Învățare Automată

Tema 1 – 2024

1. Analiza setului de date

După ce am analizat setul de date primit, am decis ca este esențial să realizăm operații de filtrare asupra datelor non-relevante pentru studiul nostru.

Am realizat următoarele schimbări pentru valorile din setul inițial:

1. Regular_fiber_diet – am eliminat valorile mai mari decât 10.
2. Sedentary_hours_daily – am schimbat numerele din folosirea “,” in folosirea “.” + am eliminat o valoare care depășeau valoarea maximă pentru acest atribut (956 ore > 24 ore).
3. Age – am eliminat numerele mai mari decât 150.
4. Main_meals_daily – am eliminat valorile mai mari decât 10.
5. Height – am schimbat numerele din folosirea “,” in folosirea “.” + am eliminat două valori care depășeau cu mult o înălțime “normală” (1915 m).
6. Water_daily – am eliminat valorile mai mari decât 10.
7. Weight – am eliminat valorile mai mari decât 300 și valorile de -1.
8. Physical_activity_level – am eliminat valorile mai mari decât 10.

Before

683	Automobil	2,341,133	no	yes	3,9	2,669,858	no	2327	1,578,521	Sometime	1,82	no	2	no	-1	2	1	Male	D2
684	Public_Tra	2	no	yes	2,49	21,125,836	Sometime	2893	1	no	1,64	no	2,115,967	no	70	0,770536	0	Male	D2
685	Automobil	1,450,218	no	yes	3,01	25,191,627	no	2893	3,985,442	Sometime	1,81	no	2,147,746	no	85,637,789	0,046836	1	Male	D2
686	Public_Tra	2,204,914	yes	yes	2,43	21,963,457	Sometime	2758	3,623,364	Sometime	1,7	no	1,815,293	no	755,771	0,989316	0	Male	D2
687	Public_Tra	1,206,276	no	yes	3,06	21	Sometime	2817	3	no	1,62	no	2,406,541	no	68,869,791	0,94984	0	Male	D2
688	Public_Tra	2	yes	yes	2,51	19,114,981	Sometime	1794	3	Sometime	1,86	no	1,015,677	no	88,965,521	0	0	Male	D2
689	Automobil	281,646	no	yes	2,62	41,823,567	Sometime	2918	336,313	Sometime	1,72	no	2,722,063	no	82,919,584	3	0	Female	D2
690	Public_Tra	2	yes	yes	3,13	21,142,432	Sometime	1771	3	Sometime	1,86	no	1,345,298	no	86,413,388	########	1	Male	D2
691	Public_Tra	2	yes	yes	2,27	21,962,426	Sometime	1613	3	Sometime	1,7	no	1,825,629	no	75	0,699592	0	Male	D2
692	Public_Tra	2	yes	yes	3,74	18,836,315	Frequently	2593	173,762	Sometime	1,75	no	2,207,978	no	80	########	1	Female	D2
693	Automobil	3	yes	no	3,2	33,700,749	Sometime	2407	1,146,052	Sometime	1,64	no	1,074,048	no	74,803,157	0,679935	0	Female	D2
694	Public_Tra	1,758,394	yes	yes	3,94	21,845,025	Sometime	2030	3,981,997	Sometime	1,61	no	2,174,248	no	68,126,955	0,920476	1	Female	D2
695	Public_Tra	2	no	yes	2,82	21	Sometime	2429	19,154	no	1,61	no	3	no	68,226,511	1	0	Male	D2
696	Automobil	2,392,665	yes	yes	3,2	36,769,646	Sometime	2580	3	Sometime	1,55	no	2,951,056	no	62,337,721	0	0	Female	D2
697	Public_Tra	2,577,427	yes	yes	3,15	21,868,932	Sometime	1546	1	Sometime	1,73	no	2	no	78,175,706	########	0	Female	D2
698	Public_Tra	2	no	yes	2,55	22,549,208	Sometime	2991	1	no	1,63	no	2,803,311	no	70	0,245354	0	Male	D2
699	Public_Tra	2	no	yes	3,09	21	Sometime	2487	1	no	1,62	no	3	no	70	1	0	Male	D2
700	Public_Tra	2	no	yes	2,46	21	Sometime	1594	1	no	1,62	no	3	no	70	1	0	Male	D2

After

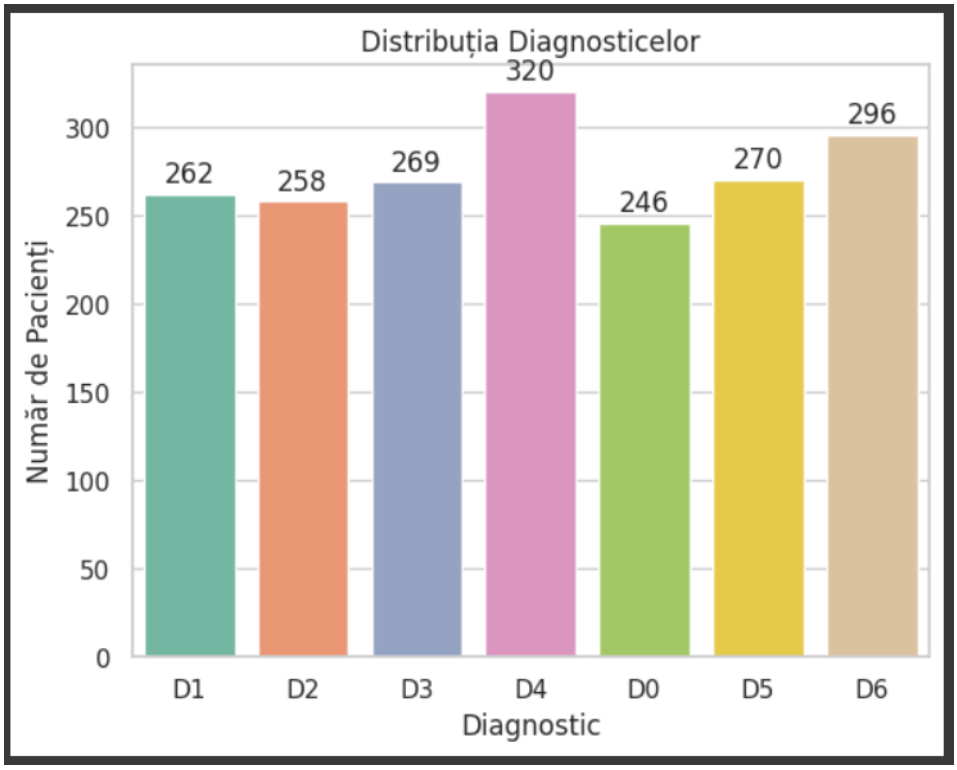
683	Automobile		no	yes	3.9		no	2327		Sometimes	1.82	no	2	no		2	1	Male	D2
684	Public_Tra	2	no	yes	2.49		Sometimes	2893	1	no	1.64	no		no	70		0	Male	D2
685	Automobile		no	yes	3.01		no	2893		Sometimes	1.81	no		no			1	Male	D2
686	Public_Transportation		yes	yes	2.43		Sometimes	2758		Sometimes	1.7	no		no			0	Male	D2
687	Public_Transportation		no	yes	3.06	21	Sometimes	2817	3	no	1.62	no		no			0	Male	D2
688	Public_Tra	2	yes	yes	2.51		Sometimes	1794	3	Sometimes	1.86	no		no		0	0	Male	D2
689	Automobile		no	yes	2.62		Sometimes	2918		Sometimes	1.72	no		no		3	0	Female	D2
690	Public_Tra	2	yes	yes	3.13		Sometimes	1771	3	Sometimes	1.86	no		no			1	Male	D2
691	Public_Tra	2	yes	yes	2.27		Sometimes	1613	3	Sometimes	1.7	no		no	75		0	Male	D2
692	Public_Tra	2	yes	yes	3.74		Frequently	2593		Sometimes	1.75	no		no	80		1	Female	D2
693	Automobile	3	yes	no	3.2		Sometimes	2407		Sometimes	1.64	no		no			0	Female	D2
694	Public_Transportation		yes	yes	3.94		Sometimes	2030		Sometimes	1.61	no		no			1	Female	D2
695	Public_Tra	2	no	yes	2.82	21	Sometimes	2429		no	1.61	no	3	no		1	0	Male	D2
696	Automobile		yes	yes	3.2		Sometimes	2580	3	Sometimes	1.55	no		no		0	0	Female	D2
697	Public_Transportation		yes	yes	3.15		Sometimes	1546	1	Sometimes	1.73	no	2	no			0	Female	D2
698	Public_Tra	2	no	yes	2.55		Sometimes	2991	1	no	1.63	no		no	70		0	Male	D2
699	Public_Tra	2	no	yes	3.09	21	Sometimes	2487	1	no	1.62	no	3	no	70	1	0	Male	D2
700	Public_Tra	2	no	yes	2.46	21	Sometimes	1594	1	no	1.62	no	3	no	70	1	0	Male	D2

În urma schimbărilor, am generat un set nou de date: cleaned_date_tema_1_iaut_2024.csv

Noul set de date poate fi folosit pentru analiza atributelor, clasei și relațiilor dintre acestea.

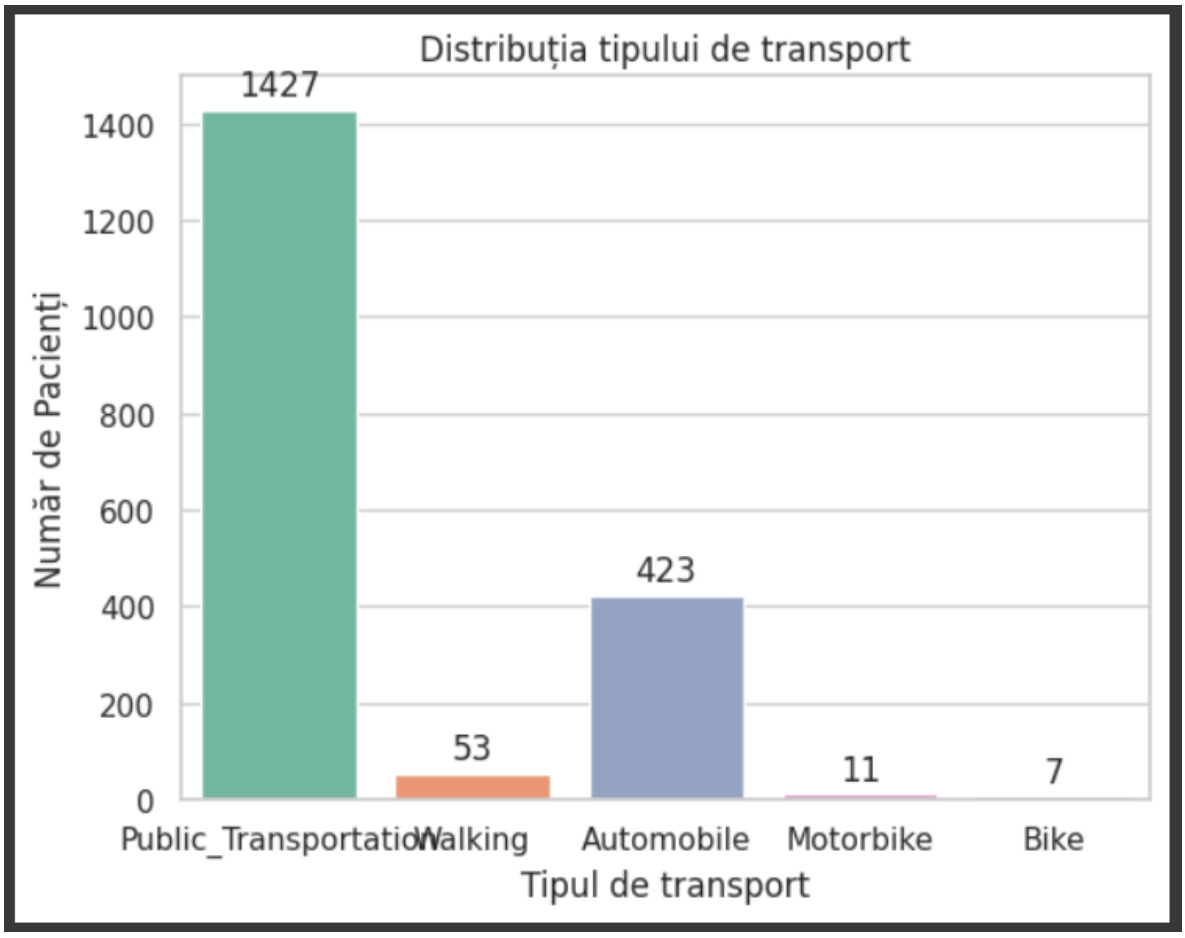
2. Relații în setul de date

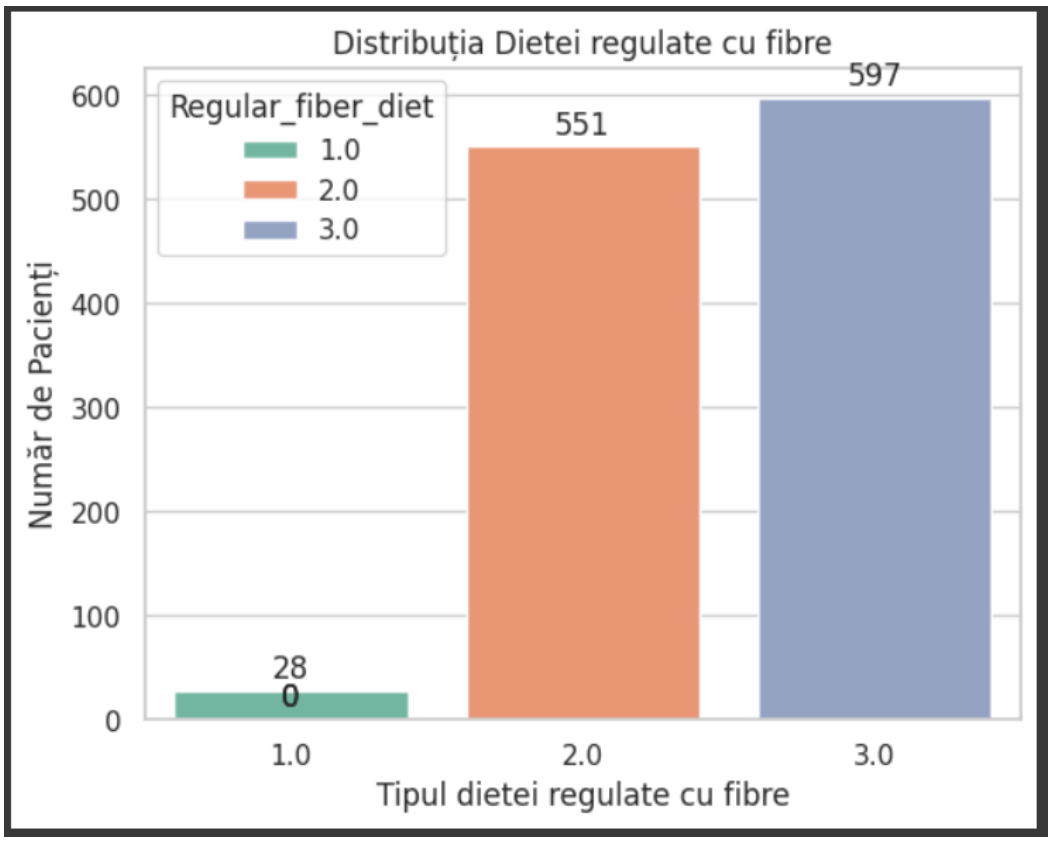
2.1. Analiza echilibrului de date

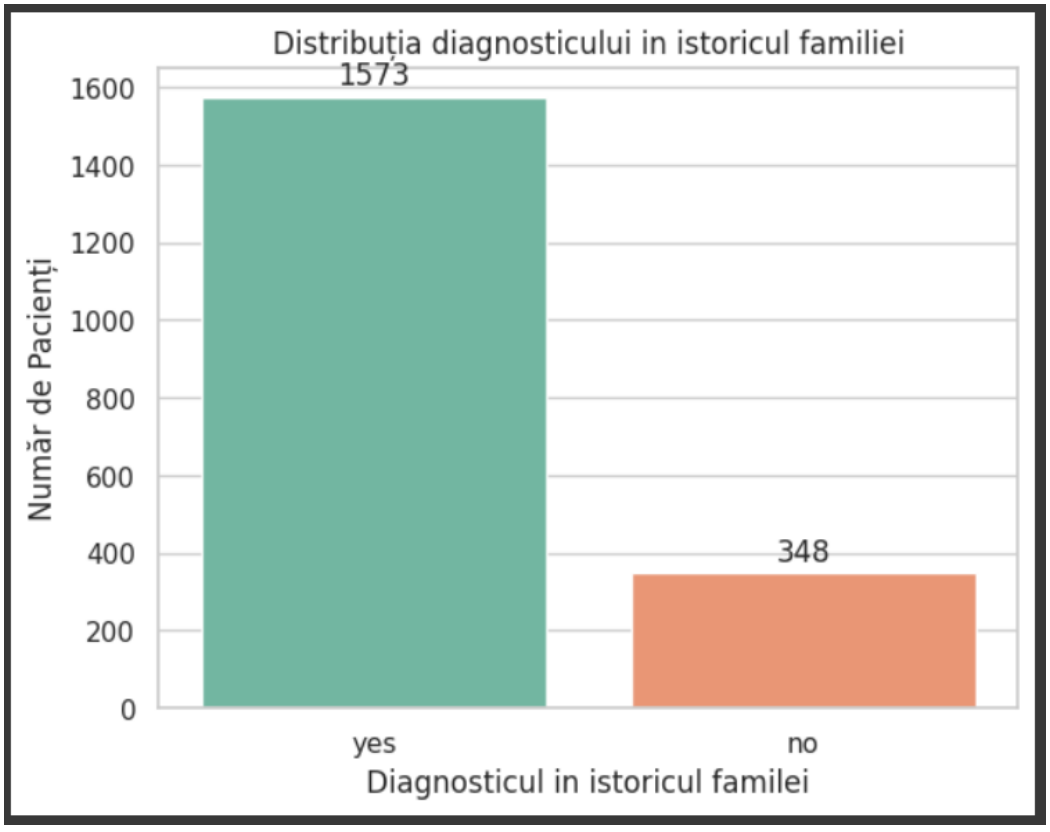


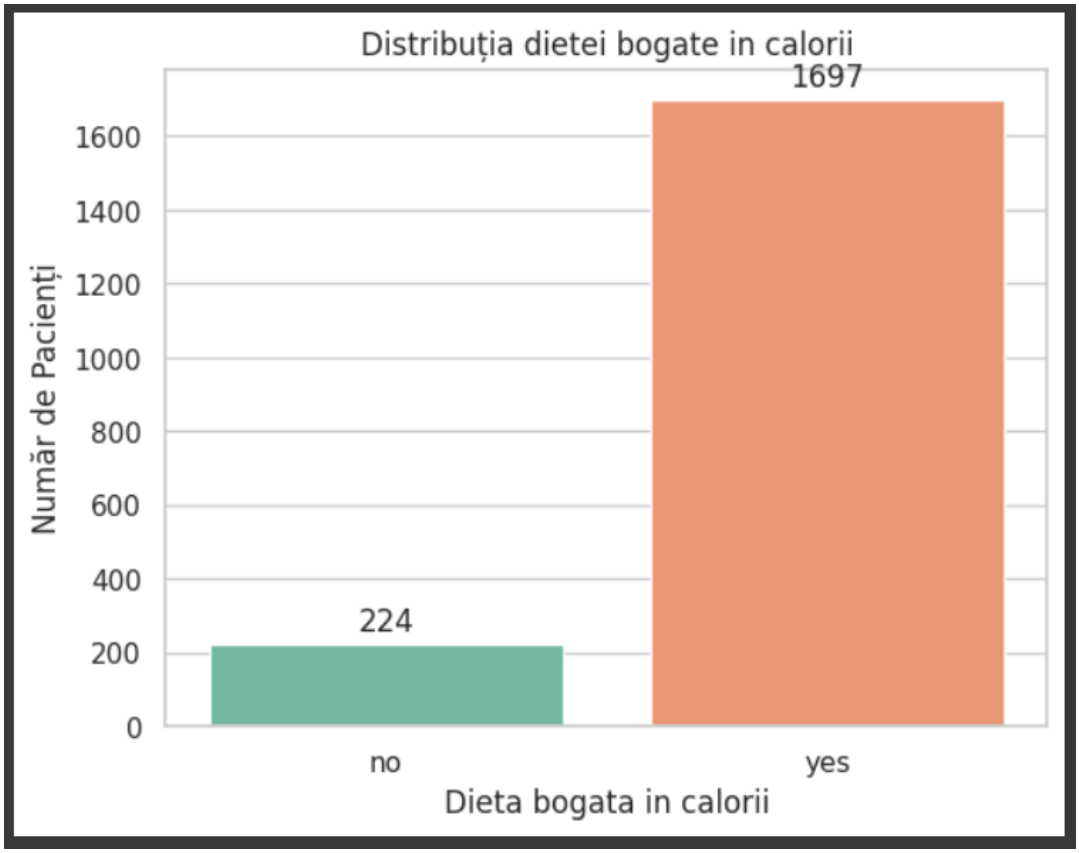
Setul de date pare să fie echilibrat, cu o mică abatere pentru D4 și D6 care prezintă mai mulți pacienți.

2.2. Vizualizarea datelor



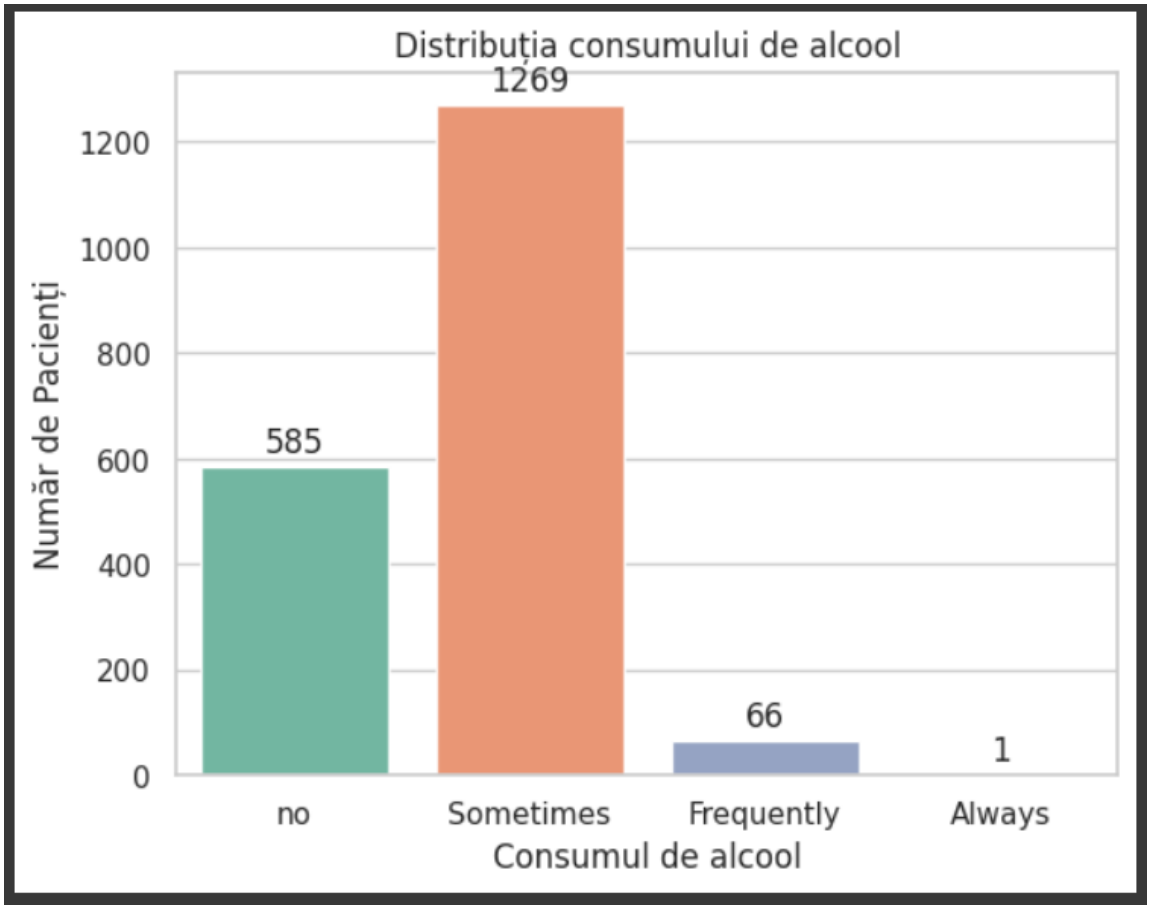

Distribuția Dietei regulate cu fibre
Regular_fiber_diet
1.0
2.0
3.0
Număr de Pacienți
600
500
400
300
200
100
0
28
0
551
597
1.0
2.0
3.0
Tipul dietei regulate cu fibre


Distribuția diagnosticului in istoricul familiei
Număr de Pacienți
1600
1400
1200
1000
800
600
400
200
0
1573
348
yes
no
Diagnosticul in istoricul familei

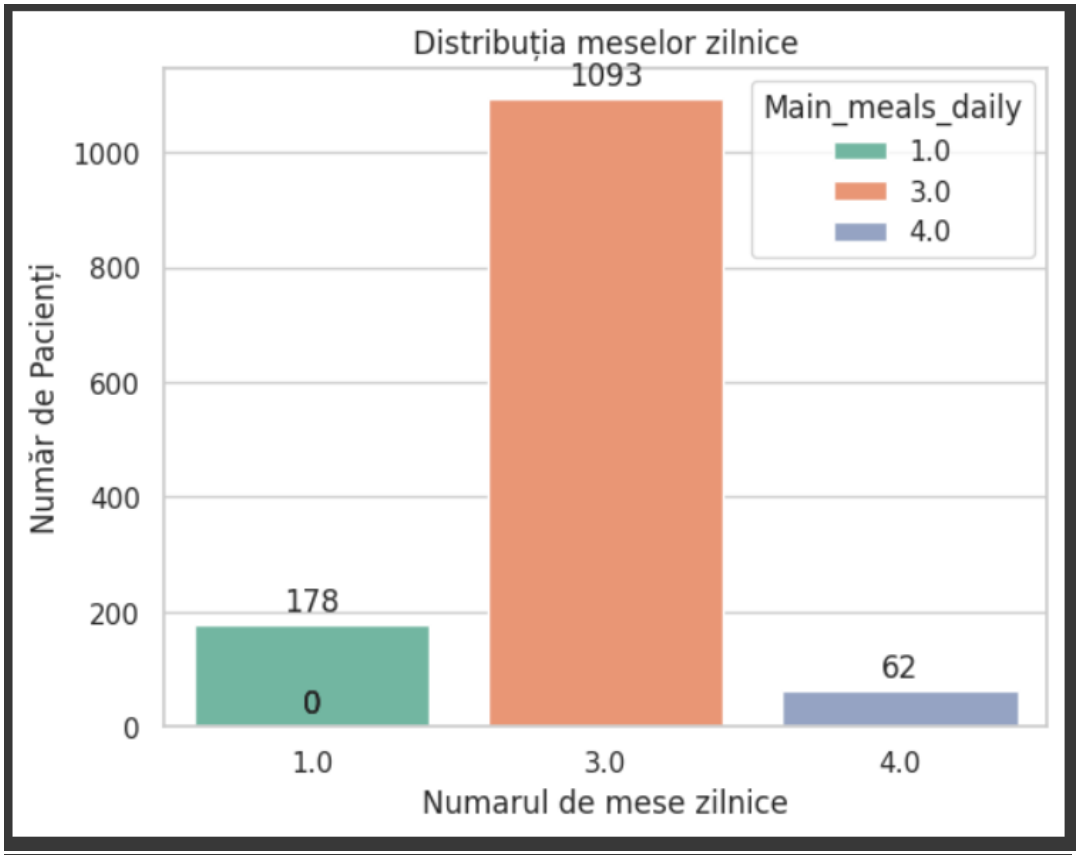


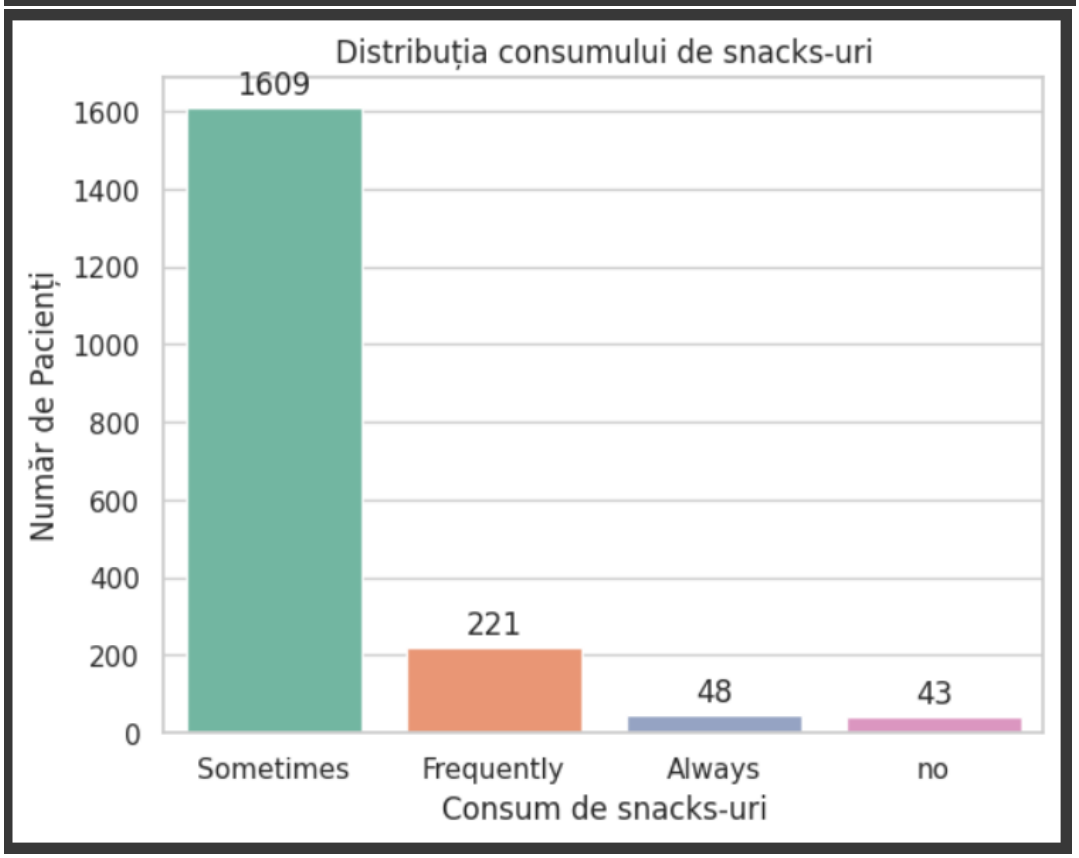

```
Valorile inregistrate pentru atributul Sedentary_hours_daily:
Valoarea medie:  3.197276041666667
Valoarea maximă:  4.67
Valoarea minimă:  2.21
Valoarea medianei:  3.13
Abaterea standard:  0.5756056835783199
Abaterea medie absolută:  0.47053131507027596
Abaterea mediană absolută:  0.43999999999999995
```

```
Valorile inregistrate pentru atributul Age:
Valoarea medie:  23.121076233183857
Valoarea maximă:  61.0
Valoarea minimă:  15.0
Valoarea medianei:  21.0
Abaterea standard:  6.122892259253898
Abaterea medie absolută:  4.217056445936979
```

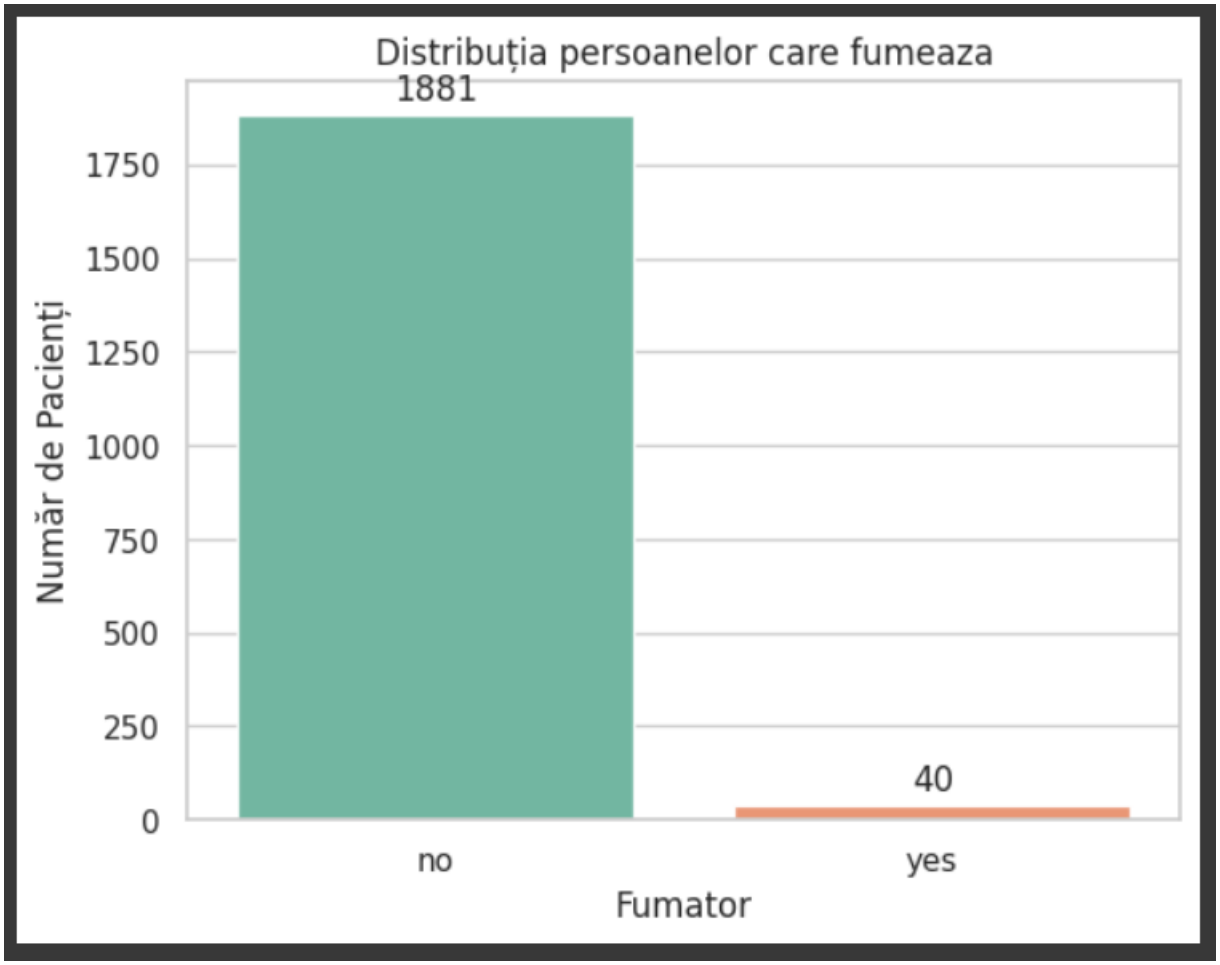


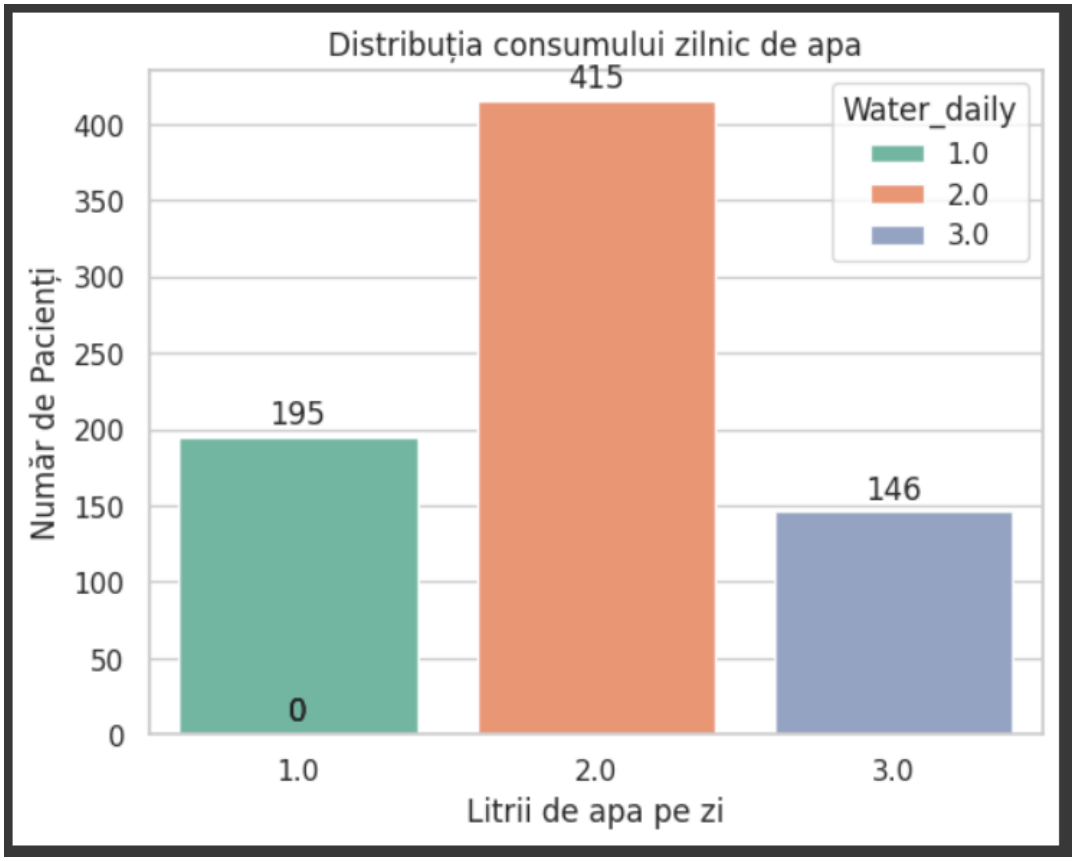

```
Valorile inregistrate pentru atributul Est_avg_calorie_intake:
Valoarea medie:  2253.68766267569
Valoarea maximă:  3000
Valoarea minimă:  1500
Valoarea medianei:  2253.0
Abaterea standard:  434.07579419142866
Abaterea medie absolută:  375.36234408538627
Abaterea mediană absolută:  380.0
Intervalul intercuartil (IQR):  757.0
```

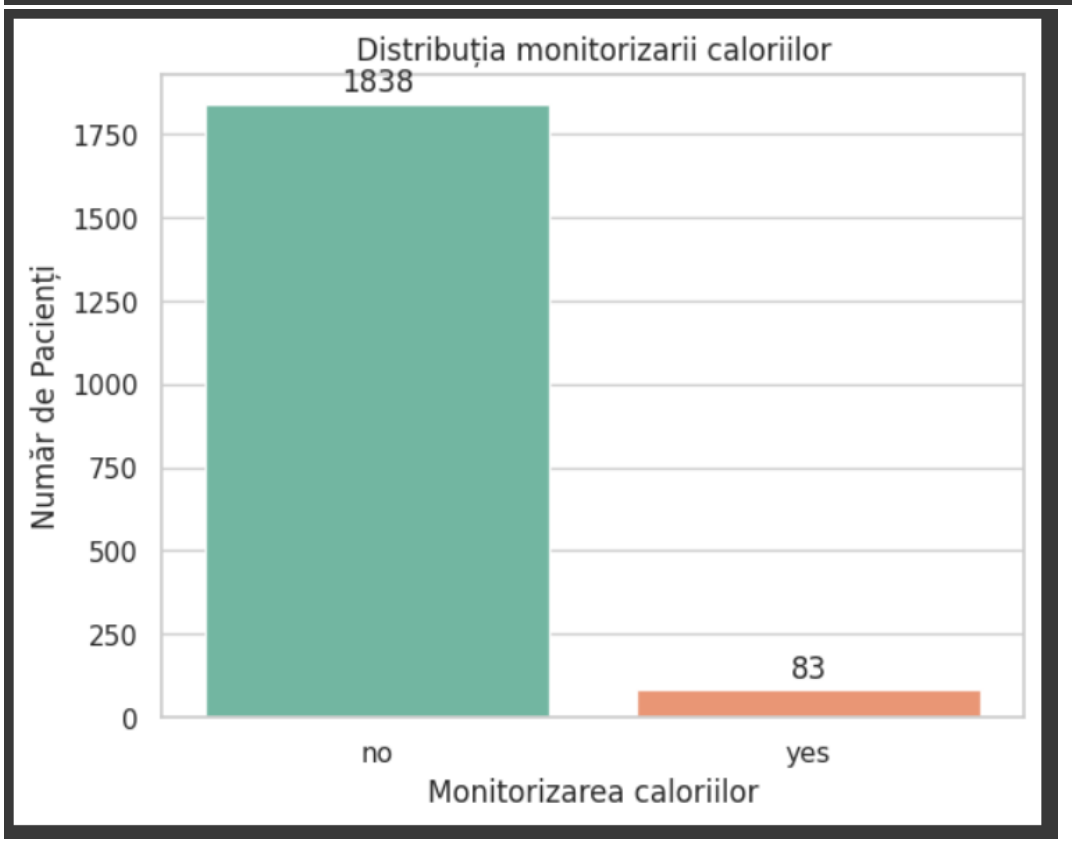

Distribuția meselor zilnice
Main_meals_daily
1.0
3.0
4.0
1093
178
0
62
1000
800
600
400
200
0
Număr de Pacienți
1.0
3.0
4.0
Numarul de mese zilnice


Distribuția consumului de snacks-uri
1609
221
48
43
1600
1400
1200
1000
800
600
400
200
0
Număr de Pacienți
Sometimes
Frequently
Always
no
Consum de snacks-uri

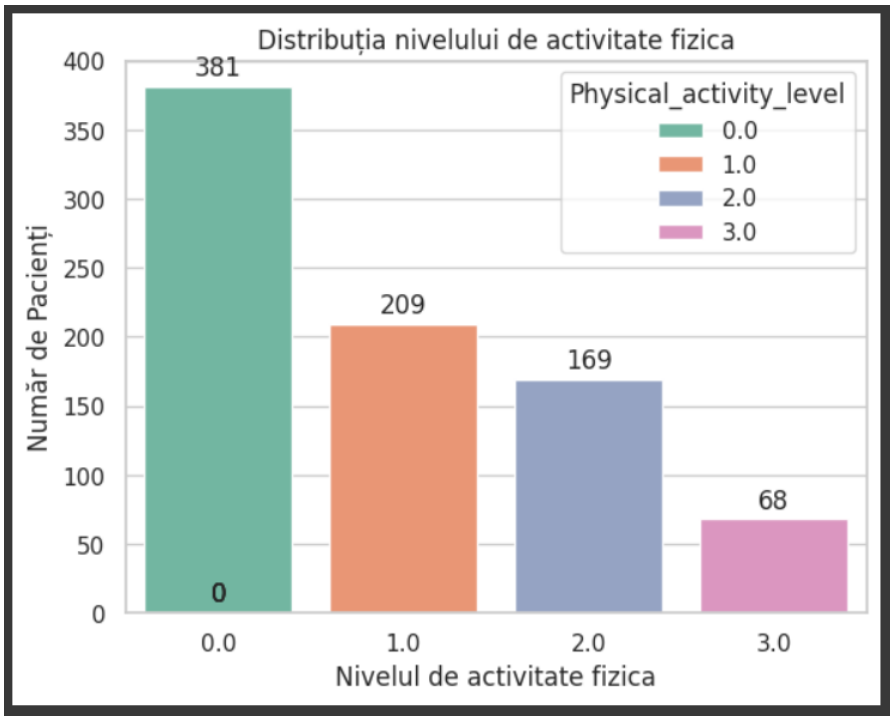
```
Valorile inregistrate pentru atributul Height:
Valoarea medie:  1.7022772277227722
Valoarea maximă:  1.98
Valoarea minimă:  1.45
Valoarea medianei:  1.7
Abaterea standard:  0.093206615563588
Abaterea medie absolută:  0.07702650743991632
Abaterea mediană absolută:  0.07000000000000006
Intervalul intercuartil (IQR):  0.14000000000000012
```

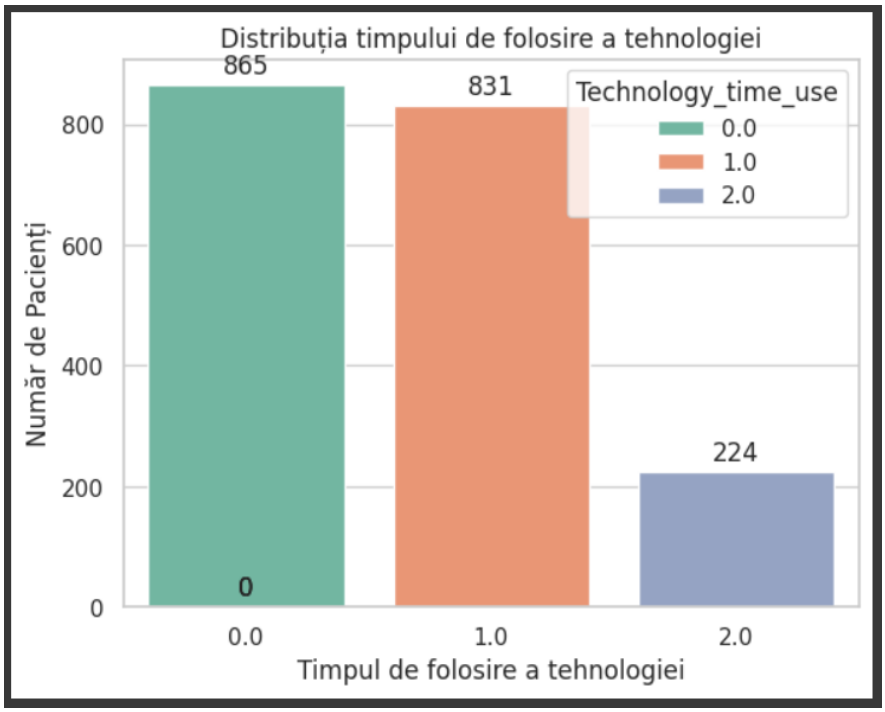


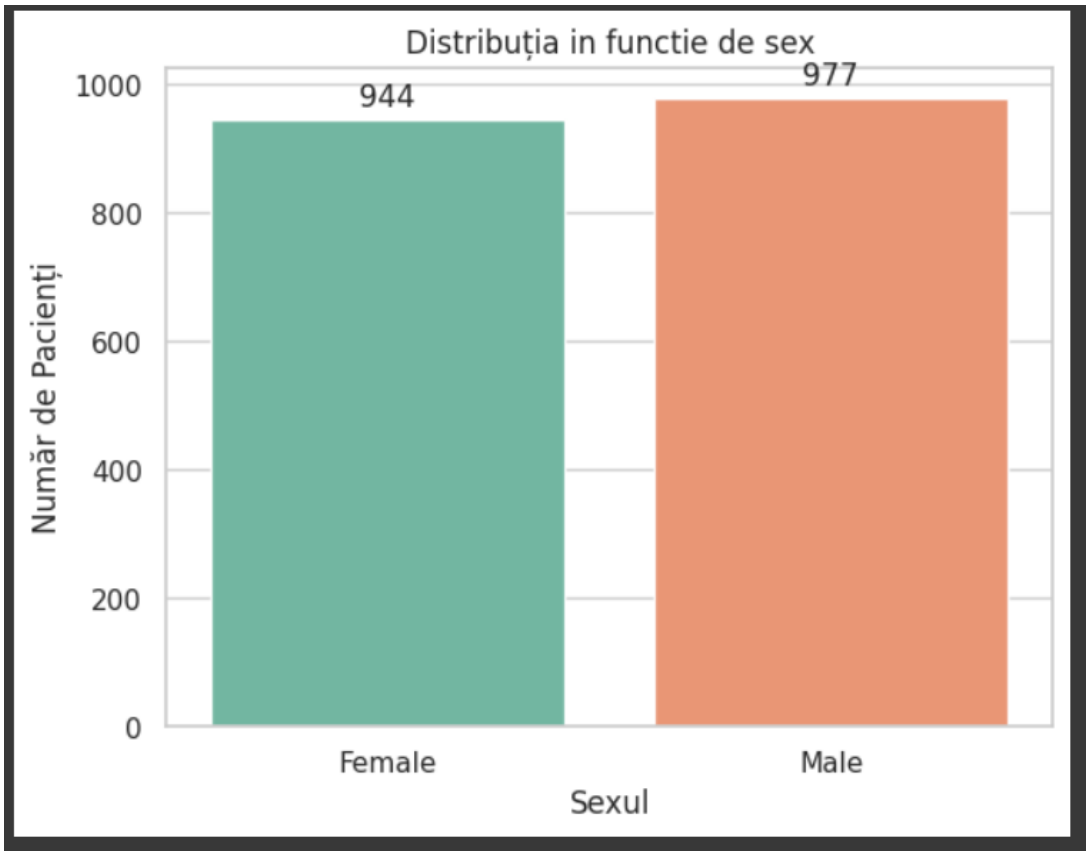

Distribuția consumului zilnic de apa
Water_daily
1.0
2.0
3.0
Număr de Pacienți
Litrii de apa pe zi
0
195
415
146
1.0
2.0
3.0


Distribuția monitorizarii caloriilor
Număr de Pacienți
Monitorizarea caloriilor
1838
83
no
yes

```
Valorile inregistrate pentru atributul Weight:
Valoarea medie:  69.32700729927005
Valoarea maximă:  130.0
Valoarea minimă:  39.0
Valoarea medianei:  69.32700729927006
Abaterea standard:  8.65248520091587
Abaterea medie absolută:  3.7281342974500156
```







Discuție:

Pentru atributele care au un număr mai mic de argumente (Transportation, Regular_fiber_diet, Diagnostic_in_family_history, High_calorie_diet, Alcohol, Main_meals_daily, Snacks, Smoker, Water_daily, Calorie_monitoring, Physical_activity_level, Technology_time_use, Gender) am decis sa folosesc un countplot:

- Putem observa numărul de argumente pentru fiecare atribut
- Putem observa direct distribuția intre argumente (numărul de pacienți)
- Putem observa dacă există preferințe generale pentru un anumit argument
- Putem observa dacă exista o corelație care sa ducă la un anumit diagnostic
- Putem observa ce argument ar trebui studiat mai mult

Pentru atributele care au un numar mai mare de argumente (Sedentary_hours_daily, Age, Est_avg_calorie_intake, Height, Weight) am decis să extrag mai multe valori pe baza setului de date (valoare medie, valoare minima, valoare maximă, abaterea standard, abaterea medie absolute, valoarea medianei, abaterea mediană absolută, interval intercuartil). Cum setul de date a trebuit "curățat" înainte să putem face analize pe el, pentru valorile lipsă am adăugat valoarea medie din acel atribut. Acest pas a fost necesar pentru a obține valoarea mediană și intervalul intercuartil.

Deoarece unele argumente au avut mult de "suferit" în urma procesului de eliminare, pentru acestea nu am putut calcula anumite valori pentru că majoritatea datelor aveau valoarea medie atribuită (în cazul intervalului intercuartil nu putem separa datele corect intre 25-75).

Cu ajutorul acestor valori putem:

- Înțelege dacă un pacient se află peste/sub media categoriei sale
- Înțelege daca atributul are o variație general bună
- Normaliza anumite valori pentru fiecare atribut
- Înțelege dacă un pacient are nevoie de "atenție" deosebită
- Corela diagnosicul cu anumite valori din atribut

Pentru o analiză mai detaliată voi exemplifica pe atributul Weight:

```
Valorile inregistrate pentru atributul Weight:
Valoarea medie:  69.32700729927005
Valoarea maximă:  130.0
Valoarea minimă:  39.0
Valoarea medianei:  69.32700729927006
Abaterea standard:  8.65248520091587
Abaterea medie absolută:  3.7281342974500156
```

Media și Mediana Greutății: Valoarea medie (69.33 kg) și valoarea medianei (69.33 kg) sunt foarte apropiate, ceea ce indică o distribuție relativ simetrică a datelor în jurul acestui centru. Asta sugerează că majoritatea pacienților au o greutate în jurul acestei valori.

Valoarea Maximă și Minimă: Greutățile variază de la un minim de 39 kg până la un maxim de 130 kg. Acest interval larg indică o varietate considerabilă în greutatea pacienților din acest set de date.

Abaterea Standard (8.65 kg): Aceasta măsură ne spune cât de dispersate sunt valorile greutății în jurul mediei. O abatere standard de 8.65 kg sugerează că majoritatea valorilor greutății se află într-un interval de ±17.3 kg (dublul abaterii standard) față de media, ceea ce acoperă majoritatea pacienților din intervalul 51.97 kg la 86.67 kg. Asta înseamnă că în acest interval se găsește aproximativ 68% din pacienți, dacă presupunem o distribuție normală.

Abaterea Medie Absolută (3.73 kg): Aceasta este o altă măsură a dispersiei care este mai puțin sensibilă la valori extreme. O abatere medie absolută de 3.73 kg indică faptul că, în medie, greutățile individuale se abat cu 3.73 kg de la greutatea medie de 69.33 kg.

Interpretarea pentru Un Pacient Specific

Dacă un pacient are o greutate apropiată de 69.33 kg: Acest pacient ar fi considerat a avea o greutate "tipică" pentru acest set de date, situându-se aproape de media populației.

Dacă un pacient are o greutate sub 51.97 kg sau peste 86.67 kg: Acești pacienți ar putea fi considerați ca având greutăți "atipice" în comparație cu majoritatea populației din setul de date, fiind mai departe de medie.

Greutate foarte apropiată de 39 kg sau 130 kg: Pacienții la aceste extreme pot necesita atenție specială, deoarece se încadrează la capetele spectrului de greutate, indicând posibile probleme de sănătate sau condiții speciale.

Analize de covarianță

```
Raport intre Diagnostic_in_family_history si Diagnostic:
(Diagnostic                    D0   D1   D2   D3   D4   D5   D6
 Diagnostic_in_family_history
 no                           131  121   72   17    6    1    0
 yes                          115  141  186  252  314  269  296,
 565.4862202238144,
 6.472223861927874e-119)
```

```
Raport intre High_calorie_diet si Diagnostic:
(Diagnostic          D0   D1   D2   D3   D4   D5   D6
 High_calorie_diet
 no                  46   70   20   70   10    7    1
 yes                200  192  238  199  310  263  295,
 208.2104438340789,
 3.387930453342451e-42)
```

```
Raport intre Smoker si Diagnostic:
(Diagnostic   D0   D1   D2   D3   D4   D5   D6
 Smoker
 no          245  249  255  265  315  257  295
 yes           1   13    3    4    5   13    1,
 30.310408327994224,
 3.431106582797794e-05)
```

```
Raport intre Alcohol si Diagnostic:
(Diagnostic   D0   D1   D2   D3   D4   D5   D6
 Alcohol
 Always        0    1    0    0    0    0    0
 Frequently    1   17   15   19   12    2    0
 Sometimes   138  149  197  129  157  204  295
 no          107   95   46  121  151   64    1,
 313.74265129715997,
 7.129430759453751e-56)
```

```
Raport intre Physical_activity_level si Diagnostic:
(Diagnostic               D0  D1  D2  D3  D4  D5  D6
 Physical_activity_level
 0.0                      32  73  36  58  76  26  80
 1.0                       6  87  37  42  35   2   0
 2.0                      57  65  22   8  11   5   1
 3.0                       3  37  10   6  12   0   0,
 273.2379231290483,
 1.4849286155403042e-47)
```

```
Raport intre Transportation si Physical_activity_level:
(Physical_activity_level  0.0  1.0  2.0  3.0
 Transportation
 Automobile                99   29   49   12
 Bike                       1    1    2    3
 Motorbike                  7    1    1    2
 Public_Transportation    263  162  103   39
 Walking                   11   16   14   12,
 56.142295163913374,
 1.1284204214354972e-07)
```

Am realizat aceste rapoarte între diverse atribute și clasă și între atribute pentru a putea observa cum se influențează unele pe altele. In principal, dorim sa observăm dacă o creștere/scădere într-un argument al unui atribut poate favoriza la apariția unui anumit diagnostic.

În rapoarte putem observa numarul de pacienți corelat între cele doua argumente (atribut și clasă/ atribut și atribut). Putem afla câte persoane dintr-o categorie al unui atribut se regăsește in categoria altui atribut. Așa putem înțelege in mod direct implicația unui atribut in clasa noastră.

Ultimele două valori sunt esențiale, ele reprezentând:

Valoarea lui (chi-pătrat) este o măsură a cât de mult așteptările (frecvențele așteptate) diferă de observații (frecvențele observate).

În general:

O valoare mare a lui χ^2 indică o diferență mare între frecvențele așteptate și cele observate, sugerând că variabilele sunt dependente (există o relație între ele).

O valoare mică a lui χ^2 sugerează că discrepanțele între frecvențele așteptate și cele observate sunt mici sau inexistente, indicând independența între variabile (lipsa unei relații).

p-value reprezintă probabilitatea de a observa o statistică de test la fel de extremă sau mai extremă decât cea observată, presupunând că ipoteza nulă este adevărată.

Ipoteza nulă într-un test Chi-pătrat este de obicei că nu există o relație (independență) între variabile.

Pentru p-value:

Un p-value mic (de obicei, mai mic decât un prag de semnificație, cum ar fi 0.05), indică faptul că diferențele observate sunt suficient de mari încât să fie improbabile să fi apărut prin întâmplare, ceea ce conduce la respingerea ipotezei nule și acceptarea că există o relație între variabile.

Un p-value mare sugerează că diferențele observate ar putea fi datorate variației întâmplătoare și, prin urmare, nu sunt suficient de convingătoare pentru a respinge ipoteza nulă; variabilele pot fi considerate independente.

Exemplu:

```
Raport intre Diagnostic_in_family_history si Diagnostic:
(Diagnostic                    D0   D1   D2   D3   D4   D5   D6
 Diagnostic_in_family_history
 no                           131  121   72   17    6    1    0
 yes                          115  141  186  252  314  269  296,
 565.4862202238144,
 6.472223861927874e-119)
```

Valoarea χ^2 este 565.49, ceea ce este destul de mare. Această valoare sugerează că există o asociere puternică între istoricul familial de diagnostic și diagnosticul actual al pacienților. Cu alte cuvinte, există diferențe semnificative între frecvențele așteptate și cele observate în categoriile de diagnostic atunci când sunt comparate cu istoricul familial.

Valoarea p-value este aproximativ 6.47 x 10^(-119), care este extrem de mică, mult sub orice nivel de semnificație convențional (cum ar fi 0.01 sau 0.05). Aceasta indică faptul că rezultatele observate sunt extrem de improbabile să fie datorate șansei, și deci putem respinge ipoteza nulă a independenței între aceste două variabile. Cu alte cuvinte, există dovezi statistice foarte puternice că există o relație între istoricul familial și diagnostic.

În concluzie, pe baza valorilor mari de χ^2 și a p-value extrem de mici, putem spune că atributul „Diagnostic_in_family_history” are un impact semnificativ asupra clasei „Diagnostic”. Acesta poate fi considerat unul dintre factorii cei mai predictivi sau cu cea mai mare influență asupra diagnosticului pacienților în acest set de date.

3. Extragerea manuală a atributelor și utilizarea algoritmilor clasici de Învățare Automată

Au fost propuși 4 algoritmi pentru analiza setului de date:

- RandomForest
- ExtraTrees
- GradientBoosted Trees
- SVM

Vom utiliza fiecare algoritm, urmărind structura următoare:

1. Tratarea valorilor lipsă

Alegem între SimpleImputer sau IterativeImputer bazat pe natura datelor și relațiile dintre atribute. SimpleImputer poate fi suficient pentru majoritatea cazurilor, dar IterativeImputer poate oferi rezultate mai bune pentru date cu relații complexe între atribute.

2. Standardizarea datelor

Alegem între StandardScaler, MinMaxScaler sau RobustScaler. Decizia depinde de prezența outlierilor și de distribuția datelor. StandardScaler este o alegere bună pentru majoritatea cazurilor, dar dacă datele tale includ multe outliere, RobustScaler ar putea fi o opțiune mai bună.

3. Selecția atributelor

Folosim Variance Threshold sau Select Percentile pentru a reduce numărul de atribute la cele mai relevante. Aceasta va ajuta la reducerea complexității modelului și la evitarea overfitting-ului.

4. Căutarea hiper-parametrilor
Utilizăm GridSearchCV pentru a testa diferite combinații de hiper-parametri pentru fiecare algoritm.

5. Antrenarea și testarea modelului

6. Evaluarea algoritmului

7. Raportarea rezultatelor

Vom urma acești pași pentru fiecare algoritm aplicat setului nostru de date pentru a putea face analize cât mai corecte si specific.

3.1. Pentru tratarea valorilor lipsă din setul de date am folosit SimpleImputer și IterativeImputer.

Acolo unde lipseau foarte puține valori (<10) am preferat să folosesc SimpeImputer cu strategia de înlocuire most_frequent sau median. Nu afectează echilibrul de date și nici nu creștem biasul considerabil.

Acolo unde lipseau foarte multe valori (ordinul sutelor) am preferat sa folosesc IterativeImputer pentru a avea un echilibru in date și pentru a nu introduce o singură valoare de sute de ori. Această problemă ar fi influențat modelul sa urmeze acea valoare. De asemenea, am făcut modificările necesare pentru ca valorile să ramană in parametrii inițiali (am rotunjit la numere întregi, am păstrat doar doua zecimale).

3.2. Pentru decizia standardizării setului de date, vom analiza necesitatea pentru fiecare algoritm propus.

RandomForest:

RandomForest construiește modelul său folosind copaci de decizie. Un copac de decizie face split-uri în date (adică împarte datele în grupuri mai mici) bazându-se pe cât de bine o anumită caracteristică poate separa clasele țintă.

Fiecare caracteristică este evaluată independent de celelalte când copacii își fac split-urile, deci scala absolută a unei caracteristici nu schimbă modul în care modelul învață structura de bază a datelor.

În plus, RandomForest este proiectat să fie robust la overfitting, în parte datorită alegerii aleatorii a caracteristicilor la fiecare split, cunoscută sub numele de "feature bagging". Acest proces reduce sensibilitatea modelului la variabilitatea specifică a oricărei caracteristici, făcându-l mai puțin susceptibil la anomalii în distribuția datelor, cum ar fi scale diferite.

ExtraTrees:

La fel ca și RandomForest, ExtraTrees este format dintr-un ansamblu de copaci de decizie. Split-urile în copaci sunt bazate pe selecția aleatoare a punctelor de split și a caracteristicilor, nu pe valorile absolute ale datelor.

Copacii de decizie analizează fiecare caracteristică independent, făcând split-uri bazate pe capacitatea de a separa eficient clasele, fără a fi influențați de scala absolută a caracteristicilor. Acest lucru înseamnă că variația în magnitudinea caracteristicilor nu afectează deciziile de split, ci doar ordinea valorilor este relevantă.

ExtraTrees introduce un nivel suplimentar de aleatorizare comparativ cu RandomForest, alegând puncte de split complet la întâmplare pentru fiecare caracteristică la fiecare nod. Această tehnică reduce și mai mult orice sensibilitate minoră la variația scalei caracteristicilor pe care copacii individuali ar putea să o aibă.

Prin utilizarea aleatorizării extreme și antrenându-se pe multiple sub-seturi ale datelor, ExtraTrees este, de asemenea, robust împotriva overfitting-ului. Aleatorizarea ajută la generalizarea modelului, făcându-l mai puțin susceptibil la variații mici în date, inclusiv la diferitele scale ale caracteristicilor.

GradientBoost Trees:

Deși GBT folosește copaci de decizie, care teoretic sunt insensibili la scala caracteristicilor, în practică, algoritmul GBT ajustează ponderile pe baza erorilor și poate fi sensibil la caracteristici cu scale foarte diferite. Acest lucru poate afecta viteza de convergență și stabilitatea algoritmului, deoarece copacii pot necesita ajustări diferite în ratele de învățare pentru diferite caracteristici.

Gradient Boosting optimizează o funcție de pierdere diferită de RandomForest și ExtraTrees, folosind o abordare de optimizare secvențială. Caracteristicile cu valori mari pot domina gradientul, făcând modelul să se concentreze disproporționat asupra acestor caracteristici. Acest lucru poate duce la o convergență mai lentă sau la necesitatea de a ajusta fin parametrii, cum ar fi rata de învățare.

GBT poate fi mai sensibil la outlieri decât algoritmi bazati exclusiv pe aleatorizare, cum ar fi RandomForest. Dacă caracteristicile cu scale mari includ și valori extreme, acestea pot influența disproporționat modelul, distorsionând performanța generală.

SVM:

SVM funcționează prin găsirea unui hiperplan care maximizează marginea între clasele de date. Deciziile despre poziția și orientarea hiperplanului sunt sensibil influențate de scalele caracteristicilor.

De exemplu, dacă o caracteristică are valori care variază între 0 și 1000, iar o altă caracteristică între 0 și 1, prima caracteristică va domina procesul de luare a deciziilor, potențial ignorând contribuția caracteristicii cu variații mai mici.

Distanțele calculate în spațiul caracteristicilor sunt direct afectate de magnitudinea fiecărei caracteristici. Caracteristicile cu valori mai mari pot distorsiona calculul distanței, afectând astfel determinarea hiperplanului optim.

Standardizarea reduce acest risc asigurând că fiecare caracteristică contribuie echitabil la calculul distanței.

Algoritmul SVM utilizează metode de optimizare (de exemplu, metode bazate pe gradienți) pentru a minimiza o funcție de pierdere. Dacă caracteristicile nu sunt pe o scală uniformă, gradientul funcției de pierdere poate fi dominat de caracteristicile cu variabilitate mai mare, ceea ce poate încetini convergența sau poate duce la convergență la soluții suboptimale.

Concluzie:

Am decis sa standardizez datele doar pentru algoritmul GradientBoost Trees și pentru algoritmul SVM. Voi folosi Standardizarea Z-score (StandardScaler din scikit-learn):

Standardizarea Z-score este un proces de normalizare a datelor care ajustează fiecare caracteristică astfel încât media să fie 0 și deviația standard să fie 1. Acest tip de standardizare este util pentru a asigura că toate caracteristicile contribuie egal la performanța modelului, în special pentru algoritmii sensibili la scala caracteristicilor, cum ar fi SVM.

Am creat o copie a setului de date pentru a o folosi pentru standardizare.

Este necesar sa abordăm și transformarea datelor categorice in numerice pentru a putea procesa datele. Astfel, vom avea doua seturi de date: unul pe care aplicăm doar transformarea datelor categorice în numerice și unul pe care aplicăm transformarea datelor categorice în numerice și standardizarea datelor.

3.3. Feature Extraction

Am decis să folosesc SelectPercentile, pentru a putea alege câte atribute păstrăm pentru noul set de date și pentru a putea folosi 2 teste diferite:

- chi2 pentru atributele categorice
- f_classif pentru atributele numerice

SelectPercentile este o metodă de selecție a caracteristicilor care selectează un procent specific de cele mai bune caracteristici, bazate pe un test statistic. Aceasta este utilă pentru eliminarea caracteristicilor care sunt cel mai puțin semnificative în predicția variabilei dependente.

chi2 (Testul Chi-pătrat):

chi2 este folosit pentru caracteristici categorice și măsoară dependența dintre fiecare caracteristică și variabila răspuns (care trebuie de asemenea să fie categorică). Este un test statistic folosit pentru a evalua dacă distribuțiile observate ale variabilelor categorice diferă semnificativ de distribuțiile așteptate, bazate pe ipoteza că caracteristicile sunt independente de variabila răspuns.

Cum funcționează: chi2 calculează suma pătratelor diferențelor între frecvențele observate și cele așteptate, normalizată prin frecvențele așteptate. Scorurile mari ale testului chi-pătrat indică o asociere puternică între caracteristică și variabila răspuns, sugerând că caracteristica este importantă pentru predicție.

f_classif (ANOVA F-value):

f_classif este utilizat pentru caracteristici continue și efectuează un test ANOVA (Analiza Variației) pentru a determina dacă mediile grupurilor sunt semnificativ diferite. Este folosit în scenarii de clasificare unde variabila răspuns este categorică.

Cum funcționează: f_classif calculează F-value pentru fiecare caracteristică, care măsoară cât de mult variază mediile grupurilor (definite de variabila răspuns) comparativ cu variația în cadrul grupurilor. Valori mai mari ale F-value sugerează că există o diferență statistic semnificativă între mediile grupurilor, și astfel caracteristica este relevantă pentru model.

Pentru început am selectat primele 8/13 atribute categorice și primele 3/5 atribute numerice. Vom varia ulterior pentru a observa diferențe în rezultate.

3.4. Alegerea hiper-parametrilor

RandomForest:

1. Numărul de arbori (n_estimators):

   Numărul de arbori este literalmente numărul de arbori individuali în pădurea aleatoare. Fiecare arbore este antrenat pe un subset de date și contribuie la predicția finală a modelului.

2. Adâncimea maximă a unui arbore (max_depth):

   Adâncimea maximă este lungimea maximă a căilor de la rădăcina arborelui până la o frunză. Un arbore cu adâncime mai mare poate capta mai multe detalii despre date, dar există și riscul de a se supra-antrena (overfitting).

3. Procentul din input folosit la antrenarea fiecărui arbore (max_features):

   Acest parametru controlează numărul de caracteristici care trebuie să fie considerate la căutarea celei mai bune diviziuni la fiecare nod al arborelui. Nu este un procent direct, ci mai degrabă un număr de caracteristici să fie alese la întâmplare.

   Am variat:

   n_estimators: 100, 200, 300

   max_depth: 10, 20, 30, None

   max_features: 0.25, sqrt, log2

Am obținut cea mai bună combinație:

{max_depth: 10, max_features: sqrt, n_estimators: 300}

Cu scorul: 0.7356910190786412

ExtraTrees:

Am folosit aceeași hiper-parametrii dar am obținut rezultate diferite.

Cea mai bună combinație:

{max_depth: 20, max_features: 0.25, n_estimators: 300}

Cu scorul: 0.7239688650112102

Diferențele pe care le putem observa în rezultatele GridSearchCV pentru RandomForestClassifier și ExtraTreesClassifier pot fi atribuite caracteristicilor intrinseci ale acestor doi algoritmi de păduri aleatoare, chiar dacă amândoi sunt bazati pe arbori de decizie:

Metoda de selecție a caracteristicilor:

În RandomForestClassifier, la fiecare nod, se alege cea mai bună divizare dintr-un subset aleatoriu de caracteristici (max_features).

În ExtraTreesClassifier (Extremely Randomized Trees), divizările sunt alese mai "aleatoriu" pentru fiecare caracteristică din subsetul aleatoriu și apoi cea mai bună dintre aceste divizări aleatoare este aleasă ca regula de divizare a nodului. Acest lucru adaugă un suplimentar de aleatoriu la model, ceea ce poate duce la o mai mare diversitate între arborii individuali.

Profunzimea arborelui:

RandomForestClassifier tinde să aibă arbori mai profund decizați, în timp ce arborii din ExtraTreesClassifier pot fi uneori mai puțin profund decizați datorită aleatorizării mai puternice, ceea ce poate duce la o reducere a riscului de supra-antrenare.

Sensibilitate la zgomot:

Datorită aleatorizării suplimentare, ExtraTreesClassifier poate fi mai puțin sensibil la zgomotul din date comparativ cu RandomForestClassifier.

RandomForestClassifier a obținut cele mai bune rezultate cu o adâncime mai mică a arborilor și utilizând 'sqrt' ca max_features, ceea ce sugerează că acesta preferă modele puțin mai simple și o selecție mai restrânsă a caracteristicilor pentru fiecare divizare.

Pe de altă parte, ExtraTreesClassifier a funcționat mai bine cu o adâncime mai mare și un procent mai mic de caracteristici (0.25) considerate la fiecare divizare, ceea ce indică faptul că un model mai complex cu o aleatorizare mai mare în selecția caracteristicilor funcționează mai bine pentru datele tale.

GradientBoostedTrees:

În contrast cu Random Forests sau Extra Trees, Gradient Boosting construiește arborii unul câte unul, unde fiecare arbore nou încearcă să corecteze erorile arborilor anteriori.

Numărul de arbori (n_estimators):

Numărul de arbori sequențiali adăugați în model. În contextul Gradient Boosting, fiecare arbore nou adaugă mai multă complexitate modelului, încercând să reducă eroarea rămasă de la arborii anteriori.

Adâncimea maximă a unui arbore (max_depth):

Limita maximă a adâncimii pentru fiecare arbore adăugat. Controlează complexitatea individuală a fiecărui arbore.

Rata de învățare (learning_rate):

Rata de învățare controlează cât de mult contribuie fiecare arbore la modelul final. Este un factor de scalare aplicat predicțiilor fiecărui arbore.

Am folosit următorii hiper-parametri:

n_estimators: 100, 200

max_depth: 10, 20

learning_rate: 0.01, 0.1

Cea mai bună combinație:

{learning_rate: 0.01, max_depth: 10, n_estimators: 200}

Cu scorul: 0.7285227801514446

SVM:

SVM funcționează construind un hiperplan sau un set de hiperplane într-un spațiu multidimensional, care poate fi folosit pentru clasificare, regresie sau alte sarcini. Intuitiv, o bună separare a claselor va fi realizată de hiperplanul care are cea mai mare distanță până la cele mai apropiate puncte de date ale oricăror clase (cunoscută sub numele de marja maximă).

Parametrul C (Regularizare):

C este un parametru de penalizare al termenului de eroare și controlează trade-off-ul dintre clasificarea corectă a punctelor de antrenament și maximizarea marjei de decizie.

Pentru valori mici ale lui C, marja va fi mai mare, chiar dacă unele puncte de date vor fi clasificate greșit. Pentru valori mari ale lui C, SVM va încerca să clasifice corect toate exemplele de antrenament, chiar și pe costul reducerii marjei, ceea ce poate duce la supra-antrenare.

Tipuri de kernel:

Linear: Nu adaugă complexitatea kernelurilor non-lineare, util pentru seturi de date mari sau când caracteristicile sunt suficiente pentru a lua o decizie.

RBF (Radial Basis Function): Acesta este cel mai frecvent utilizat și poate mânui cazuri în care relația dintre clasă și caracteristici este complexă.

Polinomial: Transformă caracteristicile într-un spațiu polinomial de grad specificat.

Sigmoid: Aduce caracteristicile la o scară similară cu funcția logistică.

Am folosit următorii hiper-parametrii:

C: 0.1, 1, 10

kernel: linear, poly, rbf, sigmoid

Cea mai bună combinație:

{C: 10, kernel: rbf}

Cu scorul: 0.696651719615889

3.5. Evaluarea algorimtilor

chi2:

Un scor mai mare de chi-squared indică o dependență mai mare între caracteristica respectivă și variabila țintă, sugerând că variabila este un predictor puternic pentru variabila țintă.

f_classif:

Un scor F mare indică faptul că media atributului variază considerabil între clase, sugerând că atributul are un rol important în distingerea între clase.

Am folosit aceste două teste pentru etapa de Feature Extraction și am decis să aleg 11/13 atribute din cele categorice și 4/5 atribute din cele numerice. Acest lucru inseamnă ca vom elimina din procesul de antrenare și testare ultimele 2 atribute categorice care au cea mai mică dependență raportată la clasa noastră și ultimul atribut din cele numerice.

Atributele eliminate sunt:

- Main_meals_daily
- Water_daily
- Est_avg_calorie_intake

```
Atribute categorice sortate după scorul chi2:
                              chi2_score
Gender                        292.146301
Calorie_monitoring            106.226250
Diagnostic_in_family_history  102.441023
Transportation                 97.847689
Regular_fiber_diet             83.427195
Physical_activity_level        53.105158
Snacks                         40.836828
Smoker                         29.679270
Technology_time_use            29.231594
High_calorie_diet              24.278573
Alcohol                        20.095790
Main_meals_daily                9.241580
Water_daily                     6.047906
Atribute numerice sortate după scorul ANOVA F-test:
                         f_classif_score
Weight                         82.858446
Height                         34.050162
Age                            14.159415
Sedentary_hours_daily           6.370526
Est_avg_calorie_intake          1.938191
```

Acestea indică cel ma mic nivel de corelație cu clasa target. Informațiile pe care le regăsim in aceste atribute sunt mult mai puțin importante pentru a determina un anumit Diagnostic decât cele pe care le oferă restul atributelor.

Dorim sa reducem costul computațional => renunțăm la aceste atribute in procesul de antrenare și testare.

În urma procedurii de Grid Search cu Cross Validation am obținut cele mai bune combinații de hiper-parametri prezentate în secțiunea anterioară. Vom folosi aceste valori pentru a face predicții pe setul de test și a le compara cu valorile exacte.

Urmează să analizăm anumite metrici pentru a înțelege cat de bune sunt predicțiile pentru algoritmi. Metricile cerute sunt următoarele:

Media:

- suma tuturor valorilor dintr-un set de date împărțită la numărul de valori din acel set
- media este utilă pentru a obține o valoare care reprezintă „centrul” unui set de numere

Varianța:

- varianța este o măsură a dispersiei sau a variabilității într-un set de numere
- ea arată cât de mult se împrăștie valorile în raport cu media
- o variantă mică indică faptul că valorile din set sunt apropiate de media (și, prin extensie, între ele), în timp ce o variantă mare indică faptul că valorile sunt mai răspândite în jurul mediei

De ce sunt importante?

Media este folosită pentru a oferi o valoare unică care rezumă setul de date, permițând compararea rapidă între diferite seturi de date sau subgrupuri de date.

Varianța este crucială pentru a înțelege cât de uniforme sunt datele. De exemplu, în contextul machine learning, o variantă mare în rezultatele unui model poate indica o instabilitate, în timp ce o variantă scăzută sugerează că modelul este consistent.

Acuratețea:

- acuratețea măsoară procentul total de predicții corecte (true positives și true negatives) din totalul predicțiilor făcute

Precizia:

- măsoară dintre toate predicțiile pozitive făcute de model, câte sunt de fapt corecte
- aceasta este o măsură a exactității pozitivelor prezise

Recall:

- recall-ul măsoară dintre toate cazurile pozitive reale, câte a reușit modelul să identifice corect
- este o măsură a capacității modelului de a detecta toate cazurile positive

Scorul F1:

- scorul F1 este media armonică între precizie și recall și este o măsură a balanței între precizie și recall
- acesta este util când avem nevoie de un singur număr pentru a evalua performanța modelului, mai ales când nu există o preferință clară între precizie și recall

RandomForest:

	D0	D1	D2	D3	D4	D5	D6
Acurateţe	0.78	0.88	0.45	0.57	0.59	0.88	**0.98**
Precizie	0.921	0.686	0.678	0.666	0.82	0.576	**0.945**
Recall	0.783	0.884	0.452	0.571	0.594	0.883	**0.981**
F1	0.846	0.773	0.542	0.615	0.689	0.697	**0.962**

	Media	Varianța
Acurateţe	0.73	0.03
Precizie	0.75	0.02
Recall	0.73	0.03
F1	0.72	0.02

45	11	2	1	0	1	0
1	46	2	2	0	1	0
2	7	19	0	5	8	1
0	3	3	27	2	14	0
0	1	2	5	41	18	2
2	1	0	5	0	52	0
0	0	0	1	0	0	52

ExtraTrees:

	D1	D2	D3	D4	D5	D6	D7
Acurateţe	0.75	0.77	0.6	0.69	0.7	0.85	**0.98**
Precizie	0.9	0.655	0.714	0.693	0.857	0.645	**0.945**
Recall	0.75	0.769	0.595	0.693	0.695	0.85	**0.981**
F1	0.818	0.707	0.649	0.693	0.768	0.733	**0.962**

	Media	Varianța
Acuratețe	0.76	0.01
Precizie	0.77	0.01
Recall	0.76	0.01
F1	0.76	0.01

45	10	3	1	0	1	0
4	40	2	4	1	0	1
1	6	25	0	3	7	0
0	2	2	3	48	12	2
0	2	2	3	48	12	2
0	1	0	6	2	51	0
0	0	0	1	0	0	52

GradientBoostedTrees:

	D1	D2	D3	D4	D5	D6	D7
Acuratețe	0.93	0.73	0.57	0.76	0.68	0.85	**0.98**
Precizie	0.877	0.791	0.6	0.725	0.824	0.662	**0.945**
Recall	0.833	0.730	0.571	0.755	0.681	0.85	**0.981**
F1	0.854	0.76	0.585	0.74	0.746	0.744	**0.962**

	Media	Varianța
Acuratețe	0.77	0.01
Precizie	0.78	0.01
Recall	0.77	0.01
F1	0.77	0.01

50	3	6	0	0	1	0
3	38	3	4	2	1	1
1	6	24	1	4	6	0
0	0	4	37	0	8	0
2	0	2	6	47	10	2
1	1	1	3	3	51	0
0	0	0	0	1	0	52

SVM:

	D0	D1	D2	D3	D4	D5	D6
Acuratețe	0.8	0.65	0.48	0.55	0.62	0.83	**0.98**
Precizie	0.827	0.739	0.588	0.574	0.796	0.543	**0.962**
Recall	0.8	0.653	0.476	0.551	0.623	0.833	**0.981**
F1	0.813	0.693	0.526	0.5625	0.699	0.657	**0.971**

	Media	Varianța
Acuratețe	0.7	0.03
Precizie	0.72	0.02
Recall	0.7	0.03
F1	0.7	0.02

48	6	4	1	0	1	0
8	34	1	6	1	1	1
2	4	20	2	5	9	0
0	1	5	27	2	14	0
0	1	3	4	43	17	1
0	0	1	6	3	50	0
0	0	0	1	0	0	52

Analiză:

Pe baza modelelor antrenate putem obține un raport al importanței fiecărui atribut din algoritmii utilizați.

Pentru RandomForest:

```
Importanța caracteristicilor:
Weight: 0.1372
Height: 0.1258
Regular_fiber_diet: 0.1176
Age: 0.0857
Gender: 0.0838
Alcohol: 0.0746
Sedentary_hours_daily: 0.0691
Snacks: 0.0620
Diagnostic_in_family_history: 0.0569
Transportation: 0.0557
Physical_activity_level: 0.0478
Technology_time_use: 0.0332
High_calorie_diet: 0.0331
Calorie_monitoring: 0.0115
Smoker: 0.0060
```

Pentru ExtraTrees, GradientBoostedTrees, SVM:

```
Importanța caracteristicilor:
Weight: 0.1443
Height: 0.1306
Regular_fiber_diet: 0.1231
Age: 0.0819
Gender: 0.0780
Alcohol: 0.0703
Sedentary_hours_daily: 0.0677
Snacks: 0.0618
Transportation: 0.0552
Diagnostic_in_family_history: 0.0531
Physical_activity_level: 0.0488
High_calorie_diet: 0.0347
Technology_time_use: 0.0331
Calorie_monitoring: 0.0115
Smoker: 0.0059
```

Putem observa cat de important este pentru predicție un atribut anume.

Alegera hiper-parametrilor este critică pentru fiecare din algoritmii studiați. Aceștia controlează comportamentul modelului și pot avea un impact semnificativ asupra performanței acestuia.

Un set bun de hiper-parametri poate îmbunătăți semnificativ performanța unui model, în timp ce un set slab ales poate duce la underfitting sau overfitting.

Dacă dispunem de putere de calcul și resursele necesare este recomandat să alocăm o parte importantă procesului de selecție a hiper-parametrilor prin metoda Grid Search cu Cross Validation. Cea mai buna combinație a hiper-parametrilor ne duce si la cea mai performantă predicție ulterioară.

Clasa D6 are cele mai bune predicții și prezintă cele mai bune metrici, in comparație cu restul claselor.

Vom studia metricile obținute pentru fiecare algoritm antrenat pentru a înțelege cât de corecte sunt predicțiile făcute pentru clase.

Acuratețea este o metrică prin care putem observa cât de bine a prezis modelul nostrum raportat la o anumită clasă. Mai exact:

Acuratețe = (TP+TN) / (TP+TN+FP+FN)

În urma analizei făcute putem deduce această metrică ca fiind un procent care ne indică cu ce valoare a reușit modelul antrenat de noi sa prezică o clasă.

Vom face un top pentru fiecare model, pentru a observa ce clase prezic cel mai bine:

RandomForest: D6, D5, D1, D0, D4, D3, D2

ExtraTrees: D6, D5, D1, D0, D4, D3, D2

GradientBoostedTrees: D6, D5, D0, D3, D1, D4, D2

SVM: D6, D5, D0, D1, D4, D3, D2

Precizia = TP / (TP+FP)

Putem observa astfel după procentul obținut câte predicții corect pozitive avem din totalul predicțiilor pozitive făcute.

În acest caz, clasa D6 tinde să obțină cele mai bune procente pentru majoritatea algoritmilor folosiți.

Recall = TP / (TP+FN)

TP (True Positive) este numărul de predicții corecte pozitive făcute de model pentru o clasă specifică.

FN (False Negative) este numărul de ori în care modelul a eșuat în a identifica corect o instanță pozitivă (adica, a clasificat greșit o instanță care ar fi trebuit să fie pozitivă).

În acest caz, clasa D6 tinde să obțină cele mai bune procente pentru majoritatea algoritmilor folosiți.

F1 score = 2 * (Precizie*Recall) / (Precizie + Recall)

Scorul F1 este o metrică folosită pentru a evalua performanța unui model de clasificare, combinând precizia și recall-ul într-o singură măsură. Este deosebit de util când avem nevoie de un echilibru între precizie și recall, și este particular important în scenarii unde distribuția claselor este neechilibrată.

În acest caz, clasa D6 tinde să obțină cele mai bune procente pentru majoritatea algoritmilor folosiți.

În matricea de confuzie putem observa foarte bine modul în care fiecare predicție s-a făut corect sau greșit.

Fiecare coloană a matricei reprezintă numărul de predicții ale unei clase, în timp ce fiecare rând reprezintă clasele reale din setul de date.

Diagonala principală (de la stânga sus la dreapta jos) arată numărul de predicții corecte pentru fiecare clasă. Valorile mari pe diagonala principală indică o performanță mai bună, deoarece modelul a reușit să clasifice corect o mare parte din instanțe.

Valoarea de pe rândul i și coloana j (în afară de diagonala principală) indică numărul de ori în care modelul a clasificat greșit instanțe din clasa reală i ca aparținând clasei j.

În acest caz, clasa D6 tinde să obțină cele mai bune valori pentru majoritatea algoritmilor folosiți.

Class	Metric	Random Forest	ExtraTrees	GradientBoostedTrees	SVM	Max Value
D0	Precision	**0.90196**	0.9	0.87719	0.82759	0.90196
D0	Recall	0.76667	0.75	**0.83333**	0.8	0.83333
D0	F1-Score	0.82883	0.81818	**0.85470**	0.81356	0.85470
D1	Precision	0.70149	0.65574	**0.79167**	0.73913	0.79167
D1	Recall	**0.90385**	0.76923	0.73077	0.65385	0.90385
D1	F1-Score	**0.78992**	0.70796	0.76	0.69388	0.78992
D2	Precision	**0.72**	0.71429	0.6	0.58824	0.72
D2	Recall	0.42857	**0.59524**	0.57143	0.47619	0.59524
D2	F1-Score	0.53731	**0.64935**	0.58537	0.52632	0.64935
D3	Precision	0.69767	0.69388	**0.72549**	0.57447	0.72549
D3	Recall	0.61224	0.69388	**0.75510**	0.55102	0.75510
D3	F1-Score	0.65217	0.69388	**0.74**	0.5625	0.74
D4	Precision	0.84	**0.85714**	0.82456	0.79630	0.85714
D4	Recall	0.60870	**0.69565**	0.68116	0.62319	0.69565
D4	F1-Score	0.70588	**0.768**	0.74603	0.69919	0.768
D5	Precision	0.58696	0.64557	**0.66234**	0.54348	0.66234
D5	Recall	**0.9**	0.85	0.85	0.83333	0.9
D5	F1-Score	0.71053	0.73381	**0.74453**	0.65789	0.74453
D6	Precision	0.91228	0.94545	0.94545	**0.96296**	0.96296
D6	Recall	0.98113	0.98113	0.98113	**0.98113**	0.98113
D6	F1-Score	0.94545	0.96296	0.96296	**0.97196**	0.97196

Putem observa ce algoritmi dau rezultate mai bune pentru anumite clase.

Concluzie:

Consider că, atât setul de date, cerințele și analizele propuse în acest studiu de caz, sunt foarte relevante și folositoare pentru înțelegerea modului în care trebuie pregătit un set de date pentru a fi antrenat și modelat în vederea obținerii finale a unor predicții și verificarea corectitudinii de execuție.