

## PROIECT PCLP3

### - Clasificarea stării de sănătate a pacienților –

Acest proiect are ca scop realizarea unui model de clasificare binară care prezice starea de sănătate (bolnav sau sanator) a unor pacienți pe baza unor factori precum: vârsta (numere întregi), sexul (valori categorice: feminin sau masculin), greutatea (numere reale), dacă fumează sau nu (1 – pacientul fumează, 0 – nu fumează), febra (numere reale), istoricul familiei (variabile categorice: 1 – pacientul are membrii ai familiei diagnosticați ca fiind bolnavi, 0 – contrar), dacă consumă sau nu fast food (1 – da, 0 – nu), stilul de viață (valori categorice: activ, moderat, sedentar). Setul de date, atât cel care antrenează modelul cât și cel de testare, sunt generate sintetic, iar algoritmul de bază folosit este regresia logistică.

Implementarea proiectului este făcută în fișierul `sursa proiect.py`, în care ne folosim de biblioteci și funcții din Python (pandas, numpy, seaborn, matplotlib, os, scikit-learn), sau de funcții definite de mine (generate\_data din fișierul `generate.py`).

Fișierul `generate.py`: definește funcția `generate_data` care creează un data frame cu  $n$  rânduri (care reprezintă  $n$  pacienți) necesar pentru implementarea modelului de clasificare. Fiecare pacient (fiecare rând) este caracterizat prin factorii descriși anterior. Datele sunt generate random, folosind un seed oarecare pentru a avea de fiecare dată același output.

Modul de generare a datelor:

Vârsta (age) : este generată aleator, cu valori întregi între 18 și 90 de ani.

Sexul (sex) : este selectat aleatoriu între "male" și "female" cu probabilitate egală

Greutatea (weight) : este generată astfel încât majoritatea valorilor să fie în jurul lui 70 cu o deviație de 10 dar fără să depășească intervalul (45, 100). Pentru a face datele mai realiste și pentru a putea trata cazul valorilor aberante, am adăugat intenționat câteva valori aberante (10) la pacienți selectați random (greuți din intervalul 110-150).

Fumatul (smokes) : generat astfel încât să fie o probabilitate de 40% ca pacientul să fumeze (smokes = 1)

Temperatura (fever) : valori random între 36 și 40. Am adăugat și aici câteva valori aberante: 5 valori din intervalul (32, 35) și 5 valori din (41, 43).

Istoria familială (family\_history) : generat astfel încât să existe o probabilitate de 30% ca să fie adevărat

Fast food (eats\_fast\_food) : identic cu smokes

Stilul de viata (lifestyle) : este selectat random dintre “active”, “moderate” si “sedentary” si apoi mapat la un scor de risc ( 0 – active, 0.5 – moderat, 1 – sedentar) pentru a-l putea lua in calcul mai usor la aflarea starii de sanatate a pacientilor.

Starea de sanatate (sick) : este eticheta finala care decide daca pacientul este sau nu bolnav (1-da, 0 – nu ) si este calculate ca o combinatie ponderata a factorilor de risc descrisi anterior. Daca suma factorilor de risc ( fiecare luat cu o anumita pondere) depaseste valoarea de prag = 2, atunci pacientul este considerat bolnav.

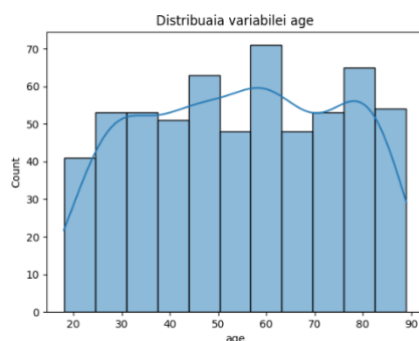
Pentru a indeplini si cerinta in care trebuie sa tratam valorile lipsa, am adaugat intentionat astfel de valori : 5% penntru coloanele fever, lifestyle si family\_history ( fiecare este de tip diferit pentru a trata modul de alegere pentru a complete datele pentru fiecare tip de date).

Functia intoarce data frame-ul de care ne vom folosi mai departe.

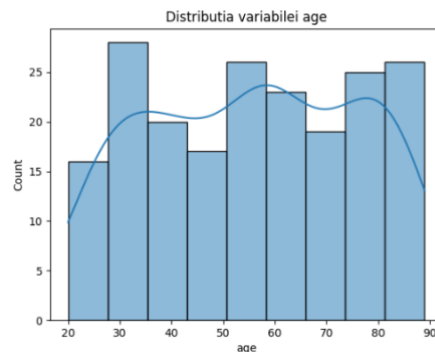
Fisierul proiect.py: pentru a descrie modul de implementare, vom imparti programul in mai multe parti:

1. Generam cele doua seturi de date (train si test) apeland functia generate\_data. Setul de train va contine 600 de exemple (pacienti), iar cel de test, 200. Datele sunt salvate in format csv in fisierele train.csv si test.csv. Cream de asemenea si folderul “grafice” in care vom salva toate graficele pe care urmeaza sa le generam.
2. Analiza datelor lipsa : am definit functia missing\_analysis care primeste ca parametru un data frame (fie cel de testare, fie cel de antrenament) si un nume (train sau test) necesar pentru afisarea intuitiva a rezultatelor. Functia verifica fiecare coloana din data frameul transmis pentru a vedea cate valori lipsa exista ( atat numeric, cat si ca procent) si creaza un data frame pentru acestea. In cazul in care nu exista valori lipsa ( caz imposibil pentru modul in care am setat sa se genereze datele) se afiseaza un mesaj corespunzator, iar in caz afirmativ, data frame-ul creat. In exterior, apelam functia pentru ambele seturi de date.
3. Statistici descriptive: analizam ambele seturi de date pentru a observa statistici precum: valori medii, valori maxime si minime.
4. Analiza distributiei variabilelor 1: in aceasta parte ne ocupam de analiza distributiei valorilor din fiecare coloana numerica (age, weight, fever) cu ajutorul histogramelor. Astfel putem observa ce categorie de pacienti predomina din cei aflati in data frame-uri. Pentru exemplul generat de mine am obtinut:

Train, age:



Test, age:

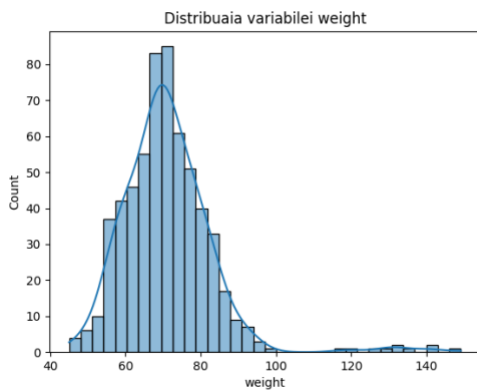


Observam ca nu exista diferenta majore intre cele doua histograme, la fiecare distributia varstei este relative uniforma. Pentru setul de train varstele sunt distribuite echilibrat, nu exista o categorie de varsta care sa predomine exagerat (desigur observam ca personale cu varsta de 58-62 sunt predominante iar cele cu varsta de 18-25 ceva mai rare, insa nu este o diferenta majora care ne poate afecta modelul). Pentru setul de train observam cateva variatii mai considerabile, datorate numarului mai redus de exemple (200 comparativ cu cele pentru train, 600).

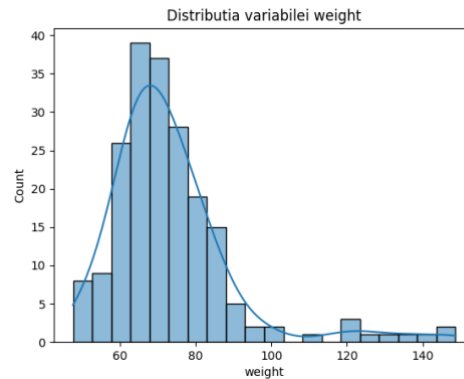
Nu sunt zone de varsta ignorate, deci nu riscam ca modelul sa nu fie antrenat pentru anumite cazuri.

Consider ca nu trebuie adusa nicio preprocesare, deoarece aveam de-a face cu valori numerice de tip continuu, fara valori lipsa si fara valori exagerate (reamintesc ca aceste valori le-am setat de mana dar in realitate pot exista).

Train, weight:



Test, weight:

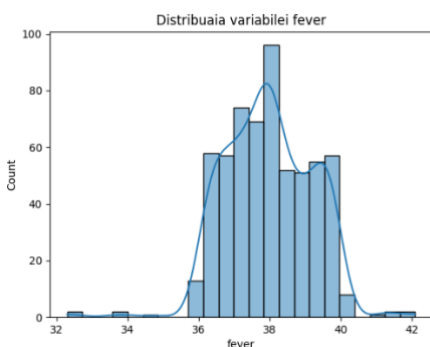


Ce observam : atat pentru train cat si pentru test, valorile se incadreaza in intervalul 45-100, mojaritatea concentrandu-se intre 65-75. Se observa si cateva valori aberante (introduce intentionat) in intervalul 110-150.

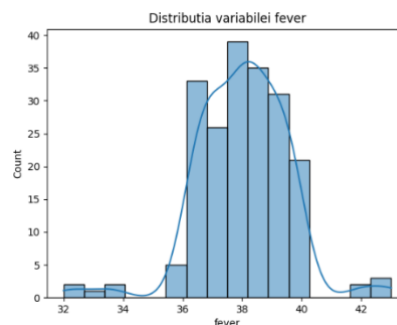
Ce idei putem formula: prezenta valorilor aberante poate influenta negativ antrenarea modelului si rezultatele date de acesta si trebuie tratate separat.

Ce proprocesari ar trebui sa aplicam: ar trebui sa detectam valorile aberante si sa gasim o modalitate de tartare a acestora, precum inlocuirea acestora cu o medie. De asemenea pentru functionarea mai corecta a algoritmului este necesara o scalare ( standard scaler) pentru ca toate variabilele sa se afle pe aceeasi scala si sa influenteze modelul intr-un mod mai echilibrat.

Train, fever:



Test, fever:

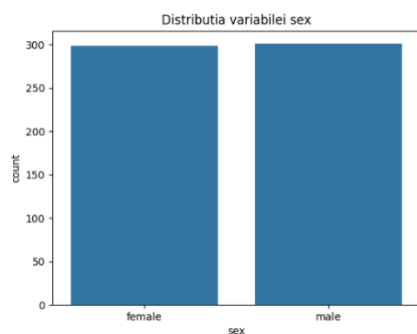


Ce observam : valorile sunt predominante in zona de 37-38, iar in intervalul 36,2- 39,4 excluzand intervalul 37-38 sunt distribuite relative uniform. Obervam si valorile aberante setate : 32-35 si 41-43  
 Ce idei putem formula: in cazul febrei, valorile aberante pot insemna risc ridicat de a fi bolnav, iar din cauza, iar in modelul nostru febrei ii este atribuit o anumita pondere in calcularea factorului de risc care poate scapa aceste cazuri de pacienti bolnavi. Asadar modelul ar trebui adaptat astfel incat sa ia mai semnificativ in considerare aceste valori aberante. De asemenea, la fel ca la age, trebuie sa scalam datele.

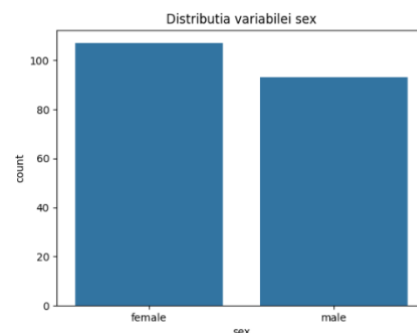
Ce preprocesari ar trebui sa aplicam: tratarea valorilor aberante, fie prin inlocuirea unei medii (nerecomandabil pentru motivul descries anterior) fie prin calcularea intr-un mod diferit a factorului de risc care sa tina cont de acestea. De asemenea pentru functionarea mai corecta a algoritmului este necesara o scalare ( standard scaler) pentru ca toate variabilele sa se afle pe aceeasi scala si sa influenteze modelul intr-un mod mai echilibrat.

5. Analiza distributiei variabilelor 2: : in aceasta parte ne ocupam de analiza distributiei valorilor din fiecare coloanal cu valori categorice (sex, smokes, family\_history, eats\_fast\_food, lifestyle) cu ajutorul graficelor de tip countplot cu scopul de a observa frecventa fiecarei valori posibile a variabilei. Pentru exemplul generat de mine am obtinut:

Train, sex:



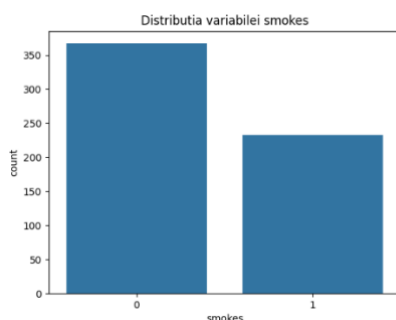
Test, sex:



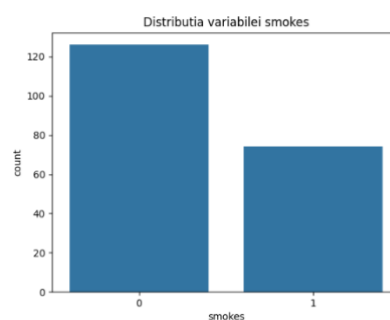
Ce observam si ce idei putem formula: pentru setul de train distributia este perfect echilibrata (datorita faptului ca am ales sa generam datele cu probabilitati egale pentru fiecare gren), iar pentru setul de train se obsrva o usoara diferenta: femeile au o frecventa mai mare. Desi am generat dateele in acelasi mod cu cele de train, aceasta diferenta poate fi causata de numarul mic de exemple.

Ce preprocesari putem aplica: putem transforma variabila categorica intr-un format numeric (1- female, 0- male) pentru calcularea factorului de risc

Train, smokes:

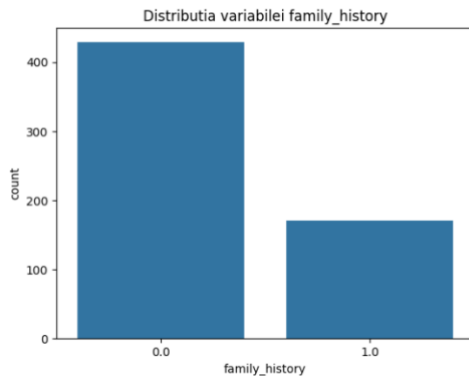


Test, smokes:

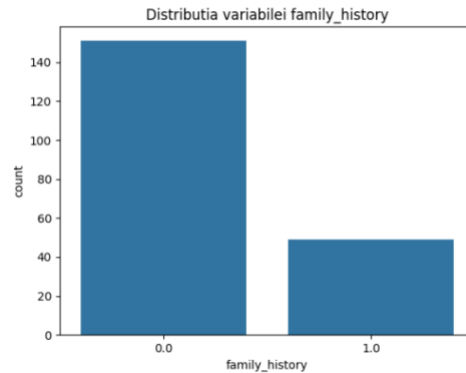


Observam ca in ambele seturi de date avem 60% personae fumatoare, 40% personae nefumatoare, lucru datorat felului in care am generat datele. Persoanele fumatoare vor avea un risc mai ridicat fata de cele nefumatoare ( in formula pe care am folosit-o in calculul factorului de risc, pentru persoanele nefumatoare nu se va mai adauga variabila smokes). Nu consider ca trebuie adusa nicio preprocesare.

Train, family\_history:

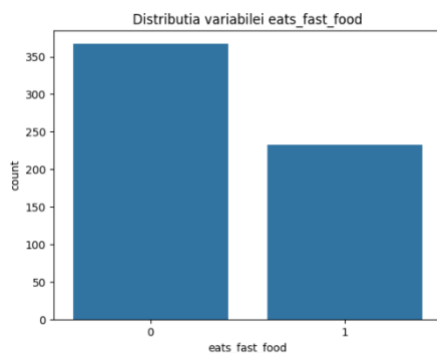


Test, family\_history:

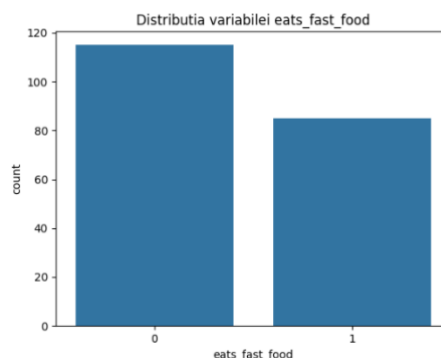


Observam ca pentru ambele seturi de date probabilitatile sunt de 70% la 30 % (nu exista si exista). Acest factor nu este unul foarte important in calcularea factorului de risc deci nu are o influenta masiva pentru model si nu este nevoie de preprocesari.

Train, eats\_fast\_food:

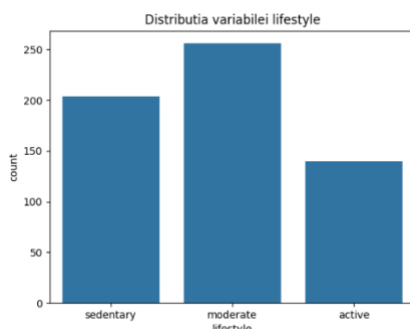


Test, eats\_fast\_food:

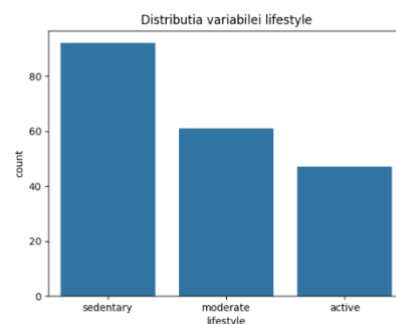


In ambele grafice observam ca majoritatea pacientilor nu consuma fast food ( 60% pentru valoarea 0 ). Aceasta varabila poate influenta destul de considerabil riscul de boala. Fiind o varabila binara nu necesita preprocesari.

Train, lifestyle:



Test, lifestyle:



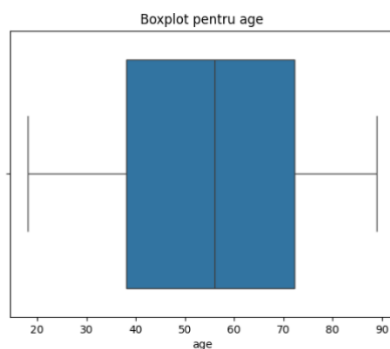
Ce observam: pentru setul de antrenament predominant este stilul de viata moderat, iar pentru cel de test, predominant sedentar. Elementul comun al celor doua seturi de date este frecventa scazuta pentru stilul de viata activ.

Ce idei putem formula: un stil de viata sedentar, poate fi un factor ce creste considerabil riscul de imbolnavire, mai ales alaturi de alti factori precum consumul de fast food sau fumatul.

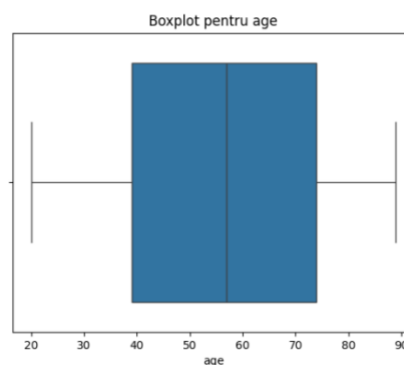
Ce preprocesari ar trebui sa aplicam: fiind o varibila categorica, este necesara o codificare pentru a putea manipula datele mai usor (label encoder). Observam ordinea aleasa: stilul de viata moderat nu este cel mai bun (comparative cu cel active care nu influenteaza factorul de risc), dar nu este la fel de rau precum cel sedentar.

6. Detectarea outlierilor: folosim boxplot-uri pentru a detecta valorile aberante. Am definit functia `detect_outliers` care primeste un data frame si evalueaza coloanele numerice ( in cazul nostru `age`, `weight` si `fever`) si genereaza pentru acestea cate un boxplot in care se pot observa valorile aberante (daca exista). Apelam aceasta functie pentru ambele seturi de date.

Train, age:

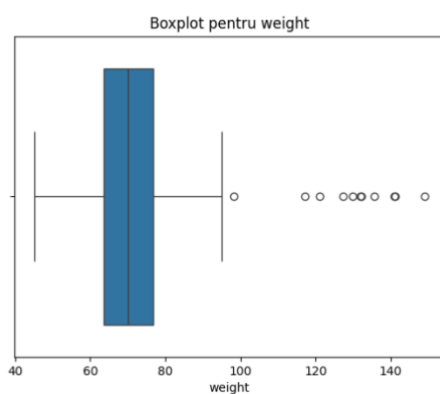


Test, age:

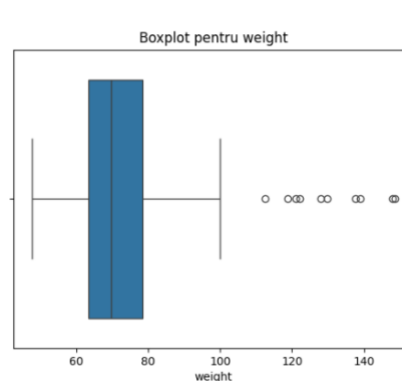


Observam ca nu exista valori aberante pentru niciun set din cele doua si, la fel ca la histograme observam o distribuire echilibrata a varstelor (medianele sunt plasate aproximativ in centru). Constatam, datorita asemanarii dintre cele doua seturi, ca modelul va fi bine antrenat (putea sa existe cazul in care antrenam modelul pentru o anumita categorie de pacienti, iar cei din setul de test sa nu se incadreze in standard, aparant erori). Nu necesita preprocesari.

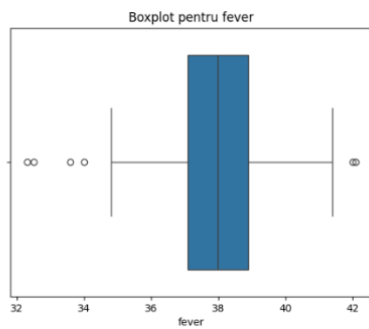
Train, weight:



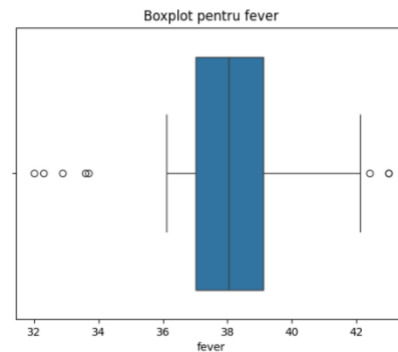
Test, weight:



Train, fever:



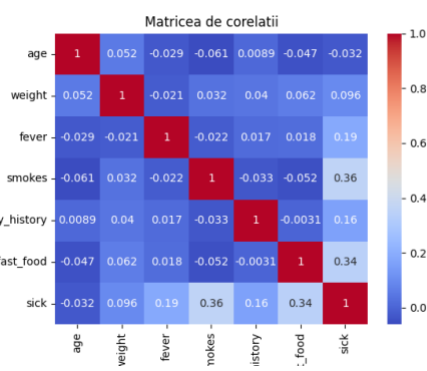
Test, fever:



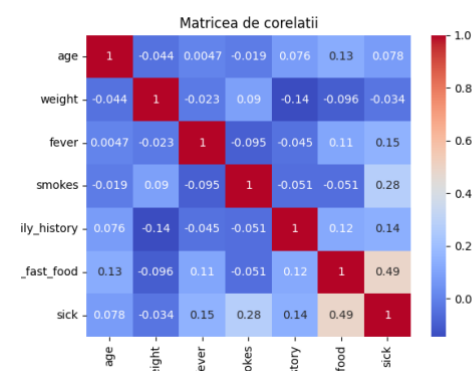
Observam ca atat pentru fever cat si pentru weight, pentru m=ambele seturi de date avem valori aberante ( in intervalele pe care le-am setat), cazuri care trebuie tratate separate pentru a nu afecta acuratetea modelului. Putem inlocui valorile aberante cu media, desi in realitate aceste valori aberante pot semnifica o afectiune posibil grava, caz in care alegem sa le normalizam.

- Analiza corelatiilor: ne vom folosi de matrice de corelatii pentru variabilele numerice. Variabila coeficientului de corelatie variaza intre -1 si 1: -1 reprezinta o corelatie negativa (cand prima variabila creste, cea de-a doua creste in acelasi ritm, 0 absenta unei corelatii (variabilele sunt independente) iau 1 o corelatie pozitiva (cand prima variabila creste, cealalta scade). Facem acest lucru pentru ambele seturi de date. Primul lucru pe care il facem este sa selectam coloanele numerice si sa calculam corelatia.

Train:



Test:



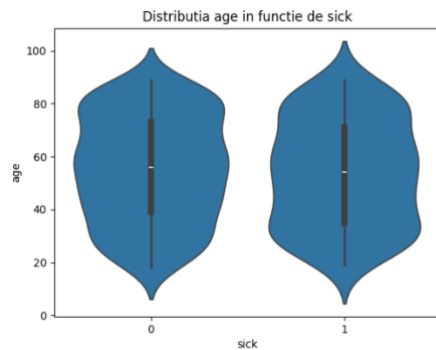
Ce observam: pentru ambele seturi de date, corelatia dintre factorii definiti (age, weight, fever, smokes, life\_history, eats\_fast\_food) este una slaba (tinde spre 0) ceea ce inseamna ca nu se influenteaza reciproc. Analizand corelatia dintre sick si celelalte variabile constatam o legatura puternica pozitiva cu smokes si eats\_fast\_food, una moderata pozitiva cu fever si family\_history). Uitandu-ne la corelatia dintre sick si age pentru train observam o corelatie negativa, ceea ce inseamna ca oamenii varstnici sunt mai putin predispusi decat cei tineri la a se imbolnavi (desi initial pare aberant, deoarece oamenii varstnici de obicei sufera de mai multe afectiuni decat cei tineri, acest rezultat se datoreaza faptului ca nu am introdus varsta ca un factor de risc atunci cand am calculat sick)

Ce idei putem formula: cei mai considerabili factori de risc sunt smokes si eats\_fast\_food, iar cei mai putin semnificativi sunt age si weight.

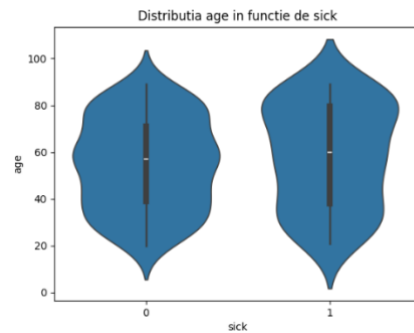
Ce preprocesari ar trebui sa aplicam: neavand corelatii mari intre variabile (excluzand sick) modelul functioneaza corect si nu trebuie sa aplicam preprocesari.

8. Analiza relatiilor cu variabila tinta: generam grafice de tip violin pentru fiecare variabila numerica, nebinara (age, weight, fever) in functie de sick pentru a evidentia relatiile dintre acestea.

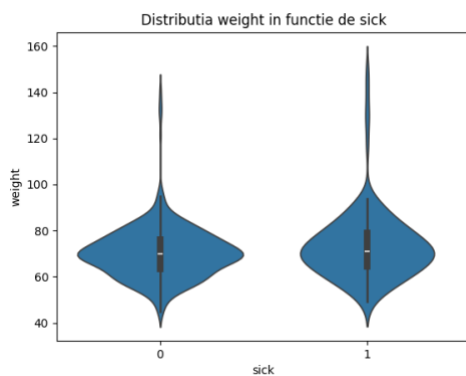
Train, age:



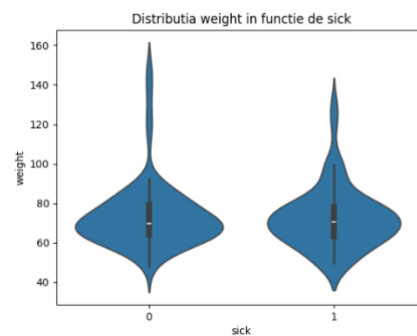
Test, age:



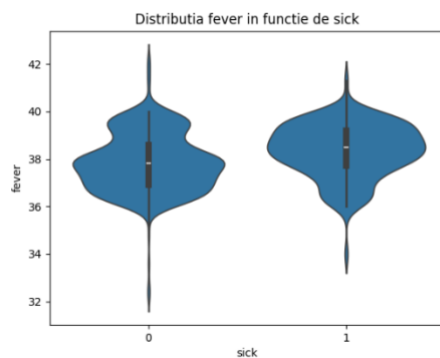
Train, weight:



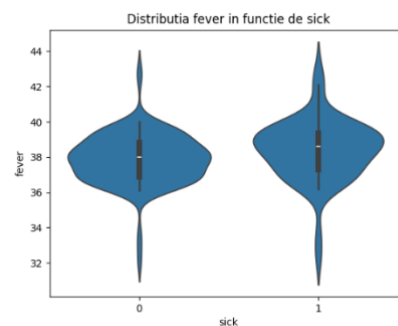
Test, weight:



Train, fever:



Test, fever:



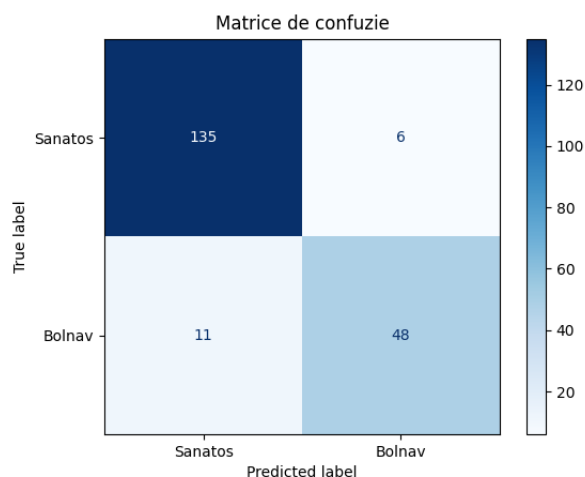
9. Antrenarea modelului: aceasta parte a proiectului se ocupa de preprocesarea datelor si de antrenarea modelului. Mai intai combinam cele doua seturi de date, pentru a aplica transformarile o singura data. Valorile catelogice sunt convertite in valori numerice folosind



label encoder, iar cele numerice (nebinare) sunt scalate cu standard scaler. După procesarea datelor, acestea sunt separate înapoi în setul de antrenament și cel de test. Fiecare set este împărțit astfel: într-o matrice ( $X_{train}$  sau  $X_{test}$ ) ce conține toate variabilele mai puțin `sick` și într-un vector ( $y_{train}$  și  $y_{test}$ ) ce conține doar variabila `sick`. Matricele  $X$  reprezintă practic inputurile (liste de pacienți, fiecare pacient fiind caracterizat prin variabile) iar  $y$  outputurile (fiecare linie reprezintă diagnosticarea pacientului). Cream modelul de regresie logistică pe care îl antrenăm cu  $X_{train}$  și  $y_{train}$ . Folosim modelul pentru a face predicții pe setul de testare ( $X_{test}$ ), rezultându-ne un output  $y_{pred}$  cu predicțiile realizate cu metoda logistic regression. Practic  $y_{test}$  reprezintă valorile reale ale setului de testare, iar  $y_{pred}$  reprezintă predicțiile modelului. Comparând acești doi vectori putem să evaluăm modelul.

- Interpretarea datelor: se realizează folosind `classification_report` care oferă evaluarea predicțiilor. Practic analizăm dacă modelul este sau nu corect: dacă este total corect atunci  $y_{test}$  și  $y_{pred}$  vor coincide. Această funcție ne returnează acuratețea, precizia, recall și f1-score. De asemenea pentru a analiza și grafic rezultatele ne vom folosi de:

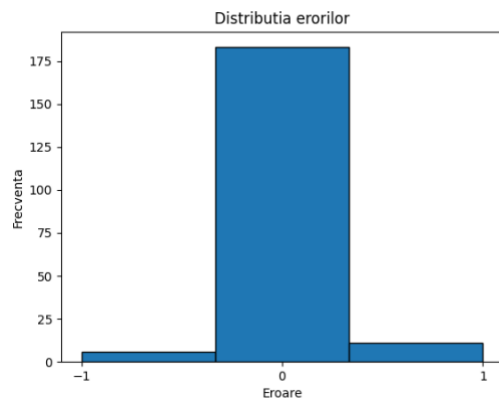
Matrice de confuzie:



Analizând acest graphic observăm că din cei 200 de pacienți:

- 135 erau sănătoși și au fost diagnosticați ca fiind sănătoși
- 48 erau bolnavi și au fost diagnosticați ca fiind bolnavi
- 6 erau sănătoși și au fost diagnosticați ca fiind bolnavi
- 11 erau bolnavi și au fost diagnosticați ca fiind sănătoși. Acest aspect este unul negativ, deoarece într-un scenariu real, pacienți care ar fi avut nevoie de tratament, nu îl vor primi, ceea ce poate aduce consecințe grave, spre deosebire de cei sănătoși și catalogați bolnavi.

Grafic de erori (histograma):



0 – prezis corect

-1 bolnav si diagnosticat snatos

1 sanatos si diagnosticat bolnav

11. Salvarea datelor: salvam predictiile intr-un fisier csv