1. Analiza echilibrului de clase

* Frecventa pentru setul de antrenare:

A graph with numbers and a bar

Description automatically generated

* Frecventa pentru setul de test:

A graph with numbers and a bar

Description automatically generated

1. Vizualizarea datelor:

A graph with different colored bars

Description automatically generated

Nu prea ne ajuta cu ninmic asa ca luam fiecare statistica in parte:

A graph with numbers and a bar

Description automatically generatedA graph with numbers and a bar

Description automatically generatedA graph with numbers and a bar

Description automatically generatedA graph with a bar and text

Description automatically generatedA graph with numbers and a bar

Description automatically generated

A graph with numbers and a bar

Description automatically generatedA graph of numbers and a bar

Description automatically generated with medium confidenceA graph with numbers and text

Description automatically generatedA graph with blue bars

Description automatically generated with medium confidence

Interpretare:  
Fig.1(Medie) – Se observa multe anomalii, ceea ce indica catre valori eronate. Aici par a fi date normale doar la atributele “Sedentary\_hours\_daily”, “Est\_avg\_calorie\_intake” si “Technology\_time\_use”.

Fig. 4, 5(Valori Minime si Maxime) – Pe Minim singura observatie este atributul “Weight” = -1. Pe Maxim ca si “Sedentary\_hours\_daily”, si “Technology\_time\_use” au valori care nu corespund comportamentului uman deci au valori eronate.

In concluzie din atributele numerice doar “Est\_avg\_calorie\_intake” are valori valide deci el ramane neschimbat.

Imputatie:

Nu avem probleme decal la EST\_avg\_calorie\_intake deci la celelalte aplicam petoda de Imputatie, cu strategia “mean” dar mai intai trebuie sa inlocuim peste tot unde avem valori eronate cu -1 ca flag. Pentru asta am creat un set de functii care fac clamp intr-un interval anumit bazat pe statistici:

A screen shot of a computer program

Description automatically generated

Dupa ce aplicam “SimpleImputer” obtinem urmatoarele statistici:  
A graph with numbers and a bar

Description automatically generatedA graph with text and numbers

Description automatically generatedA graph with a bar

Description automatically generatedA graph with numbers and text

Description automatically generatedA graph with text and numbers

Description automatically generated

A graph with text and numbers

Description automatically generatedA graph with numbers and a bar

Description automatically generatedA graph with text on it

Description automatically generatedA graph with a bar

Description automatically generated

Interpretare:

A graph with numbers and a bar

Description automatically generated

Obtinem indivizi cu parametri “average” in directia favorabila deoarece avem indivizi tineri (23 ani) cu consum de calorii situat in norma recomandata (2000 - 2500), cel din urma corespunde cu un parametru “accurate” in cea ce priveste greutatea (69 kg). Atributul “Regular\_fiber\_diet” pare usore nefavorabil deoarece 25-34g reprezinta norma sanatoasa insa desi am facut clamp [0, 1000] avem 2.48 valoare medie deci avem indivizi care consuma prea putine fibre, lucru care este acceptabil. “Physical\_activity\_level” normal este intre 1.4-2.4 deci avem populatie usor sedentara.

A graph with text and numbers

Description automatically generated

O abatere standard mică indică faptul că valorile tind să fie aproape de media (sau valoarea medie) a setului, în timp ce o abatere standard mare indică faptul că valorile sunt răspândite pe o gamă mai largă.

Deci avem o gama larga a atributelor pentru “Est\_avg\_calorie\_intake”, “Weight” si poate "Age"

A graph with a bar

Description automatically generated

Observam diferente minore fata de abaterea standard ceea ce indica ca nu avem valori extreme mult mai mari decat medie cee ace indica o distributie echilibrata.

Abaterea Standard – diferentele valoare-medie se ridicau la patrat => obtineam o valoare mult mai mare cee ace ar fi influentat mult abaterea.

Abaterea Medie – nu ridica la patrat diferentele deci obtinem o evaluare mai precisa in ceea ce priveste distributia valorilor.

A graph with numbers and text

Description automatically generatedA graph with text and numbers

Description automatically generated

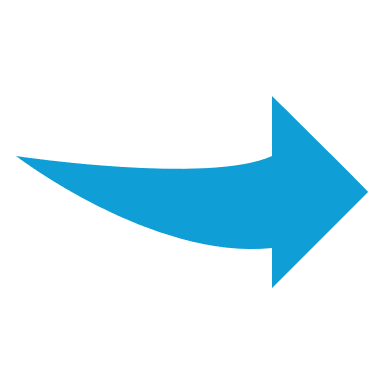
Valorile minime nu sufera modificari decat in cazurile valorilor lipsa(-1) ceea ce este ok. Observam imbunatatiri la valorile maxime, de observant ca am ales intervale de clamp bune deoarece nu avem limite in acele valori => cele care depaseau limitele logice, le depaseau cu mult deci erau clar valori eronate iar cele care se afla in interval nu se afla la pe limita (weight clamp[0 - 200] iar weight max = 130 deci nu am eliminat potentiali 201, 202 care ar putea fi indivizi valizi insa care ar afecta o analiza “average” deoarece ei nu sunt “average”).

Las mai jos comparatia dintre valorile initiale si cele actuale:

A graph with a bar and text

Description automatically generatedA graph with numbers and a bar

Description automatically generated



A graph with numbers and text

Description automatically generatedA graph with text and numbers

Description automatically generated

A graph with text and numbers

Description automatically generated

O valoare mica ar indica faptul ca datele noastre contin multi indivizi similari deci vom obtine rezultate care vor fi valide pe gama restransa de indivizi. Desi suntem tentati sa credem asta valorile nu sunt exact mici deoarece 0.53cm este o diferenta de inaltime decenta la fel si 46 in cazul varstei. La fel si in cazul altor attribute cum ar fi “Physical\_activity\_level” = 3 (luand in considerare ca norma e 1.4 – 2.4), 3 indica o gama larga a valorilor. O problema poate fi observata la “Regular\_fiber\_diet” pe care l-am observat inca de la medie.

A graph with numbers and a bar

Description automatically generatedA graph with numbers and a bar

Description automatically generated

Deoarece Mediana apropiata de Medie => avem un set de date echilibrat. (set = [1, 3, 3, **6**, 7, 8, 9] Mediana = 6) Inseamna ca avem valori peste medie si sub medie in numar echilibrat.

A graph with text on it

Description automatically generated

Valorile mari ale MAD indică că datele sunt foarte dispersate în jurul medianei, adică există multe valori care sunt mult mai mari sau mult mai mici decât mediana. Pe de altă parte, valorile mici ale MAD indică că majoritatea datelor sunt aproape de mediana.

Deci observam valori foarte mici pentru multe attribute de pe grafic, in evidenta sarind cele = 0. Acest lucru se datoreaza faptului ca avem multe valori appropriate de mediana, ba chiar = cu mediana deoarece am avut multe valori eronate in acel atribut si le-am inlocuit cu media prin procedeul de Imputatie. De aici tragem concluzia ca setul nostrum de date devine usor “inaccurate” deoarece dam multor attribute, valori egale, ceea ce nu corespunde cu un studiu “accurate”. Dar lucram cu ce avem.

A graph with a bar

Description automatically generated

IQR = Q3 – Q1

* Q1  valoarea sub care se află 25% din date.
* Q3  valoarea sub care se află 75% din date.

Este mai puțin sensibil la valori aberante și, prin urmare, poate fi mai util.

Observam si aici multe valori = 0, lucru care a fost observant si mai sus deci atributele = 0 sunt destul de “inacurate”.

Nominale:

A graph with blue rectangles

Description automatically generatedA graph of a diet

Description automatically generated with medium confidenceA graph with a bar

Description automatically generatedA graph with blue rectangles

Description automatically generatedA graph of blue rectangular bars

Description automatically generated with medium confidenceA graph with blue rectangles

Description automatically generatedA graph with blue rectangles

Description automatically generatedA graph with blue rectangles

Description automatically generatedA graph with numbers and a bar

Description automatically generated

Covarianta:  
A screenshot of a computer

Description automatically generated

Covarianța pozitivă: variabile tind sa creasca sau sa scada impreuna

Covarianta negative: una dintre variabile tinde sa creasca atunci cand cealalta scade.

|  |  |
| --- | --- |
| **Column1** | **Sedentary\_hours\_daily** |
| **Regular\_fiber\_diet** | -0.014642591 |

“Regular\_fiber\_diet” tinde sa creasca atunci cand “Sedentary\_hours\_daily” scade, si invers.

3.2 - Analia VarianceTreshold

|  |  |
| --- | --- |
| SVM: | |
| Initial.shape:  (1921, 18) | |
| Threshold:  0.1 | Threshold:  0.2 |
| Reduced.shape:  (1921, 15)  ['Height', 'Smoker', 'Calorie\_monitoring'] | Reduced.shape:  (1921, 11)  ['Regular\_fiber\_diet','Diagnostic\_in\_family\_history',  'High\_calorie\_diet', 'Height', 'Smoker','Water\_daily', 'Calorie\_monitoring'] |
| Hiper-Parametri:  {'C': 10, 'kernel': 'rbf'} | Hiper-Parametri:  {'C': 10, 'kernel': 'rbf'} |
| Classification Report:  precision recall f1-score support  D0 0.81 0.73 0.77 60  D1 0.63 0.71 0.67 52  D2 0.51 0.50 0.51 42  D3 0.62 0.57 0.60 49  D4 0.81 0.62 0.70 69  D5 0.65 0.85 0.73 60  D6 0.96 0.98 0.97 53  accuracy 0.72 385  macro avg 0.71 0.71 0.71 385  weighted avg 0.73 0.72 0.72 385 | Classification Report:  precision recall f1-score support  D0 0.80 0.62 0.70 60  D1 0.55 0.60 0.57 52  D2 0.56 0.52 0.54 42  D3 0.46 0.45 0.45 49  D4 0.72 0.45 0.55 69  D5 0.51 0.80 0.62 60  D6 0.88 0.98 0.93 53  accuracy 0.63 385  macro avg 0.64 0.63 0.62 385  weighted avg 0.65 0.63 0.63 385 |

|  |  |
| --- | --- |
| RandomForest: | |
| Initial.shape:  (1921, 18) | |
| Threshold:  0.1 | Threshold:  0.2 |
| Reduced.shape:  (1921, 15) | Reduced.shape:  (1921, 11) |
| Hiper-Parametri:  {'n\_estimators': 100, 'max\_depth': 20, 'max\_features': 'log2'} | Hiper-Parametri:  {'n\_estimators': 50, 'max\_depth': 10, 'max\_features': 'sqrt'} |
| Classification Report:  precision recall f1-score support  D0 0.94 0.75 0.83 60  D1 0.68 0.94 0.79 52  D2 0.59 0.52 0.56 42  D3 0.68 0.61 0.65 49  D4 0.77 0.67 0.71 69  D5 0.73 0.85 0.78 60  D6 0.96 0.98 0.97 53  accuracy 0.77 385  macro avg 0.76 0.76 0.76 385  weighted avg 0.77 0.77 0.76 385 | Classification Report:  precision recall f1-score support  D0 0.86 0.63 0.73 60  D1 0.67 0.81 0.73 52  D2 0.55 0.43 0.48 42  D3 0.50 0.49 0.49 49  D4 0.68 0.49 0.57 69  D5 0.59 0.87 0.70 60  D6 0.88 0.98 0.93 53  accuracy 0.68 385  macro avg 0.68 0.67 0.66 385  weighted avg 0.68 0.68 0.67 385 |

|  |  |
| --- | --- |
| ExtraTrees: | |
| Initial.shape:  (1921, 18) | |
| Threshold:  0.1 | Threshold:  0.2 |
| Reduced.shape:  (1921, 15)  ['Height', 'Smoker', 'Calorie\_monitoring'] | Reduced.shape:  (1921, 11)  ['Regular\_fiber\_diet','Diagnostic\_in\_family\_history',  'High\_calorie\_diet', 'Height', 'Smoker','Water\_daily', 'Calorie\_monitoring'] |
| Hiper-Parametri:  {'n\_estimators': 100, 'max\_depth': 20, 'max\_features': 'sqrt' } | Hiper-Parametri:  {,'n\_estimators': 50', max\_depth': 10, 'max\_features': 'None'} |
| Classification Report:  precision recall f1-score support  D0 0.88 0.77 0.82 60  D1 0.68 0.87 0.76 52  D2 0.60 0.50 0.55 42  D3 0.69 0.71 0.70 49  D4 0.84 0.67 0.74 69  D5 0.69 0.83 0.76 60  D6 0.96 0.98 0.97 53  accuracy 0.77 385  macro avg 0.76 0.76 0.76 385  weighted avg 0.77 0.77 0.76 385 | Classification Report:  precision recall f1-score support  D0 0.91 0.65 0.76 60  D1 0.71 0.75 0.73 52  D2 0.52 0.52 0.52 42  D3 0.50 0.49 0.49 49  D4 0.70 0.51 0.59 69  D5 0.60 0.87 0.71 60  D6 0.87 0.98 0.92 53  accuracy 0.68 385  macro avg 0.69 0.68 0.67 385  weighted avg 0.70 0.68 0.68 385 |

|  |  |
| --- | --- |
| GradientBoostedTrees: | |
| Initial.shape:  (1921, 18) | |
| Threshold:  0.1 | Threshold:  0.2 |
| Reduced.shape:  (1921, 15) | Reduced.shape:  (1921, 11) |
| Hiper-Parametri:  {'n\_estimators': 100, 'max\_depth': 20, 'learning\_rate': 0.1} | Hiper-Parametri:  {'n\_estimators': 50, 'max\_depth': 10, 'learning\_rate': 0.1} |
| Classification Report:  precision recall f1-score support  D0 0.88 0.75 0.81 60  D1 0.72 0.88 0.79 52  D2 0.66 0.50 0.57 42  D3 0.62 0.71 0.67 49  D4 0.73 0.62 0.67 69  D5 0.74 0.85 0.79 60  D6 0.96 0.98 0.97 53  accuracy 0.76 385  macro avg 0.76 0.76 0.75 385  weighted avg 0.76 0.76 0.76 385 | Classification Report:  precision recall f1-score support  D0 0.87 0.65 0.74 60  D1 0.72 0.85 0.78 52  D2 0.50 0.33 0.40 42  D3 0.53 0.49 0.51 49  D4 0.55 0.52 0.53 69  D5 0.60 0.80 0.69 60  D6 0.87 0.98 0.92 53  accuracy 0.67 385  macro avg 0.66 0.66 0.65 385  weighted avg 0.67 0.67 0.66 385 |

Concluzie:  
In acest caz procedura de VarianceThreshold nu a fost prea utila deoarece obtinem acc mai mica decat in cazul in care nu am aplicat procedura. Las mai jos rezultatele pentru varianta obisnuita:

Cele mai bune rezultate obtinute sunt pe varianta fara Variance Threshold, deci aici vom face si interpretarea performantelor obtinute:

SVM:

Pentru GridSearch:

Hiper-parametri: {'C': 10, 'kernel': 'rbf'}

Classification Report:

precision recall f1-score support

D0 0.85 0.77 0.81 60

D1 0.60 0.71 0.65 52

D2 0.61 0.60 0.60 42

D3 0.64 0.61 0.62 49

D4 0.84 0.62 0.72 **69**

D5 0.68 0.87 0.76 60

D6 **0.98** **0.98** **0.98** 53

accuracy 0.74 385

macro avg 0.74 0.74 0.73 385

weighted avg 0.75 0.74 0.74 385

acc: 0.74 > 0.72

Manual:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | 0.1 | 1 | 10 | 100 |
| Linear acc | 0.62 | 0.63 | 0.63 | 0.63 |
| Poly acc | 0.36 | 0.63 | 0.69 | 0.70 |
| Rbf acc | 0.58 | 0.70 | 0.74 | 0.76 |
| Sigmoid acc | 0.52 | 0.52 | 0.46 | 0.41 |

Best Hiper-Parametri: {'C': 100, 'kernel': 'rbf'}

Classification Report:

precision recall f1-score support

D0 0.89 0.70 0.79 60

D1 0.60 0.83 0.69 52

D2 0.61 0.60 0.60 42

D3 0.67 0.71 0.69 49

D4 0.81 0.64 0.72 **69**

D5 0.77 0.85 0.81 60

D6 **0.98** **0.98** **0.98** 53

accuracy 0.76 385

macro avg 0.76 0.76 0.75 385

weighted avg 0.77 0.76 0.76 385

Confusion Matrix:

A chart of different colored squares

Description automatically generated

Hiper-parametrii pot influenta negative acuratetea predictiilor ca in cazul: {'C': 0.1, 'kernel': 'poly'} acc = 0.36, trebuie de considerat si complexitatea temporala care creste odata cu C, deci in cazul unui set gigantic am considera alegerea unei configuratiei: {'C': 1, 'kernel': 'rbf'} unde acc 0.70 (best 0.76). Deci intelegerea comportamentului setului nostrum de date ne poate ajuta sa evitam complexitatea temporala generata de GridSearch la fel si analizarea trade-off-urilor poate imbunatati performanta.

|  |  |
| --- | --- |
| Clasa | Sum |
| D0 | 2.38 |
| D1 | 2.12 |
| D2 | 1.81 |
| D3 | 2.07 |
| D4 | 2.17 |
| D5 | 2.43 |
| D0 | 2.94 |

Cele mai bune predictii de clasa sunt la D6, precizie = 0.98, un recall = 0.98 si scor F1 = 0.98. Apoi suntem tentati sa spunem clasa D0 deoarece are precizie = 0.89 insa avem recall mic = 0.70 si scor F1 = 0.79 (idea e sa maximizam aceste valori) iar D0 = 2.38 in timp ce D5 = 2.43. Ne asiguram de acest lucru inspectand si matricea de confuzie si observam multe exemple clasificate gresit in cazul D0.

D2 are cea mai rea preddictie.

RandomForest:

Pentru GridSearch:

Hiper-parametri: {'max\_depth': None, 'max\_features': 'sqrt', 'n\_estimators': 100}

Classification Report:

precision recall f1-score support

D0 **0.96** 0.77 0.85 60

D1 0.67 0.94 0.78 52

D2 0.75 0.64 0.69 42

D3 0.75 0.73 0.74 49

D4 0.89 0.72 0.80 **69**

D5 0.80 0.93 0.86 60

D6 **0.96** **0.98** **0.97** 53

accuracy 0.82 385

macro avg 0.83 0.82 0.81 385

weighted avg 0.83 0.82 0.82 385

acc: 0.82 > 0.77

Manual:

A screenshot of a table

Description automatically generated

Best Hiper-Param: {'n\_estimators': 100, 'max\_depth': 20, 'max\_features': 'sqrt'}

Classification Report:

precision recall f1-score support

D0 **0.98** 0.77 0.86 60

D1 0.67 0.94 0.78 52

D2 0.82 0.67 0.74 42

D3 0.78 0.73 0.76 49

D4 0.85 0.74 0.79 **69**

D5 0.79 0.93 0.85 60

D6 0.96 **0.98** **0.97** 53

accuracy 0.83 385

macro avg 0.84 0.82 0.82 385

weighted avg 0.84 0.83 0.83 385

Confusion Matrix:

A chart of a number and a number

Description automatically generated with medium confidence

Aici observam o distributie mai echilibrata a acuratetii decat in cazul SVM. Acc\_min aprox 0.76 iar acc\_max = 0.83. Analizand acest lucru putem allege o configuratie a hiperparametrilor cu un n\_estimators si max\_depth mai mici precum: {'max\_depth': None, 'max\_features': 'None', 'n\_estimators': 10} unde acc = 0.8

Obtinem un trade-off convenabil intre complexitate si performanta. Deci si in acest caz hiper-parametrii pot afecta considerabil performanta in dependenta de ce urmarim (pe un set mare, viteza iar pe un set mic acuratete), in cazul setului nostru unde avem putine date putem sa urmarim acuratetea rezultatului si sa evitam costul unui GridSearch.

|  |  |
| --- | --- |
| Clasa | Sum |
| D0 | 2.61 |
| D1 | 2.39 |
| D2 | 2.23 |
| D3 | 2.27 |
| D4 | 2.38 |
| D5 | 2.57 |
| D6 | 2.91 |

Cele mai bune predictii de clasa sunt D6 si D0 iar cele mai slabe sunt D2 si D3. Observam astfel o imbunatatire considerabila fata de SVM.

ExtraTrees:

Pentru GridSearch:

Hiper-parametri: {'max\_depth': 30, 'max\_features': 'log2', 'n\_estimators': 100}

Classification Report:

precision recall f1-score support

D0 0.96 0.77 0.85 60

D1 0.68 0.85 0.75 52

D2 0.68 0.67 0.67 42

D3 0.76 0.76 0.76 49

D4 0.89 0.78 0.83 **69**

D5 0.75 0.85 0.80 60

D6 **0.98** **0.98** **0.98** 53

accuracy 0.81 385

macro avg 0.81 0.81 0.81 385

weighted avg 0.82 0.81 0.81 385

acc: 0.81 > 0.77

Manual:

A screenshot of a computer

Description automatically generated

Best Hiper-Param: {'n\_estimators': 50, 'max\_depth': 20, 'max\_features': 'sqrt'}

Classification Report:

precision recall f1-score support

D0 **0.98** 0.78 0.87 60

D1 0.70 0.88 0.78 52

D2 0.67 0.71 0.69 42

D3 0.82 0.73 0.77 49

D4 0.88 0.74 0.80 **69**

D5 0.76 0.88 0.82 60

D6 0.96 **0.98** **0.97** 53

accuracy 0.82 385

macro avg 0.82 0.82 0.81 385

weighted avg 0.83 0.82 0.82 385

Confusion Matrix:

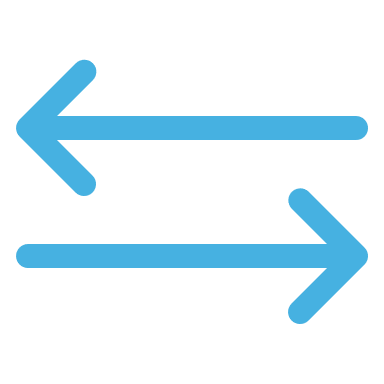
A chart of a number of colored squares

Description automatically generated with medium confidence

Se poate observa faptul ca n\_estimatori = 50 obtine cel mai convenabil trade-off deoarece pentru:  
{'n\_estimators': 50, 'max\_depth': 20, 'max\_features': 'sqrt'} acc = 0.8207

{'n\_estimators': 50, 'max\_depth': 20, 'max\_features': 'log2'} acc = 0.8103

{'n\_estimators': 50, 'max\_depth': 20, 'max\_features': 'None'} acc = 0.8103



{'n\_estimators': 100, 'max\_depth': 30, 'max\_features': 'sqrt'} acc = 0.8129

{'n\_estimators': 100, 'max\_depth': 30, 'max\_features': 'log2'} acc = 0.8103

{'n\_estimators': 100, 'max\_depth': 30, 'max\_features': 'None'} acc = 0.8207

Observam aproape aceleasi valori => mai rentabil sa folosim: {'n\_estimators': 50, 'max\_depth': 20} deoarece optinem performanta crescuta la un cost mai mic.

|  |  |
| --- | --- |
| Clasa | Sum |
| D0 | 2.63 |
| D1 | 2.36 |
| D2 | 2.07 |
| D3 | 2.32 |
| D4 | 2.34 |
| D5 | 2.42 |
| D6 | 2.91 |

Cele mai bune predictii de clasa sunt D6 si D0 iar cele mai slabe sunt D2 si D3. Observam comportament similar cu RandomForest deoarece avem diferenta acc\_max – acc\_min neglijabila.

GradientBoostedTrees:

Pentru GridSearch:

Hiper-parametri: {'n\_estimators': 50, 'max\_depth': 10, 'learning\_rate': 0.1}

Classification Report:

precision recall f1-score support

D0 0.94 0.75 0.83 60

D1 0.74 0.88 0.81 52

D2 0.67 0.57 0.62 42

D3 0.77 0.73 0.75 49

D4 0.82 0.74 0.78 **69**

D5 0.76 0.95 0.84 60

D6 **0.95** **0.98** **0.96** 53

accuracy 0.81 385

macro avg 0.81 0.80 0.80 385

weighted avg 0.81 0.81 0.81 385

acc: 0.81 > 0.76

Manual:

A screenshot of a spreadsheet

Description automatically generated

Best Hiper-Param: {'n\_estimators': 100, 'max\_depth': 10, 'learning\_rate': 0.1}

Classification Report:

precision recall f1-score support

D0 0.94 0.77 0.84 60

D1 0.75 0.87 0.80 52

D2 0.71 0.60 0.65 42

D3 0.76 0.76 0.76 49

D4 0.85 0.77 0.81 **69**

D5 0.75 0.95 0.84 60

D6 **0.96** **0.98** **0.97** 53

accuracy 0.82 385

macro avg 0.82 0.81 0.81 385

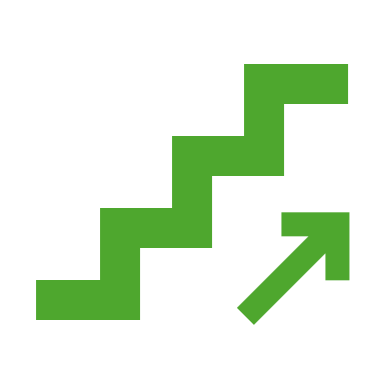
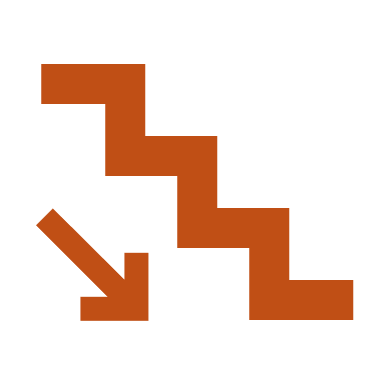
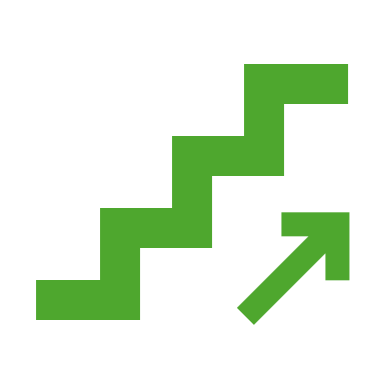
weighted avg 0.82 0.82 0.82 385

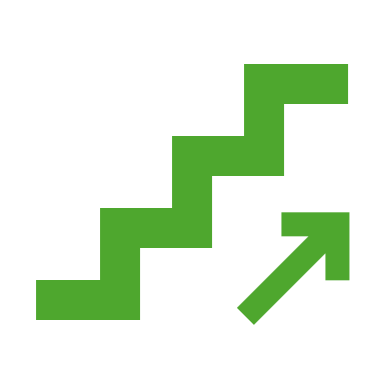
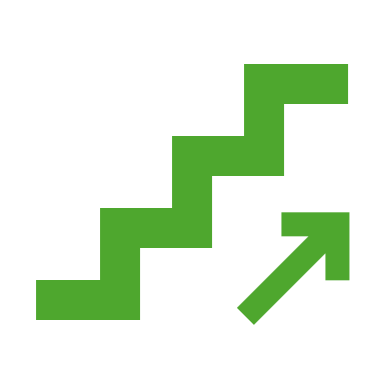
Confusion Matrix:

A chart of a number and a number

Description automatically generated with medium confidence

Observam ca acuratetea predictiei creste odata cu cresterea learning\_rate-ului, lucru surprinzator deoarece, un learning rate mai mare va face ca algoritmul sa convearga mai rapid deci obtinem performante mai bune. In schimb, nu observam liniaritate intre cresterea estimatorilor si a max\_dept-ului deoarece avem comportamente de genul:

{'n\_estimators': 100, 'max\_depth': 10, 'learning\_rate': 0.1} acc = 0.818  {'n\_estimators': 100, 'max\_depth': 20, 'learning\_rate': 0.1} acc = 0.807  {'n\_estimators': 100, 'max\_depth': 30, 'learning\_rate': 0.1} acc = 0.810 

Trade-off gasit e la mijloc de ex:  
{'n\_estimators': 50, 'max\_depth': 20, 'learning\_rate': 0.1} acc = 0.807 {'n\_estimators': 50, 'max\_depth': 30, 'learning\_rate': 0.1} acc = 0.812 

Aici putem concluziona faptul ca ar fi mai sigur folosirea GridSearch.

|  |  |
| --- | --- |
| Clasa | Sum |
| D0 | 2.55 |
| D1 | 2.42 |
| D2 | 1.96 |
| D3 | 2.28 |
| D4 | 2.43 |
| D5 | 2.54 |
| D6 | 2.91 |

Cele mai bune predictii de clasa sunt D6 si D0 iar cele mai slabe sunt D2 si D3. Aici diferenta acc\_max – acc\_min este mai mare decat la RandomForest si ExtraTrees deci, cum am mai spus putem prefera strategia GridSearch.

Dupa observatiile de la IQR:  
Decid sa elimin Age, Weight, Physical\_activity\_level

dataset\_numeric = dataset\_numeric.drop("Age", axis=1)

dataset\_numeric = dataset\_numeric.drop("Weight", axis=1)

dataset\_numeric = dataset\_numeric.drop("Physical\_activity\_level", axis=1)

SVM:

{'C': 10, 'kernel': 'rbf'}

Classification Report:

precision recall f1-score support

D0 0.84 0.72 0.77 60

D1 0.47 0.58 0.52 52

D2 0.54 0.52 0.53 42

D3 0.57 0.51 0.54 49

D4 0.77 0.62 0.69 69

D5 0.66 0.83 0.74 60

D6 0.98 0.98 0.98 53

accuracy 0.69 385

macro avg 0.69 0.68 0.68 385

weighted avg 0.70 0.69 0.69 385

acc: 0.69 < 0.72 < 0.74

Confusion Matrix:

A chart of different colored squares

Description automatically generated

RandomForest:

{'max\_depth': None, 'max\_features': 'sqrt', 'n\_estimators': 100}

Classification Report:

precision recall f1-score support

D0 0.94 0.73 0.82 60

D1 0.64 0.79 0.71 52

D2 0.60 0.67 0.63 42

D3 0.73 0.65 0.69 49

D4 0.78 0.67 0.72 69

D5 0.77 0.90 0.83 60

D6 0.96 0.98 0.97 53

accuracy 0.77 385

macro avg 0.77 0.77 0.77 385

weighted avg 0.78 0.77 0.77 385

acc: 0.77 = 0.77 < 0.82

Confusion Matrix:

A chart of a number of colored squares

Description automatically generated with medium confidence

ExtraTrees:

{'max\_depth': 30, 'max\_features': None, 'n\_estimators': 50}

Classification Report:

precision recall f1-score support

D0 0.92 0.75 0.83 60

D1 0.63 0.69 0.66 52

D2 0.66 0.69 0.67 42

D3 0.62 0.65 0.63 49

D4 0.74 0.72 0.73 69

D5 0.82 0.83 0.83 60

D6 0.96 0.98 0.97 53

accuracy 0.76 385

macro avg 0.76 0.76 0.76 385

weighted avg 0.77 0.76 0.77 385

acc: 0.76 < 0.77 < 0.81

Confusion Matrix:

A chart of different colored squares

Description automatically generated

GradientBoostedTrees:

{'learning\_rate': 0.1, 'max\_depth': 20, 'n\_estimators': 100}

Classification Report:

precision recall f1-score support

D0 0.84 0.70 0.76 60

D1 0.57 0.65 0.61 52

D2 0.65 0.62 0.63 42

D3 0.64 0.59 0.62 49

D4 0.78 0.67 0.72 69

D5 0.75 0.95 0.84 60

D6 0.95 0.98 0.96 53

accuracy 0.74 385

macro avg 0.74 0.74 0.73 385

weighted avg 0.75 0.74 0.74 385

acc: 0.74 < 0.76 < 0.81

Confusion Matrix:

A chart of different colored squares

Description automatically generated