Raport Tema 1 - Învățare Automată

Nume: Vîrlan Mihnea-Alexandru

Grupa: 341C4

3.1. Analiza setului de date

Cele 18 atribute ale pacienților din setul de date au prezentat o provocare, întrucât o parte considerabilă din acestea au prezentat anomalii, valori eronate, care dacă nu sunt tratate corect pot avea un impact negativ asupra performanțelor modelelor de învățare automată. În cele ce urmează, voi prezenta fiecare atribut într-o comparație before vs after. Comparația after apare numai acolo unde atributele au prezentat anomalii.

Observație: la corectarea anomaliilor din setul de date am decis să merg pe o strategie de imputare, desfășurată astfel:

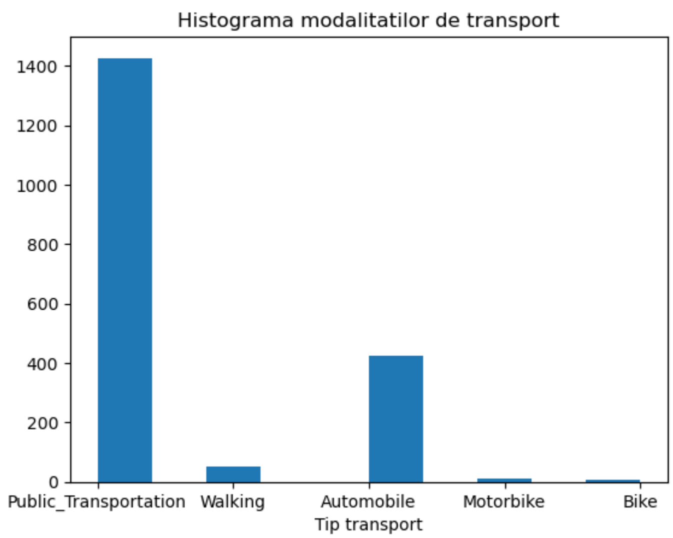
1. Identificăm indecșii din setul de date unde un atribut prezintă valori anormale după inspectarea manuală. Vrem să obținem subsecvențe consecutive unde apar anomalii.

2. Dându-se o subsecvență de indecși cu anomalii, în care 2 indecși consecutivi au diferența 1 și sunt ordonați crescător, anomaliile vor fi înlcouite corespunzător pe baza datelor normale până la apariția anomaliilor.

De exemplu: dacă anomalii pentru un atribut A au fost semnalate la indecșii 300,301,302 și 400, pentru subsecvența 300,301,302 ne vom raporta la datele normale de la indecșii 0 până la 299 inclusiv, iar după ce acestea au fost tratate, vor fi folosite pentru a trata anomalia de la indicele 400 luând toate datele de la indicele 0 la 399 inclusiv.

- Transportation

  Acest atribut indică preferințele pacienților în ceea ce privește modalitatea de transport. Fiindcă valorile posibile sunt Public_Transportation, Bike, Wallking, Automobile și Motorbike am dedus că aceasta este o variabilă categorică, drept pentru care se poate face o histogramă.

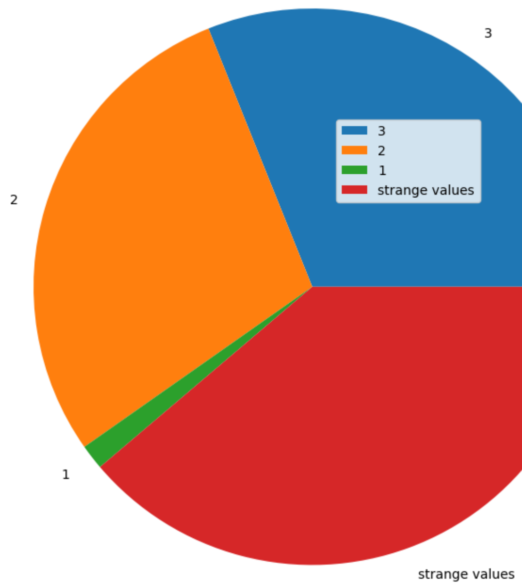


Se poate observa că majoritatea pacienților ce se prezintă la clinică preferă transportul în comun (autobuze, troleibuze, tramvaie etc.), asociat cu lipsa efortului fizic, ceea ce duce în timp la probleme de sănătate.

Acest atribut nu a prezentat outlieri.

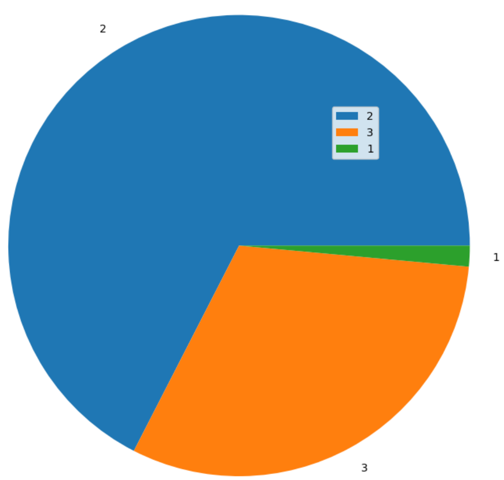
- Regular fiber_diet

Intuitiv acest atribut ar trebui să indice care este nivelul de fibre prezent în hrana consumată de fiecare pacient, ceea ce înseamnă că atributul este discret fie sub forma: Scăzut,Mediu,Mare sau 1,2,3. Din nefericire observăm în setul de date că apar niște valori anormale. Fiind vorba despre un atribut categoric, virgulele din acele numere indică ordinul miilor,zecilor de mii etc. Inițial, o histogramă pe asemenea valori e imposibil de plotat și am decis să fac un pie chart.



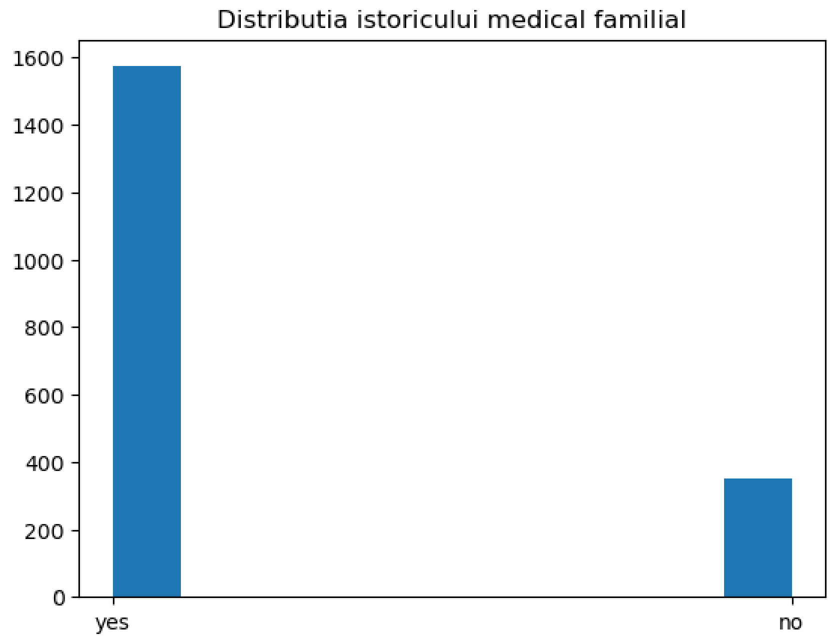
Se constată cu îngrijorare prezența semnificativă a valorilor anormale, cu un factor de alterare serios, ceea ce impune modificarea acestor atribute anormale.

Aplicăm transformarea datelor enunțată mai sus în care vom folosi pentru imputare clasa cea mai frecventă și obținem următorul pie-chart.



- Diagnostic_in_family_history

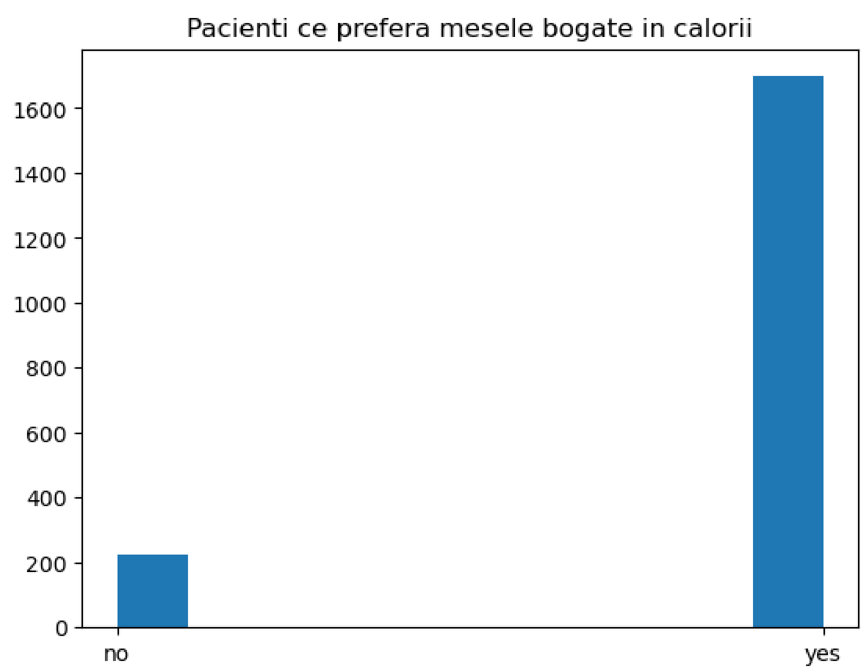
S-a constatat că este o variabilă categorică binară, deci poate fi făcută o histogramă.



Se constată că prezența unui istoric medical în familia crește șansele prezentării la medic.

- High_calorie_diet

Este o varibilă categorică binară, deci se poate face o histogramă



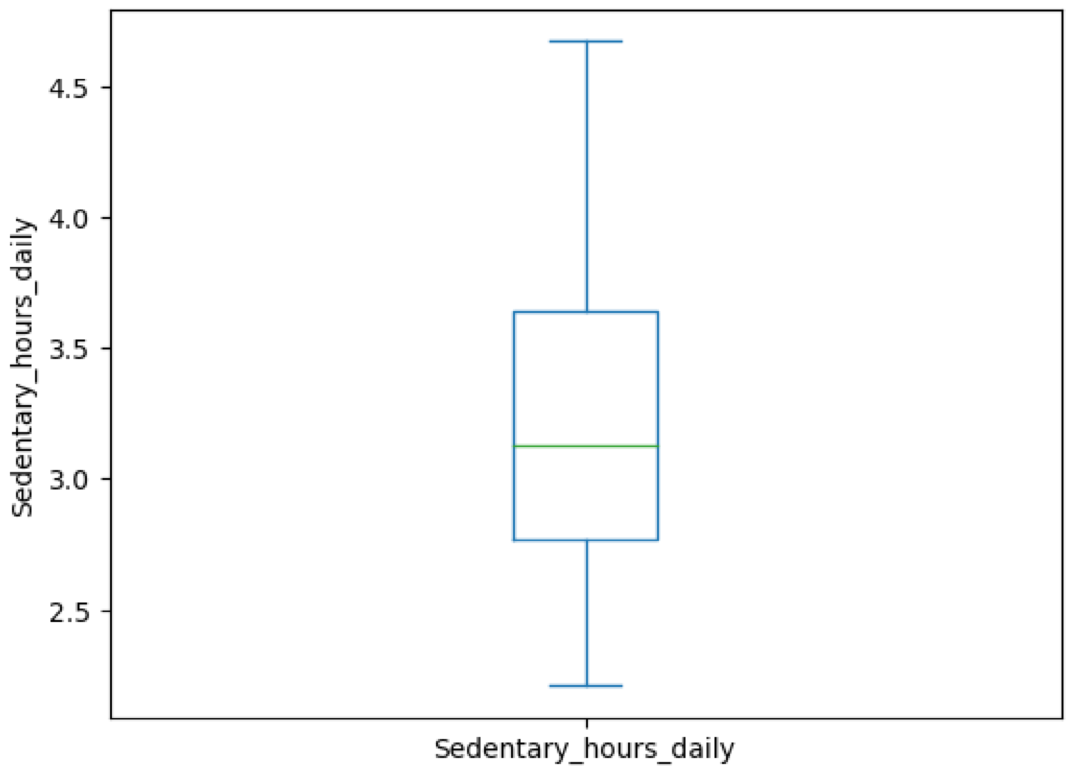
Graficul arată astfel că, la nivelul setului de date pacienții care au meniuri bogate în calorii au șanse mai mari de a se prezenta la medic.

- Sedentary_hours_daily

Am observat că acest atribut numeric arată pentru fiecare pacient media numărului de ore în care nu face mișcare.

Am constatat un outlier, anume o valoare de 956.58 care apare doar o singură dată, deci putem șterge fără probleme înregistrarea respectivă din setul de date.

După eliminarea outlier-ului, să facem un boxplot



Valorile statistice

```
count     1920.000000
mean         3.197276
std          0.575756
min          2.210000
25%          2.770000
50%          3.130000
75%          3.640000
max          4.670000
Name: Sedentary_hours_daily, dtype: float64
```

Valoarea IQR pentru acest atribut este 0.87, lucru care înseamnă că pentru 50% din aceste date, împrăștierea este una acceptabilă.

- Age

Acest atribut numeric prezintă, din păcate outlieri. După efectuarea unei sortări, analizând primele 100 de valori unice ale vârstelor am constatat niște valori nepermis de mari, pe care trebuie să le înlocuim. Vom folosi o imputare pe baza mediei după criteriul anunțat la început. Diferența cheie este că vom considera doar partea întreagă a acesteia.

```
In [53]:  1 sorted_by_age.index[:100]

Out[53]: Index([     15,      16,      17,      18,      19,      20,      21,      22,
                 23,      24,      25,      26,      27,      28,      29,      30,
                 31,      32,      33,      34,      35,      36,      37,      38,
                 39,      40,      41,      44,      45,      51,      52,      55,
                 56,      61,   19627,   19685,  176739,  203756,  209859,  210285,
             213928,  216548,  240409,  245822,  252984,  253292,  330816,  477061,
            1630687, 1638009, 1706713, 1708525, 1725813, 1770368, 1804892, 1810682,
            1813782, 1820634, 1848207, 1853084, 1883919, 1888061, 1888485, 1891505,
            1894093, 1916097, 1917614, 1921164, 1921638, 1947219, 1947554, 1959904,
            1972925, 1986997, 1988036, 1994814, 1996247, 1997166, 1997981, 2031094,
            2070768, 2081158, 2100829, 2101245, 2102064, 2102497, 2120563, 2128253,
            2167315, 2167447, 2172127, 2172738, 2181119, 2182165, 2190012, 2195994,
            2218881, 2239251, 2277789, 2287795],
           dtype='int32', name='Age')
```

Din nefericire un număr semnificativ de vârste sunt eronate

```
In [57]: abnormal_ages=dataset[dataset["Age"]>61]
         abnormal_ages
```

Out[57]:

	Transportation	Regular_fiber_diet	Diagnostic_in_family_history	High_calorie_diet	Sedentary_hours_daily	Age	Alcohol	Est_avg_calorie_intak
457	Public_Transportation	3	yes	yes	3.14	25196214	Sometimes	165
458	Public_Transportation	3	yes	yes	3.81	18503343	Sometimes	215
460	Public_Transportation	3	yes	yes	3.17	21853826	Sometimes	262
461	Public_Transportation	3	yes	yes	2.39	2190012	Sometimes	155
462	Public_Transportation	3	yes	yes	3.64	18306615	Sometimes	173
...	...	...	...	...	...	...	...	.
1916	Public_Transportation	3	yes	yes	3.08	20976842	Sometimes	274
1917	Public_Transportation	3	yes	yes	3.00	21982942	Sometimes	297
1918	Public_Transportation	3	yes	yes	3.26	22524036	Sometimes	242
1919	Public_Transportation	3	yes	yes	3.61	24361936	Sometimes	237
1920	Public_Transportation	3	yes	yes	3.83	23664709	Sometimes	233

1252 rows × 19 columns

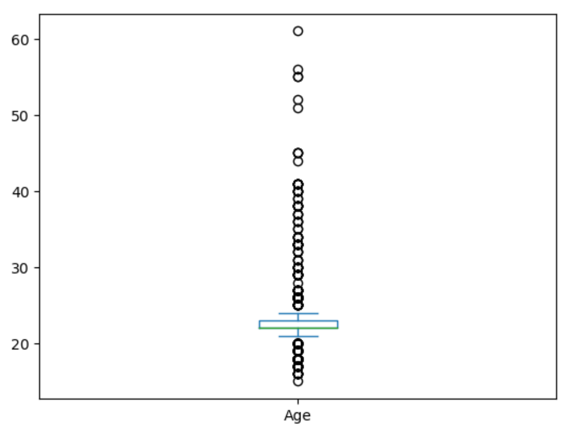
La final obținem următoarele statistici

```
In [44]: dataset["Age"].describe()

Out[44]: count    1920.000000
         mean       22.489062
         std         3.651991
         min        15.000000
         25%        22.000000
         50%        22.000000
         75%        23.000000
         max        61.000000
         Name: Age, dtype: float64
```

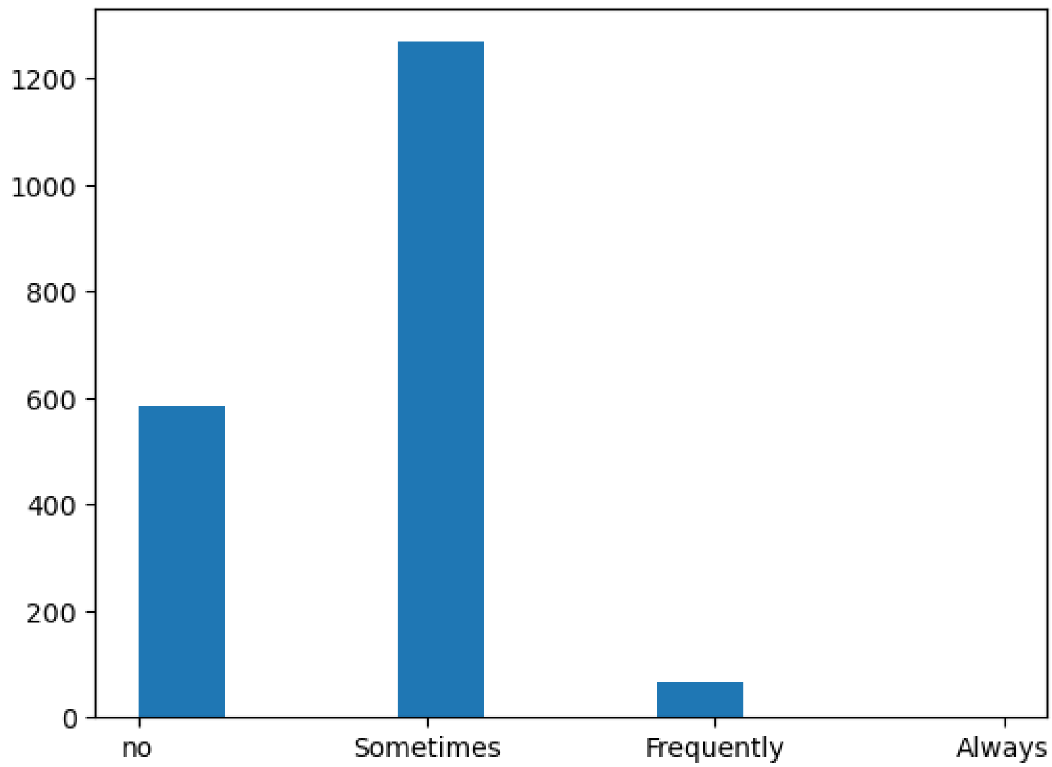
Valoarea IQR este 1.

La final, facem un boxplot și constatăm că rămânem în continuare cu outlieri, de data asta, avand inclusiv vârste considerabil tinere (30,40 de ani). Acest lucru că o contaminare a unui atribut cu valori nepermis de mari duce la dezechilibre majore.



- Alcohol

Este o variabilă categorică având 4 valori posibile, deci se poate plota o histogramă.

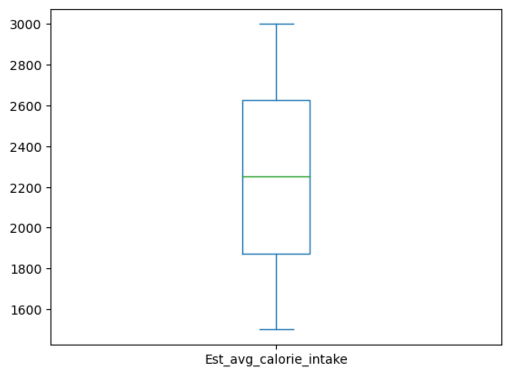


Vedem că majoritatea pacienților consumă ocazional alcool.

- Est_avg_calorie_intake

Este un atribut numeric prin care vedem care este aportul caloric mediu al unui pacient. Nu s-au constatat valori anormale, deci putem face un boxplot.

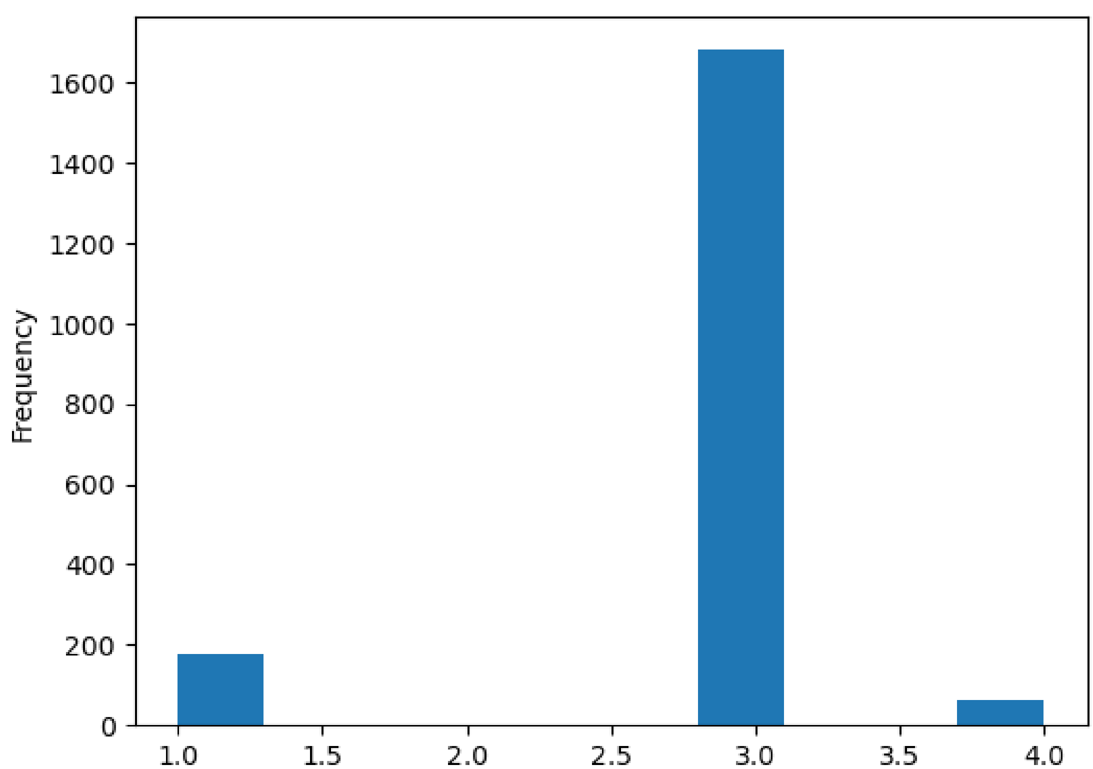
```
count    1920.000000
mean     2253.405208
std       434.012263
min      1500.000000
25%      1870.250000
50%      2252.500000
75%      2628.000000
max      3000.000000
Name: Est_avg_calorie_intake, dtype: float64
```



Vedem că nu există outlieri, deci putem presupune corectitudinea datelor acestui atribut. Valoarea IQR este 757.75, deci pentru 50% din pacienți avem o varietate considerabilă pentru acest atribut.

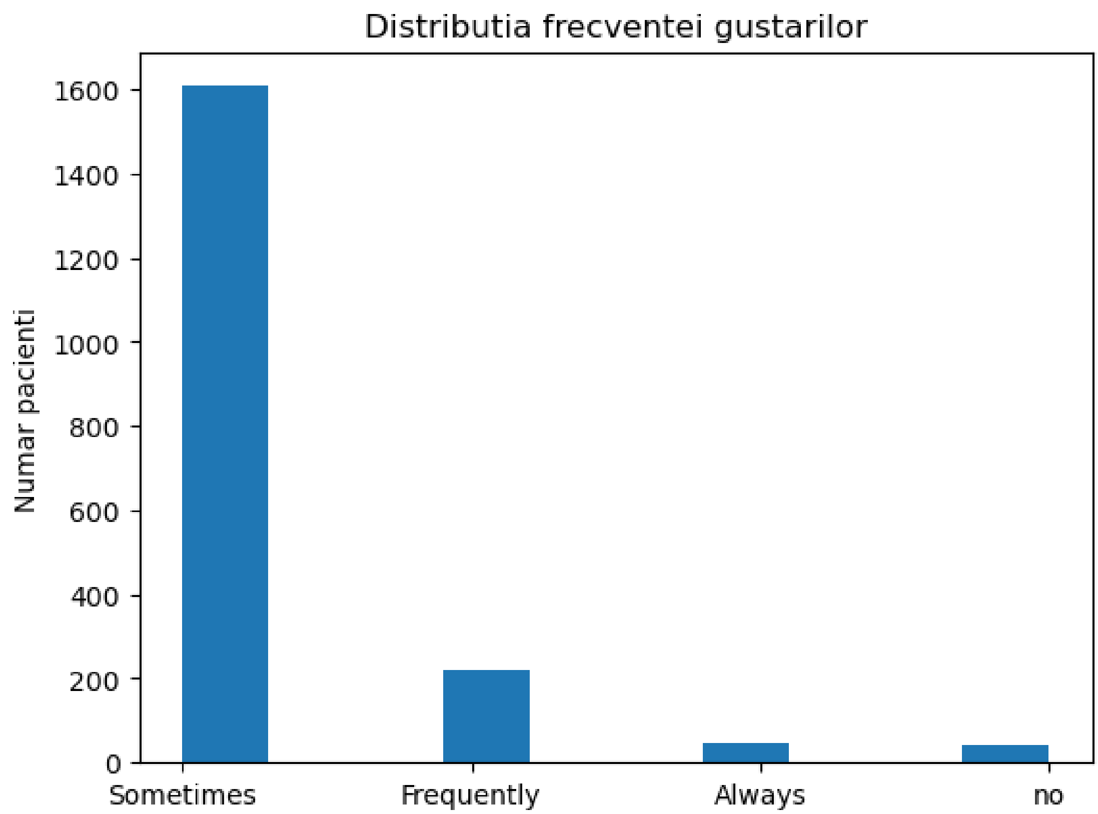
- Main_meals_daily

Este un atribut care arată pentru fiecare pacient, câte din cele 3 mese principale sunt servite. Aici, ar trebui să avem valori discrete foarte mici, dar setul de date este contaminat. Putem considera acest atribut ca fiind categoric. La final, obținem această histogramă.



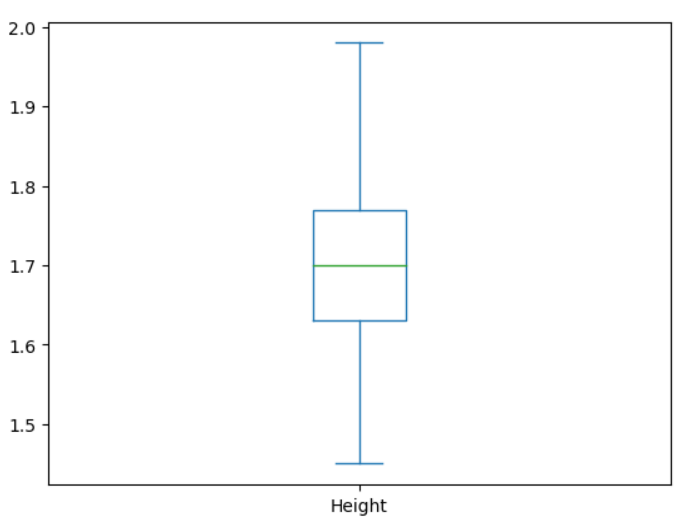
- Snacks

Este o variabilă categorică, cu 4 valori posibile, deci putem face o histogramă.



- Height este un atribut numeric, număr real. În setul de date apar niște valori greșite pe care vrem să le corectăm.

După corecție avem acest boxplot



Valori statistice

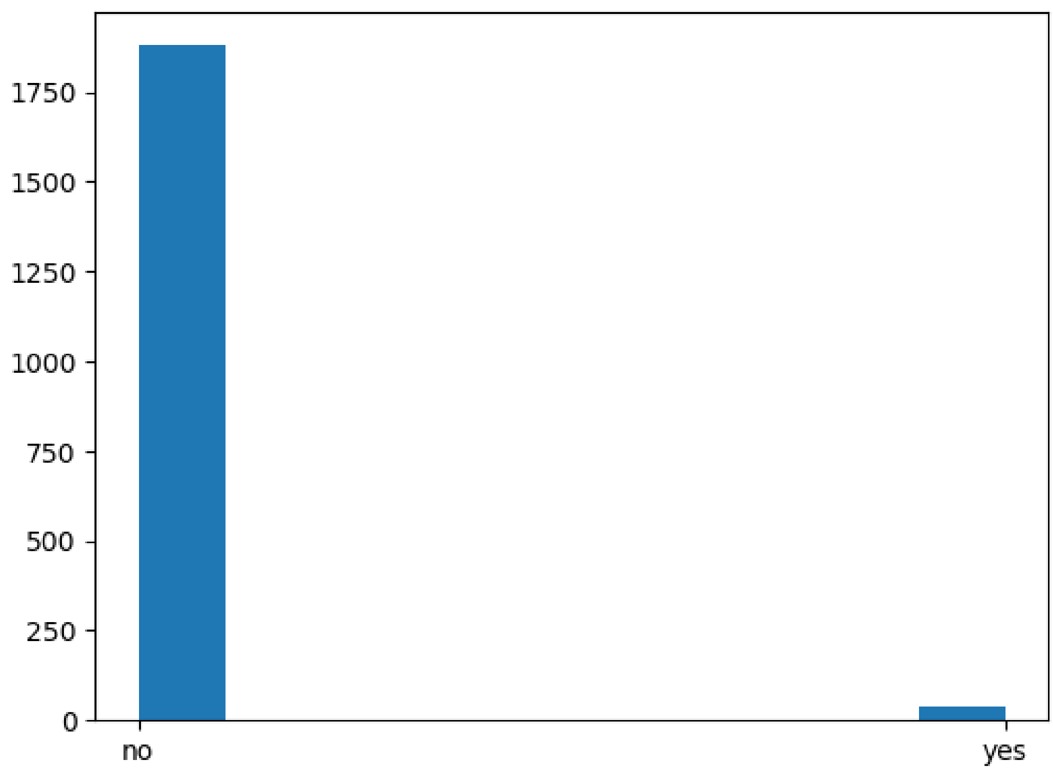
```
]:  dataset["Height"].astype(float).describe()
```

```
]: count    1920.000000
   mean        1.702297
   std         0.093224
   min         1.450000
   25%         1.630000
   50%         1.700000
   75%         1.770000
   max         1.980000
   Name: Height, dtype: float64
```

IQR este 0.14

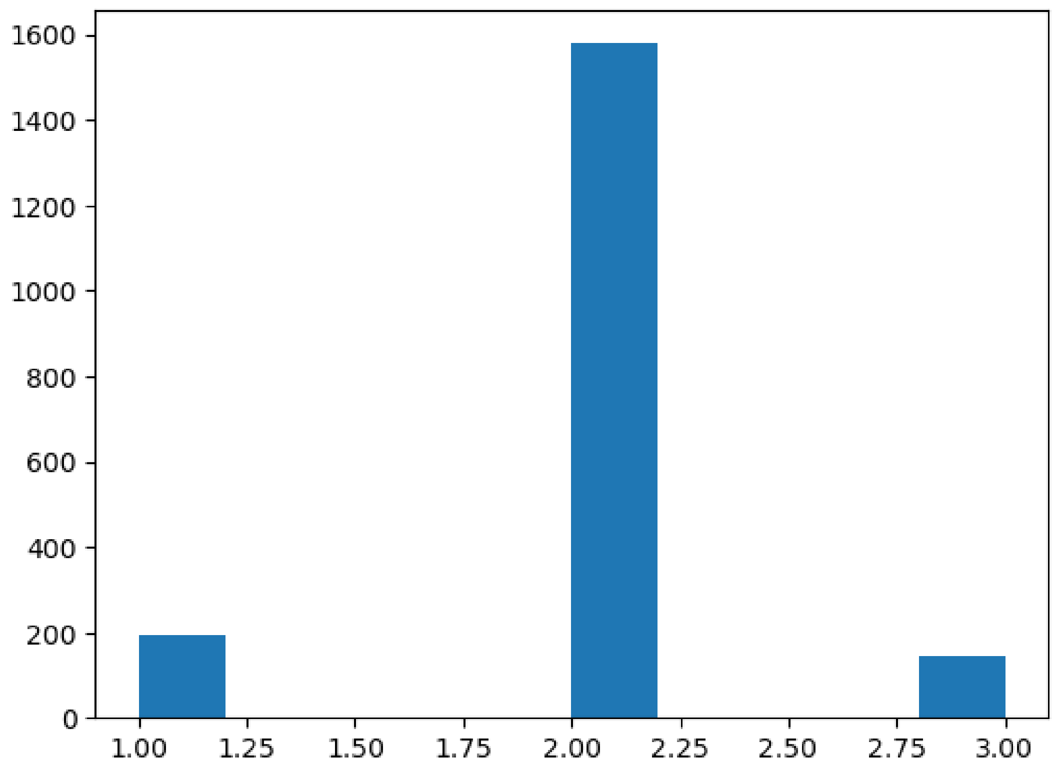
- Smoker

Variabilă categorică binară, deci putem plota histograma

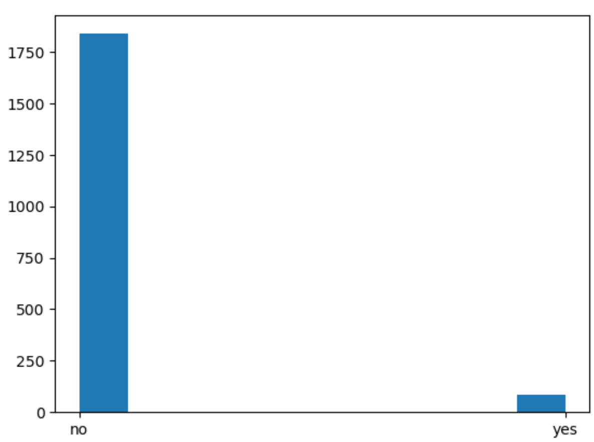


- Water_daily

Ar trebui să fie un atribut categoric, care arată frecvența consumului de lichide. Avem inclusiv valori eronate. După corecție



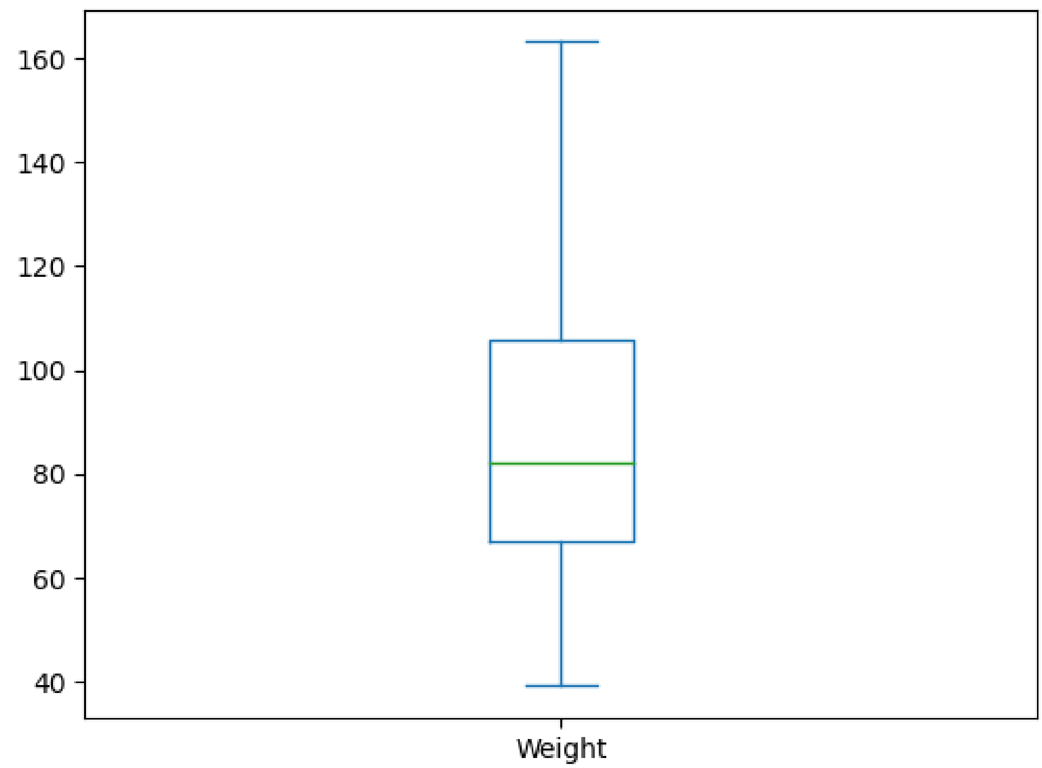
- Calorie_monitoring



Este o variabilă categorică binară, deci se poate plota o histogramă.

- Weight

Este un atribut numeric, care în setul de date conține și valori eronate care trebuie corectate. După un număr de iterații am ajuns la următorul boxplot

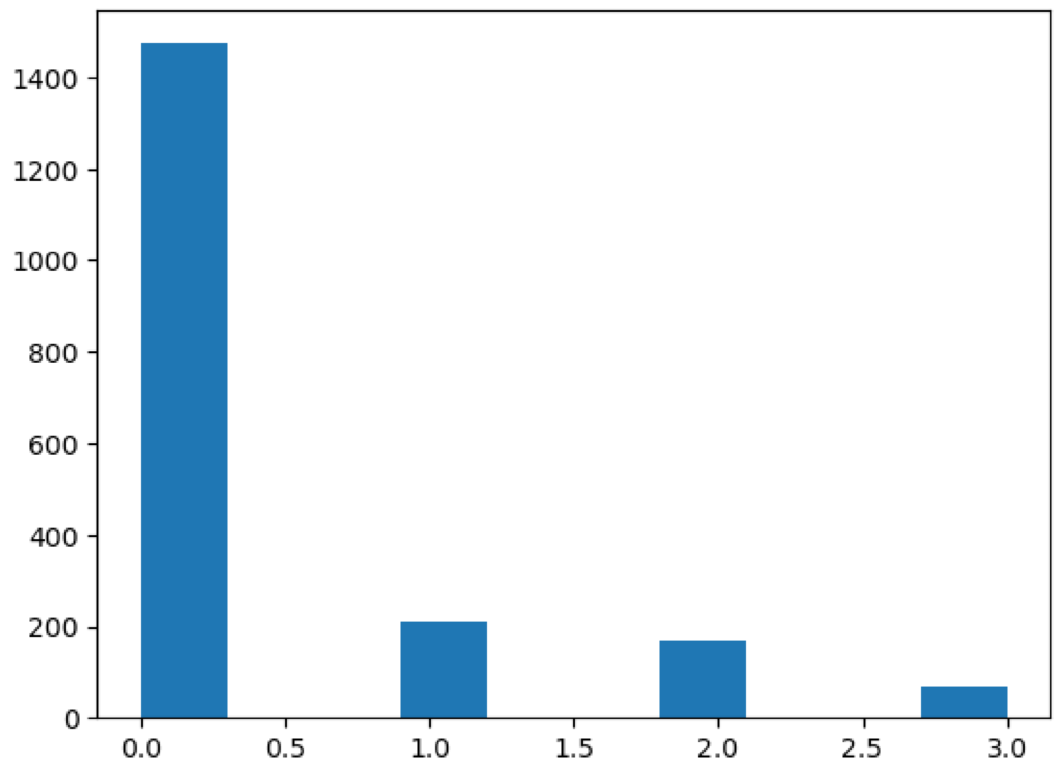



```
count    1920.000000
mean       86.582119
std        26.596913
min        39.000000
25%        66.739474
50%        82.015473
75%       105.984787
max       165.057269
Name: Weight, dtype: float64
```

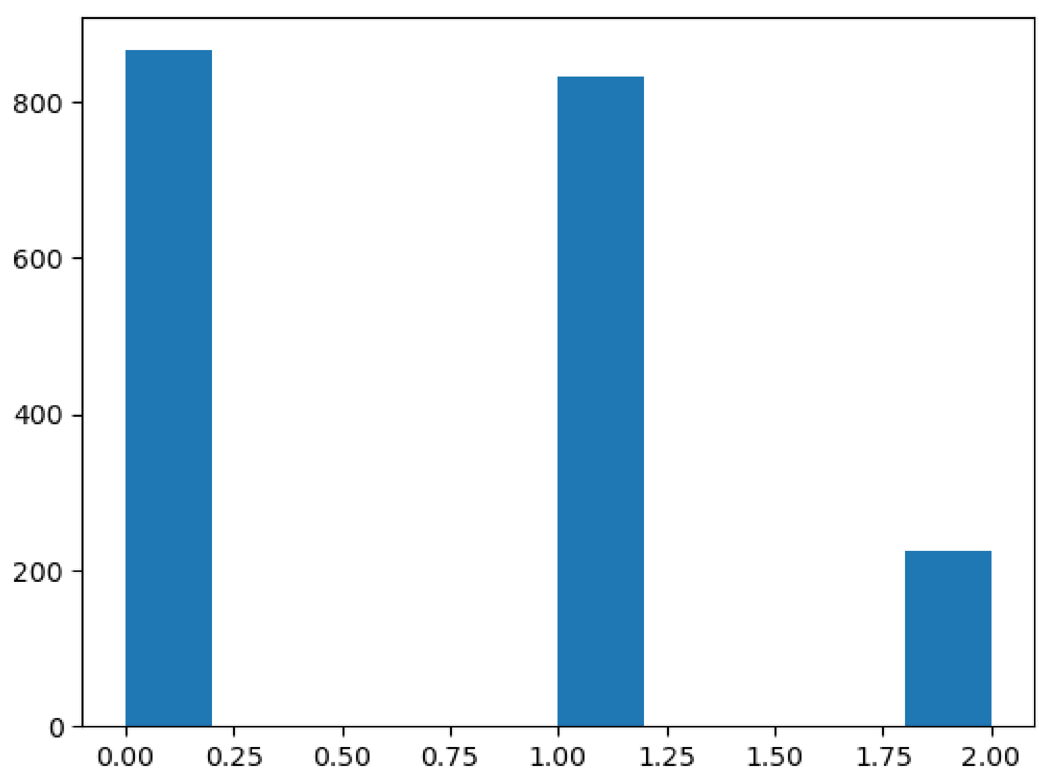
IQR este 39.1

- Physical_activity_level

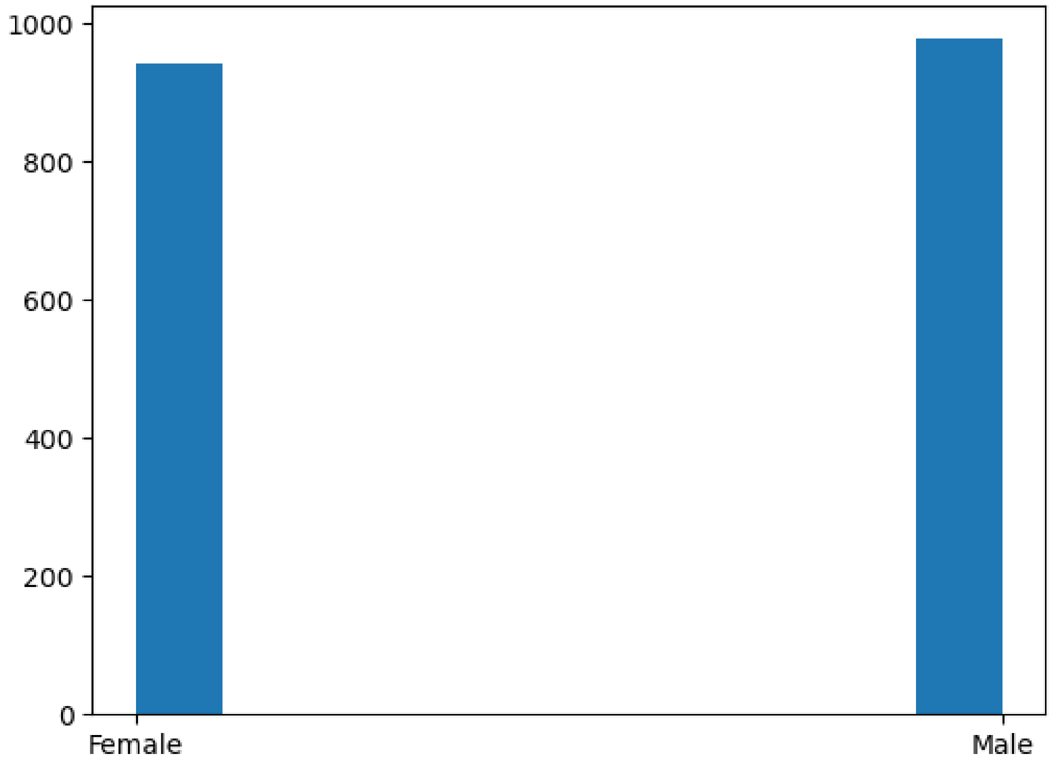
O variabilă categorică ce avusese valori eronate care au trebuit corectate, la final obținându-se histograma



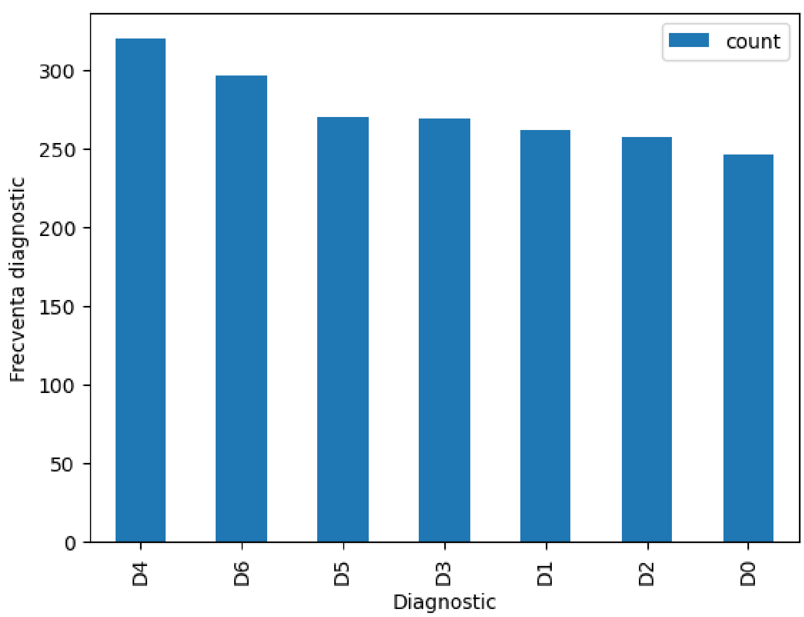
- Technology_time_use



- Gender

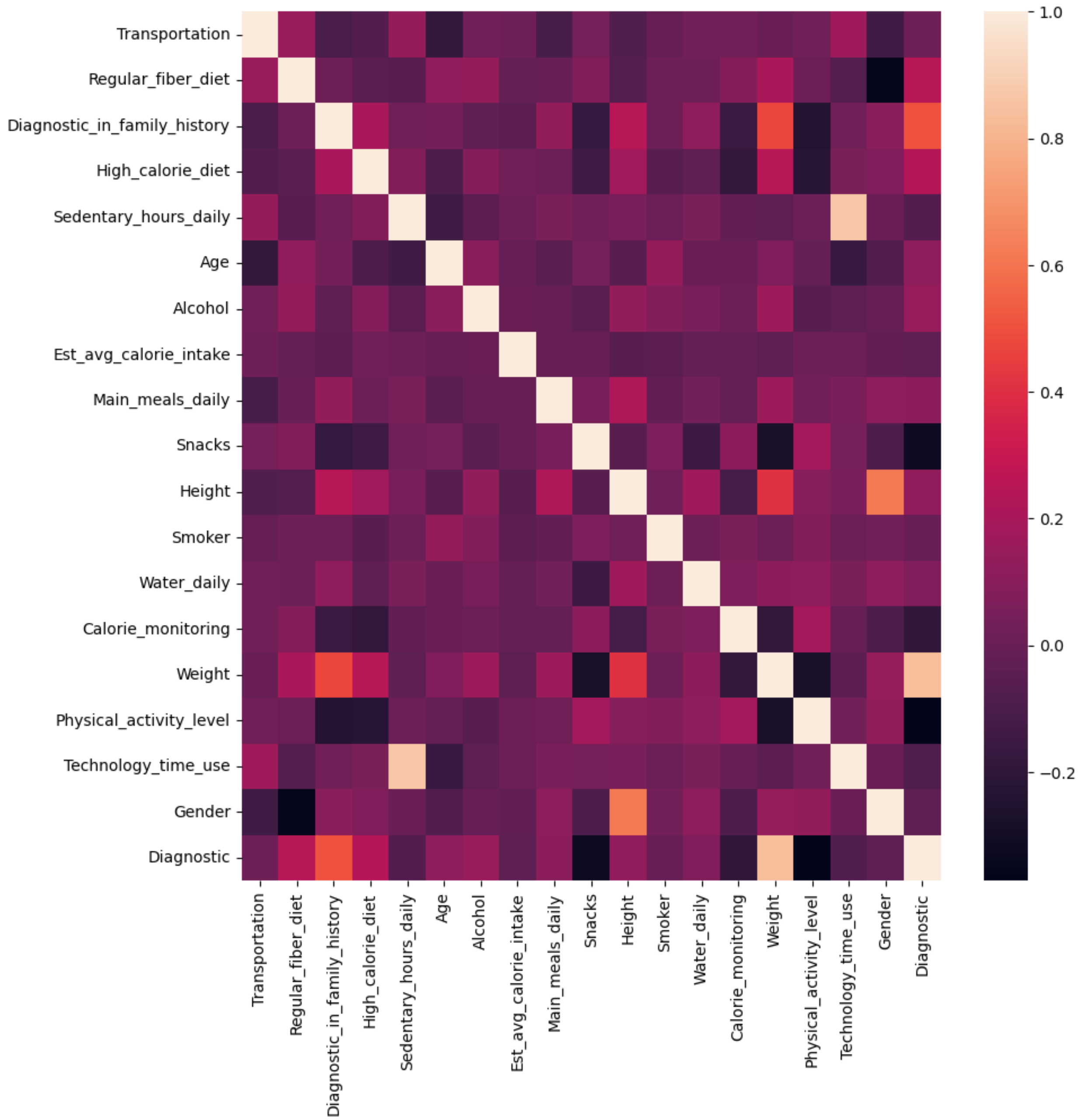


Distribuția clasei diagnostic



Se constată din acest bar-plot că distribuția claselor este aproape echilibrată, ceea ce înseamnă că algoritmii de învățare automată nu vor favoriza o clasă în detrimentul alteia.

Corelații între atribute și între atribute și clasă



Ca să ajung la diagrama corelației între atribute și între atribute și clasă, am transformat atributele care aveau valori sub formă de stringuri în numere folosind OrdinalEncoder din scikit-learn. Să analizăm diagrama.

- Vedem că diagnosticul este puternic afectat o serie de atribute, cele mai semnificative fiind Weight (o masă corporală mai mare crește posibilitatea unui anumit diagnostic) și prezența unui diagnostic în familie, din care deducem impactul genetic în stabilirea diagnosticului. Tot niște atribute semnificative în stabilirea diagnosticului sunt: prezența fibrelor în dietă și concentrația de calorii din dietă. Vedem însă și un rol al gustărilor și al nivelului de activitate fizică. Aparent un aport moderat de gustări ar reduce șansele unui anumit diagnostic. În mod firesc o activitate fizică redusă crește șansele unui diagnostic mai sever.
- Atributul Weight e în strânsă legătură cu istoricul medical de familie și cu înălțimea, ceea ce întărește ipoteza că masa corporală este factor cheie în stabilirea diagnosticului.
- Înălțimea e strâns legată de sexul pacientului, fiind foarte posibil ca pacienții de sex masculin să fie mai înalți decât pacienții de sex feminin.
- Atributele de folosirea tehnologiei și sedentarism sunt legate una de alta, dar niciuna dintre ele nu au un impact deosebit asupra diagnosticului

3.2 Feature Engineering și evaluarea de algoritmi

La etapa de feature engineering vom folosi Variance Threshold. Variance Threshold este o metodă de extragere de caracteristici în care sunt excluse atributele a căror varianță este sub un anumit prag. Acele variațe excluse nu au o contribuție semnificativă în predicția atributului.

Vom explora în continuare ce atribute rămân dacă variem pragul.

- Pentru prag 0

```
In [313]:
variance_threshold=VarianceThreshold(threshold=0.0)
variance_threshold.fit(features,labels)
new_features=variance_threshold.get_support(indices=True)
new_dataset=dataset.iloc[:,new_features]
dataset.columns[new_features]
#new_dataset.to_numpy()
#train_val_features,test_features,train_val_y,test_y=train_test_split(new_dataset.to_numpy(),labels,
                                                                      #random_state=42,test_size=0.2)

Out[313]: Index(['Transportation', 'Regular_fiber_diet', 'Diagnostic_in_family_history',
       'High_calorie_diet', 'Sedentary_hours_daily', 'Age', 'Alcohol',
       'Est_avg_calorie_intake', 'Main_meals_daily', 'Snacks', 'Height',
       'Smoker', 'Water_daily', 'Calorie_monitoring', 'Weight',
       'Physical_activity_level', 'Technology_time_use', 'Gender'],
      dtype='object')
```

Vedem că toate atributele au o varianță nenulă

- Pentru prag 0.1

```
In [338]:
variance_threshold=VarianceThreshold(threshold=0.1)
variance_threshold.fit(features)
new_features_indexes=variance_threshold.get_support(indices=True)
new_dataset=dataset.iloc[:,new_features_indexes]
#print(dataset.columns[new_features_indexes])
all_features_indexes=np.arange(0,features.shape[1])
excluded_feature_indexes=np.array(list(set(all_features_indexes)-set(new_features_indexes)))
print("Excluded features",dataset.columns[excluded_feature_indexes])
print("Correlation of excluded features with the target variable")
for col in dataset.columns[excluded_feature_indexes]:
    print("Corr between Diagnostic and {} is {}".format(col,dataset.corr()["Diagnostic"][col]))
#new_dataset.to_numpy()
#train_val_features,test_features,train_val_y,test_y=train_test_split(new_dataset.to_numpy(),labels,
                                                                      #random_state=42,test_size=0.2)

Excluded features Index(['Height', 'Smoker', 'Calorie_monitoring'], dtype='object')
Correlation of excluded features with the target variable
Corr between Diagnostic and Height is 0.13019007655111536
Corr between Diagnostic and Smoker is -0.005093486025003062
Corr between Diagnostic and Calorie_monitoring is -0.18964450668598895
```

Observăm că s-a scos atributul smoker care aparent nu are o corelație puternică cu ținta, dar sunt scoase Calorie_monitoring care pare să aibă o oarecare influență, dar și Height.

- Pragul de 0.2

```
labels=dataset.iloc[:,-1].to_numpy()
```

```
n [339]:
variance_threshold=VarianceThreshold(threshold=0.2)
variance_threshold.fit(features)
new_features_indexes=variance_threshold.get_support(indices=True)
new_dataset=dataset.iloc[:,new_features_indexes]
#print(dataset.columns[new_features_indexes])
all_features_indexes=np.arange(0,features.shape[1])
excluded_feature_indexes=np.array(list(set(all_features_indexes)-set(new_features_indexes)))
print("Excluded features",dataset.columns[excluded_feature_indexes])
print("Correlation of excluded features with the target variable")
for col in dataset.columns[excluded_feature_indexes]:
    print("Corr between Diagnostic and {} is {}".format(col,dataset.corr()["Diagnostic"][col]))
#new_dataset.to_numpy()
#train_val_features,test_features,train_val_y,test_y=train_test_split(new_dataset.to_numpy(),labels,
                                                                      #random_state=42,test_size=0.2)

Excluded features Index(['Diagnostic_in_family_history', 'High_calorie_diet', 'Height', 'Smoker',
       'Water_daily', 'Calorie_monitoring'],
      dtype='object')
Correlation of excluded features with the target variable
Corr between Diagnostic and Diagnostic_in_family_history is 0.5046920587060038
Corr between Diagnostic and High_calorie_diet is 0.24348928273131878
Corr between Diagnostic and Height is 0.13019007655111536
Corr between Diagnostic and Smoker is -0.005093486025003062
Corr between Diagnostic and Water_daily is 0.0704308834592036
Corr between Diagnostic and Calorie_monitoring is -0.18964450668598895
```

Vedem că la acest prag, atribute care influențează mai mult diagnosticul, precum Diagnostic_in_family_history și High_calorie_diet sunt excluse. E de menționat că atributul de diagnositic are o corelație de aproximativ 0.5 cu ținta, iar pentru High_calorie_diet avem corelație de 0.24. De aici vedem că un atribut important este exclus.

- Pragul de 0.3

```
In [337]:
variance_threshold=VarianceThreshold(threshold=0.3)
variance_threshold.fit(features)
new_features_indexes=variance_threshold.get_support(indices=True)
new_dataset=dataset.iloc[:,new_features_indexes]
#print(dataset.columns[new_features_indexes])
all_features_indexes=np.arange(0,features.shape[1])
excluded_feature_indexes=np.array(list(set(all_features_indexes)-set(new_features_indexes)))
print("Excluded features",dataset.columns[excluded_feature_indexes])
print("Correlation of excluded features with the target variable")
for col in dataset.columns[excluded_feature_indexes]:
    print("Corr between Diagnostic and {} is {}".format(col,dataset.corr()["Diagnostic"][col]))
#new_dataset.to_numpy()
#train_val_features,test_features,train_val_y,test_y=train_test_split(new_dataset.to_numpy(),labels,
                                                                     #random_state=42,test_size=0.2)
```

```
Excluded features Index(['Regular_fiber_diet', 'Diagnostic_in_family_history',
       'High_calorie_diet', 'Alcohol', 'Snacks', 'Height', 'Smoker',
       'Water_daily', 'Calorie_monitoring', 'Gender'],
      dtype='object')
Correlation of excluded features with the target variable
Corr between Diagnostic and Regular_fiber_diet is 0.2494076964096828
Corr between Diagnostic and Diagnostic_in_family_history is 0.5046920587060038
Corr between Diagnostic and High_calorie_diet is 0.24348928273131878
Corr between Diagnostic and Alcohol is 0.14977667835302858
Corr between Diagnostic and Snacks is -0.3260519395456708
Corr between Diagnostic and Height is 0.13019007655111536
Corr between Diagnostic and Smoker is -0.005093486025003062
Corr between Diagnostic and Water_daily is 0.0704308834592036
Corr between Diagnostic and Calorie_monitoring is -0.18964450668598895
Corr between Diagnostic and Gender is -0.03651470293582483
```

Observăm de aici că încă un atribut important este, din păcate scos, aume Regular_fiber_diet. Un lucru interesant este la Snacks, căci în ciuda corelării negative, valoarea acesteia în modul nu poate fi neglijată.

Ținând cont de constatările făcute am decis că nu mai este cazul să creștem pragul, întrucât riscăm să eliminăm și alte atribute care ar putea influența predicția. Vom alege un prag sub 0.2 pentru a păstra atributele Diagnostic_in_family și High_calorie_diet.

Aleg drept prag valoarea de 0.1025, fiindcă doresc ca diagnositcul să fie influențat inclusiv de atribute considerate esențiale, printre care Regular_fiber_diet, Diagnostic_in_family_history,High_calorie_diet, Snacks, Weight și Physical_activity_label.

```
variance_threshold=VarianceThreshold(threshold=0.1025)
variance_threshold.fit(features)
new_features_indexes=variance_threshold.get_support(indices=True)
new_dataset=dataset.iloc[:,new_features_indexes]
#print(dataset.columns[new_features_indexes])
all_features_indexes=np.arange(0,features.shape[1])
excluded_feature_indexes=np.array(list(set(all_features_indexes)-set(new_features_indexes)))
print("Excluded features",dataset.columns[excluded_feature_indexes])
print("Correlation of excluded features with the target variable")
for col in dataset.columns[excluded_feature_indexes]:
    print("Corr between Diagnostic and {} is {}".format(col,dataset.corr()["Diagnostic"][col]))
print("Considered columns for prediction: ",dataset.columns[new_features_indexes])
#new_dataset.to_numpy()
#train_val_features,test_features,train_val_y,test_y=train_test_split(new_dataset.to_numpy(),labels,
                                                                      #random_state=42,test_size=0.2)
```

```
Excluded features Index(['Height', 'Smoker', 'Calorie_monitoring'], dtype='object')
Correlation of excluded features with the target variable
Corr between Diagnostic and Height is 0.13019007655111536
Corr between Diagnostic and Smoker is -0.005093486025003062
Corr between Diagnostic and Calorie_monitoring is -0.18964450668598895
Considered columns for prediction:  Index(['Transportation', 'Regular_fiber_diet', 'Diagnostic_in_family_history',
       'High_calorie_diet', 'Sedentary_hours_daily', 'Age', 'Alcohol',
       'Est_avg_calorie_intake', 'Main_meals_daily', 'Snacks', 'Water_daily',
       'Weight', 'Physical_activity_level', 'Technology_time_use', 'Gender'],
      dtype='object')
```

În continuare, împărțim setul de date în setul de antrenare și validare, respectiv testare. 80% din date vor fi folosite pentru antrenare și validare, iar restul de 20% pentru testare.

Pentru că în noul set de date, după transformarea atributelor categorice sub formă de stringuri în numere, avem atribute categorice de o cifră, numere raționale la Sedentary_hours_daily și Weight, respectiv valori de ordinul miilor la Est_avg_calorie_intake, considerăm că e necesară standardizarea datelor.

Testarea algoritmilor

Conform documentației din scikit-learn, pentru o clasificare multi-clasă, pentru metricile f1,precision și recall voi folosi varianta macro, pentru că la acea variantă pentru fiecare clasă se iau metricile corespunzătoare și se împarte la numărul claselor.

- SVM

  Variem parametrul C de regularizare între 0.1 și 1.4 inclusiv cu pas 0.1

  Variem tipul kernelului între linear,rbf,sigmoid și poly

index	params	mean_test_accuracy	std_test_accuracy	mean_test_f1_macro	std_test_f1_macro	mean_test_recall_macro	std_test_recall_macro	mean_test_precision_macro	std_test_precision_macro
0	"{'C': 0.1, 'kernel': 'linear'}"	0.695327637	0.029655392	0.68390591	0.032501616	0.693294429	0.030357666	0.695191587	0.03362185
1	"{'C': 0.1, 'kernel': 'rbf'}"	0.620464064	0.021148039	0.596664381	0.022105908	0.61579248	0.021063566	0.65773513	0.030954597
2	"{'C': 0.1, 'kernel': 'poly'}"	0.476564152	0.020485413	0.442577223	0.020589312	0.456377378	0.020122839	0.622948995	0.067397035
3	"{'C': 0.1, 'kernel': 'sigmoid'}"	0.552749693	0.018781063	0.523108166	0.022351037	0.545805102	0.018722494	0.553722271	0.041579224
4	"{'C': 0.2, 'kernel': 'linear'}"	0.708993612	0.022869453	0.70097879	0.025177132	0.707543091	0.024023788	0.710018947	0.026382826
5	"{'C': 0.2, 'kernel': 'rbf'}"	0.648447481	0.021823562	0.635099125	0.025377407	0.645044197	0.02229566	0.656938151	0.025112856
6	"{'C': 0.2, 'kernel': 'poly'}"	0.576837007	0.022937287	0.572216252	0.020981323	0.566025331	0.023617533	0.681740881	0.024214176
7	"{'C': 0.2, 'kernel': 'sigmoid'}"	0.574882609	0.024534619	0.552298115	0.023396987	0.567604599	0.023525537	0.567037507	0.028775927
8	"{'C': 0.30000000000000004, 'kernel': 'linear'}"	0.718103558	0.022561785	0.709323485	0.025615185	0.717038148	0.023477262	0.717466434	0.026741675
9	"{'C': 0.30000000000000004, 'kernel': 'rbf'}"	0.68164474	0.022352767	0.675877323	0.023309066	0.680415392	0.022418472	0.690450631	0.024760958
10	"{'C': 0.30000000000000004, 'kernel': 'poly'}"	0.595067473	0.015879781	0.590843386	0.014578385	0.58763157	0.015297631	0.66535575	0.013580193
11	"{'C': 0.30000000000000004, 'kernel': 'sigmoid'}"	0.597662761	0.025835039	0.58072632	0.026792934	0.590841649	0.025675149	0.595269012	0.033528481
12	"{'C': 0.4, 'kernel': 'linear'}"	0.717454207	0.017781447	0.71095814	0.017752842	0.716407668	0.018868444	0.716710057	0.021207811

13	"{'C': 0.4, 'kernel': 'rbf'}"	0.700522442	0.021083201	0.696642298	0.020534023	0.700927681	0.020804938	0.708057989	0.023847362
14	"{'C': 0.4, 'kernel': 'poly'}"	0.616548923	0.023755621	0.613002011	0.0218743 3	0.609843823	0.022881772	0.666555301	0.019582351
15	"{'C': 0.4, 'kernel': 'sigmoid'}"	0.600905284	0.023907437	0.584872206	0.024446667	0.594252866	0.023200222	0.597233587	0.032463917
16	"{'C': 0.5, 'kernel': 'linear'}"	0.719404374	0.0224655	0.714871639	0.024247648	0.718353896	0.023518027	0.7198093	0.026400232
17	"{'C': 0.5, 'kernel': 'rbf'}"	0.707030754	0.019870684	0.703100101	0.019671028	0.708020985	0.019633938	0.710841037	0.02081999
18	"{'C': 0.5, 'kernel': 'poly'}"	0.626306104	0.017914955	0.623697706	0.016256409	0.619800309	0.017583952	0.669450793	0.015973591
19	"{'C': 0.5, 'kernel': 'sigmoid'}"	0.599610813	0.031084016	0.584793457	0.034037801	0.59197807	0.031037888	0.595488467	0.043835625
20	"{'C': 0.6, 'kernel': 'linear'}"	0.720713651	0.021986228	0.716771623	0.023725611	0.720085782	0.022651843	0.721017156	0.02621689
21	"{'C': 0.6, 'kernel': 'rbf'}"	0.71223402	0.020734767	0.708692689	0.020392735	0.713830085	0.020367478	0.715254805	0.020557712
22	"{'C': 0.6, 'kernel': 'poly'}"	0.644536571	0.015476216	0.643596301	0.0152473 6	0.639155228	0.015482695	0.68028847	0.019210349
23	"{'C': 0.6, 'kernel': 'sigmoid'}"	0.602214561	0.025074606	0.589453297	0.029477559	0.595068819	0.025696608	0.59750706	0.034316383
24	"{'C': 0.7000000000000001, 'kernel': 'linear'}"	0.72332163	0.021304503	0.719303345	0.022464968	0.722636184	0.021697373	0.724079084	0.025280303
25	"{'C': 0.7000000000000001, 'kernel': 'rbf'}"	0.719391683	0.024073097	0.716220666	0.023607761	0.720900626	0.024411803	0.722776188	0.022466584
26	"{'C': 0.7000000000000001, 'kernel': 'poly'}"	0.653648631	0.01111091	0.653446731	0.012057037	0.648840819	0.011700993	0.683979588	0.015237529
27	"{'C': 0.7000000000000001,	0.598309996	0.022849324	0.584738527	0.026343402	0.590845724	0.023652361	0.592337885	0.031127872

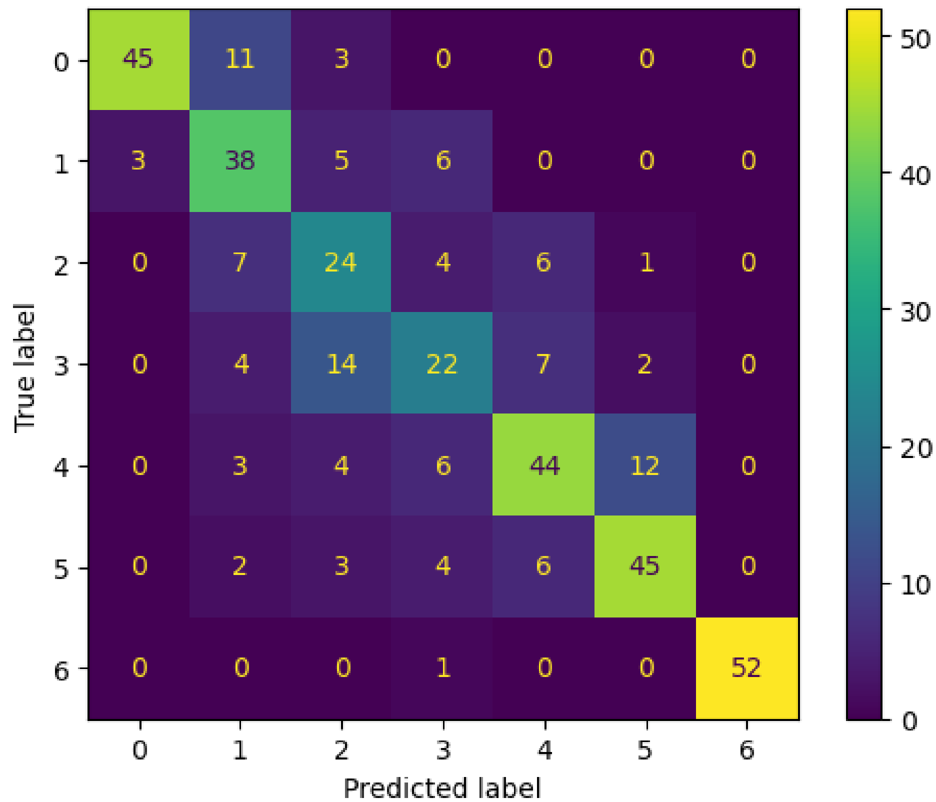
	'kernel': 'sigmoid'}"								
28	"{'C': 0.8, 'kernel': 'linear'}"	0.721367232	0.020155893	0.717353207	0.021406624	0.720735133	0.02074073	0.722581319	0.024237144
29	"{'C': 0.8, 'kernel': 'rbf'}"	0.723292017	0.024531678	0.720385485	0.024075854	0.724751812	0.025236235	0.725672659	0.02207199
30	"{'C': 0.8, 'kernel': 'poly'}"	0.662117687	0.014119904	0.660835194	0.014541332	0.657557043	0.015322847	0.688292139	0.015084445
31	"{'C': 0.8, 'kernel': 'sigmoid'}"	0.595714709	0.027870496	0.581384579	0.028910648	0.588354733	0.028337245	0.586486631	0.030006067
32	"{'C': 0.9, 'kernel': 'linear'}"	0.721363002	0.020221528	0.717719591	0.021366031	0.720906539	0.020590184	0.722625471	0.023746727
33	"{'C': 0.9, 'kernel': 'rbf'}"	0.726545116	0.022754571	0.72422487	0.022150687	0.727911902	0.022868223	0.729372108	0.020860105
34	"{'C': 0.9, 'kernel': 'poly'}"	0.666028597	0.01546942	0.663944833	0.016359739	0.662013728	0.01675772	0.686422997	0.015983203
35	"{'C': 0.9, 'kernel': 'sigmoid'}"	0.583334743	0.012472012	0.569533234	0.01183885	0.576615298	0.011703259	0.572800987	0.012514164
36	"{'C': 1.0, 'kernel': 'linear'}"	0.722014468	0.020215394	0.718191078	0.021273474	0.72158753	0.020511761	0.722540035	0.023762413
37	"{'C': 1.0, 'kernel': 'rbf'}"	0.727848048	0.023980639	0.726110089	0.023677154	0.728999002	0.024250834	0.73138451	0.02287064
38	"{'C': 1.0, 'kernel': 'poly'}"	0.668628115	0.014421093	0.666019872	0.015958883	0.664818876	0.015778879	0.688176308	0.015069346
39	"{'C': 1.0, 'kernel': 'sigmoid'}"	0.584633445	0.018531358	0.571263223	0.018182872	0.577718587	0.018081527	0.574668072	0.020030853
40	"{'C': 1.1, 'kernel': 'linear'}"	0.724618216	0.023431222	0.721349567	0.025201466	0.724309596	0.023774241	0.725382482	0.02732452
41	"{'C': 1.1, 'kernel': 'rbf'}"	0.732408308	0.027404513	0.730691466	0.026983979	0.733304377	0.027183572	0.735188341	0.02513228
42	"{'C': 1.1, 'kernel': 'poly'}"	0.680995389	0.015006249	0.678230876	0.015961328	0.67731363	0.015668545	0.700813931	0.0150832
43	"{'C': 1.1, 'kernel': 'sigmoid'}"	0.57226617	0.019804595	0.561145068	0.018950777	0.566561509	0.019890614	0.563078072	0.020557792
44	"{'C': 1.2000000000000	0.723968865	0.021987784	0.721118982	0.023378276	0.723542615	0.022419822	0.725606568	0.025200039

	002, 'kernel': 'linear'}"								
45	"{'C': 1.2000000000000002, 'kernel': 'rbf'}"	0.73826727	0.02541224	0.736528614	0.025044786	0.739077427	0.025649979	0.741423168	0.023440513
46	"{'C': 1.2000000000000002, 'kernel': 'poly'}"	0.686856466	0.015985277	0.684266205	0.016515649	0.68349201	0.015984557	0.704823524	0.017737613
47	"{'C': 1.2000000000000002, 'kernel': 'sigmoid'}"	0.574876264	0.015025645	0.562703632	0.014295602	0.568529321	0.015611186	0.565227779	0.014845864
48	"{'C': 1.3000000000000003, 'kernel': 'linear'}"	0.724620331	0.021612198	0.721550995	0.023407733	0.724236894	0.022196859	0.725782763	0.025340569
49	"{'C': 1.3000000000000003, 'kernel': 'rbf'}"	0.739572317	0.025915016	0.737833638	0.025428125	0.740317808	0.025943882	0.743117325	0.02427947
50	"{'C': 1.3000000000000003, 'kernel': 'poly'}"	0.690105334	0.016886605	0.687317023	0.017657082	0.686786256	0.016798396	0.704509537	0.017474425
51	"{'C': 1.3000000000000003, 'kernel': 'sigmoid'}"	0.568359491	0.011189645	0.556286489	0.009856841	0.562374755	0.011220528	0.557325913	0.009201618
52	"{'C': 1.4000000000000001, 'kernel': 'linear'}"	0.723315284	0.018641208	0.719775028	0.02053 8	0.722814248	0.019489501	0.72357188	0.021833886
53	"{'C': 1.4000000000000001,	**0.744126232**	0.027265592	**0.742185964**	0.026731841	**0.744969314**	0.027473048	**0.747999166**	0.025905638

	'kernel': 'rbf'}"								
54	"{'C': 1.4000000000000001, 'kernel': 'poly'}"	0.692059732	0.014596686	0.689154054	0.015041967	0.688811252	0.014296219	0.705571089	0.016290551
55	"{'C': 1.4000000000000001, 'kernel': 'sigmoid'}"	0.555983756	0.015920292	0.54476358	0.012504851	0.549989249	0.014802125	0.546693595	0.011198056

Observăm că un kernel care a jucat un rol central în obținerea celui mai bun rezultat este rbf. Faptul că a fost folosit RBF arată complexitatea atributelor, precum și neliniaritatea acestora, iar valoarea C de 1.4 arată că a fost necesară o valoare semnificativă pentru a reduce overfitting-ul rezultând o margine de separare mai mică și care să ducă la o clasificare mai bună. Parametrii au fost influențați de asemenea și de atribute care aveau o corelație mică cu clasa. Se poate observa din tabel că folosirea kernelului RBF duce la rezultate mai bune în comparație cu alte funcții kernel.

Pentru configurația de la linia 53 matricea de confuzie e următoarea



Din matricea de confuzie vedem că diagnosticul de indice 6 a obținut rezultate mai bune.

Random Forest

index	params	mean_test_accuracy	std_test_accuracy	mean_test_f1_macro	std_test_f1_macro	mean_test_recall_macro	std_test_recall_macro	mean_test_precision_macro	std_test_precision_macro
0	"{'max_depth': 2, 'max_samples': 0.1, 'n_estimators': 10}"	0.503301747	0.045509531	0.447177023	0.058632422	0.493780861	0.048721151	0.510074137	0.097249271
1	"{'max_depth': 2, 'max_samples': 0.1, 'n_estimators': 60}"	0.618492745	0.0241802	0.586783629	0.026219811	0.614056451	0.022632181	0.630668726	0.034479705
2	"{'max_depth': 2, 'max_sam	0.617832819	0.016423124	0.581799672	0.018963043	0.61511521	0.016055	0.646967993	0.038702384

	ples': 0.1, 'n_estimators': 110}"								
3	"{'max_depth': 2, 'max_samples': 0.1, 'n_estimators': 160}"	0.618484284	0.016644561	0.582694168	0.016956525	0.615056377	0.016150073	0.651183095	0.018225572
4	"{'max_depth': 2, 'max_samples': 0.1, 'n_estimators': 210}"	0.619778755	0.016678315	0.580920227	0.017231491	0.617472209	0.015497856	0.641068694	0.030176533
5	"{'max_depth': 2, 'max_samples': 0.1, 'n_estimators': 260}"	0.622386734	0.014785179	0.581310925	0.020207935	0.619731389	0.013844649	0.646562135	0.028097594
6	"{'max_depth': 2, 'max_samples': 0.30000000000000004, 'n_estimators': 10}"	0.560560937	0.019199765	0.520747085	0.024039597	0.550576128	0.019552808	0.577037396	0.026612063
7	"{'max_depth': 2, 'max_samples': 0.30000000000000004, 'n_estimators': 60}"	0.610670925	0.018953667	0.57864866	0.020737157	0.607268271	0.019222172	0.651647244	0.039953518
8	"{'max_depth': 2, 'max_samples': 0.30000000000000004, 'n_estimators': 110}"	0.619787216	0.023680387	0.5849343	0.026726288	0.616919943	0.022809259	0.681097086	0.048638453
9	"{'max_depth': 2, 'max_samples': 0.300000	0.622384619	0.021885169	0.586788222	0.024809233	0.618617449	0.022228026	0.684628647	0.048408772

	00000000004, 'n_estimators': 160}"								
10	"{'max_depth': 2, 'max_samples': 0.30000000000000004, 'n_estimators': 210}"	0.626287068	0.019849255	0.588671151	0.020652665	0.623737208	0.019648869	0.682914057	0.044778338
11	"{'max_depth': 2, 'max_samples': 0.30000000000000004, 'n_estimators': 260}"	0.628903507	0.008841887	0.590946204	0.009957911	0.626397398	0.007698587	0.682932749	0.034675777
12	"{'max_depth': 2, 'max_samples': 0.5000000000000001, 'n_estimators': 10}"	0.581393037	0.020587933	0.550360035	0.021816675	0.573243146	0.021363691	0.613662416	0.052184502
13	"{'max_depth': 2, 'max_samples': 0.5000000000000001, 'n_estimators': 60}"	0.615231186	0.018184366	0.585345014	0.016360341	0.611690503	0.016985929	0.662016932	0.047202726
14	"{'max_depth': 2, 'max_samples': 0.5000000000000001, 'n_estimators': 110}"	0.612627438	0.017412771	0.576007027	0.017357961	0.609117972	0.015504934	0.657938857	0.041905033
15	"{'max_depth': 2, 'max_samples': 0.500000	0.617179238	0.015955894	0.579156769	0.016006868	0.614235338	0.014494852	0.664031062	0.03599293

	0000000001, 'n_estimators': 160}"								
16	"{'max_depth': 2, 'max_samples': 0.50000000000000001, 'n_estimators': 210}"	0.619129405	0.021983576	0.579453477	0.023306061	0.616364914	0.020971428	0.666479362	0.046488082
17	"{'max_depth': 2, 'max_samples': 0.50000000000000001, 'n_estimators': 260}"	0.624994712	0.009935865	0.584065718	0.012334042	0.622778621	0.009438307	0.668532289	0.043678799
18	"{'max_depth': 2, 'max_samples': 0.70000000000000001, 'n_estimators': 10}"	0.574867803	0.025889066	0.53435695	0.03009013	0.566621529	0.028332754	0.612773219	0.053228142
19	"{'max_depth': 2, 'max_samples': 0.70000000000000001, 'n_estimators': 60}"	0.614581835	0.018633739	0.584822366	0.018189713	0.609738477	0.017893976	0.651491961	0.037130088
20	"{'max_depth': 2, 'max_samples': 0.70000000000000001, 'n_estimators': 110}"	0.621094378	0.017892173	0.588798836	0.020076308	0.617621146	0.016873994	0.679631071	0.043548359
21	"{'max_depth': 2, 'max_samples': 0.700000	0.615886882	0.017682628	0.580457524	0.01889518	0.612034852	0.017195453	0.667113618	0.040497847

	0000000001, 'n_estimators': 160}"								
22	"{'max_depth': 2, 'max_samples': 0.70000000000000001, 'n_estimators': 210}"	0.620436567	0.014215864	0.582869387	0.013047037	0.617354967	0.012987829	0.672555964	0.043277485
23	"{'max_depth': 2, 'max_samples': 0.70000000000000001, 'n_estimators': 260}"	0.626297644	0.007460416	0.587986836	0.009342663	0.623981757	0.006595923	0.677109621	0.038643136
24	"{'max_depth': 2, 'max_samples': 0.90000000000000001, 'n_estimators': 10}"	0.583347434	0.021803059	0.547572589	0.02481296	0.575149902	0.023029396	0.617580089	0.04581182
25	"{'max_depth': 2, 'max_samples': 0.90000000000000001, 'n_estimators': 60}"	0.608069292	0.018333944	0.574121422	0.01879333	0.603383263	0.017599197	0.650833897	0.033346042
26	"{'max_depth': 2, 'max_samples': 0.90000000000000001, 'n_estimators': 110}"	0.617834934	0.023595657	0.58249386	0.028531046	0.613828537	0.023027619	0.663064293	0.042751495
27	"{'max_depth': 2, 'max_samples': 0.900000	0.620436567	0.016431555	0.586365357	0.01714582	0.615987982	0.017104856	0.675862215	0.021952034

	0000000001, 'n_estimators': 160}"								
28	"{'max_depth': 2, 'max_samples': 0.90000000000000001, 'n_estimators': 210}"	0.624996827	0.012343335	0.588055162	0.013056528	0.622192031	0.011315746	0.681365091	0.032405888
29	"{'max_depth': 2, 'max_samples': 0.90000000000000001, 'n_estimators': 260}"	0.628252041	0.007950579	0.590407395	0.014495837	0.626130353	0.008226997	0.678196957	0.037988123
30	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 10}"	0.567077711	0.028280172	0.540279636	0.026237386	0.563160467	0.029262938	0.575353478	0.047326357
31	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 60}"	0.664080545	0.019132797	0.644896298	0.0189283	0.662954285	0.017561014	0.666061134	0.013977577
32	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 110}"	0.671240323	0.017263097	0.652269496	0.017612747	0.670874673	0.016785955	0.677723534	0.020152284
33	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 160}"	0.676449934	0.0211735	0.658494717	0.022546208	0.676190695	0.020049593	0.681364029	0.02530911
34	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 210}"	0.673835611	0.012041106	0.654064745	0.012362769	0.673473313	0.010692838	0.681331154	0.010714732

35	"{'max_depth': 3, 'max_samples': 0.1, 'n_estimators': 260}"	0.679694573	0.015615033	0.659148653	0.019868372	0.680241612	0.015114668	0.687275771	0.019089303
36	"{'max_depth': 3, 'max_samples': 0.30000000000000004, 'n_estimators': 10}"	0.629561318	0.015502719	0.60633802	0.015837662	0.626083166	0.01143688	0.623303135	0.016609325
37	"{'max_depth': 3, 'max_samples': 0.30000000000000004, 'n_estimators': 60}"	0.689491941	0.031159575	0.671521701	0.037164991	0.689739643	0.031435327	0.693504792	0.034788496
38	"{'max_depth': 3, 'max_samples': 0.30000000000000004, 'n_estimators': 110}"	0.672539024	0.016013991	0.651612154	0.018565672	0.672750468	0.015153185	0.684151821	0.023348684
39	"{'max_depth': 3, 'max_samples': 0.30000000000000004, 'n_estimators': 160}"	0.679694573	0.0137 3522	0.660073091	0.014430787	0.67986105	0.013191727	0.694035312	0.018759963
40	"{'max_depth': 3, 'max_samples': 0.30000000000000004, 'n_estimators': 210}"	0.680995389	0.015561614	0.659965674	0.018175656	0.681374317	0.015373479	0.694066291	0.017310644
41	"{'max_depth': 3, 'max_sam	0.684902069	0.014625836	0.665717641	0.017598124	0.685238178	0.014630949	0.694417364	0.018625035

	ples': 0.30000000000000004, 'n_estimators': 260}"								
42	"{'max_depth': 3, 'max_samples': 0.5000000000000001, 'n_estimators': 10}"	0.634111003	0.024214599	0.612625063	0.023469618	0.630770655	0.021480896	0.626492144	0.024881651
43	"{'max_depth': 3, 'max_samples': 0.5000000000000001, 'n_estimators': 60}"	0.688167858	0.025950654	0.672686122	0.024776925	0.687790382	0.024224421	0.689015978	0.027454691
44	"{'max_depth': 3, 'max_samples': 0.5000000000000001, 'n_estimators': 110}"	0.67773806	0.022976264	0.655364934	0.025301439	0.678373441	0.021519756	0.687798907	0.025505835
45	"{'max_depth': 3, 'max_samples': 0.5000000000000001, 'n_estimators': 160}"	0.6907568	0.01452742	0.672778398	0.013403788	0.691218978	0.013279558	0.698512482	0.015315155
46	"{'max_depth': 3, 'max_samples': 0.5000000000000001, 'n_estimators': 210}"	0.68229409	0.021129129	0.661704789	0.022234292	0.682824704	0.020476346	0.691019652	0.023119109
47	"{'max_depth': 3, 'max_sam	0.690763146	0.016957595	0.672553332	0.01847325	0.691869835	0.016370921	0.700182194	0.019772997

	ples': 0.50000000000001, 'n_estimators': 260}"								
48	"{'max_depth': 3, 'max_samples': 0.70000000000001, 'n_estimators': 10}"	0.630204323	0.015171562	0.608133743	0.012532568	0.627251107	0.012129224	0.616491685	0.014949116
49	"{'max_depth': 3, 'max_samples': 0.70000000000001, 'n_estimators': 60}"	0.689460214	0.015619452	0.674439062	0.014603107	0.688786094	0.014031561	0.687933039	0.018019978
50	"{'max_depth': 3, 'max_samples': 0.70000000000001, 'n_estimators': 110}"	0.686854351	0.016161254	0.668175711	0.017572697	0.686897326	0.014806503	0.686286741	0.020854965
51	"{'max_depth': 3, 'max_samples': 0.70000000000001, 'n_estimators': 160}"	0.691410381	0.015243021	0.673570226	0.013634979	0.69136958	0.014107838	0.693690941	0.014910799
52	"{'max_depth': 3, 'max_samples': 0.70000000000001, 'n_estimators': 210}"	0.680343923	0.016159088	0.659495345	0.018709688	0.680779447	0.015662412	0.685898209	0.022346876
53	"{'max_depth': 3, 'max_sam	0.688812979	0.019853385	0.670594034	0.023265675	0.689909819	0.019486051	0.697715133	0.025478395

	ples': 0.70000000000000001, 'n_estimators': 260}"								
54	"{'max_depth': 3, 'max_samples': 0.90000000000000001, 'n_estimators': 10}"	0.624366513	0.018678161	0.597874173	0.025620481	0.619778565	0.016780485	0.609200686	0.028267031
55	"{'max_depth': 3, 'max_samples': 0.90000000000000001, 'n_estimators': 60}"	0.682308896	0.020472666	0.665868793	0.021630823	0.681564309	0.019770814	0.678704526	0.026681249
56	"{'max_depth': 3, 'max_samples': 0.90000000000000001, 'n_estimators': 110}"	0.684256948	0.017240872	0.667402474	0.018205515	0.684309441	0.015787561	0.685903494	0.021934093
57	"{'max_depth': 3, 'max_samples': 0.90000000000000001, 'n_estimators': 160}"	0.694009899	0.019219594	0.676141648	0.020536279	0.694398997	0.018320768	0.69533096	0.019566625
58	"{'max_depth': 3, 'max_samples': 0.90000000000000001, 'n_estimators': 210}"	0.69011168	0.021331889	0.670612666	0.024311132	0.690711863	0.020235358	0.699282675	0.026098112
59	"{'max_depth': 3, 'max_sam	0.697284149	0.020191588	0.680859068	0.022980479	0.698274215	0.018936352	0.701104648	0.022499085

	ples': 0.90000000000001, 'n_estimators': 260}"								
60	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 10}"	0.611352003	0.036373652	0.590033626	0.037589394	0.609799585	0.039223387	0.621857553	0.041119795
61	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 60}"	0.69921105	0.012801902	0.688792249	0.012808356	0.700372349	0.011427475	0.702381071	0.013902889
62	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 110}"	0.706379288	0.019159364	0.694578176	0.020197563	0.708368833	0.017807313	0.707194632	0.023402197
63	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 160}"	0.722008122	0.01934535	0.712597846	0.020533114	0.724486787	0.018665196	0.722231184	0.022478417
64	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 210}"	0.719404374	0.021101697	0.709515156	0.021677501	0.721328429	0.020175725	0.719263868	0.025152404
65	"{'max_depth': 4, 'max_samples': 0.1, 'n_estimators': 260}"	0.714194763	0.017851428	0.703081459	0.019092092	0.716091498	0.017292276	0.715209754	0.021262946
66	"{'max_depth': 4, 'max_samples': 0.30000000000000004, 'n_estimators': 10}"	0.636752824	0.032038178	0.613932645	0.036545938	0.635204434	0.034352765	0.642813903	0.042659425
67	"{'max_depth': 4,	0.718120479	0.029233641	0.707155968	0.033711149	0.719310394	0.029355338	0.720192688	0.033341356

	'max_samples': 0.30000000000000004, 'n_estimators': 60}"								
68	"{'max_depth': 4, 'max_samples': 0.30000000000000004, 'n_estimators': 110}"	0.721360887	0.029036009	0.709615346	0.031581373	0.72221258	0.028874795	0.723046887	0.036738384
69	"{'max_depth': 4, 'max_samples': 0.30000000000000004, 'n_estimators': 160}"	0.727221964	0.015585242	0.716846163	0.014440294	0.728527906	0.0153	0.729603427	0.018300791
70	"{'max_depth': 4, 'max_samples': 0.30000000000000004, 'n_estimators': 210}"	0.729172131	0.013295229	0.719067106	0.011808651	0.730739667	0.013295078	0.730806904	0.01432994
71	"{'max_depth': 4, 'max_samples': 0.30000000000000004, 'n_estimators': 260}"	0.73243369	0.018743149	0.723222531	0.018796148	0.734347324	0.01845343	0.734515412	0.022718146
72	"{'max_depth': 4, 'max_samples': 0.50000000000000001, 'n_estimators': 10}"	0.643891451	0.027374346	0.615669069	0.03076129	0.642249558	0.027186175	0.632368097	0.029189456
73	"{'max_depth': 4,	0.717460552	0.033761041	0.705524365	0.037107728	0.71844445	0.033437871	0.71897519	0.038397908

	'max_samples': 0.50000000000000001, 'n_estimators': 60}"								
74	"{'max_depth': 4, 'max_samples': 0.50000000000000001, 'n_estimators': 110}"	0.726564152	0.030400112	0.716548762	0.033392772	0.72752188	0.030000655	0.731096026	0.034812941
75	"{'max_depth': 4, 'max_samples': 0.50000000000000001, 'n_estimators': 160}"	0.732423114	0.025499994	0.723648197	0.026188595	0.733465373	0.024925477	0.736326522	0.027768331
76	"{'max_depth': 4, 'max_samples': 0.50000000000000001, 'n_estimators': 210}"	0.734375397	0.026101094	0.726018721	0.027458651	0.735522269	0.025723766	0.737411879	0.028125822
77	"{'max_depth': 4, 'max_samples': 0.50000000000000001, 'n_estimators': 260}"	0.738939887	0.028943958	0.730545955	0.030743431	0.7404544	0.028753649	0.741405349	0.032240557
78	"{'max_depth': 4, 'max_samples': 0.70000000000000001, 'n_estimators': 10}"	0.639989001	0.016966106	0.613181987	0.023699065	0.639281481	0.017065928	0.638858822	0.033237927
79	"{'max_depth': 4,	0.721365117	0.024116509	0.712415548	0.025014084	0.72244845	0.023469666	0.720174516	0.024335738

	'max_samples': 0.70000000000001, 'n_estimators': 60}"								
80	"{'max_depth': 4, 'max_samples': 0.70000000000001, 'n_estimators': 110}"	0.724620331	0.026144503	0.715490937	0.028139674	0.725606362	0.025032949	0.725215761	0.026535088
81	"{'max_depth': 4, 'max_samples': 0.70000000000001, 'n_estimators': 160}"	0.72787766	0.023286055	0.718289885	0.024638308	0.729067091	0.022556494	0.728030634	0.024175601
82	"{'max_depth': 4, 'max_samples': 0.70000000000001, 'n_estimators': 210}"	0.73242946	0.022031697	0.724294235	0.020995022	0.733763774	0.020725428	0.733459024	0.019822717
83	"{'max_depth': 4, 'max_samples': 0.70000000000001, 'n_estimators': 260}"	0.730477178	0.019999634	0.721385731	0.019529404	0.732139837	0.018698895	0.729861149	0.020798324
84	"{'max_depth': 4, 'max_samples': 0.90000000000001, 'n_estimators': 10}"	0.649107407	0.039891997	0.631785317	0.041606172	0.647416701	0.038773997	0.653944707	0.037125372
85	"{'max_depth': 4,	0.719404374	0.020176346	0.711181191	0.020621548	0.719791409	0.020121587	0.721492481	0.018306637

	'max_samples': 0.9000000000000001, 'n_estimators': 60}"								
86	"{'max_depth': 4, 'max_samples': 0.9000000000000001, 'n_estimators': 110}"	0.73698972	0.02218614	0.729443579	0.023220826	0.737619045	0.021463993	0.738767456	0.022232089
87	"{'max_depth': 4, 'max_samples': 0.9000000000000001, 'n_estimators': 160}"	0.740898515	0.021689426	0.733767021	0.022585363	0.74187677	0.020960206	0.740977486	0.022364099
88	"{'max_depth': 4, 'max_samples': 0.9000000000000001, 'n_estimators': 210}"	0.745458776	0.026322223	0.738439381	0.026568496	0.746634778	0.025642935	0.746801984	0.025920756
89	"{'max_depth': 4, 'max_samples': 0.9000000000000001, 'n_estimators': 260}"	0.742850797	0.01990701	0.735998404	0.020593077	0.744379266	0.019196142	0.744716837	0.020498541
90	"{'max_depth': 5, 'max_samples': 0.1, 'n_estimators': 10}"	0.638673379	0.021078644	0.621037271	0.026150419	0.638687624	0.021713945	0.645675989	0.021177268
91	"{'max_depth': 5, 'max_samples': 0.1,	0.724616101	0.024606203	0.716955639	0.024009488	0.726520553	0.023961057	0.726490218	0.025135718

	'n_estimators': 60}"								
92	"{'max_depth': 5, 'max_samples': 0.1, 'n_estimators': 110}"	0.736338255	0.019064151	0.72937196	0.017905987	0.737413074	0.018310299	0.736599047	0.022160825
93	"{'max_depth': 5, 'max_samples': 0.1, 'n_estimators': 160}"	0.737643301	0.023653154	0.731248746	0.02289123	0.738657067	0.023453511	0.736905452	0.024655799
94	"{'max_depth': 5, 'max_samples': 0.1, 'n_estimators': 210}"	0.743504378	0.030236773	0.736921617	0.029404785	0.744910317	0.029960867	0.742608312	0.032415979
95	"{'max_depth': 5, 'max_samples': 0.1, 'n_estimators': 260}"	0.744149499	0.029166709	0.737658343	0.028774834	0.745589754	0.029203291	0.742697873	0.03048988
96	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 10}"	0.688176319	0.023862798	0.676318954	0.024214363	0.68781509	0.025981952	0.68614975	0.02133465
97	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 60}"	0.752614324	0.020154706	0.747296937	0.020511273	0.753359675	0.01983756	0.755235247	0.017441769
98	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 110}"	0.764340708	0.022223099	0.759581385	0.021811247	0.765345216	0.021717449	0.76712086	0.02311635

99	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 160}"	0.763033546	0.018636483	0.758111266	0.017680563	0.764053807	0.017916348	0.764347592	0.019449943
100	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 210}"	0.76238208	0.020130963	0.757148029	0.020056243	0.763408094	0.019530765	0.763471343	0.021669298
101	"{'max_depth': 5, 'max_samples': 0.30000000000000004, 'n_estimators': 260}"	0.761734845	0.023792006	0.756963164	0.023257892	0.762889262	0.023073639	0.76271275	0.025064578
102	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 10}"	0.701178138	0.029600576	0.690774003	0.032480851	0.701019248	0.02863733	0.699988512	0.029424944
103	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 60}"	0.765645755	0.027820413	0.76064488	0.027678824	0.766516727	0.028522471	0.76720801	0.028880512
104	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 110}"	0.764990059	0.025763888	0.760212103	0.025319504	0.765947627	0.025519194	0.769055744	0.025458214

105	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 160}"	0.771502602	0.028632583	0.767169947	0.029085439	0.772277983	0.028538898	0.774719802	0.030561682
106	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 210}"	0.776703752	0.024663439	0.77276383	0.024481217	0.77747181	0.024505461	0.78099737	0.025122758
107	"{'max_depth': 5, 'max_samples': 0.5000000000000001, 'n_estimators': 260}"	0.771500486	0.025015806	0.76743265	0.025698996	0.772294062	0.024904209	0.775080034	0.027659969
108	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 10}"	0.714194763	0.027799096	0.70057571	0.02919262	0.715124924	0.028304836	0.711196184	0.031050202
109	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 60}"	0.764342823	0.026866351	0.759493175	0.02714922	0.765379712	0.027448107	0.767243574	0.028257804
110	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 110}"	0.770191209	0.019711797	0.766321953	0.018166582	0.771600247	0.019266865	0.775138193	0.019576672

111	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 160}"	0.769541859	0.017253545	0.765053812	0.016989621	0.770628262	0.017170579	0.772195405	0.018815868
112	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 210}"	0.772797073	0.021622138	0.768532319	0.021194359	0.774042794	0.021467994	0.775517669	0.023010205
113	"{'max_depth': 5, 'max_samples': 0.7000000000000001, 'n_estimators': 260}"	0.77084479	0.021630483	0.766634475	0.021566465	0.77191504	0.021171027	0.772575245	0.021685089
114	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 10}"	0.710298659	0.030150088	0.699511833	0.029962707	0.710943744	0.031780972	0.722671253	0.033716486
115	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 60}"	0.761085494	0.028121805	0.755896131	0.028446676	0.76264543	0.027984376	0.764104405	0.026959908
116	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 110}"	0.763037777	0.024136889	0.758287663	0.02370863	0.764527974	0.023367679	0.76488778	0.022267036

117	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 160}"	0.770201785	0.023141751	0.766050159	0.023559241	0.771183612	0.02299744	0.771709193	0.024919175
118	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 210}"	0.768245273	0.022607428	0.763961805	0.022814526	0.769174811	0.022245466	0.769232858	0.023829125
119	"{'max_depth': 5, 'max_samples': 0.9000000000000001, 'n_estimators': 260}"	0.77019967	0.023621522	0.765902019	0.02394382	0.771240104	0.02354934	0.77058668	0.023105724
120	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 10}"	0.664059393	0.027550247	0.649601524	0.031725717	0.661738284	0.027403557	0.660503188	0.03300148
121	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 60}"	0.73634037	0.022879077	0.730552577	0.021324549	0.736790159	0.022711483	0.74184391	0.022058857
122	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 110}"	0.755211726	0.024110836	0.749912427	0.022513812	0.7563514	0.02281185	0.755124581	0.022482449
123	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 160}"	0.760427683	0.027887093	0.754721594	0.026610336	0.761212943	0.027137265	0.760843185	0.026321198

124	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 210}"	0.765635179	0.019949644	0.760826156	0.017705288	0.766347457	0.019204055	0.766430627	0.01758442
125	"{'max_depth': 6, 'max_samples': 0.1, 'n_estimators': 260}"	0.763033546	0.022832557	0.757974721	0.020878056	0.7640047 9	0.022081171	0.762004365	0.021539721
126	"{'max_depth': 6, 'max_samples': 0.300000000000004, 'n_estimators': 10}"	0.708354837	0.028341798	0.69768901	0.032785886	0.708689974	0.027231409	0.707377219	0.028840345
127	"{'max_depth': 6, 'max_samples': 0.300000000000004, 'n_estimators': 60}"	0.77019544	0.021112417	0.765597377	0.020947886	0.770752372	0.020889465	0.773077708	0.018076099
128	"{'max_depth': 6, 'max_samples': 0.300000000000004, 'n_estimators': 110}"	0.769541859	0.026554024	0.765427988	0.02534522	0.770337844	0.026021388	0.771294182	0.024569882
129	"{'max_depth': 6, 'max_samples': 0.300000000000004, 'n_estimators': 160}"	0.778651804	0.024577595	0.774785326	0.023442956	0.779242755	0.024108782	0.78187185	0.023307389
130	"{'max_depth': 6, 'max_samples': 0.30000000000000	0.779313846	0.0312 0249	0.775458701	0.030818349	0.779723019	0.030531199	0.781583479	0.031824808

	004, 'n_estimators': 210}"								
131	"{'max_depth': 6, 'max_samples': 0.30000000000000004, 'n_estimators': 260}"	0.781266128	0.028732845	0.77754862	0.028100295	0.782332429	0.028114682	0.782292696	0.029044629
132	"{'max_depth': 6, 'max_samples': 0.5000000000000001, 'n_estimators': 10}"	0.706391979	0.028518362	0.696066722	0.0293053	0.708321398	0.028095361	0.706732171	0.029040971
133	"{'max_depth': 6, 'max_samples': 0.5000000000000001, 'n_estimators': 60}"	0.779958966	0.022411048	0.777214023	0.022017724	0.781020311	0.022281019	0.782162656	0.021855238
134	"{'max_depth': 6, 'max_samples': 0.5000000000000001, 'n_estimators': 110}"	0.781921824	0.027832537	0.778838742	0.027157748	0.782497112	0.027526164	0.78430482	0.02734994
135	"{'max_depth': 6, 'max_samples': 0.5000000000000001, 'n_estimators': 160}"	0.785824273	0.023709503	0.782864589	0.023666504	0.7863598	0.023890477	0.788001517	0.022285152
136	"{'max_depth': 6, 'max_samples': 0.50000000000000	0.78712509	0.020271075	0.784165714	0.020300195	0.788056264	0.020408015	0.789487048	0.020418892

	01, 'n_estimators': 210}"								
137	"{'max_depth': 6, 'max_samples': 0.50000000000001, 'n_estimators': 260}"	0.785170692	0.021544062	0.782290293	0.021216224	0.786295582	0.021474671	0.787549229	0.019899251
138	"{'max_depth': 6, 'max_samples': 0.70000000000001, 'n_estimators': 10}"	0.748690723	0.009704855	0.742249094	0.009583331	0.749972357	0.009289315	0.747826991	0.010344062
139	"{'max_depth': 6, 'max_samples': 0.70000000000001, 'n_estimators': 60}"	0.772801303	0.018382998	0.769247422	0.01822927	0.773820592	0.018237704	0.775390699	0.017184287
140	"{'max_depth': 6, 'max_samples': 0.70000000000001, 'n_estimators': 110}"	0.772799188	0.022368577	0.769450749	0.021583833	0.773680452	0.021383895	0.774832771	0.021535557
141	"{'max_depth': 6, 'max_samples': 0.70000000000001, 'n_estimators': 160}"	0.781913364	0.020037249	0.778557832	0.019190927	0.782414485	0.019203425	0.785085169	0.01931467
142	"{'max_depth': 6, 'max_samples': 0.70000000000000	0.781907018	0.023063991	0.778920654	0.022739948	0.782632817	0.022749371	0.783734517	0.022224833

	01, 'n_estimators': 210}"								
143	"{'max_depth': 6, 'max_samples': 0.70000000000001, 'n_estimators': 260}"	0.780610432	0.019684101	0.777425868	0.019632523	0.78123757	0.01914219	0.781546478	0.01903651
144	"{'max_depth': 6, 'max_samples': 0.90000000000001, 'n_estimators': 10}"	0.743493803	0.018637968	0.738577743	0.016815562	0.745155032	0.018732724	0.742111989	0.016801135
145	"{'max_depth': 6, 'max_samples': 0.90000000000001, 'n_estimators': 60}"	0.78321841	0.023293448	0.780803562	0.022566047	0.784565698	0.023588372	0.785559	0.023285201
146	"{'max_depth': 6, 'max_samples': 0.90000000000001, 'n_estimators': 110}"	0.781261898	0.0233119	0.778728848	0.02276766	0.782487267	0.023280173	0.783279284	0.022705503
147	"{'max_depth': 6, 'max_samples': 0.90000000000001, 'n_estimators': 160}"	0.783869876	0.024960333	0.781333676	0.024547017	0.785127414	0.025198594	0.787099479	0.023482177
148	"{'max_depth': 6, 'max_samples': 0.90000000000000	0.78712509	0.022266523	0.784352718	0.02215797	0.788404484	0.022114206	0.789365478	0.022210385

	01, 'n_estimators': 210}"								
149	"{'max_depth': 6, 'max_samples': 0.900000000000001, 'n_estimators': 260}"	0.787122975	0.02050607	0.784153864	0.020210703	0.788056264	0.020716927	0.788921262	0.020335588
150	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 10}"	0.676454165	0.037698728	0.666246214	0.04063473	0.675426833	0.037308667	0.674218936	0.038915813
151	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 60}"	0.755869538	0.022257669	0.750544827	0.022078697	0.757104389	0.020803669	0.752849236	0.022330646
152	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 110}"	0.760423453	0.017208442	0.755853401	0.01524772	0.76112286	0.015912024	0.75836631	0.017021718
153	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 160}"	0.771492026	0.017901695	0.766774201	0.016808988	0.772173455	0.016798235	0.769186869	0.017579595
154	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 210}"	0.77084056	0.015383169	0.766184904	0.013894282	0.771153634	0.014925339	0.769054078	0.015668619
155	"{'max_depth': 7, 'max_samples': 0.1, 'n_estimators': 260}"	0.766938111	0.017146674	0.762132617	0.015703921	0.767602753	0.015969082	0.76370602	0.016345111
156	"{'max_depth': 7, 'max_samples':	0.743498033	0.024051111	0.73773469	0.02585341	0.744409264	0.025116565	0.73949486	0.023665109

	0.300000000000000004, 'n_estimators': 10}"								
157	"{'max_depth': 7, 'max_samples': 0.300000000000000004, 'n_estimators': 60}"	0.792319895	0.019884607	0.789388794	0.019943785	0.792668902	0.019891191	0.792368446	0.019580941
158	"{'max_depth': 7, 'max_samples': 0.300000000000000004, 'n_estimators': 110}"	0.789075257	0.019103843	0.786194633	0.018158543	0.789465702	0.018425636	0.789376271	0.018246098
159	"{'max_depth': 7, 'max_samples': 0.300000000000000004, 'n_estimators': 160}"	0.79622869	0.015972385	0.793040978	0.015061324	0.796242898	0.01576199	0.798312521	0.015711985
160	"{'max_depth': 7, 'max_samples': 0.300000000000000004, 'n_estimators': 210}"	0.796880156	0.013830285	0.793819583	0.012784662	0.797004874	0.013412833	0.79838257	0.012506433
161	"{'max_depth': 7, 'max_samples': 0.300000000000000004, 'n_estimators': 260}"	0.791668429	0.014684273	0.787840106	0.01414663	0.792039348	0.014187916	0.79155653	0.013684313
162	"{'max_depth': 7, 'max_samples':	0.751954397	0.010549925	0.747956308	0.010312455	0.752387115	0.010573583	0.751934908	0.010940784

	0.50000000000000001, 'n_estimators': 10}"								
163	"{'max_depth': 7, 'max_samples': 0.50000000000000001, 'n_estimators': 60}"	0.800156521	0.02768506	0.797438911	0.027225856	0.799541799	0.027785655	0.802064717	0.025462087
164	"{'max_depth': 7, 'max_samples': 0.50000000000000001, 'n_estimators': 110}"	0.800799526	0.022512896	0.798611196	0.021798315	0.80081618	0.022131924	0.802204064	0.021096865
165	"{'max_depth': 7, 'max_samples': 0.50000000000000001, 'n_estimators': 160}"	0.798849359	0.023401812	0.796120199	0.02281471	0.798623077	0.023089748	0.800813318	0.023476215
166	"{'max_depth': 7, 'max_samples': 0.50000000000000001, 'n_estimators': 210}"	0.804052625	0.022349875	0.801659235	0.021913279	0.803799309	0.022324935	0.806541332	0.021004975
167	"{'max_depth': 7, 'max_samples': 0.50000000000000001, 'n_estimators': 260}"	0.803396929	0.020017962	0.800394464	0.018951719	0.803209602	0.01975752	0.804996866	0.018395875
168	"{'max_depth': 7, 'max_samples':	0.756533694	0.034979765	0.752459782	0.033673305	0.758229736	0.034494425	0.75738025	0.032927228

	0.70000000000001, 'n_estimators': 10}"								
169	"{'max_depth': 7, 'max_samples': 0.70000000000001, 'n_estimators': 60}"	0.783865646	0.028345334	0.781369149	0.028484728	0.785059025	0.02796496	0.786072493	0.029191952
170	"{'max_depth': 7, 'max_samples': 0.70000000000001, 'n_estimators': 110}"	0.789726723	0.024647968	0.787440398	0.023577684	0.790629862	0.024173307	0.790629169	0.023537671
171	"{'max_depth': 7, 'max_samples': 0.70000000000001, 'n_estimators': 160}"	0.793635518	0.020484358	0.791181149	0.020016365	0.794128288	0.019951882	0.795441565	0.020654469
172	"{'max_depth': 7, 'max_samples': 0.70000000000001, 'n_estimators': 210}"	0.794282753	0.016624601	0.791382803	0.016158235	0.794781995	0.015973318	0.795430409	0.015610417
173	"{'max_depth': 7, 'max_samples': 0.70000000000001, 'n_estimators': 260}"	0.797533737	0.015223138	0.794651381	0.01445581	0.797693932	0.014941895	0.79831024	0.013181152
174	"{'max_depth': 7, 'max_samples':	0.764319557	0.025477559	0.761898	0.025761969	0.764510683	0.026792697	0.766158936	0.026186803

	0.90000000000001, 'n_estimators': 10}"								
175	"{'max_depth': 7, 'max_samples': 0.90000000000001, 'n_estimators': 60}"	0.79687381	0.0131423	0.794762919	0.011825486	0.797344416	0.013507418	0.79991798	0.007444503
176	"{'max_depth': 7, 'max_samples': 0.90000000000001, 'n_estimators': 110}"	0.802743348	0.016180664	0.800334652	0.015361103	0.803094595	0.015910759	0.804589533	0.015504625
177	"{'max_depth': 7, 'max_samples': 0.90000000000001, 'n_estimators': 160}"	0.80469563	0.015982317	0.802078586	0.015817388	0.804775189	0.016342564	0.806271543	0.014766951
178	"{'max_depth': 7, 'max_samples': 0.90000000000001, 'n_estimators': 210}"	0.805349211	0.017519515	0.802288145	0.017124124	0.805212908	0.017715107	0.806510609	0.016077541
179	"{'max_depth': 7, 'max_samples': 0.90000000000001, 'n_estimators': 260}"	0.805998562	0.017199559	0.80293529	0.01703865	0.805940181	0.017377228	0.807213813	0.016279452
180	"{'max_depth': 8, 'max_samples': 0.1,	0.678402217	0.027065215	0.665562019	0.026885744	0.677950147	0.025858022	0.669252639	0.030292188

	'n_estimators': 10}"								
181	"{'max_depth': 8, 'max_samples': 0.1, 'n_estimators': 60}"	0.755209611	0.01598774	0.750578105	0.015373372	0.755673918	0.015294963	0.752448188	0.015685319
182	"{'max_depth': 8, 'max_samples': 0.1, 'n_estimators': 110}"	0.770191209	0.025109109	0.7660314	0.023760992	0.770382426	0.024558404	0.769357379	0.022437146
183	"{'max_depth': 8, 'max_samples': 0.1, 'n_estimators': 160}"	0.774097889	0.018254752	0.768904345	0.016858585	0.774379744	0.017875593	0.771334877	0.017536487
184	"{'max_depth': 8, 'max_samples': 0.1, 'n_estimators': 210}"	0.776693177	0.019163558	0.7720344	0.018020966	0.776352651	0.019051999	0.775636577	0.018341764
185	"{'max_depth': 8, 'max_samples': 0.1, 'n_estimators': 260}"	0.775390245	0.015960257	0.770745258	0.014300993	0.775747237	0.015670264	0.773917378	0.014680365
186	"{'max_depth': 8, 'max_samples': 0.30000000000000004, 'n_estimators': 10}"	0.738939887	0.015876716	0.735916423	0.016036015	0.739771074	0.015146204	0.736464375	0.018200932
187	"{'max_depth': 8, 'max_samples': 0.30000000000000004, 'n_estimators': 60}"	0.786458818	0.022804342	0.783565095	0.021986576	0.786540517	0.021636839	0.786007674	0.021350491
188	"{'max_depth': 8, 'max_samples':	0.791670544	0.019390253	0.788604468	0.018472677	0.791565434	0.018702528	0.791968984	0.017764173

	0.30000000000000004, 'n_estimators': 110}"								
189	"{'max_depth': 8, 'max_samples': 0.30000000000000004, 'n_estimators': 160}"	0.799488134	0.018618194	0.796179117	0.018278132	0.799057845	0.018591567	0.799492956	0.018887822
190	"{'max_depth': 8, 'max_samples': 0.30000000000000004, 'n_estimators': 210}"	0.797533737	0.023989876	0.794070612	0.02383477	0.79703213	0.023755752	0.797576869	0.025164495
191	"{'max_depth': 8, 'max_samples': 0.30000000000000004, 'n_estimators': 260}"	0.797533737	0.021957788	0.794051953	0.021674919	0.797233621	0.022002042	0.797229217	0.021826645
192	"{'max_depth': 8, 'max_samples': 0.50000000000000001, 'n_estimators': 10}"	0.764334363	0.022112199	0.759503967	0.020879849	0.7657675	0.020782182	0.760765028	0.022432567
193	"{'max_depth': 8, 'max_samples': 0.50000000000000001, 'n_estimators': 60}"	0.792981937	0.021524623	0.790550285	0.020961106	0.793423986	0.021337746	0.793104254	0.020427528
194	"{'max_depth': 8, 'max_samples':	0.800145945	0.021891333	0.797377128	0.021785463	0.800344589	0.02182637	0.800020117	0.020610889

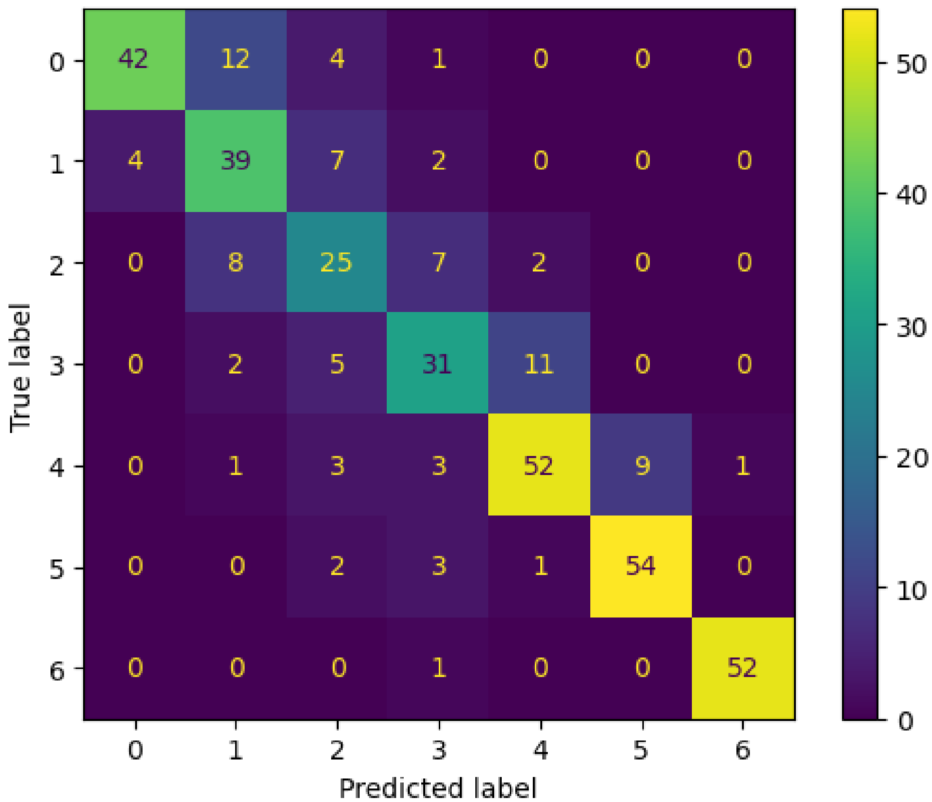
	0.500000 00000000 01, 'n_estima tors': 110}"								
1 9 5	"{'max_de pth': 8, 'max_sam ples': 0.500000 00000000 01, 'n_estima tors': 160}"	0.80274 3348	0.0148 1125	0.799581 415	0.01437 3694	0.8027016 37	0.014981 396	0.80222273 5	0.0125236 72
1 9 6	"{'max_de pth': 8, 'max_sam ples': 0.500000 00000000 01, 'n_estima tors': 210}"	0.80534 4981	0.0168 32821	0.802246 715	0.01661 8357	0.8048564 35	0.017198 086	0.80623004	0.0150817 68
1 9 7	"{'max_de pth': 8, 'max_sam ples': 0.500000 00000000 01, 'n_estima tors': 260}"	0.80729 5148	0.0156 97378	0.804531 642	0.01513 7858	0.8071540 7	0.016273 345	0.80804156 3	0.0126264 95
1 9 8	"{'max_de pth': 8, 'max_sam ples': 0.700000 00000000 01, 'n_estima tors': 10}"	0.76368 0782	0.0176 64817	0.759925 01	0.01793 424	0.7646945 16	0.017516 87	0.75969620 2	0.0181890 97
1 9 9	"{'max_de pth': 8, 'max_sam ples': 0.700000 00000000 01, 'n_estima tors': 60}"	0.79754 4312	0.0185 8593	0.795651 469	0.01813 445	0.7984735 07	0.017935 021	0.7972289	0.0171534 87
2 0 0	"{'max_de pth': 8, 'max_sam ples':	0.79884 7244	0.0188 0249	0.796733 258	0.01809 5229	0.7992849 91	0.018781 517	0.79966252 3	0.0169877 5

	0.700000000000000001, 'n_estimators': 110}"								
201	"{'max_depth': 8, 'max_samples': 0.700000000000000001, 'n_estimators': 160}"	0.800797411	0.019724753	0.798151377	0.019311779	0.801089237	0.019288101	0.801359751	0.018402592
202	"{'max_depth': 8, 'max_samples': 0.700000000000000001, 'n_estimators': 210}"	0.804708321	0.023914025	0.802585713	0.023598555	0.804989239	0.023826845	0.805668548	0.022327526
203	"{'max_depth': 8, 'max_samples': 0.700000000000000001, 'n_estimators': 260}"	0.806007022	0.019841915	0.803978146	0.019242776	0.806300134	0.019931377	0.806748248	0.018409075
204	"{'max_depth': 8, 'max_samples': 0.900000000000000001, 'n_estimators': 10}"	0.767595922	0.018471465	0.76435998	0.018637765	0.767418731	0.018681408	0.765848007	0.020286558
205	"{'max_depth': 8, 'max_samples': 0.900000000000000001, 'n_estimators': 60}"	**0.81251322**	0.016127878	**0.811035181**	0.016242644	**0.812825099**	0.015696488	**0.814591074**	0.014729322
206	"{'max_depth': 8, 'max_samples':	0.806656373	0.017837043	0.80450905	0.017391095	0.806846727	0.01761985	0.807374608	0.015470476

	0.90000000000000001, 'n_estimators': 110}"								
207	"{'max_depth': 8, 'max_samples': 0.90000000000000001, 'n_estimators': 160}"	0.809260121	0.020707253	0.806986273	0.020236216	0.809493602	0.020459188	0.808684599	0.018718991
208	"{'max_depth': 8, 'max_samples': 0.90000000000000001, 'n_estimators': 210}"	0.810565168	0.018825518	0.808003581	0.01861276	0.810441367	0.018555354	0.81132597	0.017607062
209	"{'max_depth': 8, 'max_samples': 0.90000000000000001, 'n_estimators': 260}"	0.806654258	0.015984704	0.804168614	0.015868993	0.806468319	0.015892521	0.807591392	0.01446623

Se constată că performanța acestor RandomForests sunt dictate direct de adâncimea maximă a arborelui, numărul de arbori și de procentul de intrări care sunt folosite la antrenare (max_samples). Un RandomForest la care fixăm numărul de estimatori și adâncimea maximă a arborelui are performanțe mai mari atunci când folosim mai multe date prin creșterea max_samples.

Pentru configurația bolduită matricea de confuzie este



Observăm că în cazul acestui RandomForest, puterea de predicție pentru clasa de indice 6 (D6) a rămas puternică. Vedem o îmbunătățire în predicțiile claselor de indici 3,4 și 5.