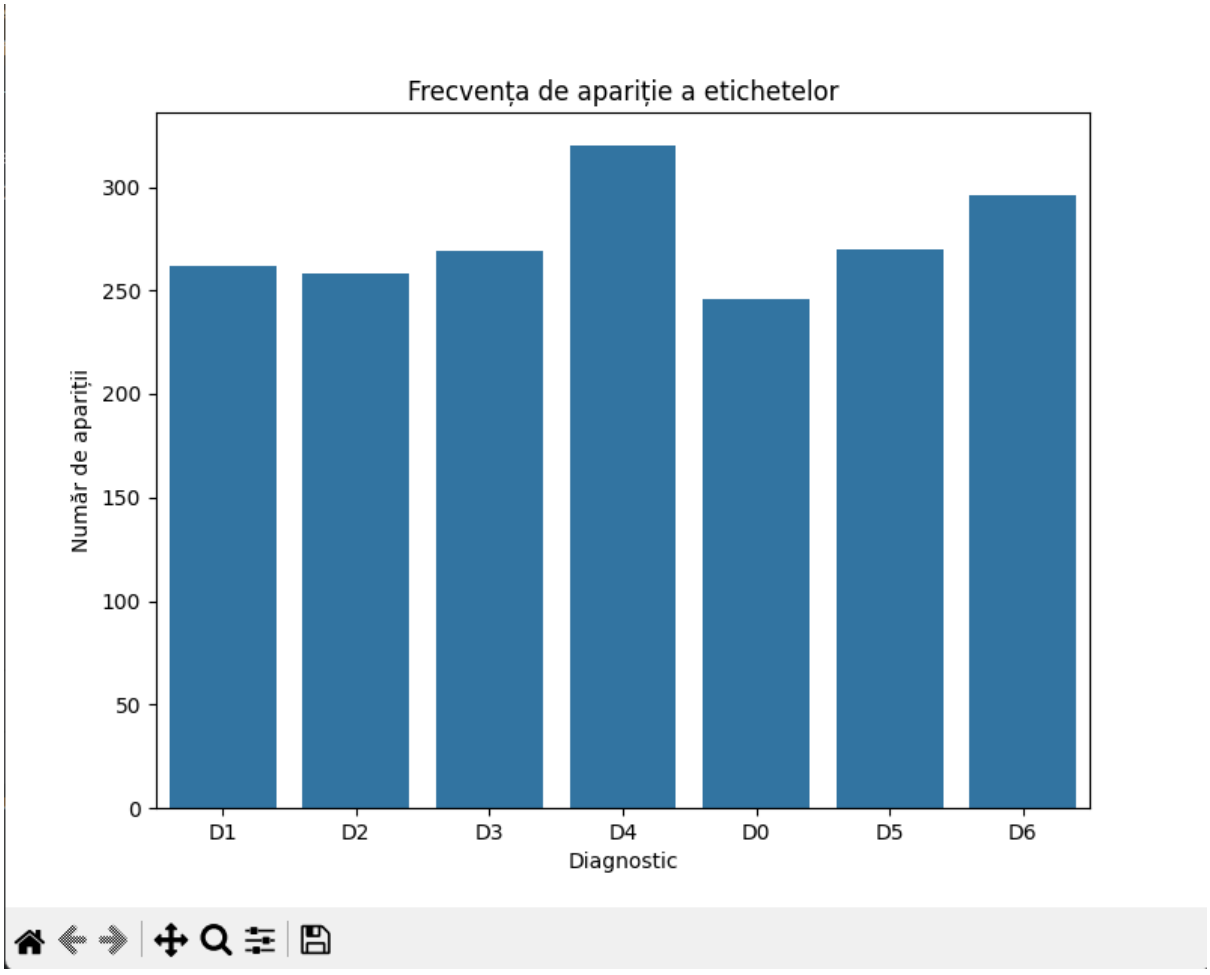
Tema 1 ML

Nitu David-Gabriel 342C2

Explorarea Datelor:

Mai intai am extras datele din fisierul CSV si le-am separat in clase numerice si clase categorice.

Primul bar plot pe care l-am facut arata de cate ori apare fiecare diagnostic(clasa).



Pentru clasele numerice am facut urmatoarele statistici din care se poate observa destul de clar ca unele medii sunt fara sens (ex: Height –> 3.57 sau Age -> 44.79, avand in vedere ca daca ne uitam prin datele din Age vom afla ca majoritatea varstelor sunt sub 30 de ani).

In plus observam ca sunt multe valori ciudate precum min Weight -1 sau max Technology_time_use 1306 (NU sunt atatea ore intr-o zi).

Statistici pentru atributele numerice:

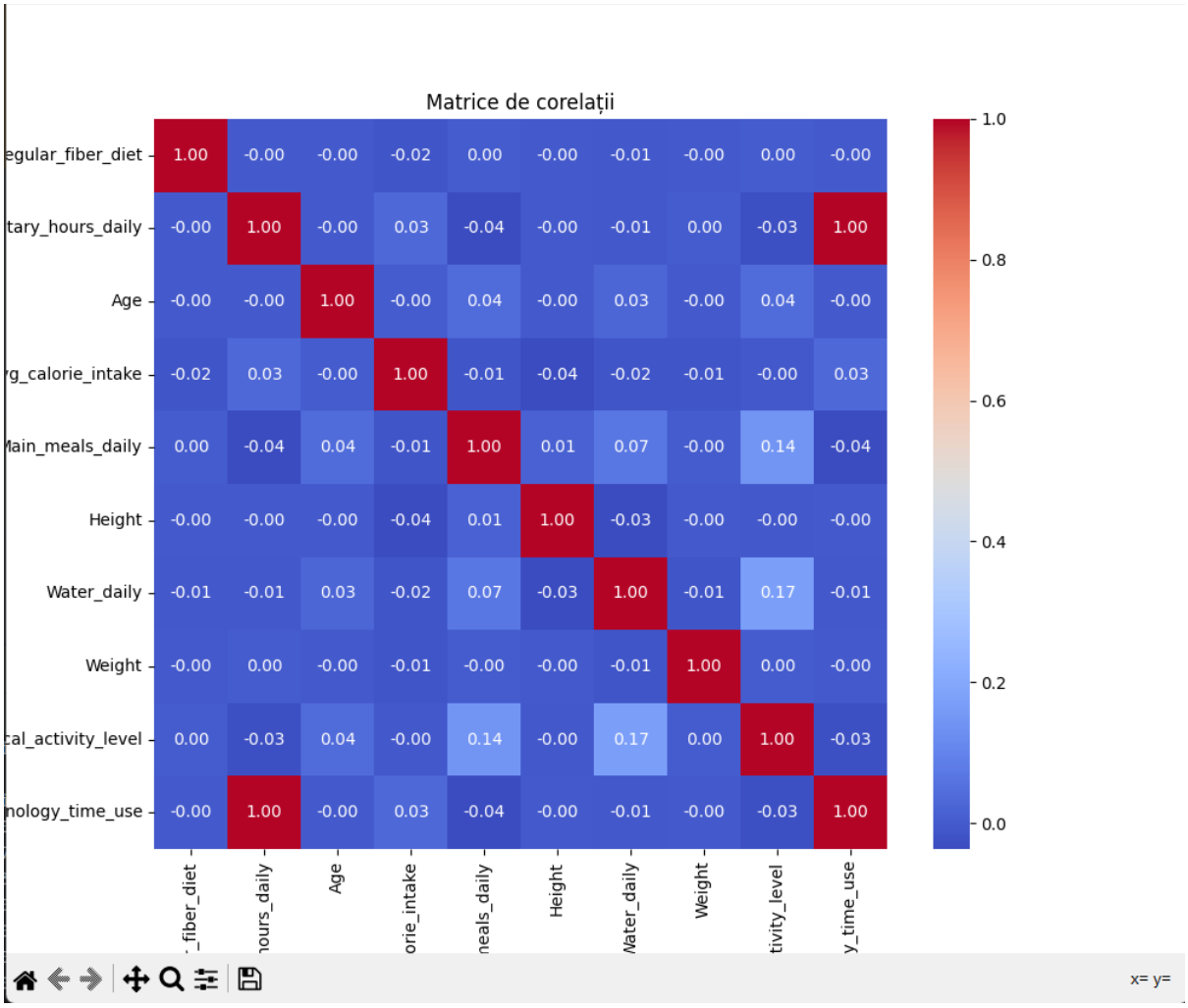
	Regular_fiber_diet	Sedentary_hours_daily	Age	Est_avg_calorie_intake
count	1921.000000	1921.000000	1921.000000	1921.000000
mean	3.844937	3.693571	44.792506	2253.687663
std	62.439617	21.759835	633.311837	434.075794
min	1.000000	2.210000	15.000000	1500.000000
25%	2.000000	2.770000	19.971660	1871.000000
50%	2.387426	3.130000	22.829753	2253.000000
75%	3.000000	3.640000	26.000000	2628.000000
max	2739.000000	956.580000	19685.000000	3000.000000

	Main_meals_daily	Height	Water_daily	Weight
count	1921.000000	1921.000000	1921.000000	1921.000000
mean	2.683472	3.573488	2.010367	205.637344
std	0.779179	58.098160	0.611034	3225.653536
min	1.000000	1.450000	1.000000	-1.000000
25%	2.658639	1.630000	1.606076	58.830710
50%	3.000000	1.700000	2.000000	80.386078
75%	3.000000	1.770000	2.480555	105.036075
max	4.000000	1915.000000	3.000000	82628.000000

	Physical_activity_level	Technology_time_use
count	1921.000000	1921.000000
mean	1.012640	1.345653
std	0.855526	29.789928
min	0.000000	0.000000
25%	0.115974	0.000000
50%	1.000000	1.000000
75%	1.683497	1.000000
max	3.000000	1306.000000

Am facut apoi matricea de corelatii. Ea ne arata cum impacteaza anumiti coeficienti ale unu atribut alti coeficienti ale altor atribute. Putem vedea astfel ce atribute au o relevanta mai mare in analiza in timp a datelor.

Se poate observa ca unele atribute influenteaza mai putin alte atribute in general. Aceste atribute au o importanta mai scazuta in prezicerea diagnosticului.



Apoi am luat atributele categorice si am extras cateva statistici folositoare, urmand apoi sa fac histograme pentru fiecare atribut.

Statistici pentru atributele categorice:

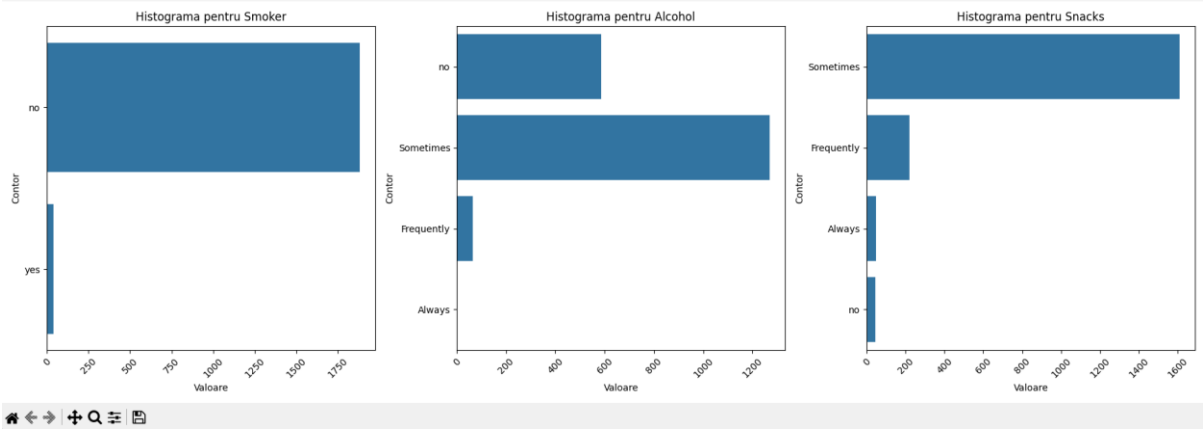
	Transportation	Diagnostic_in_family_history	High_calorie_diet
count	1921	1921	1921
unique	5	2	2
top	Public_Transportation	yes	yes
freq	1427	1573	1697

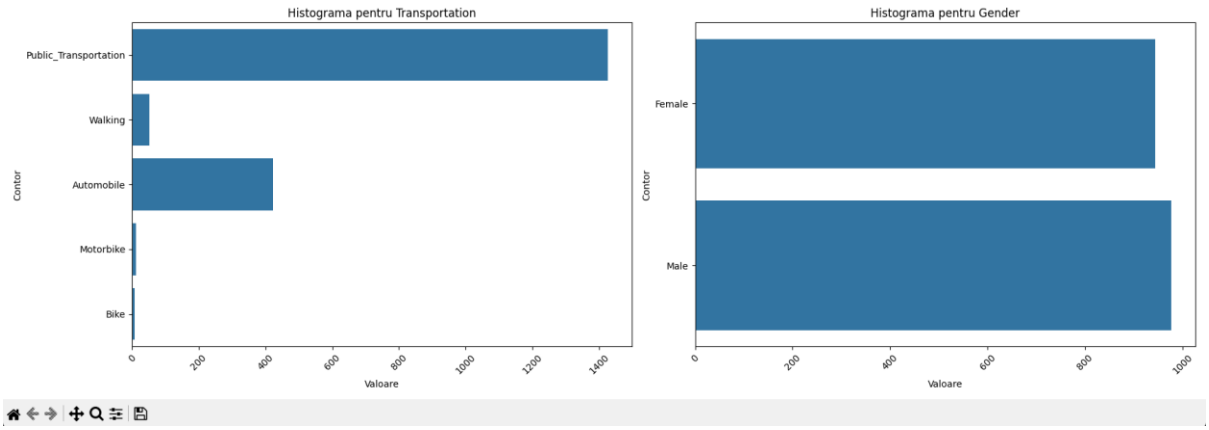
	Alcohol	Snacks	Smoker	Calorie_monitoring	Gender
count	1921	1921	1921	1921	1921
unique	4	4	2	2	2
top	Sometimes	Sometimes	no	no	Male
freq	1269	1609	1881	1838	977

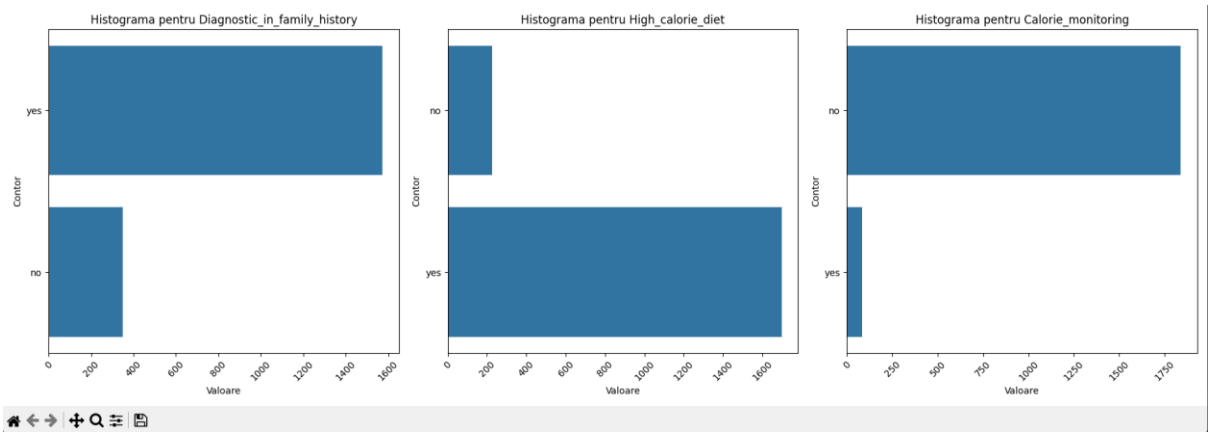
Foarte importante sunt campurile unique(care ne spune cate valori unice are fiecare atribut) si freq.

In histograme putem vedea cum sunt distribuite valorile fiecarui atribut peste setul de date. Astfel ne putem dezvolta o intelegere mai buna a masurilor pe care trebuie sa le luam pentru a avea algoritmi cat mai competenti.

Din histograme precum Smoker putem observa cat de drastica este diferenta dintre distributia celor doi coeficienti, lucru care se clasifica ca varianta insuficienta, facand astfel atributul sa nu fie luat in calcul de algoritm. Acest pas va fi efectuat la Feature Selection.







Am continuat apoi prin a rezolva problemele enumerate mai sus:
-Am inlocuit -1 din Weight cu valoarea mediana

-Am eliminat valorile exagerate prin calcularea de z scores si stabilirea unui threshold maxim acceptat

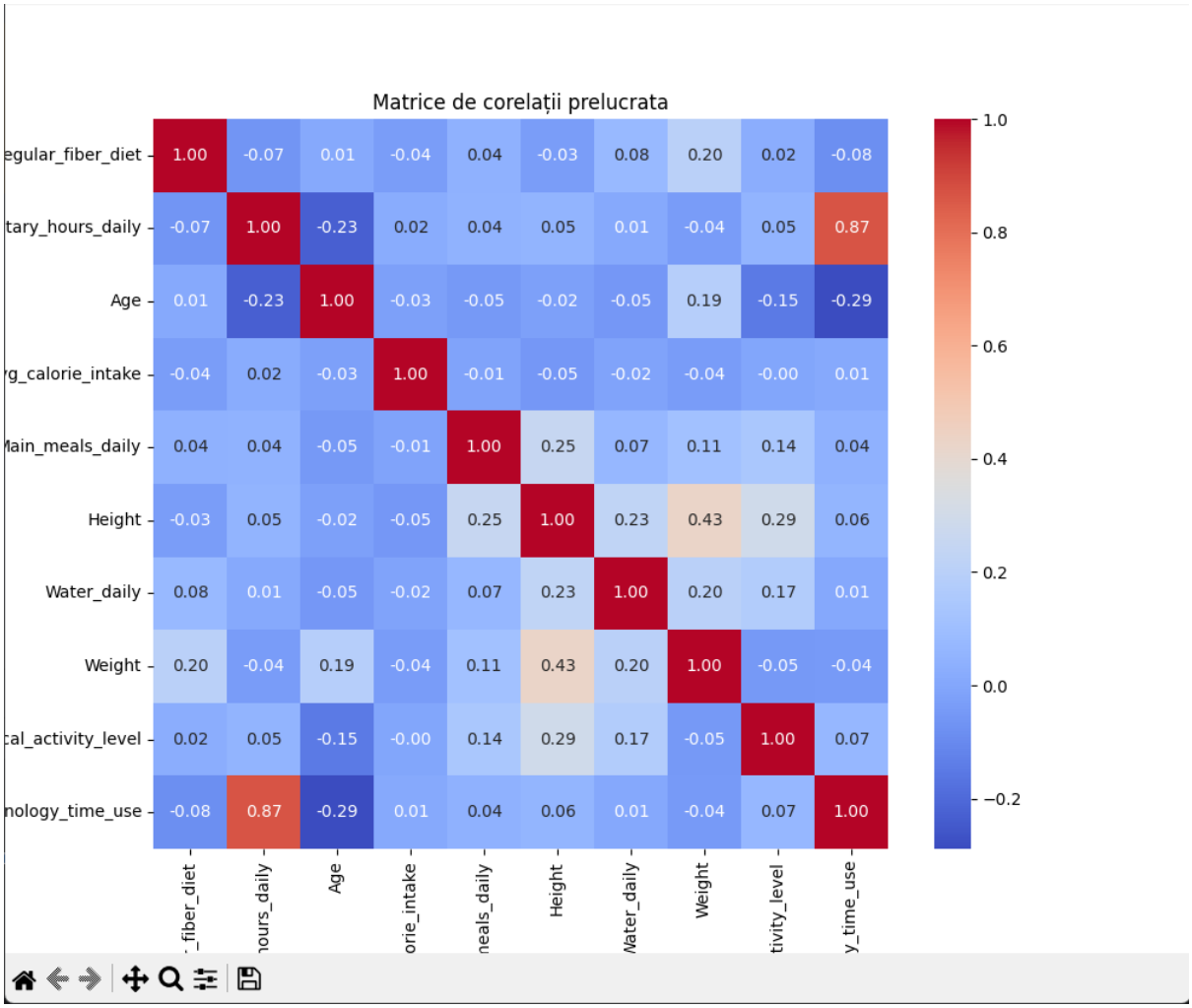
Apoi am refacut statisticile numerice si matricea de corelatie, pentru a vedea ce relatii sunt intre atribute pe setul actual de date, obtinut in urma micilor ajustari.

Statistici pentru atributele numerice prelucrate:

	Regular_fiber_diet	Sedentary_hours_daily	Age	Est_avg_calorie_intake
count	1912.000000	1912.000000	1912.000000	1912.000000
mean	2.420848	3.196308	24.362449	2254.305962
std	0.533787	0.575331	6.414437	433.767146
min	1.000000	2.210000	15.000000	1500.000000
25%	2.000000	2.770000	19.977773	1872.000000
50%	2.392422	3.130000	22.830929	2253.500000
75%	3.000000	3.632500	26.000000	2628.000000
max	3.000000	4.670000	61.000000	3000.000000

	Main_meals_daily	Height	Water_daily	Weight
count	1912.000000	1912.000000	1912.000000	1912.000000
mean	2.682859	1.702176	2.011031	86.433823
std	0.779635	0.093208	0.611325	24.931396
min	1.000000	1.450000	1.000000	39.000000
25%	2.658599	1.630000	1.605576	68.552572
50%	3.000000	1.700000	2.000000	83.337721
75%	3.000000	1.770000	2.482454	105.016878
max	4.000000	1.980000	3.000000	165.057269

	Physical_activity_level	Technology_time_use
count	1912.000000	1912.000000
mean	1.012089	0.666318
std	0.855361	0.675159
min	0.000000	0.000000
25%	0.115201	0.000000
50%	1.000000	1.000000
75%	1.683057	1.000000
max	3.000000	2.000000



Într-o matrice de corelații, o valoare de -1 indică o corelație negativă perfectă între cele două variabile. Acest lucru înseamnă că atunci când o variabilă crește, cealaltă variabilă scade într-o manieră perfectă. La 1 e fix invers. Astfel cu cat un numar e mai mare (sau mai mic) intr-o matrice de corelatii, cu atat are coloana influenta mai mare asupra liniei(sau invers).

Astfel putem vedea ca unele linii sunt destul de neutre(cum ar fi Avg_calorie_intake), ele neavand un impact prea mare asupra altor atributee.

Apoi am luat atributele categorice si le-am transformat in atribute numerice prin label encoding, pentru a ajuta algoritmii sa functioneze corect.

Am pastrat codificarea in urmatorul dictionar:

```
Codificările pentru coloana 'Transportation':
{'Automobile': 0, 'Bike': 1, 'Motorbike': 2, 'Public_Transportation': 3, 'Walking': 4}
Codificările pentru coloana 'Diagnostic_in_family_history':
{'no': 0, 'yes': 1}
Codificările pentru coloana 'High_calorie_diet':
{'no': 0, 'yes': 1}
Codificările pentru coloana 'Alcohol':
{'Always': 0, 'Frequently': 1, 'Sometimes': 2, 'no': 3}
Codificările pentru coloana 'Snacks':
{'Always': 0, 'Frequently': 1, 'Sometimes': 2, 'no': 3}
Codificările pentru coloana 'Smoker':
{'no': 0, 'yes': 1}
```

```
Codificările pentru coloana 'Calorie_monitoring':
{'no': 0, 'yes': 1}
Codificările pentru coloana 'Gender':
{'Female': 0, 'Male': 1}
Codificările pentru coloana 'Diagnostic':
{'D0': 0, 'D1': 1, 'D2': 2, 'D3': 3, 'D4': 4, 'D5': 5, 'D6': 6}
```

Am refacut apoi statisticile pentru toate atributele(deoarece toate sunt numerice acum):

```
Statistici pentru atribute dupa labelEncoding :
       Regular_fiber_diet  Sedentary_hours_daily          Age  Est_avg_calorie_intake
count         1912.000000            1912.000000  1912.000000             1912.000000
mean             2.420848               3.196308    24.362449             2254.305962
std              0.533787               0.575331     6.414437              433.767146
min              1.000000               2.210000    15.000000             1500.000000
25%              2.000000               2.770000    19.977773             1872.000000
50%              2.392422               3.130000    22.830929             2253.500000
75%              3.000000               3.632500    26.000000             2628.000000
max              3.000000               4.670000    61.000000             3000.000000
```

	Main_meals_daily	Height	Water_daily	Weight
count	1912.000000	1912.000000	1912.000000	1912.000000
mean	2.682859	1.702176	2.011031	86.433823
std	0.779635	0.093208	0.611325	24.931396
min	1.000000	1.450000	1.000000	39.000000
25%	2.658599	1.630000	1.605576	68.552572
50%	3.000000	1.700000	2.000000	83.337721
75%	3.000000	1.770000	2.482454	105.016878
max	4.000000	1.980000	3.000000	165.057269

Physical_activity_level	Technology_time_use	Transportation	Diagnostic_in_family_history	High_calorie_diet
1912.000000	1912.000000	1912.000000	1912.000000	1912.000000
1.012089	0.666318	2.350941	0.817992	0.882845
0.855361	0.675159	1.272319	0.385952	0.321689
0.000000	0.000000	0.000000	0.000000	0.000000
0.115201	0.000000	3.000000	1.000000	1.000000
1.000000	1.000000	3.000000	1.000000	1.000000
1.683057	1.000000	3.000000	1.000000	1.000000
3.000000	2.000000	4.000000	1.000000	1.000000

Alcohol	Snacks	Smoker	Calorie_monitoring	Gender	Diagnostic
1912.000000	1912.000000	1912.000000	1912.000000	1912.000000	1912.000000
2.268828	1.856695	0.020397	0.043410	0.508368	3.117155
0.518533	0.466981	0.141393	0.203832	0.500061	1.986843
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2.000000	2.000000	0.000000	0.000000	0.000000	1.000000
2.000000	2.000000	0.000000	0.000000	1.000000	3.000000
3.000000	2.000000	0.000000	0.000000	1.000000	5.000000
3.000000	3.000000	1.000000	1.000000	1.000000	6.000000

Pentru a obtine setul de date final, cu care vom antrena algoritmii, am facut ultimul pas: Feature Selection. Am folosit VarianceThreshold pentru a elimina atributele care au cea mai mica variatie intre coeficienti. Am ales un threshold destul de ridicat pentru ca in urma a mai multor rulari am obtinut cam aceleasi rezultate si cu mai multe atribute si cu numarul ales de mine. Atributele ale caror coeficienti nu variaza sunt mai putin folositoare unui algoritm de prezicere decat cele variate. Ele mai mult ingreuneaza si trag in jos performanta algoritmului dpdv temporal. Asadar renuntarea la ele este o optimizare.

```
Dimensiunile datelor înainte de selecția caracteristicilor: (1912, 19)
Dimensiunile datelor după selecția caracteristicilor: (1912, 7)
Statistici pentru atributele selectate in urma VarianceThreshold:
                Age  Est_avg_calorie_intake  Main_meals_daily       Weight  Physical_activity_level  Transportation   Diagnostic
count  1912.000000             1912.000000       1912.000000  1912.000000              1912.000000     1912.000000  1912.000000
mean     24.362449             2254.305962          2.682859    86.433823                 1.012089        2.350941     3.117155
std       6.414437              433.767146          0.779635    24.931396                 0.855361        1.272319     1.986843
min      15.000000             1500.000000          1.000000    39.000000                 0.000000        0.000000     0.000000
25%      19.977773             1872.000000          2.658599    68.552572                 0.115201        3.000000     1.000000
50%      22.830929             2253.500000          3.000000    83.337721                 1.000000        3.000000     3.000000
75%      26.000000             2628.000000          3.000000   105.016878                 1.683057        3.000000     5.000000
max      61.000000             3000.000000          4.000000   165.057269                 3.000000        4.000000     6.000000
```

Am separat apoi setul de date in X si Y(Y fiind targetul = diagnosticul), apoi am separat in X_train si X_test restpectiv y_train si y_test, acestea reprezentand procente din setul de date complet folosite ba pentru antrenare ba pentru testare.

Am ales pentru fiecare hiper-parametru specificat un numar de optiuni, apoi am rulat fiecare algoritm cronometrandu-l. Am afisat timul de rulare, cel mai bun scor si hiper-parametrii optimi pentru fiecare algoritm.

Cei mai time-consuming au fost clar SVM si GradientBoostedTrees, cu o medie de 300-450 de secunde per algoritm. Este adevarat ca acesta este timpul total necesar pentru rularea fiecarei combinatie de parametrii, dar tot au durat mult mai mult decat ceilalti algoritmi.

Cel mai bun scor l-a avut RandomForest, depasindu-i cu putin pe ExtraTrees si GradientBoostedTrees. Cel din urma a fost SVM, lucru la care ma asteptam.

```
Timp pentru SVM: 409.47858691215515
Cel mai bun scor pentru SVM: 0.6239301403621558
Hiper-parametrii optimi pentru SVM: {'C': 0.1, 'kernel': 'linear'}
Timp pentru RandomForest: 17.61211323738098
Cel mai bun scor pentru RandomForest: 0.8136269152469732
Hiper-parametrii optimi pentru RandomForest: {'max_depth': None, 'max_features': 'sqrt', 'n_estimators': 200}
Timp pentru ExtraTrees: 11.115164995193481
Cel mai bun scor pentru ExtraTrees: 0.804483017250616
Hiper-parametrii optimi pentru ExtraTrees: {'max_depth': None, 'max_features': 'log2', 'n_estimators': 100}
Timp pentru GradientBoostedTrees: 319.3848693370819
Cel mai bun scor pentru GradientBoostedTrees: 0.8011786135219114
Hiper-parametrii optimi pentru GradientBoostedTrees: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
```

Mai jos este un tabel care prezinta media și varianța pentru acuratețea generală de clasificare, precizie / recall / F1 la nivelul fiecărei clase în parte pentru fiecare algoritm cu fiecare set de parametrii.

SVM

Configurație hiper-parametrii: {'C': 0.1, 'kernel': 'linear'}

Clasă 1

Precision: 0.7937, Recall: 0.7937, F1: 0.7937

Clasă 2

Precision: 0.6286, Recall: 0.4151, F1: 0.5000

Clasă 3

Precision: 0.3235, Recall: 0.5500, F1: 0.4074

Clasă 4

Precision: 0.3111, Recall: 0.2593, F1: 0.2828

Clasă 5

Precision: 0.5606, Recall: 0.6167, F1: 0.5873

Clasă 6

Precision: 0.9348, Recall: 0.7414, F1: 0.8269

Clasă 7

Precision: 0.8000, **Recall: 0.8727**, **F1: 0.8348**

Metrică	**Medie**	**Varianță**
Acuratețe	**0.6239**	0.0217
Acuratețe	0.1681	0.0017
Acuratețe	0.1681	0.0017
Acuratețe	0.6174	0.0180
Acuratețe	0.1674	0.0013
Acuratețe	0.1367	0.0113
Acuratețe	0.5958	0.0197
Acuratețe	0.2544	0.0248
Acuratețe	0.1262	0.0096
Acuratețe	0.6037	0.0243
Acuratețe	0.4597	**0.0318**
Acuratețe	0.1256	0.0105

RandomForest

Configurație hiper-parametrii: {'max_depth': None, 'max_features': 'sqrt', 'n_estimators': 200}

Clasă 1

Precision: 0.9800, Recall: 0.7778, F1: 0.8673

Clasă 2

Precision: 0.7455, Recall: 0.7736, F1: 0.7593

Clasă 3

Precision: 0.6957, Recall: 0.8000, F1: 0.7442

Clasă 4

Precision: 0.6786, Recall: 0.7037, F1: 0.6909

Clasă 5

Precision: 0.7966, Recall: 0.7833, F1: 0.7899

Clasă 6

Precision: 0.9167, Recall: 0.9483, F1: 0.9322

Clasă 7

Precision: 0.9474, **Recall: 0.9818**, **F1: 0.9643**

Metrică	**Medie**	**Varianță**
Acuratețe	0.7986	0.0345
Acuratețe	0.8090	0.0202
Acuratețe	**0.8136**	0.0224
Acuratețe	0.8032	0.0252
Acuratețe	0.8110	0.0254
Acuratețe	0.8058	0.0293
Acuratețe	0.7907	0.0175
Acuratețe	0.7874	0.0291
Acuratețe	0.7907	0.0231
Acuratețe	0.7861	0.0234
Acuratețe	0.7861	0.0300
Acuratețe	0.8005	0.0206
Acuratețe	0.8051	0.0355

Acuratețe	0.8117	0.0283
Acuratețe	0.8077	0.0274
Acuratețe	0.8077	0.0171
Acuratețe	0.8019	**0.0370**
Acuratețe	0.8110	0.0234

ExtraTrees

Configurație hiper-parametrii: {'max_depth': None, 'max_features': 'log2', 'n_estimators': 100}

Clasă 1

Precision: 0.9074, Recall: 0.7778, F1: 0.8376

Clasă 2

Precision: 0.7222, Recall: 0.7358, F1: 0.7290

Clasă 3

Precision: 0.6250, Recall: 0.7500, F1: 0.6818

Clasă 4

Precision: 0.6727, Recall: 0.6852, F1: 0.6789

Clasă 5

Precision: 0.8070, Recall: 0.7667, F1: 0.7863

Clasă 6

Precision: 0.9138, Recall: 0.9138, F1: 0.9138

Clasă 7

Precision: 0.9123, **Recall: 0.9455**, **F1: 0.9286**

Metrică	**Medie**	**Varianță**
Acuratețe	0.7888	0.0279
Acuratețe	0.8019	0.0335
Acuratețe	0.8032	0.0267
Acuratețe	0.7986	0.0351
Acuratețe	**0.8045**	0.0344
Acuratețe	0.7973	0.0323
Acuratețe	0.7829	0.0214
Acuratețe	0.7606	0.0383
Acuratețe	0.7731	0.0296
Acuratețe	0.7659	**0.0421**
Acuratețe	0.7678	0.0248
Acuratețe	0.7718	0.0279
Acuratețe	0.7966	0.0321

Acuratețe	0.7992	0.0322
Acuratețe	0.8005	0.0279
Acuratețe	0.7979	0.0303
Acuratețe	0.8005	0.0182
Acuratețe	0.7979	0.0249

GradientBoostedTrees

Configurație hiper-parametrii: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}

Clasă 1

Precision: 0.9200, Recall: 0.7302, F1: 0.8142

Clasă 2

Precision: 0.6780, Recall: 0.7547, F1: 0.7143

Clasă 3

Precision: 0.6977, Recall: 0.7500, F1: 0.7229

Clasă 4

Precision: 0.8000, Recall: 0.7407, F1: 0.7692

Clasă 5

Precision: 0.7778, Recall: 0.8167, F1: 0.7967

Clasă 6

Precision: 0.9167, Recall: 0.9483, F1: 0.9322

Clasă 7

Precision: 0.9310, **Recall: 0.9818**, **F1: 0.9558**

Metrică	**Medie**	**Varianță**
Acuratețe	0.6213	**0.0302**
Acuratețe	0.6678	0.0216
Acuratețe	0.7037	0.0201
Acuratețe	0.7011	0.0235
Acuratețe	0.7266	0.0345
Acuratețe	0.7620	0.0290
Acuratețe	0.7390	0.0172
Acuratețe	0.7580	0.0090
Acuratețe	0.7724	0.0139
Acuratețe	0.7515	0.0140
Acuratețe	0.7750	0.0183
Acuratețe	0.7835	0.0225
Acuratețe	0.7770	0.0188

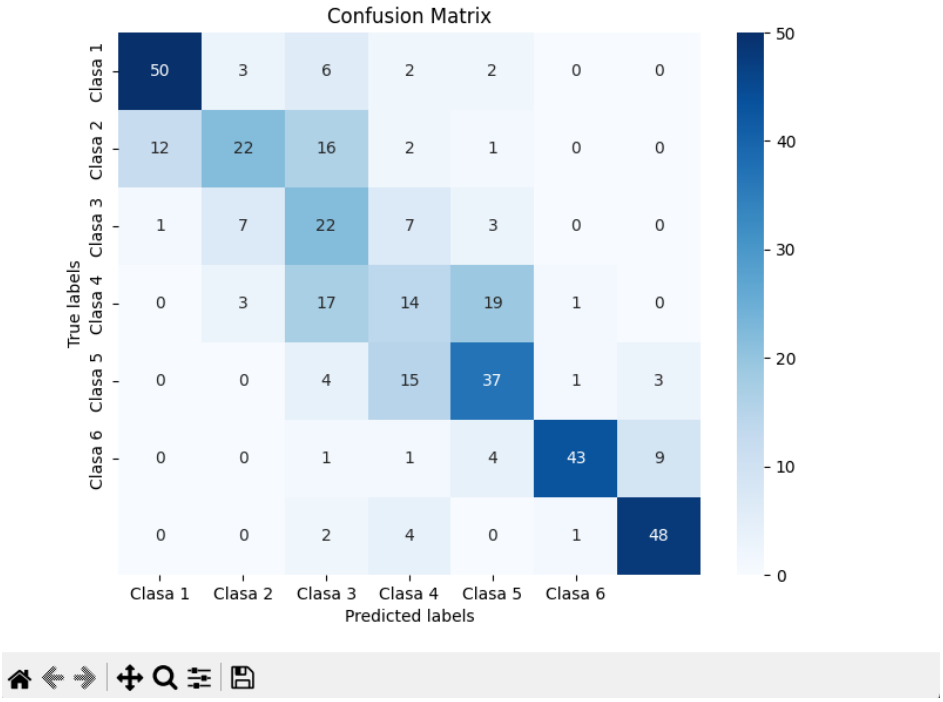
Acuratețe	0.7711	0.0167
Acuratețe	0.7855	0.0123
Acuratețe	0.7848	0.0179
Acuratețe	0.7914	0.0120
Acuratețe	0.7992	0.0172
Acuratețe	0.7796	0.0170
Acuratețe	0.7809	0.0251
Acuratețe	0.7861	0.0134
Acuratețe	0.7901	0.0134
Acuratețe	0.7940	0.0142
Acuratețe	0.7907	0.0086
Acuratețe	0.7933	0.0140
Acuratețe	**0.8012**	0.0075
Acuratețe	0.8005	0.0117

Recall/F1/Precision sunt metrici care arata cat de bune sunt rezultatele algoritmului. Cu cat ele sunt mai mari, cu atat algoritmul are un procent de rezultate mai bun. Putem oberva ca in mare parte din cazuri, Clasa 7 (D6) este cea mai predictibila, iar clasa care da cele mai mari batai de cap algoritmilor variaza. Cu aceste date putem vizualiza problemele algoritmului, ca apoi sa incercam sa le rezolvam si sa il imbunatatim.

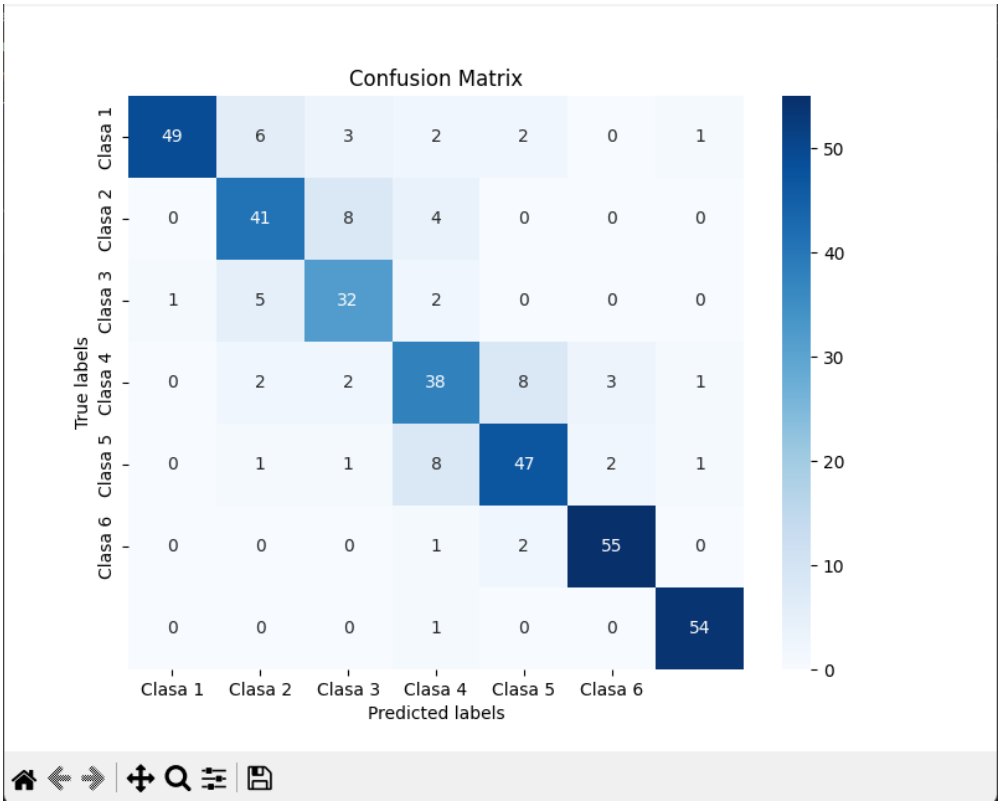
Impactul hiper-parametrilor asupra performantei algoritmilor este unul foarte mare, lucru usor sesizabil din tabelul de mai sus, in care putem vedea cu bold media cea mai mare (care este intalnita la combinatia optima de hiper-parametrii) si varianta cea mai mare (care e intalnita la cea mai instabila configuratie de hiper-parametrii, intrucat varianta mare = fluctuatie mare a performantei algoritmului in functie de atribute).

In cele din urma am realizat cate o matrice de confuzie pentru cea mai buna varianta a hiper-parametrilor pentru fiecare algoritm.

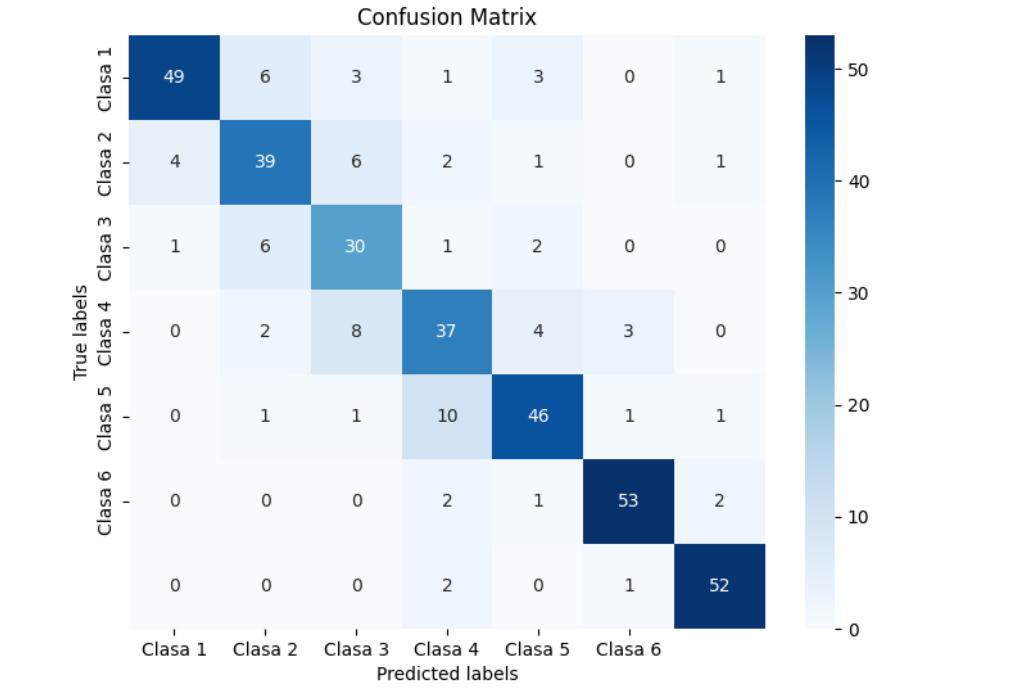
SVM



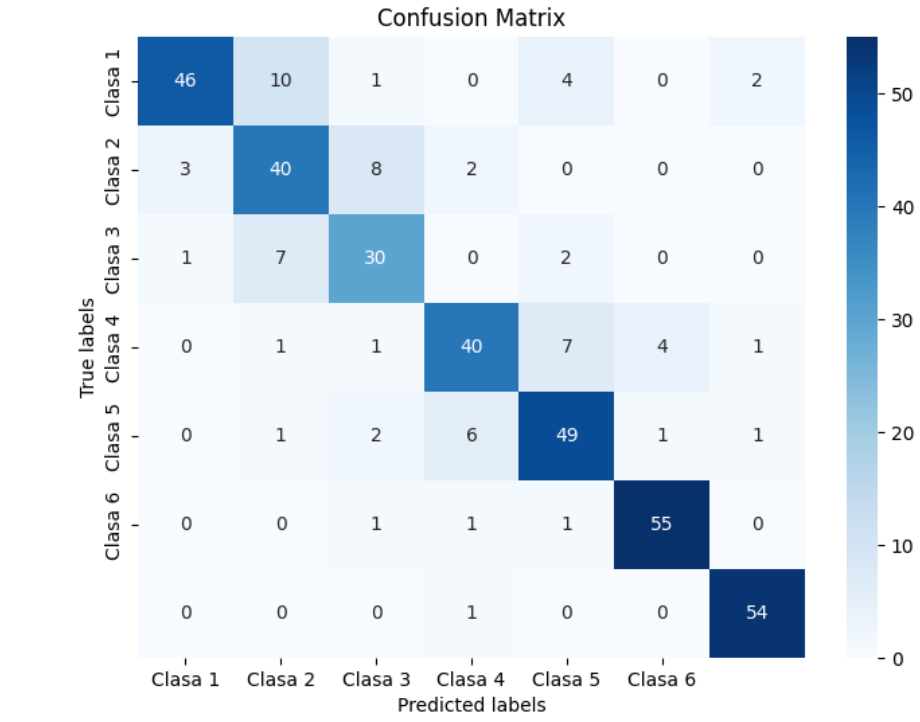
RandomForest



ExtraTrees



GradientBoostedTrees



Intr-o matrice de confuzie, putem vedea cat de des e incurcata o clasa cu alta. Pe axa True labels e clasa corecta(diagnosticul corect), iar pe axa Predicted labels e clasa prezisa de algoritm. Astfel putem vedea unde se produc greseli, si putem incerca sa rezolvam aceste confuzii intre clase direct.

Putem observa clar cate confuzii face SVM comparativ cu ceilalti algoritmi, facand foarte multe erori in primele 5 clase. Cea mai des incurcata clasa variaza de la algoritm la algoritm, principalele fiind 2, 3 si 4.