Chiselita Sebastian, Seria CA, Grupa 314

Proiect atribuire bautura

Scopul modelului creat de mine este sa prezica ce bautura i se potriveste unei persoane in funtie de preferinte, factori externi si stari de spirit.

Detaliile implementarii:

1) Tipul problemei:

Setul de date este destinate unei **probleme de clasificare**.

2) Structura setului de date:

Subsetul de antrenare are 599 de instante (randuri);

Subsetul de testare are 241 de instante (randuri).

3) Numărul minim de caracteristici:
Fiecare instanta are 11 coloane relevante, inclusive coloana tinta. Avem mai multe tipuri de date: numere intregi, numere reale, valori categorice sub forma de siruri de caractere.

4) Salvare dataseturilor:

Subsetul de antrenare a fost salvat in fisierul train.csv;

Subsetul de testare a fost salvat in fisierul test.csv.

5) Documentatie:

Setul de date este sintetic si a fost construit cu ajutorul unui fisier sursa .py. Am generat valori aleatoare pentru mai multe cateogorii care reprezinta preferintele unei persoane, precum si factori externi ca sa determine ce bautura i se potriveste la momentul respectiv:

'nr_ore_somn': cate ore a dormit individul noaptea precedenta?

'multa_treaba': sunt multe de realizat in present?

'energie': cat de energica se simte persoana?

'fericire': cat de fericit este omul?

'alcool': prefera bauturile alcoolice?

'cat_de_acru': cata acrime prefera?

'cantitate_zahar': cat zahat prefera?

'vreme': cum e vremea afara?

'temperatura': cate grade sunt afara?

'stres': cat de stresat esti?

Ulterior, am pus niste conditii intr-o functie care sa determine bautura potrivita in functie de acesti factori randomizati. Am construit data frame-ul cu dimensiunea corespunzatoare si am aplicat functia de determinare a bauturii.

La final am impartit setul de date intr-un set de antrenare si altul de testare, mai mic.

6) Analiza exploratorie a datelor (EDA complex):

Toata datele din analiza exploratorie au fost realizate cu fisierul sursa *analiza_exploratorie.py.*

a) Analiza valorilor lipsă:

In datele create nu exista valori lipsa, deorece le-am creat sintetic astfel incat fiecare casuta sa aiba o valoare, dupa cum se observa si in output-ul fisierului sursa:

```
Valori lipsa in train:
 nr_ore_somn            0
multa_treaba            0
energie                 0
fericire                0
alcool                  0
cat_de_acru             0
cantitate_zahar         0
vreme                   0
temperatura             0
stres                   0
bautura_pentru_tine     0
dtype: int64
```

```
Valori lipsa in test:
 nr_ore_somn            0
multa_treaba            0
energie                 0
fericire                0
alcool                  0
cat_de_acru             0
cantitate_zahar         0
vreme                   0
temperatura             0
stres                   0
bautura_pentru_tine     0
dtype: int64
```

b) Statistici descriptive:

Pentru ambele seturi am afisat **descrierea numerica**, sub forma de tablel ce contine informatii precum: count , media, min, max.

Descriere numerica train:

	nr_ore_somn	fericire	temperatura	stres
count	599.000000	599.000000	599.000000	599.000000
mean	7.332220	5.050568	9.969950	4.928564
std	2.315748	2.905037	11.749294	2.885661
min	4.000000	0.010000	-10.000000	0.010000
25%	5.000000	2.480000	0.000000	2.435000
50%	7.000000	5.030000	10.000000	4.980000
75%	9.000000	7.550000	20.000000	7.360000
max	11.000000	9.990000	29.000000	9.970000

Descriere numerica test:

	nr_ore_somn	fericire	temperatura	stres
count	241.000000	241.000000	241.000000	241.000000
mean	7.688797	5.137759	10.049793	5.110539
std	2.344906	2.909124	11.843670	2.838162
min	4.000000	0.090000	-10.000000	0.030000
25%	6.000000	2.640000	0.000000	2.750000
50%	8.000000	5.340000	11.000000	5.110000
75%	10.000000	7.760000	21.000000	7.520000
max	11.000000	10.000000	29.000000	9.960000

Pentru valorile categorice am afisat **numarul de elemente pentru fiecare categorie**:

Numar de elemente pe categorie train:

multa_treaba	2
energie	2
alcool	2
cat_de_acru	2
cantitate_zahar	2
vreme	3
bautura_pentru_tine	10

dtype: int64

Numar de elemente pe categorie test:

multa_treaba	2
energie	2
alcool	2
cat_de_acru	2
cantitate_zahar	2
vreme	3
bautura_pentru_tine	10

dtype: int64

De asemenea, pentru fiecare categorie, am afisat **de cate ori apare fiecare element**:

Descriere valori categorice train:

Categoria: multa_treaba

Numar aparitii elemente:

multa_treaba	
da	318
nu	281

Name: count, dtype: int64

Categoria: energie

Numar aparitii elemente:

energie	
multa	300
putina	299

Name: count, dtype: int64

Categoria: alcool

Numar aparitii elemente:

alcool	
nu	309
da	290

Name: count, dtype: int64

Categoria: cat_de_acru

Numar aparitii elemente:

cat_de_acru	
multa	308
putina	291

Name: count, dtype: int64

Categoria: cantitate_zahar

Numar aparitii elemente:

cantitate_zahar	
mult	306
putin	293

Name: count, dtype: int64

Categoria: vreme

Numar aparitii elemente:

vreme	
insorit	207
racoros	203
ploios	189

Name: count, dtype: int64

Categoria: bautura_pentru_tine

Numar aparitii elemente:

bautura_pentru_tine	
cocktail	128
bere	74
ceai	63
limonada	58
suc	58
cafea	57
vin	48
ciocolata calda	42
apa	41
compot	30

Name: count, dtype: int64

Descriere valori categorice test:

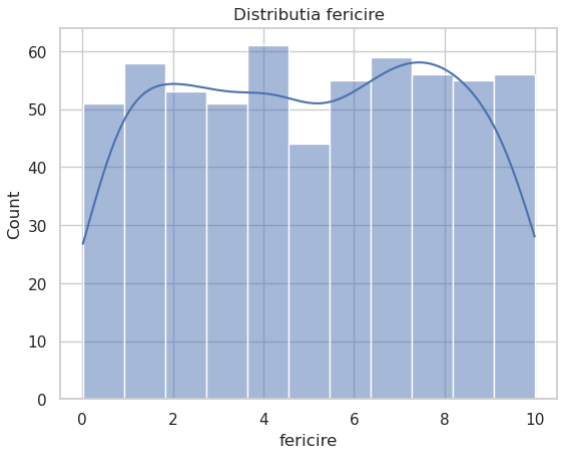
```
Categoria: multa_treaba

Numar aparitii elemente:
multa_treaba
nu    121
da    120
Name: count, dtype: int64

Categoria: energie

Numar aparitii elemente:
energie
multa     128
putina    113
Name: count, dtype: int64

Categoria: alcool

Numar aparitii elemente:
alcool
da    124
nu    117
Name: count, dtype: int64

Categoria: cat_de_acru

Numar aparitii elemente:
cat_de_acru
putina    125
multa     116
Name: count, dtype: int64
```

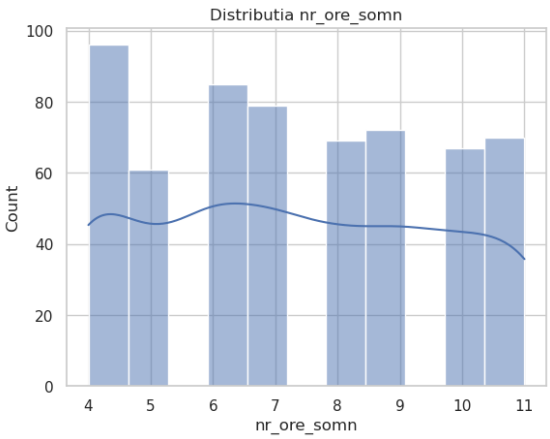
```
Categoria: cantitate_zahar

Numar aparitii elemente:
cantitate_zahar
mult     123
putin    118
Name: count, dtype: int64

Categoria: vreme

Numar aparitii elemente:
vreme
racoros    91
ploios     80
insorit    70
Name: count, dtype: int64

Categoria: bautura_pentru_tine

Numar aparitii elemente:
bautura_pentru_tine
cocktail           59
bere               33
suc                33
ceai               25
cafea              19
ciocolata calda    18
vin                18
apa                16
compot             13
limonada            7
Name: count, dtype: int64
```

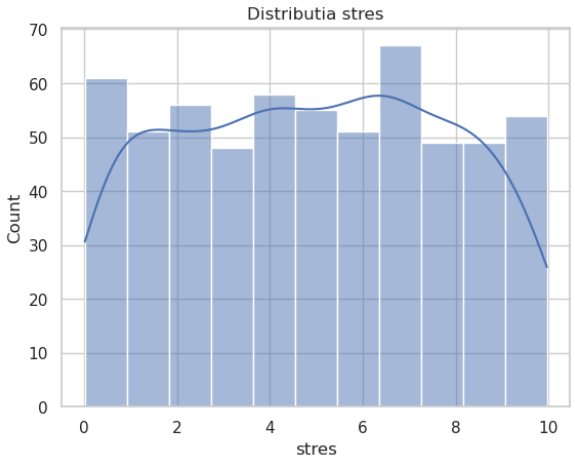
c) Analiza distributiei variabilelor:

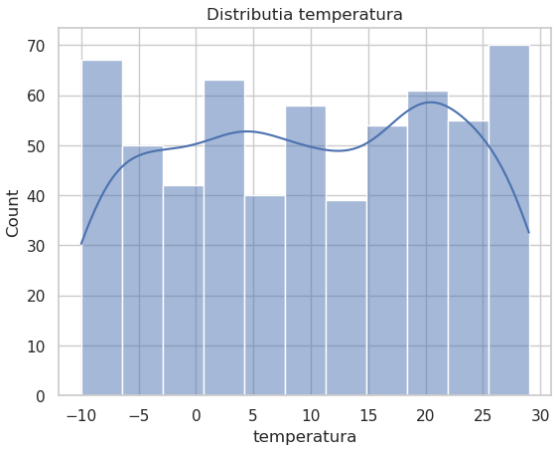
Histogramele pentru categoriile numerice:

Subsetul de antrenare:

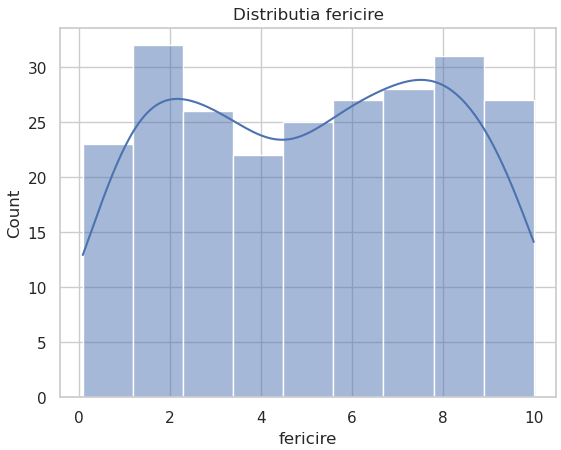


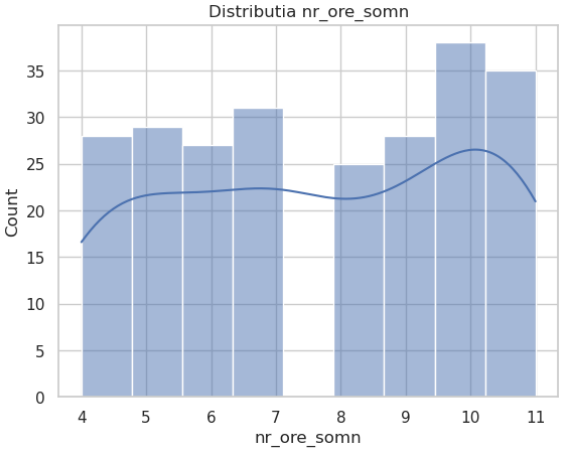


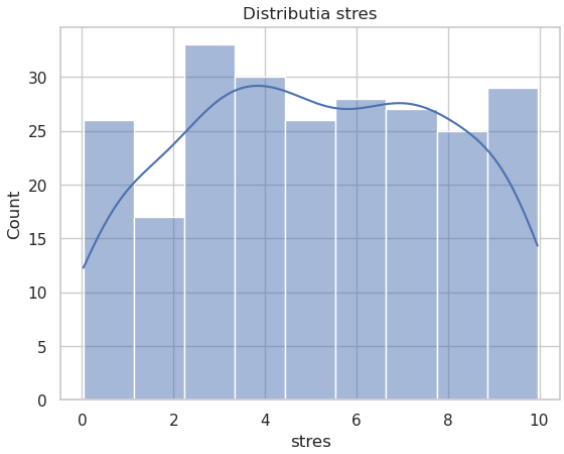


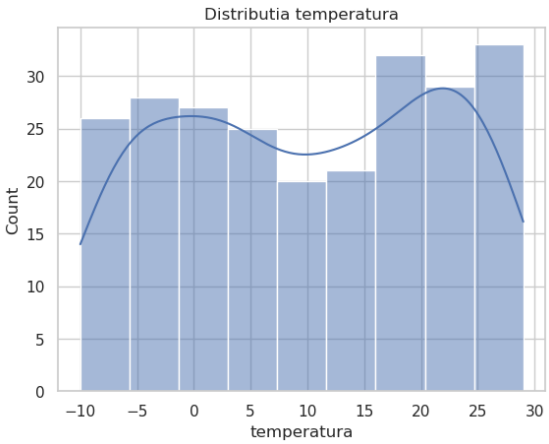


Subsetul de testare:









Observatii:

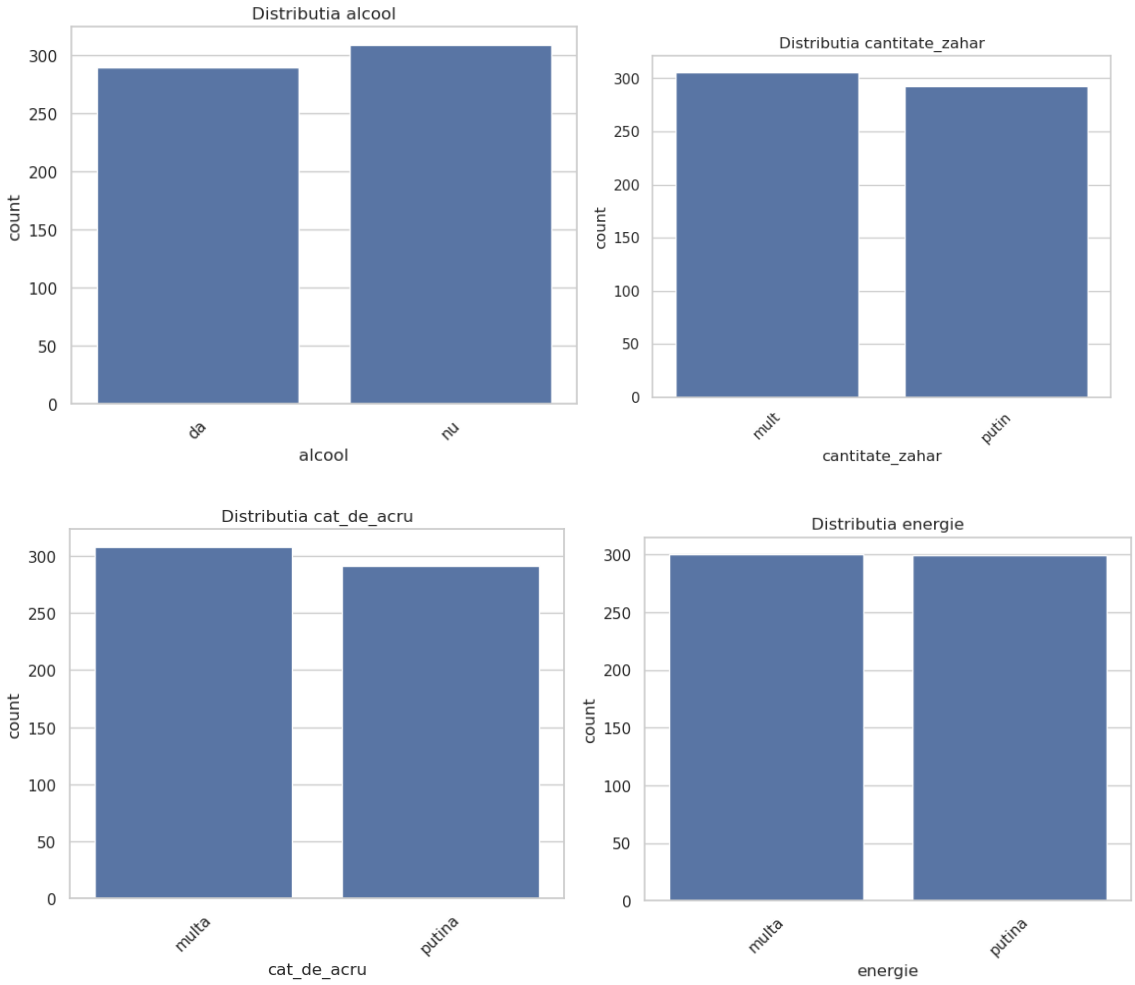
Din moment ce am generat aceste date randomizat, cu sanse egale de a aparea oricare valoare posibila din domeniu, observam o distributie relativ uniforma pentru setul de antrenare unde

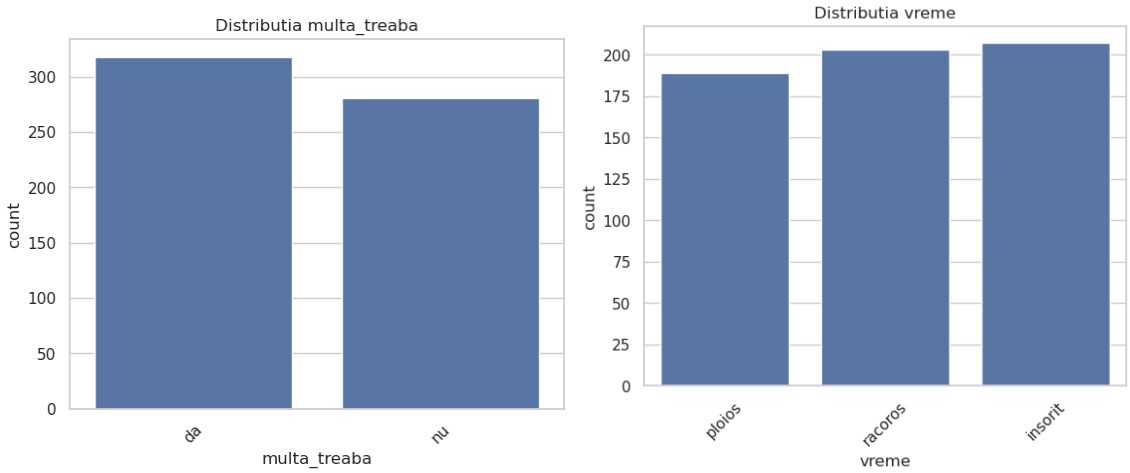
exista mai multe date, pe cand in setul de testare exista niste discrepante mai mari (putine valori pentru nivel mic de stres si pentru nivel mediu de temperatura).

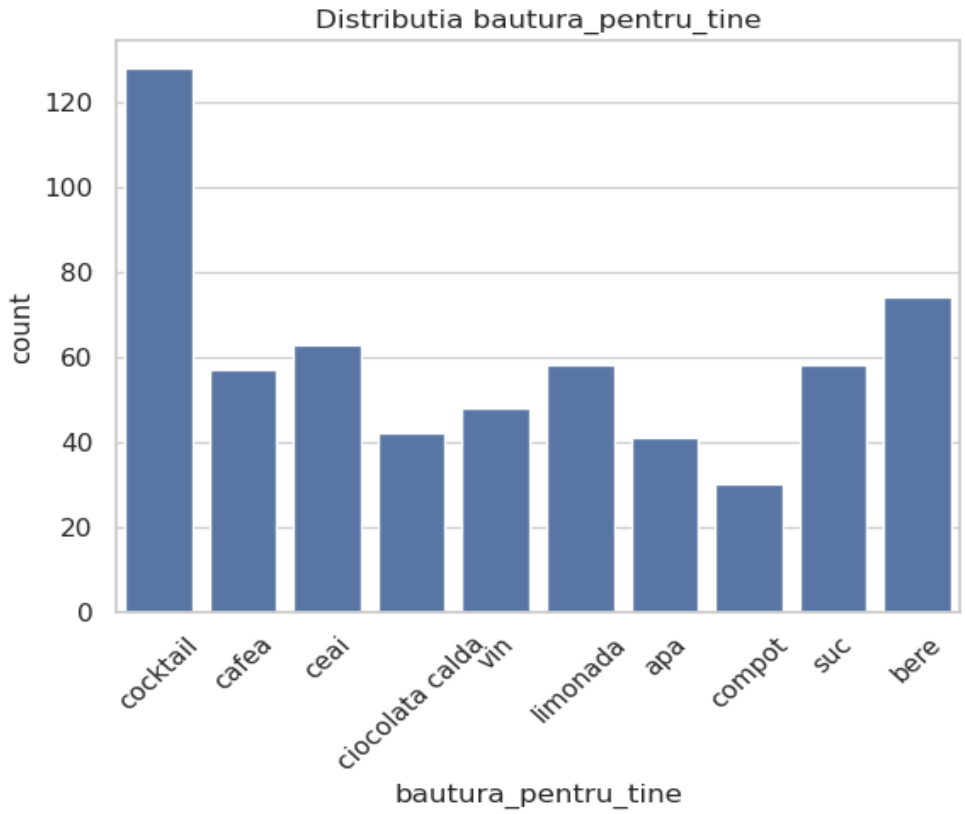
Era de asteptat distributia uniforma, dar lipsa unor valori in setul de testare poate influenta performanta modelelor.

Grafice de tip countplot pentru variabilele categorice:

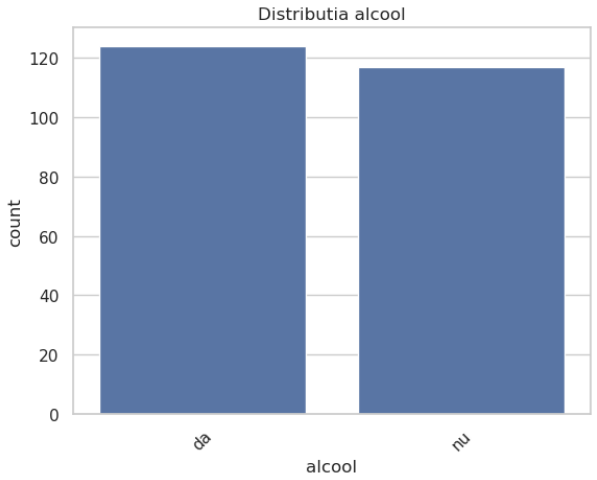
Subsetul de antrenare:

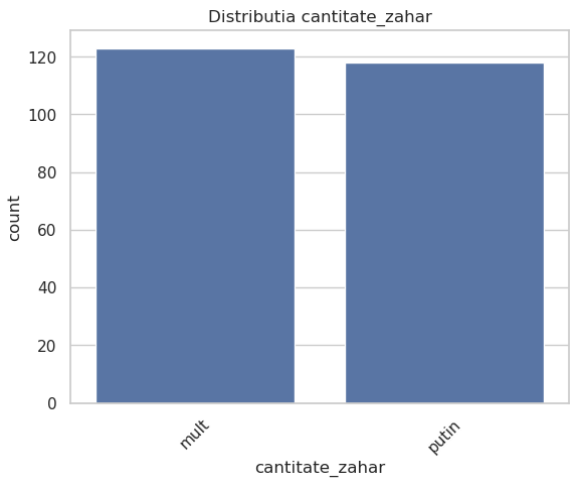


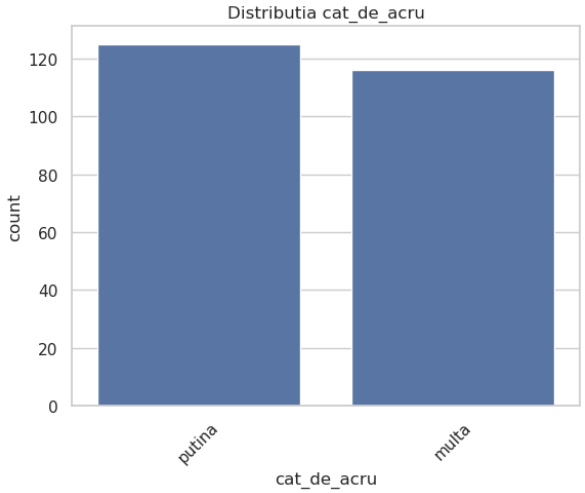


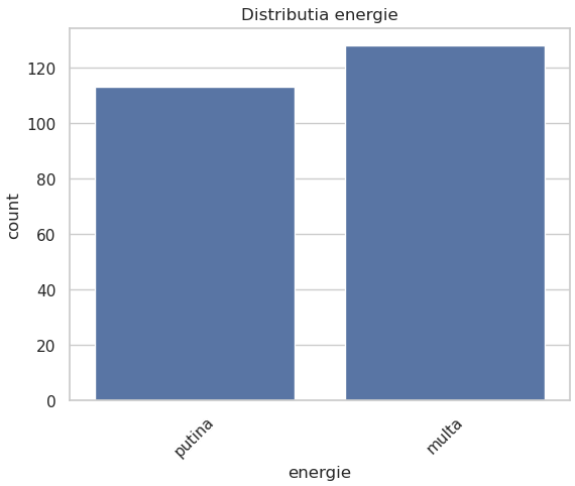


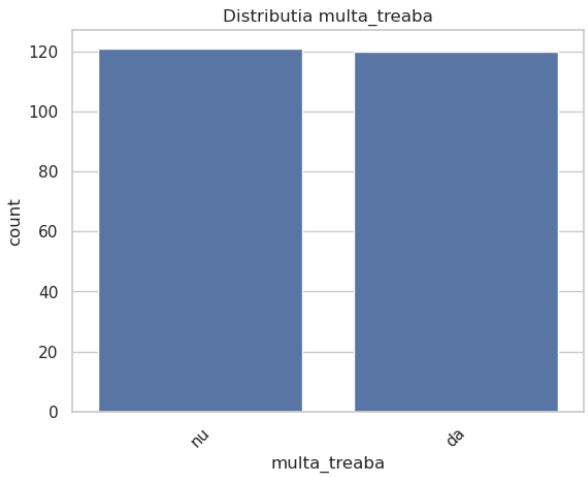
Subsetul de testare:

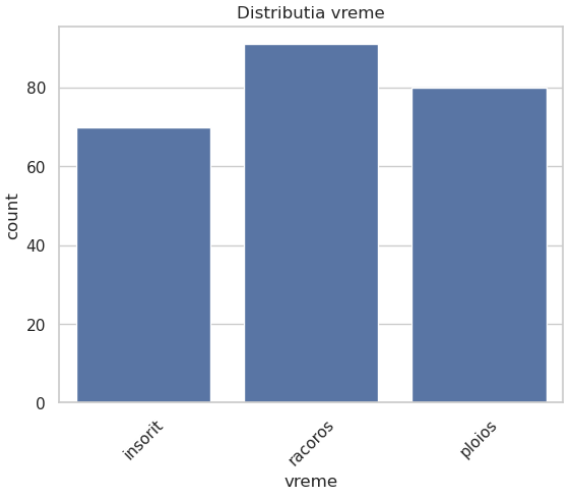

Distributia alcool
count
alcool
da
nu

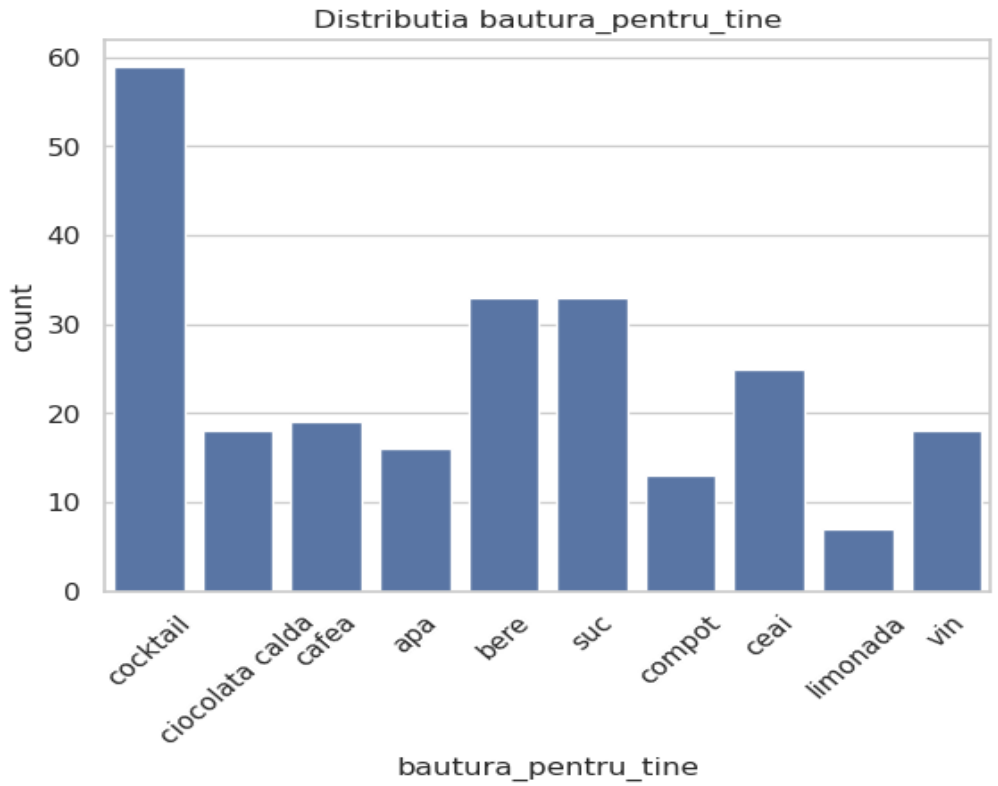

Distributia cantitate_zahar
count
cantitate_zahar
mult
putin


Distributia cat_de_acru
count
cat_de_acru
putina
multa


Distributia energie
count
energie
putina
multa


Distributia multa_treaba
count
multa_treaba
nu
da


Distributia vreme
count
vreme
insorit
racoros
ploios



Observatii:

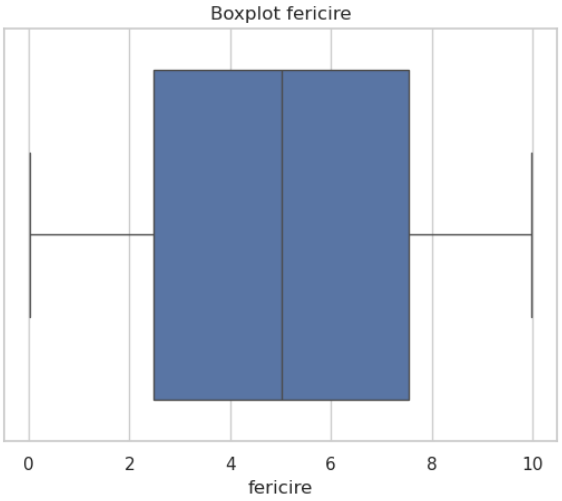
In ambele subseturi, majoritatea categoriilor au o distributie aproape egala a elementelor. Acest fapt se datoreaza numarului mic de elemente distince pentru un numar marice de date.

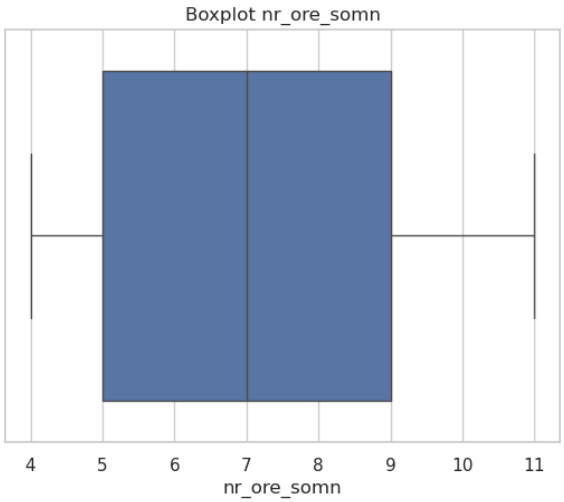
Exceptia o gasim in cazul categoriei tinta, *bautura_pentru_tine*. Aici avem 10 elemente, si se observa dominanta elementului cocktail, avand aproape dublul apariei elementului al doilea cel mai prezent. Intre cele 2 substeuri, exista discrepante pentru *limonada* (bautura apare foarte putin in setul de test, si e semnificativ mai prezenta in setul de antrenament.

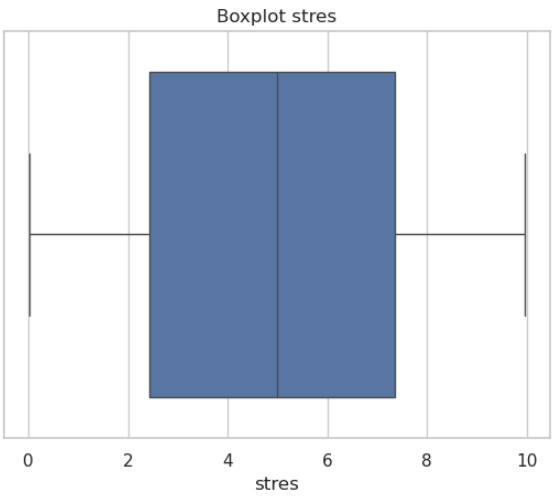
d) Detectarea outlierilor:

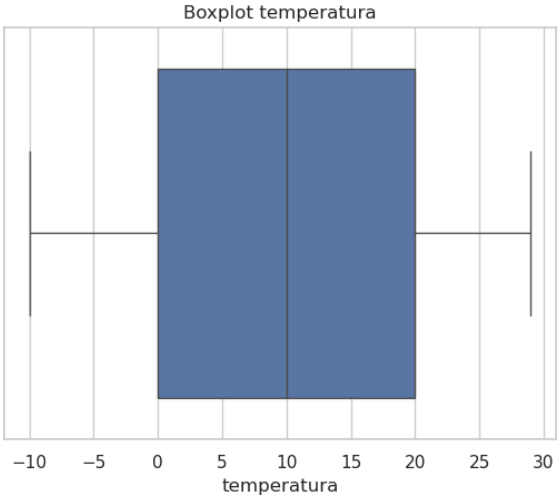
Utilizarea boxplot-urilor pentru identificarea valorilor numerice aberante:

Subsetul de antrenare:

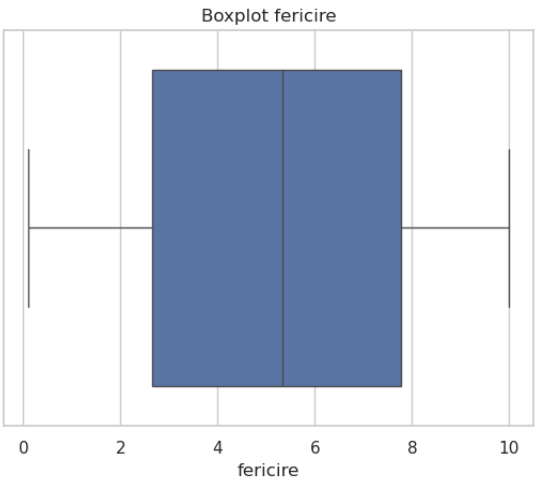


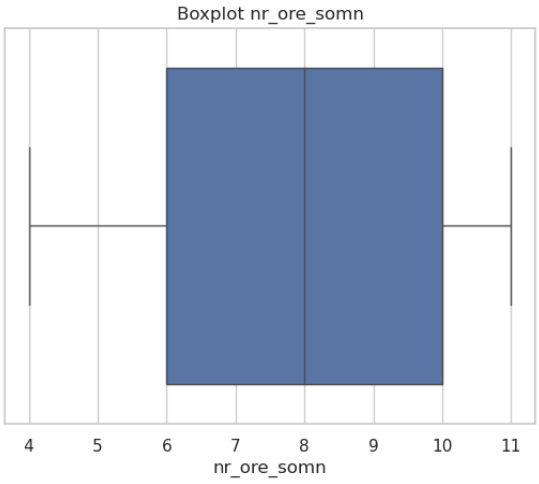


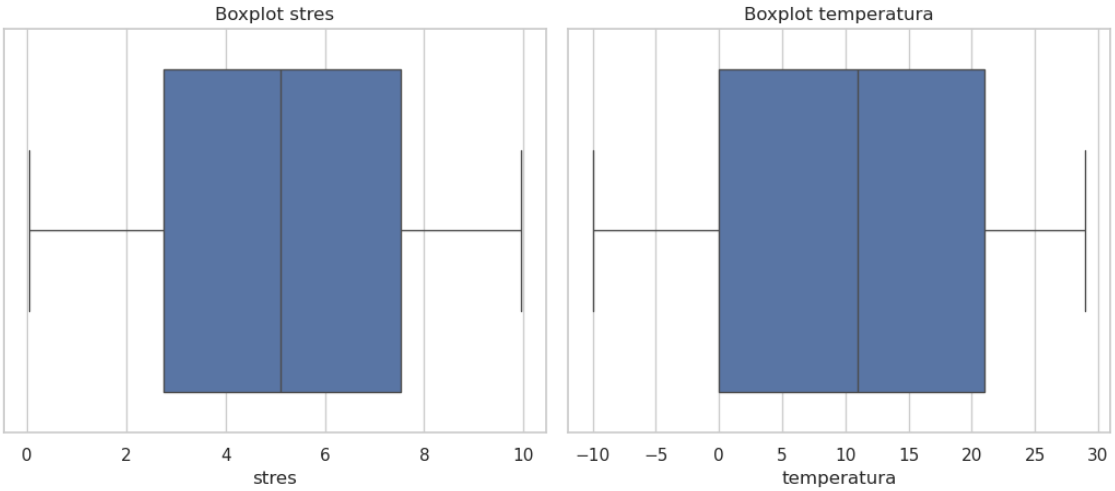




Subsetul de testare:







Observatii:

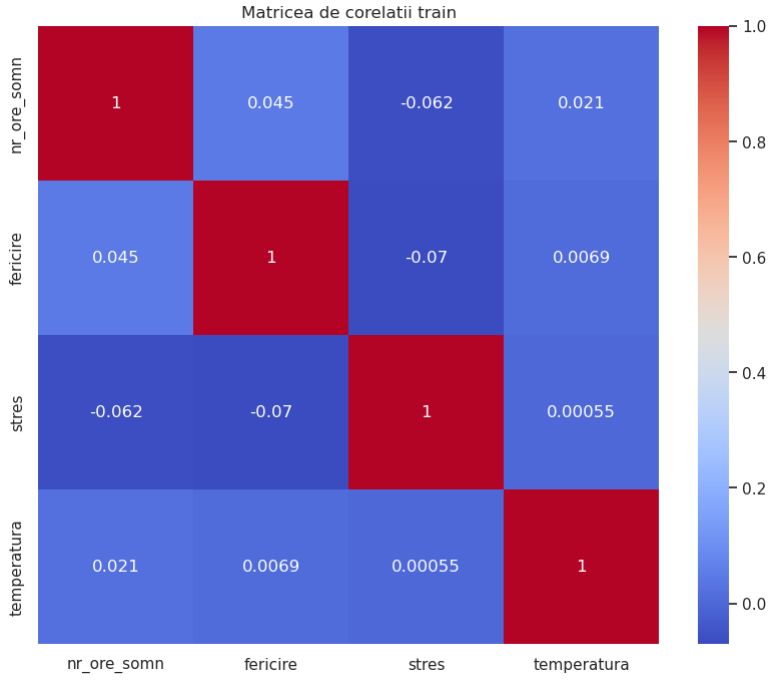
Deoarece am generat datele sintetic si am avut delimitari bine definite nu exista valori aberante pentru subseturile noastre si nu este necesar un tratament pentru outlieri.

Faptul ca nu exista outlieri poate ascunde relatii nerealiste intre varabile.

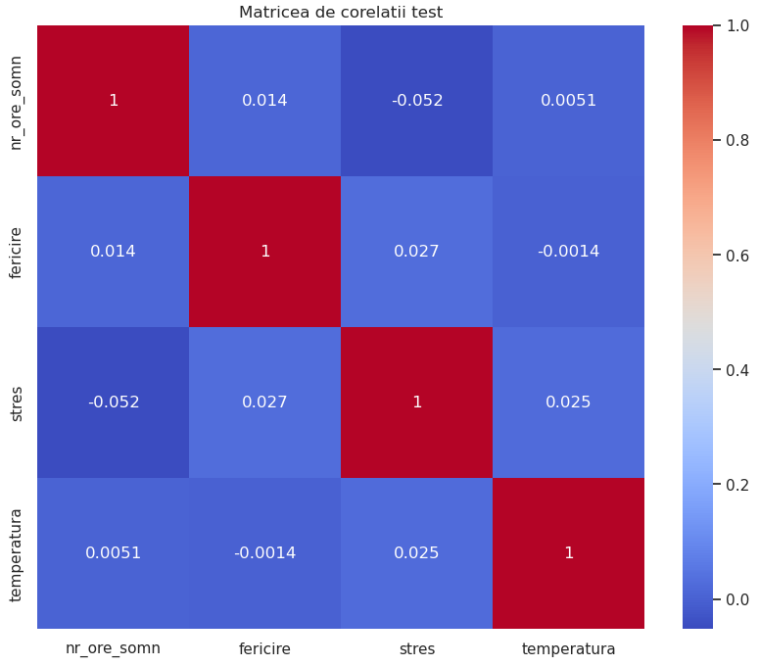
e) Analiza corelatiilor:

Utilizarea matricei de corelatii (heatmap) pentru variabilele numerice:

Subsetul de antrenare:



Subsetul de testare:



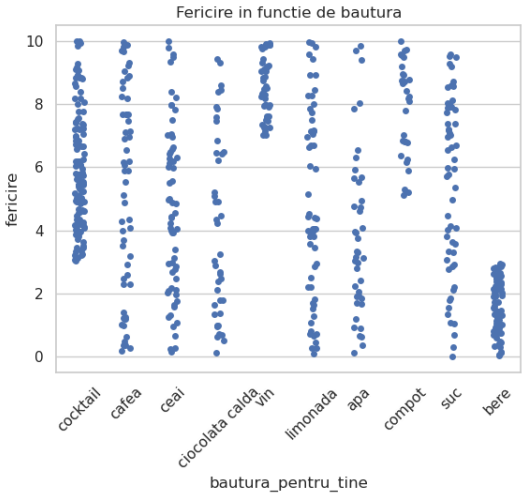
Observatii:

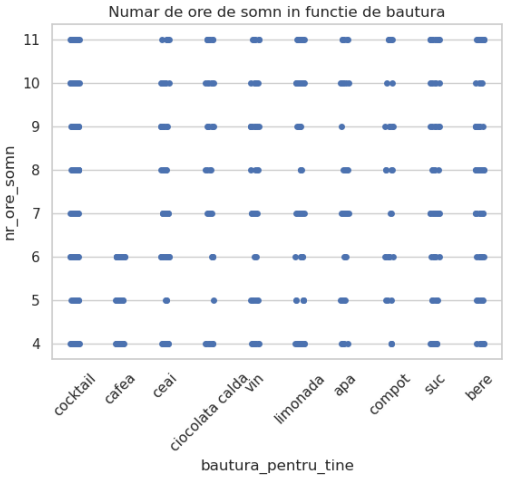
Datele au fost generate sintetic si independente unele fata de celelalte, deci nu exista corelatii semnificative intre ele. Putem concluziona ca nicio variabila numerica nu este redundanta in acest caz.

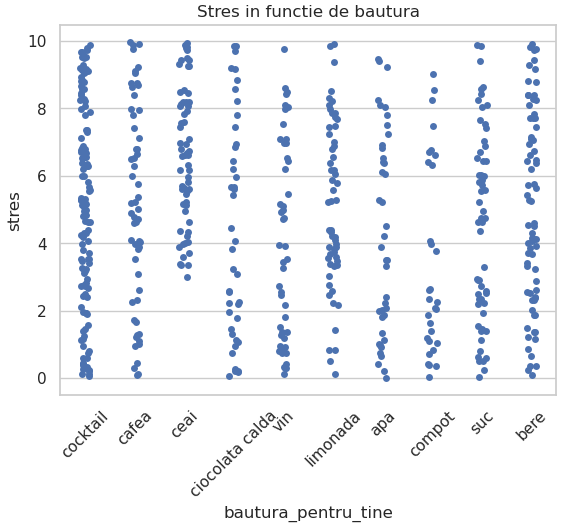
f) Analiza relatiilor cu variabila tintă:

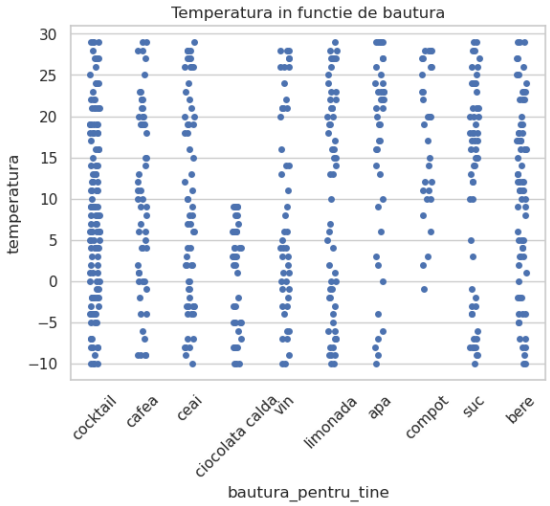
Utilizatea scatter plots pentru relatia dintre caracteristici si variabila tintă:

Subsetul de antrenare:

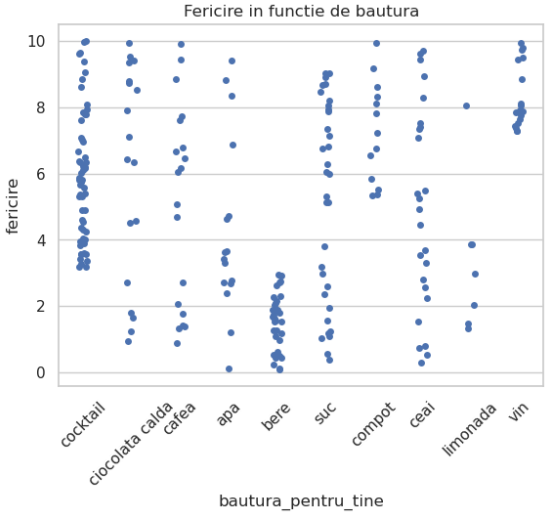


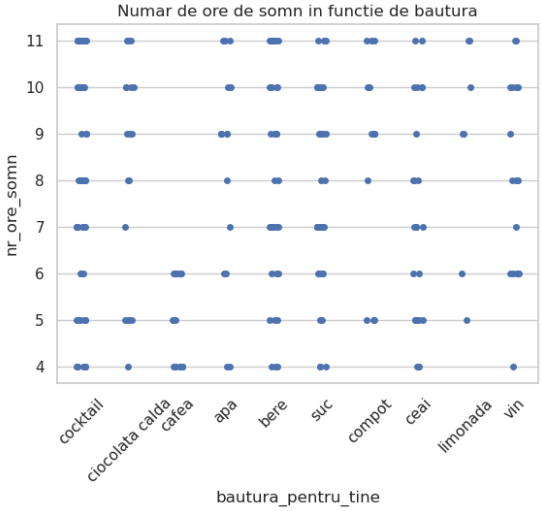


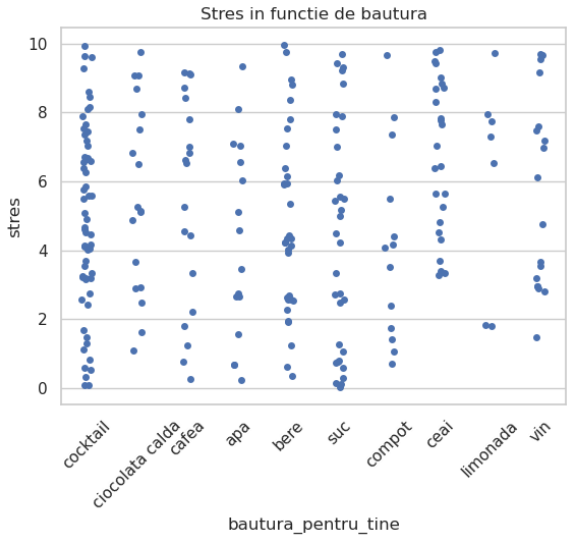


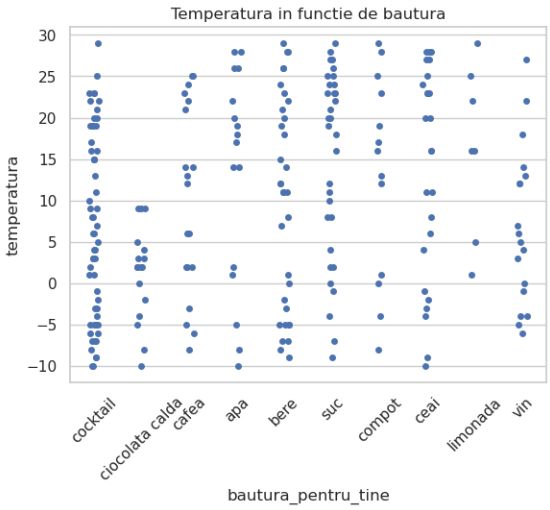


Subsetul de testare:









Observatii:

Se vad niste tendinte clare cauzate de conditiile de claficare din fisierul *build_sintetic_dataset.py*. Unele dintre acestea sunt: berea apare numai la un nivel mic de fericire, pe cand cocktail-ul, vinul si compotul sunt asociate unui nivel ridicat de fericire. Cafeaua apare numai la ore mici de somn. Ceaiul iese la iveala numai cand nivelul de stres e redicat, iar compotul la un nivel scazut, desi exista exceptii. Ciocolata calda se bea cand e frig afara, iar compotul in mare parte cand e mai cald.

Modelul de clasificare va avea performanta buna atata timp cat aceste reguli raman consistente, insa unele informatii pot parea superficiale.

7. Antrenarea si evaluarea unui model de bază:

Avand o problema de clasificare, am decis sa fac un random forest.

Interpretari:

Acuratete:

0.946058091286307

Avem o acuratete ridicata, insa aceasta poate fi inselatoare din cauza ca unele claase sunt mul mai frecvente (cocktail fata de limonada), asa ca e nevoie sa analizam si raportul de clasificare.

Raportul de clasificare :

Raport de clasificare:

	precision	recall	f1-score	support
apa	0.92	0.69	0.79	16
bere	0.97	1.00	0.99	33
cafea	1.00	0.68	0.81	19
ceai	0.89	1.00	0.94	25
ciocolata calda	1.00	0.94	0.97	18
cocktail	0.95	1.00	0.98	59
compot	0.92	0.92	0.92	13
limonada	0.70	1.00	0.82	7
suc	0.97	1.00	0.99	33
vin	1.00	1.00	1.00	18
accuracy			0.95	241
macro avg	0.93	0.92	0.92	241
weighted avg	0.95	0.95	0.94	241

Din precizie observam ca modelul a reusit sa prezica bine majoritatea categoriilor, mai putin *limonda*, confirmand efectul daunator de a avea numai 7 instante ale acestei bauturi in subsetul de testare.

Recall-ul ne dezvaluie cate din datele reale au fost prezise corect. Observam un procent scazut la apa, ceea ce consider ca se datoreaza faptului ca apa este folosita si ca o optiune atunci cand u s-a gasit nimic altceva pentru bauturile non-alcoolice, lucru greu de gestionat pentru model. De asemenea, cafeaua nu are un recall bun desi precizia este maxima, insemnand ca deseori modelul alege alta bautura cand trebuia sa aleaga cafeaua.

F1-score-ul este media armonica dinte cele 2 categorii precedente. In general scorurile sunt foarte mari, exceptand cazurile apa, cafea si limonada.

Grafice relevante:

Matricea de confuzie:

Aceasta matrice ne dezvaluie ca in mare parte, modelul a prezis bine bauturile, cu mici exceptii. Cele mai notabile greseli sunt : apa prezisa ca un cocktail (ambele sunt bauturi accesate cand nu se gaseste nimic altceva si pot exista incurcaturi intre ele) si cafeaua prezisa ca o limonda.