
6 rf = RandomForestClassifier(random_state=42)
7
8 # Define the hyperparameters grid
9 param_grid = {
10     'n_estimators': [15, 25, 50, 100, 200],
11     'max_depth': [None, 10, 20, 30],
12     'min_samples_split': [2, 10, 15, 20],
13     'min_samples_leaf': [1, 4, 8],
14     'bootstrap': [False]
15 }
16
17 # Initialize the GridSearchCV with training data only
18 grid_search = GridSearchCV(estimator=rf, param_grid=param_grid,
19                             cv=5, n_jobs=-1, verbose=1)
20
21 # Fit the grid search to the training data
22 grid_search.fit(x_train, y_train)
23
24 # Print the best parameters and best score from training data
25 print(f"Best Parameters: {grid_search.best_params_}")
26 print(f"Best Training Score: {grid_search.best_score_}")
```

Prin această abordare:

- am explorat mai multe valori posibile pentru **numărul de arbori** (*n\_estimators*), **adâncimea maximă** (*max\_depth*), **dimensiunea minimă a nodurilor** (*min\_samples\_split* și *min\_samples\_leaf*) și opțiunea de eșantionare bootstrap;
- am utilizat o validare încrucișată (**5-fold cross-validation**) pentru a evita supraînvățarea și pentru a obține o evaluare mai realistă a performanței;
- am ales **Grid Search** deoarece garantează testarea tuturor combinațiilor specificate, ceea ce este potrivit atunci când spațiul de parametri este relativ restrâns;
- scorul returnat de `grid_search.best_score_` reflectă performanța medie în timpul antrenării, iar `grid_search.best_params_` indică setul optim de hiperparametri care va fi utilizat pentru modelul final.

Motivația acestei etape este de a îmbunătăți precizia și stabilitatea modelului `RandomForest`, asigurându-ne că parametrii aleși nu sunt arbitrar selectați, ci rezultați în urma unei proceduri

sistematice de optimizare.

În urma rulării `GridSearchCV` cu validare încrucișată pe 5 folduri pentru fiecare dintre cei 240 de candidați (ai realizat astfel un total de 1200 de antrenări), modelul `RandomForest` a returnat cei mai buni hiperparametri și scorul asociat, după cum urmează:

- **bootstrap: False** – modelul nu utilizează eșantionare cu înlocuire la generarea arborilor, ceea ce poate ajuta la reducerea varianței într-un set relativ mic de date.
- **max\_depth: 10** – fiecare arbore poate avea maxim 10 niveluri, limitând complexitatea și prevenind supraînvățarea.
- **min\_samples\_leaf: 4** – fiecare frunză trebuie să conțină cel puțin 4 instanțe, ceea ce ajută la generalizare.
- **min\_samples\_split: 15** – un nod poate fi divizat doar dacă are cel puțin 15 instanțe, prevenind crearea de noduri foarte mici.
- **n\_estimators: 100** – modelul utilizează 100 de arbori, un compromis între performanță și costul de calcul.

Scorul mediu obținut pe datele de antrenament a fost **0.693**, indicând o performanță rezonabilă pentru această etapă inițială de modelare. Această valoare reflectă acuratețea medie a modelului pe foldurile de validare încrucișată.

Rezultatele obținute confirmă că procesul de *grid search* a permis identificarea unei configurații de hiperparametri care echilibrează complexitatea modelului și capacitatea sa de generalizare, fiind astfel potrivită pentru antrenarea modelului final pe întregul set de date.

### 3.2.6 Compararea mai multor modele de clasificare

Pentru a evalua performanța mai multor algoritmi de clasificare și a selecta modelul optim, am utilizat următoarele clasificatoare:

- **Random Forest (RF)**: un ansamblu de arbori de decizie, care utilizează medierea predicțiilor pentru a reduce varianța;
- **Gradient Boosting Classifier (GBC)**: un model secvențial de arbori care corectează erorile arborilor anteriori, crescând acuratețea;
- **XGBoost (XGB)**: un algoritm de boosting eficient, optimizat pentru viteză și performanță pe seturi mari de date;
- **Explainable Boosting Machine (EBM)**: un model interpretabil, bazat pe boosting, care permite explicarea contribuției fiecărei variabile la predicție;
- **TabNet (TBN)**: un model bazat pe rețele neuronale cu atenție, optimizat pentru date tabulare, cu capacitate ridicată de reprezentare.

Codul Python folosit pentru antrenarea și evaluarea acestor clasificatoare este prezentat mai jos:

```

1 from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
2 import xgboost as xgb
3 from interpret.glassbox import ExplainableBoostingClassifier
4 from pytorch_tabnet.tab_model import TabNetClassifier
5
6 # Inițializare clasificatoare
7 rf = RandomForestClassifier(n_estimators=100, n_jobs=-1,
8                             random_state=random_state)
9 gbc = GradientBoostingClassifier(random_state=random_state)
10 _xgb = xgb.XGBClassifier(random_state=random_state)
11 ebm = ExplainableBoostingClassifier(random_state=random_state)
12 tbn = TabNetClassifier(verbose=0, seed=random_state)
13
14 # Listă de clasificatoare împreună cu eticheta și flag pentru explicabilitate
15 clfs = [(rf, 'rf', True), (gbc, 'gbc', True), (_xgb, 'xgb', True),
16         (ebm, 'ebm', True), (tbn, 'tbn', False)]
17
18 # Datele de antrenament și test
19 dataset = x_train, y_train, x_test, y_test
20
21 # Funcția care antrenează fiecare model și generează rapoarte de clasificare
22 classify_report(clfs, dataset)

```

## Explicații și motivație

Scopul acestei etape este de a compara performanța mai multor modele de clasificare pe același set de date. Am inclus atât algoritmi clasici (RF, GBC, XGB), cât și modele interpretabile (EBM) și deep learning tabular (TabNet), pentru a evalua:

- Precizia și robustețea modelelor clasice comparativ cu cele moderne;
- Capacitatea modelelor interpretabile (RF, GBC, XGB, EBM) de a oferi insight-uri asupra contribuției fiecărei variabile;
- Eficiența modelelor de tip deep learning (TabNet) pe date tabulare fără preprocesare extinsă.

Flag-ul True/False asociat fiecărui clasificator indică dacă modelul este compatibil cu metode de explicabilitate (SHAP, LIME, EBM interpretabil). Această analiză permite atât compararea performanței, cât și integrarea unor metode explicabile pentru a înțelege deciziile modelelor.

## Rezultate și evaluare a modelelor

După antrenarea și testarea clasificatoarelor pe setul de date, am obținut următoarele rezultate de performanță, prezentate în Tabelul 2:

Name	F1-score	Accuracy
Random Forest (rf)	0.6819	0.6792
TabNet (tbn)	0.7019	0.5434
XGBoost (xgb)	0.7075	0.6965
Explainable Boosting Machine (ebm)	0.7268	0.7197
Gradient Boosting Classifier (gbc)	0.7273	0.7139

Table 2: Performanța clasificatoarelor pe setul de test.

## Interpretarea rezultatelor

Din tabelul de mai sus observăm următoarele:

- **Gradient Boosting Classifier (GBC)** și **EBM** au obținut cele mai bune valori de F1-score (aproximativ 0.727 și 0.727 respectiv) și acuratețe, demonstrând performanță superioară pe setul de date.
- **XGBoost** are performanțe bune, cu F1-score de 0.708 și acuratețe de 0.697, fiind ușor sub GBC și EBM.
- **Random Forest** are rezultate moderate, F1-score de 0.682 și acuratețe de 0.679, indicând că ansamblul de arbori fără optimizare extinsă nu este la fel de performant.
- **TabNet** prezintă un F1-score rezonabil (0.702), însă acuratețea mai scăzută (0.543) sugerează că modelul deep learning necesită mai multe date sau preprocesare suplimentară pentru a generaliza corespunzător.

Aceste rezultate confirmă că:

1. Modelele de boosting (GBC, EBM, XGB) oferă o performanță mai bună decât modelele clasice de tip bagging (RF) în acest context.
2. Modelele interpretabile, cum este EBM, pot atinge performanțe comparabile cu GBC, oferind în același timp posibilitatea de a explica deciziile.
3. Modelele bazate pe deep learning tabular (TabNet) pot avea nevoie de seturi mai mari de date sau de ajustări de hiperparametri pentru a concura cu modelele de boosting pe date medii.

În continuare, analiza de explicabilitate va fi aplicată în special pe modelele EBM și GBC, pentru a identifica variabilele cu cel mai mare impact asupra predicțiilor.

### 3.2.7 Explicabilitatea modelului folosind LIME

Pentru a înțelege deciziile modelului Random Forest (`selected_model = rf`), am utilizat metoda **LIME (Local Interpretable Model-agnostic Explanations)**. Aceasta permite explicarea predicțiilor pentru instanțe individuale, identificând contribuția fiecărei variabile la decizia modelului.

```
1  from lime import lime_tabular
2
3  # Initializare explainer pentru date tabulare
4  exp_lime = lime_tabular.LimeTabularExplainer(
5      np.array(x_train),
6      feature_names=x_train.columns,
7      class_names=['NO', 'YES'],
8      mode='classification'
9  )
10
11 # Funcție pentru generarea explicațiilor pentru un batch de instanțe
12 def lime_explain_instance_step(i, lim):
13     warnings.filterwarnings("ignore")
14     out = []
15     for k in range(i, lim):
16         e = exp_lime.explain_instance(x_test.values[k], selected_model.predict_proba)
17         out.append(e)
18     return out
19
20 # Generare explicații în paralel pentru toate instanțele de test
21 total = len(x_test)
22 out = Parallel(n_jobs=n_jobs)(
23     delayed(lime_explain_instance_step)(
24         i, min(total, i+int(total/n_jobs))
25     ) for i in range(0, total, int(total/n_jobs))
26 )
27 lime_explanations_list = list(itertools.chain(*out))
28
29 # Vizualizare câteva instanțe și etichetele reale
30 display(x_test.iloc[2:7])
31 display(y_test.iloc[2:7])
```

## Explicații și motivație

- Am ales **LIME** deoarece oferă explicații locale: pentru fiecare instanță de test, identifică variabilele care au influențat cel mai mult predicția modelului.
- Inițial, am creat un obiect `LimeTabularExplainer`, specificând datele de antrenament, numele variabilelor și clasele (*NO* pentru lipsa diabetului, *YES* pentru prezența diabetului).
- Funcția `lime_explain_instance_step` permite generarea explicațiilor pentru un batch de instanțe, evitând calculul secvențial care ar fi mai lent.
- Folosirea **paralelizării** cu `joblib.Parallel` accelerează procesul pentru întregul set de test.
- Rezultatele sunt stocate în `lime_explanations_list`, fiecare element fiind un obiect care poate fi vizualizat grafic sau numeric pentru a interpreta contribuția fiecărei variabile.
- Prin afișarea câtorva instanțe din `x_test` și `y_test` se pot corela explicațiile locale cu etichetele reale, oferind insight-uri asupra modului în care modelul ia deciziile.

Această analiză de explicabilitate este esențială pentru a justifica deciziile modelului și pentru a crește încrederea în predicțiile sale, mai ales în contextul aplicațiilor medicale.

Pentru a ilustra modul în care modelul Random Forest ia decizii, am analizat două instanțe de test:

qa	ps	ma0.5	ma0.6	ma0.7	ma0.8	ma0.9	ma1.0	ex0.5	ex1.0	eucmac
1	1	44	44	44	44	42	29	25.28	0.185	0.5005
1	1	47	44	41	39	34	24	137.88	2.66	0.4878

Table 3: Valorile principale ale variabilelor pentru două instanțe de test.

Etichetele reale pentru aceste instanțe sunt:

- Instanța 328: **0** (fără diabet)
- Instanța 692: **1** (diabet)

### 3.2.8 Exemplu de interpretare LIME pentru o instanță

Pentru instanța 328, modelul Random Forest a returnat următoarele probabilități pentru clase:

- **NO (fără diabet): 0.48**
- **YES (diabet): 0.52**

Predicția finală a fost **YES**, ceea ce indică o decizie aproape echilibrată, cu probabilități foarte apropiate.

Valorile variabilelor pentru această instanță sunt prezentate în Tabelul 4:

Feature	Value
exm1	0.00
exm2	0.00
ma0.6	44.00
ma1.0	29.00
ex0.5	25.28
ma0.5	44.00
ex0.7	1.64
ex0.9	0.00
ma0.8	44.00
ex1.0	0.00

Table 4: Valorile variabilelor pentru instanța 328.

## Interpretarea explicațiilor LIME

LIME a identificat contribuțiile locale ale variabilelor la predicția modelului, cu cele mai relevante caracteristici:

- $\text{exm1} \leq 0.00$  contribuie pozitiv la clasa *YES* cu 0.06;
- $\text{exm2} \leq 0.00$  contribuie pozitiv cu 0.03;
- Intervalul  $34.00 < \text{ma0.6} \leq 44.00$  contribuie pozitiv cu 0.03;
- Intervalul  $17.00 < \text{ma1.0} \leq 29.00$  contribuie pozitiv cu 0.03;
- Intervalul  $22.03 < \text{ex0.5} \leq 25.28$  contribuie pozitiv cu 0.02;
- Alte variabile ( $\text{ma0.5}$ ,  $\text{ex0.7}$ ,  $\text{ex0.9}$ ,  $\text{ma0.8}$ ,  $\text{ex1.0}$ ) au contribuții mai mici, între 0.01 și 0.02.

Aceasta arată că predicția aproape egală între clase a fost influențată în principal de valorile foarte mici ale  $\text{exm1}$  și  $\text{exm2}$ , precum și de intervalele mai mari pentru  $\text{ma0.6}$  și  $\text{ma1.0}$ . Astfel, analiza LIME permite interpretarea modelului la nivel individual, evidențiind variabilele cu impact major asupra deciziei.

Pentru a înțelege modul în care modelul Random Forest a luat decizia pentru instanța 328, am utilizat metoda LIME (Local Interpretable Model-agnostic Explanations), care evidențiază contribuția fiecărei variabile la predicția finală. Valorile reale ale caracteristicilor, împreună cu probabilitățile estimate pentru clasele *NO* și *YES*, sunt prezentate vizual în Figura 11. Această reprezentare permite observarea clară a variabilelor cu cel mai mare impact asupra deciziei modelului și facilitează interpretarea.

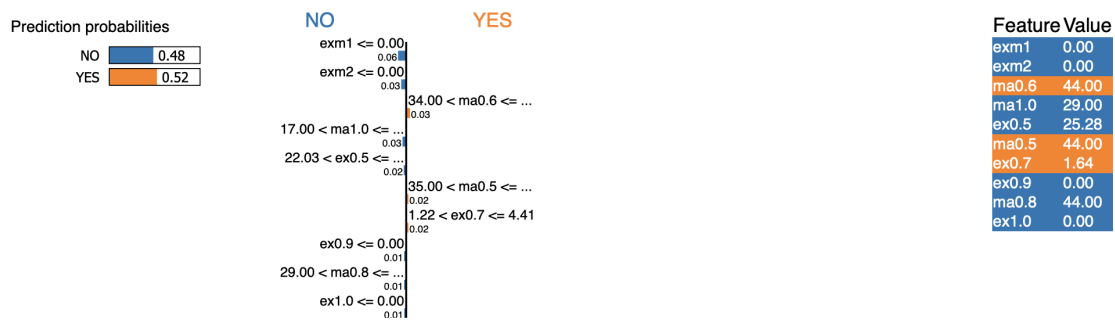


Figure 11: Explicația locală a predicției modelului Random Forest pentru instanța 328 utilizând LIME.

### 3.2.9 Explicabilitatea modelului folosind SHAP

Pentru a analiza contribuția variabilelor la predicțiile modelului Random Forest la nivel global și local, am utilizat metoda **SHAP (SHapley Additive exPlanations)**. Aceasta oferă o estimare a impactului fiecărei caracteristici asupra predicțiilor modelului, pe baza conceptelor de valoare Shapley din teoria jocurilor.

```

1 import shap
2
3 # Inițializare interactivă pentru vizualizări
4 shap.plots.initjs()
5
6 # Crearea unui explainer pentru modelul Random Forest
7 exp_shap = shap.TreeExplainer(selected_model)
8
9 # Calcularea valorilor SHAP pentru setul de test
10 shap_values = exp_shap.shap_values(x_test)
11
12 # Generarea unei vizualizări force plot pentru o instanță specifică
13 shap_html = shap.force_plot(
14     exp_shap.expected_value[0],
15     shap_values[id_to_explain][:, 0],
16     x_test.iloc[id_to_explain],
17     matplotlib=True
18 )

```

### Explicații și motivație

- `shap.TreeExplainer(selected_model)` creează un explainer specializat pentru modelele bazate pe arbori de decizie (Random Forest, XGBoost, etc.), optimizat pentru viteza de calcul.

- `shap_values = exp_shap.shap_values(x_test)` calculează pentru fiecare instanță și fiecare caracteristică contribuția acesteia la predicția modelului.
- `shap.force_plot` generează un grafic interactiv sau static care vizualizează modul în care fiecare caracteristică a influențat predicția finală pentru o instanță individuală (`id_to_explain`).
- Valorile SHAP permit identificarea variabilelor cu impact pozitiv sau negativ asupra predicției, oferind o interpretabilitate detaliată și intuitivă.

Această metodă completează analiza LIME, oferind o vedere globală și locală asupra importanței caracteristicilor și explicând deciziile modelului într-un mod ușor de interpretat, esențial pentru aplicațiile medicale în care transparența este critică.

Rezultatele analizelor de explicabilitate SHAP sunt prezentate vizual în Figura 12. Aceasta arată modul în care fiecare variabilă a influențat predicția modelului pentru instanța selectată, evidențiind contribuțiile pozitive și negative și facilitând interpretarea deciziei modelului într-un mod intuitiv și detaliat.

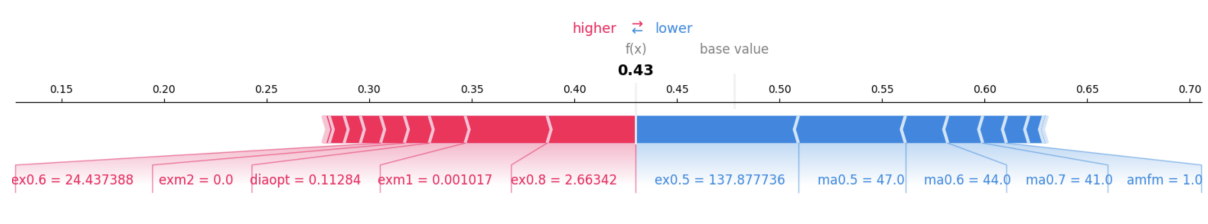


Figure 12: Vizualizarea contribuțiilor variabilelor la predicția modelului Random Forest pentru o instanță specifică, folosind metoda SHAP.

### 3.2.10 Vizualizarea detaliată a contribuțiilor variabilelor folosind SHAP Waterfall

Pentru o interpretare mai detaliată a predicției modelului Random Forest pentru o instanță individuală, am folosit **SHAP Waterfall plot**. Aceasta permite vizualizarea pas cu pas a modului în care fiecare caracteristică influențează predicția finală.

```

1 id_to_explain = 9
2 output_to_explain = 0
3
4 # Vizualizare waterfall pentru instanța specifică
5 shap.plots.waterfall(
6     exp_shap_values[id_to_explain, :, output_to_explain],
7     max_display=10
8 )

```

- `id_to_explain = 9` indică instanța de test analizată.
- `output_to_explain = 0` specifică clasa pentru care se vizualizează contribuțiile (de exemplu, clasa *NO* sau *YES*).
- `shap.plots.waterfall` afișează cum fiecare caracteristică împinge predicția de la valoarea de bază (`expected_value`) către probabilitatea finală, prezentând efectele pozitive și negative în ordine descrescătoare a impactului.
- Parametrul `max_display=10` limitează afișarea la cele 10 variabile cu cel mai mare impact, pentru claritate și lizibilitate.

Această vizualizare permite observarea detaliată a contribuțiilor individuale ale variabilelor și facilitează interpretarea deciziei modelului pentru o instanță specifică, complementând analiza SHAP Force plot și LIME. Rezultatele sunt prezentate vizual în Figura 13.

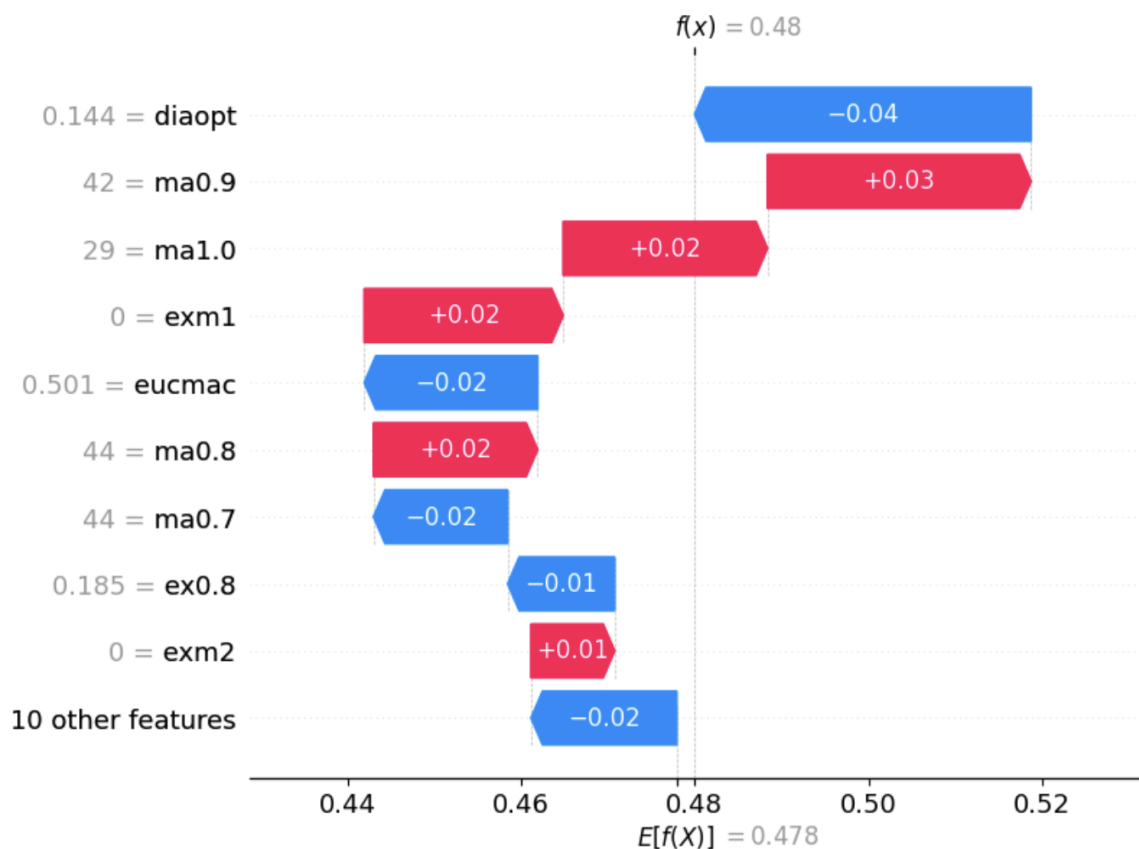


Figure 13: Contribuția fiecărei variabile la predicția modelului Random Forest pentru instanța 9, vizualizată cu SHAP Waterfall.

### **3.3 Corpus**

#### **3.3.1 Descriptives**

### **3.4 (Neural) Architecture**

### **3.5 Performance Metrics**

## **4 REZULTATE EXPERIMENTALE**

## **5 DISCUȚII**

### **5.1 Performance Comparison**

### **5.2 Limitations**

## **6 CONCLUZII ȘI DEZVOLTĂRI ULTERIOARE**

## REFERENCES

- Khater, T., Hussain, A., Bendardaf, R., Talaat, I. M., Tawfik, H., Ansari, S., & Mahmoud, S. (2023). An explainable artificial intelligence model for the classification of breast cancer. *IEEE Access*.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.
- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning* (pp. 3319–3328).