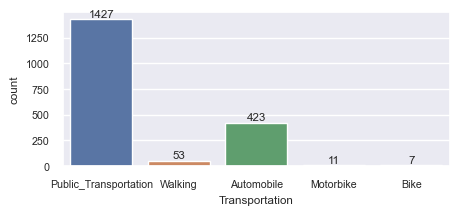
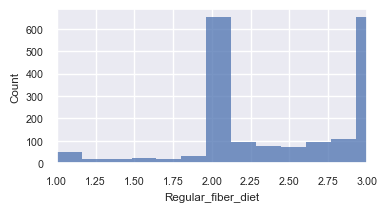
Tema ML

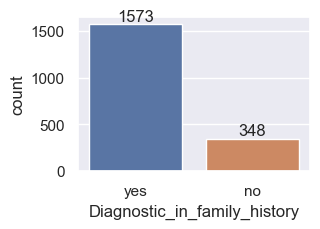
Veliscu Robert-Valentin

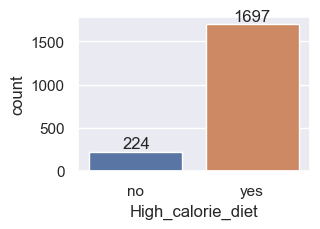
341C4

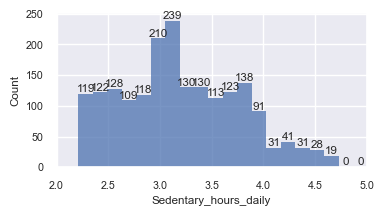
3.1.1 Diagrame count / histograme pentru fiecare atribut:



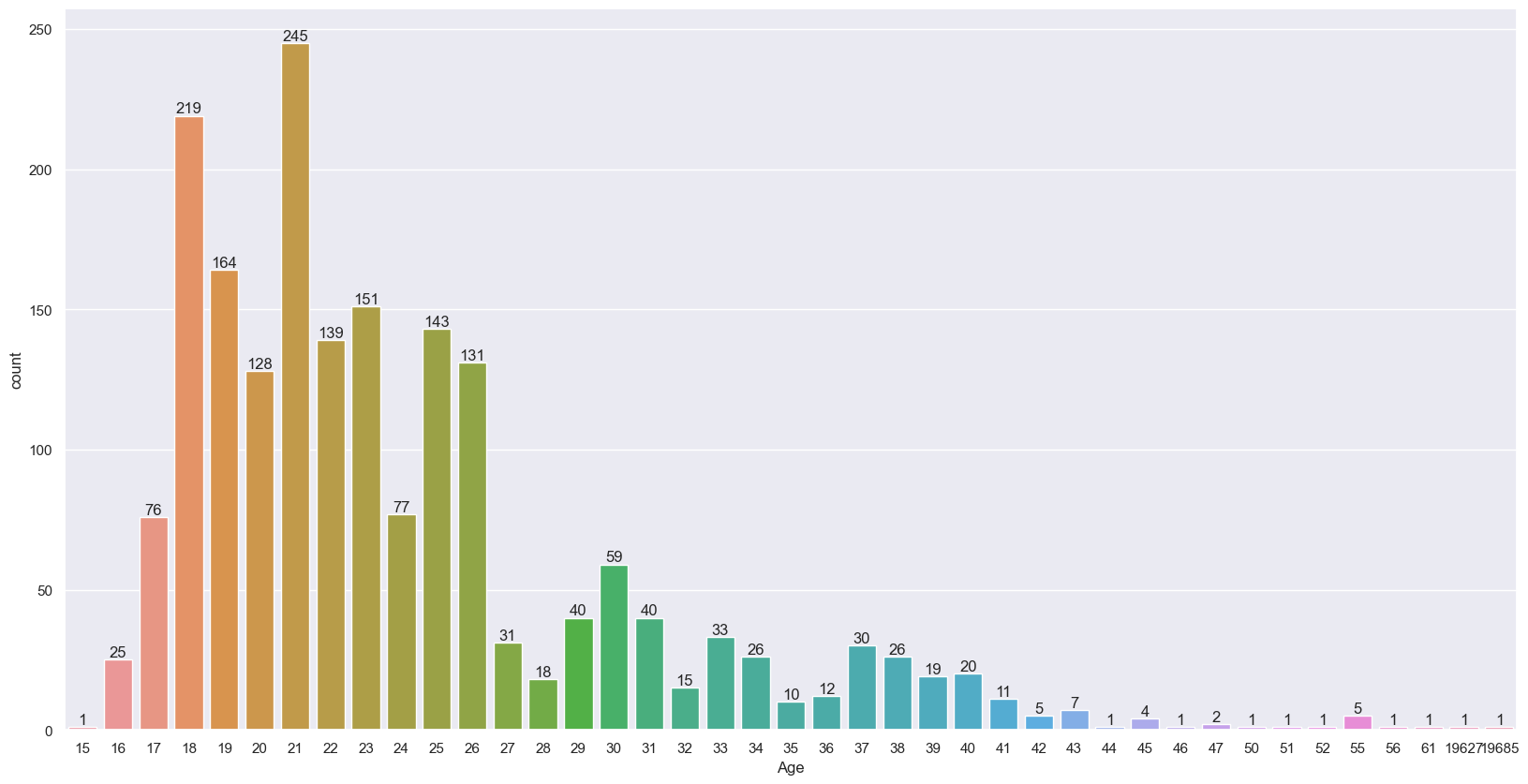




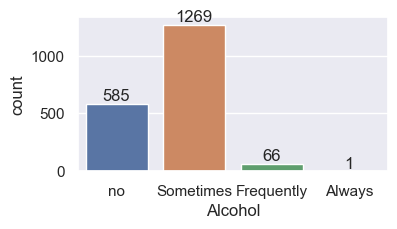


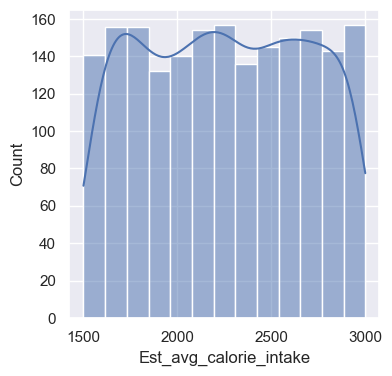


Pentru sedentary\_hours\_daily am setat limitele pe axa x intre 2 si 5, unde se aflau majoritatea pentru a putea viualiza histograma, deoarece era un outlier (greseala) cu o valoare foarte mare.

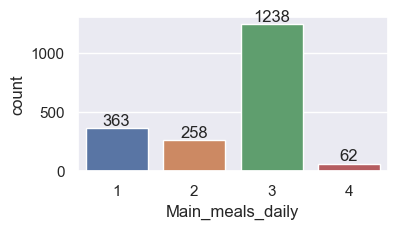


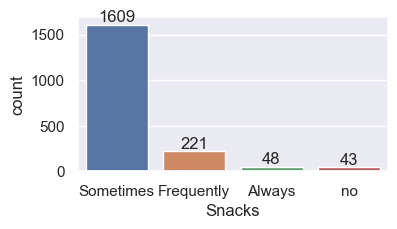
Pentru Age am rotunjit valorile la partea intreaga pentru a putea vizualiza mai bine si pentru a avea o relevanta mai mare.

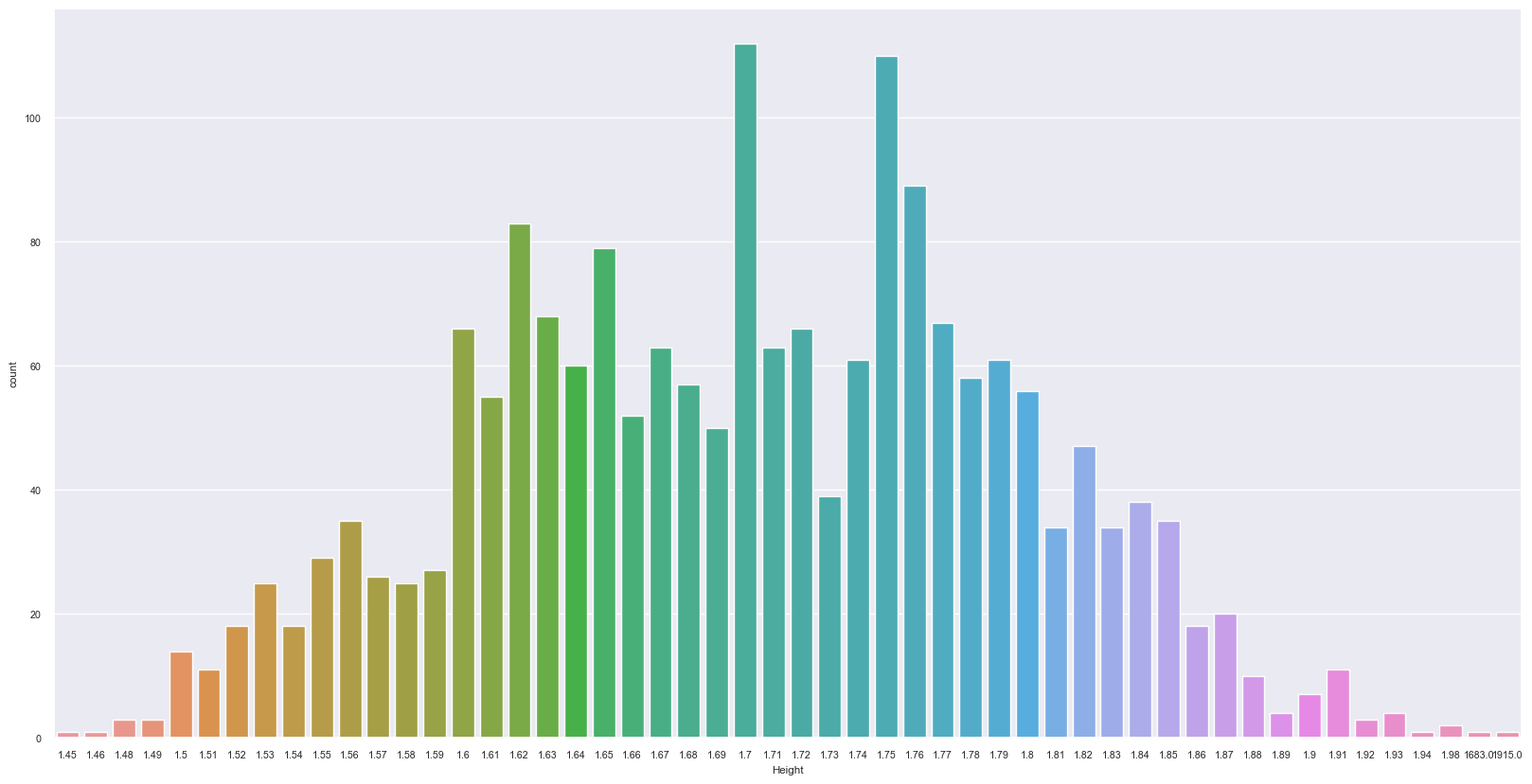


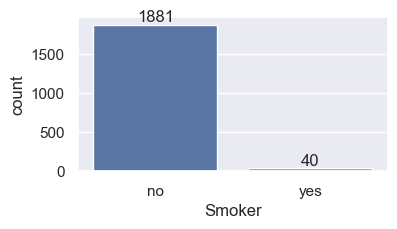


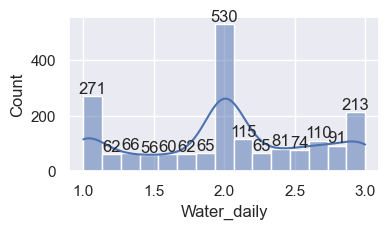
Pentru main meals daily am rotunjit din nou (fie iei masa fie nu)

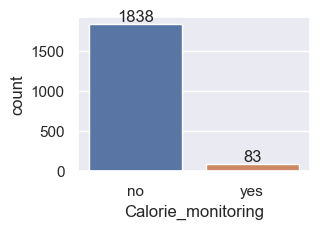


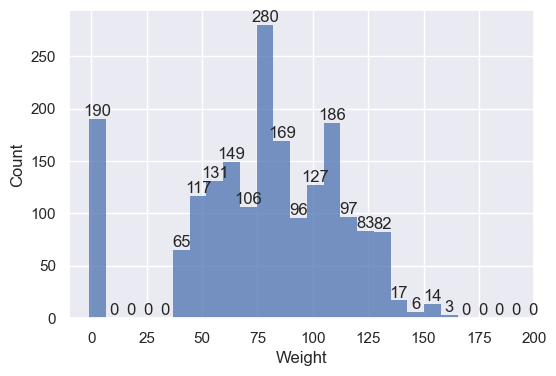


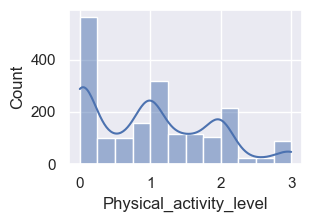




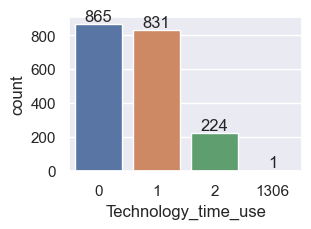


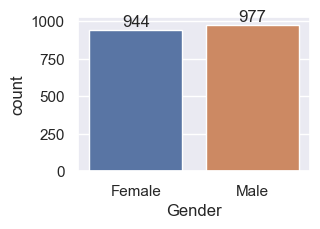


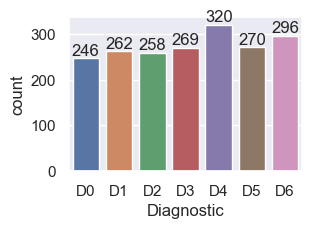




Putem observa un outlier la tech\_time\_use







3.1.2



Din valorile statistice extrase pentru atributele numerice putem observa:

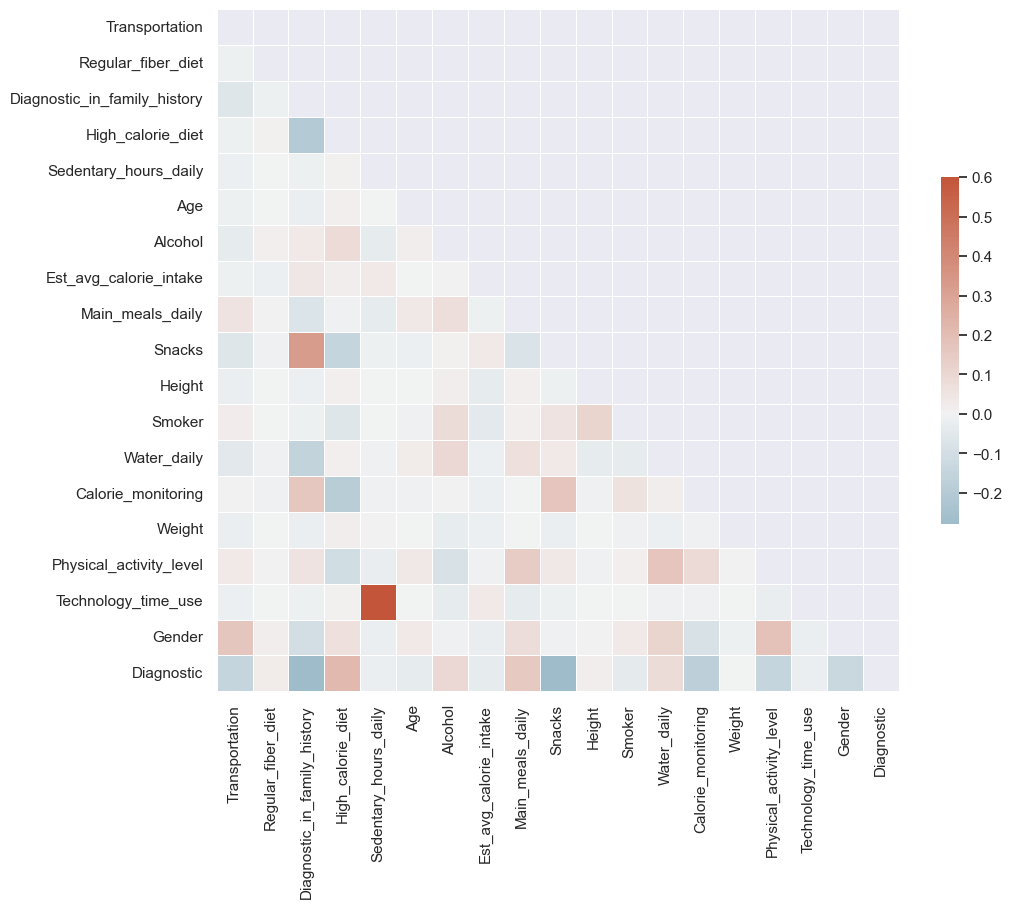
Multe dintre atribute au outlieri care influenteaza mult abaterea standard

Atributele au valori cu ordine de marimi diferite => necesita standardizare

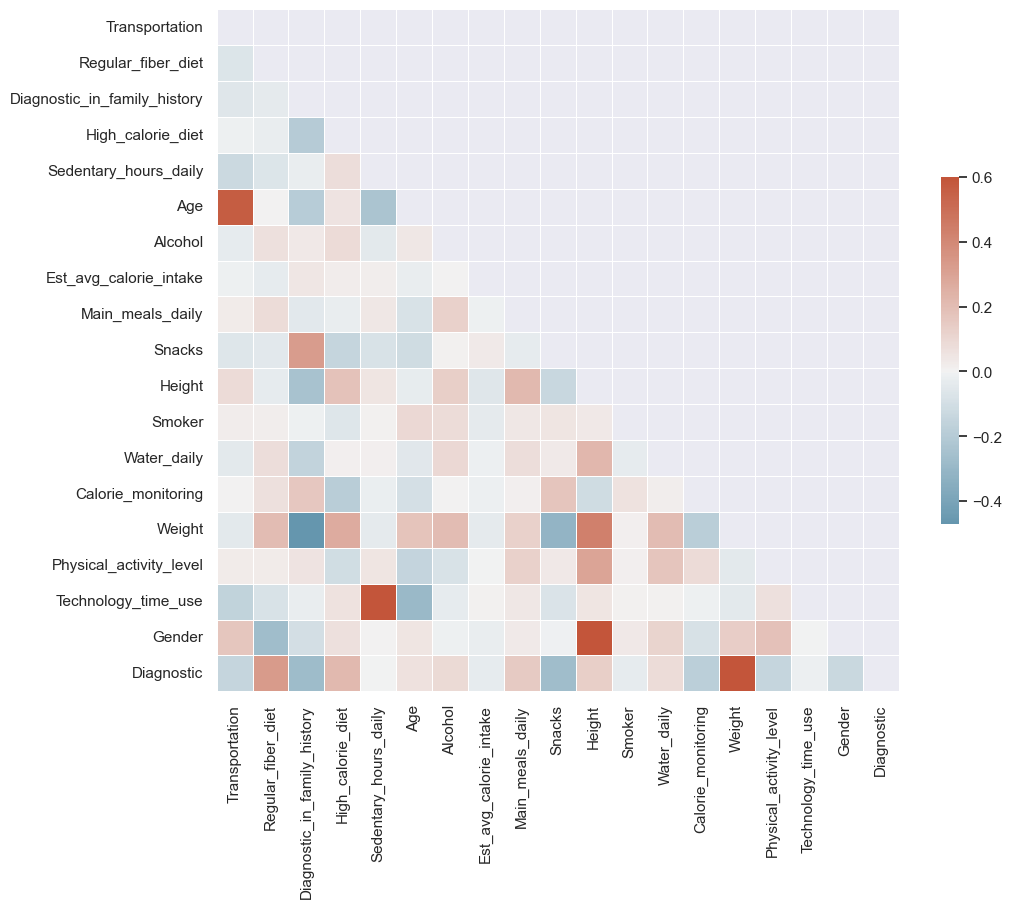


Valori unice:

* Transportation: [Public\_Transportation, Walking, Automobile, Motorbike,Bike],
* Diagnostic\_in\_family\_history: [yes, no],
* High\_calorie\_diet: [no, yes],
* Alcohol: [no, Sometimes, Frequently, Always],
* Snacks: [Sometimes, Frequently, Always, no],
* Smoker: [no, yes],
* Calorie\_monitoring: [no, yes],
* Gender: [Female, Male],
* Diagnostic: [D1, D2, D3, D4, D0, D5, D6]



Matricea de corelatie dupa tratarea valorilor lipsa si tratarea outlierilor:



Pentru tratarea valorilor lipsa am folosit SimpleImputer cu strategia mean.

Outlierii i-am scalat la ordinul de marime correct (am considerat ca a fost un typo / formatare automata a excel-ului), cu exceptia sedentary\_hours\_daily unde l-am inlocuit cu cea mai mare valoare (dintre celelalte valori) si technology\_time\_use unde l-am inlocuit cu 1.

Valorile din matricea de corelatie pentru Diagnostic sortate crescator in functie de valoarea absoluta:

Sedentary\_hours\_daily -0.002829

Technology\_time\_use -0.015710

Est\_avg\_calorie\_intake -0.036043

Smoker -0.038645

Age 0.061441

Water\_daily 0.081613

Alcohol 0.094033

Gender -0.132271

Height 0.134737

Physical\_activity\_level -0.144771

Transportation -0.146906

Main\_meals\_daily 0.151026

Calorie\_monitoring -0.177705

High\_calorie\_diet 0.218737

Snacks -0.274317

Diagnostic\_in\_family\_history -0.277579

Regular\_fiber\_diet 0.329348

Weight 0.598639

Diagnostic 1.000000

Valorile din matricea de covarianta pentru Diagnostic sortate crescator in functie de valoarea absoluta:

Sedentary\_hours\_daily -0.003254

Smoker -0.011013

Technology\_time\_use -0.021162

Height 0.025091

Calorie\_monitoring -0.072110

Alcohol 0.097250

Water\_daily 0.099499

Gender -0.131971

High\_calorie\_diet 0.140110

Diagnostic\_in\_family\_history -0.213364

Physical\_activity\_level -0.247120

Main\_meals\_daily 0.250748

Transportation -0.257017

Snacks -0.329642

Regular\_fiber\_diet 0.350327

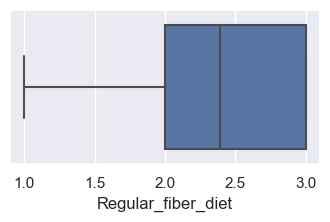
Age 0.780691

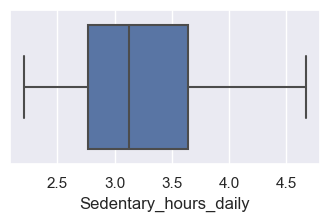
Diagnostic 3.980978

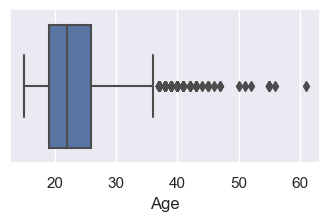
Weight 29.727358

Est\_avg\_calorie\_intake -31.216057

Boxplot-uri pentru fiecare atribut dupa tratarea valorilor lipsa si outlierilor (majori):

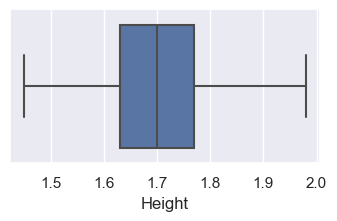


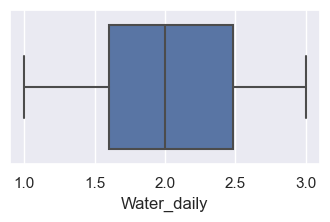


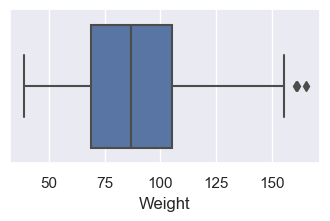


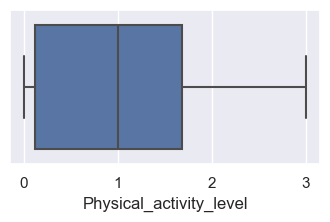


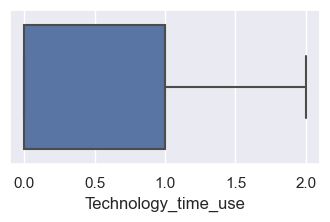












3.2

Pentru tratarea valorilor lipsa am folosit SimpleImputer cu strategia mean.

Pentru scalarea datelor am folosit RobustScaler deoarece la Age si Weight inca am cativa outlieri

Pentru partea de feature selection am folosit VarianceThreshold si SelectPercentile cu functia de scor mutual information si f\_classif.

* Pentru VarianceThreshold am folosit un threshold de 50% ( am eliminat atributele care in 50% din cazuri aveau aceeasi valore => nu ofereau insight). Astfel am redus datasetul la urmatoarele attribute:

['Transportation', 'Regular\_fiber\_diet', 'Sedentary\_hours\_daily', 'Age',

'Alcohol', 'Est\_avg\_calorie\_intake', 'Main\_meals\_daily', 'Snacks',

'Height', 'Water\_daily', 'Weight', 'Physical\_activity\_level',

'Technology\_time\_use']

* Pentru SelectPercentile cu functia de scor f\_classif am folostit un procent de 50% (am luat doar atributele cu cele mai bune 50% scoruri) si am redus dataset-ul la urmatoarele attribute:

['Regular\_fiber\_diet', 'Diagnostic\_in\_family\_history',

'High\_calorie\_diet', 'Age', 'Main\_meals\_daily', 'Snacks', 'Height',

'Weight', 'Gender']

* Pentru SelectPercentile cu functia de scor mutual information am folostit un procent de 50% (am luat doar atributele cu cele mai bune 50% scoruri) si am redus dataset-ul la urmatoarele attribute:

['Regular\_fiber\_diet', 'Age', 'Main\_meals\_daily', 'Snacks', 'Height', 'Water\_daily', 'Weight', 'Physical\_activity\_level', 'Gender']

In variance threshold doar elimin atributele care au in majoritate aceeasi valoare, fara a tine cont de relatia acestora cu target-ul. Spre exemplu Sedentary\_hours\_daily se afla in atributele mentinute de variance threshold dar nu in celelalte 2 cazuri deoarece, desi are o variatie de valori, aceasta nu are o relevanta ridicata pentru clasificare (asa cum se vede si din matricea de corelatie).

Diferenta intre f\_classif si mutual information este functia care determina impactul atributelor asupra target-ului, iar pe baza valorilor date de aceasta functie putem alege atributele ramase. Astfel putem observa cateva diferente intre atributele ramase.

Algoritmi

SVM:

Best scores:

Selected using variance threshold: 0.845697787554465 {'C': 10, 'kernel': 'rbf'}

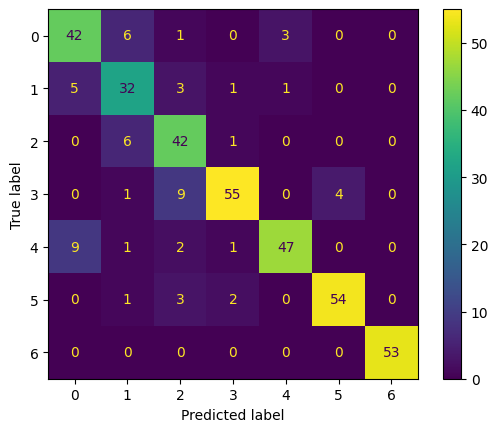
Selected using mutual information: 0.8593785693134226 {'C': 10, 'kernel': 'rbf'}

Selected using f-classif: 0.8626295528575658 {'C': 10, 'kernel': 'rbf'}

No selection: 0.8632746732095267 {'C': 10, 'kernel': 'rbf'}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | clasa | precision | recall | f1 |
| variance threshold | 0 | 0.653061 | 0.615385 | 0.633663 |
| 1 | 0.634615 | 0.785714 | 0.702128 |
| 2 | 0.72 | 0.734694 | 0.727273 |
| 3 | 0.8 | 0.753623 | 0.776119 |
| 4 | 0.862745 | 0.733333 | 0.792793 |
| 5 | 0.857143 | 0.9 | 0.878049 |
| 6 | 0.945455 | 0.981132 | 0.962963 |
| mutual information | 0 | 0.744681 | 0.673077 | 0.707071 |
| 1 | 0.644444 | 0.690476 | 0.666667 |
| 2 | 0.725806 | 0.918367 | 0.810811 |
| 3 | 0.949153 | 0.811594 | 0.875 |
| 4 | 0.839286 | 0.783333 | 0.810345 |
| 5 | 0.933333 | 0.933333 | 0.933333 |
| 6 | 0.946429 | 1 | 0.972477 |
| f-classif | 0 | 0.75 | 0.807692 | 0.777778 |
| 1 | 0.680851 | 0.761905 | 0.719101 |
| 2 | 0.7 | 0.857143 | 0.770642 |
| 3 | 0.916667 | 0.797101 | 0.852713 |
| 4 | 0.921569 | 0.783333 | 0.846847 |
| 5 | 0.931034 | 0.9 | 0.915254 |
| 6 | 1 | 1 | 1 |
| no feature selection | 0 | 0.6 | 0.692308 | 0.642857 |
| 1 | 0.659574 | 0.738095 | 0.696629 |
| 2 | 0.781818 | 0.877551 | 0.826923 |
| 3 | 0.913793 | 0.768116 | 0.834646 |
| 4 | 0.897959 | 0.733333 | 0.807339 |
| 5 | 0.887097 | 0.916667 | 0.901639 |
| 6 | 0.962963 | 0.981132 | 0.971963 |

Matricea de confuzie pentru f-classif:



Random Forest:

Selected using variance threshold: 0.8867337873852531 {'criterion': 'entropy', 'max\_depth': 8, 'max\_features': 'sqrt', 'n\_estimators': 500}

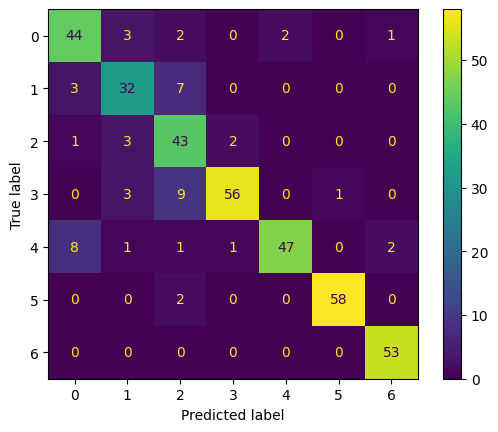
Selected using mutual information: 0.8899720800372265 {'criterion': 'entropy', 'max\_depth': 8, 'max\_features': 'sqrt', 'n\_estimators': 500}

Selected using f-classif: 0.8893100384957062 {'criterion': 'entropy', 'max\_depth': 8, 'max\_features': 'sqrt', 'n\_estimators': 500}

No selection: 0.9016921189559625 {'criterion': 'entropy', 'max\_depth': 8, 'max\_features': 'sqrt', 'n\_estimators': 500}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Random Forest | clasa | precision | recall | f1 |
| variance threshold | 0 | 0.75 | 0.807692 | 0.777778 |
| 1 | 0.744186 | 0.761905 | 0.752941 |
| 2 | 0.727273 | 0.816327 | 0.769231 |
| 3 | 0.904762 | 0.826087 | 0.863636 |
| 4 | 0.96 | 0.8 | 0.872727 |
| 5 | 0.95 | 0.95 | 0.95 |
| 6 | 0.896552 | 0.981132 | 0.936937 |
| mutual information | 0 | 0.785714 | 0.846154 | 0.814815 |
| 1 | 0.761905 | 0.761905 | 0.761905 |
| 2 | 0.671875 | 0.877551 | 0.761062 |
| 3 | 0.949153 | 0.811594 | 0.875 |
| 4 | 0.959184 | 0.783333 | 0.862385 |
| 5 | 0.983051 | 0.966667 | 0.97479 |
| 6 | 0.946429 | 1 | 0.972477 |
| f-classif | 0 | 0.754098 | 0.884615 | 0.814159 |
| 1 | 0.842105 | 0.761905 | 0.8 |
| 2 | 0.692308 | 0.918367 | 0.789474 |
| 3 | 0.90625 | 0.84058 | 0.87218 |
| 4 | 0.957447 | 0.75 | 0.841121 |
| 5 | 1 | 0.933333 | 0.965517 |
| 6 | 0.981481 | 1 | 0.990654 |
| no feature selection | 0 | 0.733333 | 0.846154 | 0.785714 |
| 1 | 0.785714 | 0.785714 | 0.785714 |
| 2 | 0.688525 | 0.857143 | 0.763636 |
| 3 | 0.933333 | 0.811594 | 0.868217 |
| 4 | 0.979167 | 0.783333 | 0.87037 |
| 5 | 0.966102 | 0.95 | 0.957983 |
| 6 | 0.963636 | 1 | 0.981481 |

Matricea de confuzie pentru mutual information:



Extra trees:

Selected using variance threshold: 0.8906341215787471 {'criterion': 'gini', 'max\_depth': 30, 'max\_samples': 0.8, 'n\_estimators': 500}

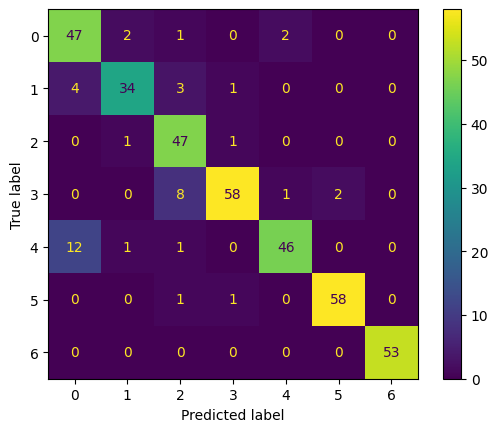
Selected using mutual information: 0.8964909683150726 {'criterion': 'entropy', 'max\_depth': 20, 'max\_samples': 0.8, 'n\_estimators': 100}

Selected using f-classif: 0.8971318583696434 {'criterion': 'gini', 'max\_depth': 20, 'max\_samples': 0.8, 'n\_estimators': 100}

No selection: 0.9010343077118321 {'criterion': 'entropy', 'max\_depth': 20, 'max\_samples': 0.8, 'n\_estimators': 300}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Extra trees | clasa | precision | recall | f1 |
| variance threshold | 0 | 0.716981 | 0.730769 | 0.72381 |
| 1 | 0.755556 | 0.809524 | 0.781609 |
| 2 | 0.788462 | 0.836735 | 0.811881 |
| 3 | 0.919355 | 0.826087 | 0.870229 |
| 4 | 0.924528 | 0.816667 | 0.867257 |
| 5 | 0.935484 | 0.966667 | 0.95082 |
| 6 | 0.896552 | 0.981132 | 0.936937 |
| mutual information | 0 | 0.777778 | 0.807692 | 0.792453 |
| 1 | 0.837209 | 0.857143 | 0.847059 |
| 2 | 0.775862 | 0.918367 | 0.841121 |
| 3 | 0.967742 | 0.869565 | 0.916031 |
| 4 | 0.923077 | 0.8 | 0.857143 |
| 5 | 0.983333 | 0.983333 | 0.983333 |
| 6 | 0.946429 | 1 | 0.972477 |
| f-classif | 0 | 0.746032 | 0.903846 | 0.817391 |
| 1 | 0.894737 | 0.809524 | 0.85 |
| 2 | 0.770492 | 0.959184 | 0.854545 |
| 3 | 0.95082 | 0.84058 | 0.892308 |
| 4 | 0.938776 | 0.766667 | 0.844037 |
| 5 | 0.966667 | 0.966667 | 0.966667 |
| 6 | 1 | 1 | 1 |
| no feature selection | 0 | 0.709091 | 0.75 | 0.728972 |
| 1 | 0.777778 | 0.833333 | 0.804598 |
| 2 | 0.77193 | 0.897959 | 0.830189 |
| 3 | 0.95 | 0.826087 | 0.883721 |
| 4 | 0.903846 | 0.783333 | 0.839286 |
| 5 | 0.935484 | 0.966667 | 0.95082 |
| 6 | 0.962963 | 0.981132 | 0.971963 |

Matricea de confuzie pentru f-classif:



GradientBoostedTrees:

Selected using variance threshold: 0.9075595414357629 {'learning\_rate': 0.3, 'max\_depth': 7, 'n\_estimators': 300}

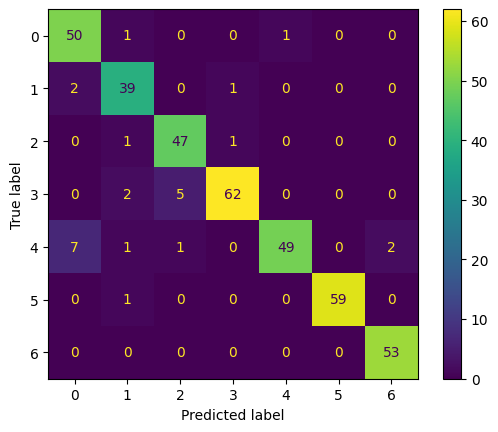
Selected using mutual information: 0.9121155717246922 {'learning\_rate': 0.1, 'max\_depth': 5, 'n\_estimators': 400}

Selected using f-classif: 0.9238271500486483 {'learning\_rate': 0.2, 'max\_depth': 5, 'n\_estimators': 300}

No selection: 0.9186217691103685 {'learning\_rate': 0.2, 'max\_depth': 7, 'n\_estimators': 500}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| XGB | clasa | precision | recall | f1 |
| variance threshold | 0 | 0.87234 | 0.788462 | 0.828283 |
| 1 | 0.770833 | 0.880952 | 0.822222 |
| 2 | 0.821429 | 0.938776 | 0.87619 |
| 3 | 0.936508 | 0.855072 | 0.893939 |
| 4 | 0.980392 | 0.833333 | 0.900901 |
| 5 | 0.966667 | 0.966667 | 0.966667 |
| 6 | 0.866667 | 0.981132 | 0.920354 |
| mutual information | 0 | 0.833333 | 0.865385 | 0.849057 |
| 1 | 0.770833 | 0.880952 | 0.822222 |
| 2 | 0.811321 | 0.877551 | 0.843137 |
| 3 | 0.967742 | 0.869565 | 0.916031 |
| 4 | 0.942308 | 0.816667 | 0.875 |
| 5 | 0.983051 | 0.966667 | 0.97479 |
| 6 | 0.929825 | 1 | 0.963636 |
| f-classif | 0 | 0.847458 | 0.961538 | 0.900901 |
| 1 | 0.866667 | 0.928571 | 0.896552 |
| 2 | 0.886792 | 0.959184 | 0.921569 |
| 3 | 0.96875 | 0.898551 | 0.932331 |
| 4 | 0.98 | 0.816667 | 0.890909 |
| 5 | 1 | 0.983333 | 0.991597 |
| 6 | 0.963636 | 1 | 0.981481 |
| no feature selection | 0 | 0.849057 | 0.865385 | 0.857143 |
| 1 | 0.880952 | 0.880952 | 0.880952 |
| 2 | 0.87037 | 0.959184 | 0.912621 |
| 3 | 0.938462 | 0.884058 | 0.910448 |
| 4 | 0.961538 | 0.833333 | 0.892857 |
| 5 | 0.983333 | 0.983333 | 0.983333 |
| 6 | 0.898305 | 1 | 0.946429 |

Matricea de confuzie pentru f-classif:



Putem observa ca in majoritatea cazurilor, feature selection-ul cu SelectPercentile cu functia de scor f-classif a fost cea mai avantajoasa.

Clasele cu cele mai bune predictii au fost clasa 3 si clasa 5.

Cel mai performant model a fost GradientBoostedTrees cu hiperparametrii: 'learning\_rate': 0.2, 'max\_depth': 5, 'n\_estimators': 300 si feature selection folosind f-classif