Prediction of Obesity

by your devoted

Leonard Desportes

Romain Girodet

Summary

I- Presentation of the BDD

Name: ObesityDataSet_raw_and_data_sinthetic

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
0	Female	21.0	1.62	64.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	0.0	1.0	no	Public_Transportation	Normal_Weight
1	Female	21.0	1.52	56.0	yes	no	3.0	3.0	Sometimes	yes	3.0	yes	3.0	0.0	Sometimes	Public_Transportation	Normal_Weight
2	Male	23.0	1.80	77.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	2.0	1.0	Frequently	Public_Transportation	Normal_Weight
3	Male	27.0	1.80	87.0	no	no	3.0	3.0	Sometimes	no	2.0	no	2.0	0.0	Frequently	Walking	Overweight_Level_I
4	Male	22.0	1.78	89.8	no	no	2.0	1.0	Sometimes	no	2.0	no	0.0	0.0	Sometimes	Public_Transportation	Overweight_Level_II

Dimension: 2111 rows × 17 columns

II- Creation of groups of variables

- **ID of the subject:** Gender, Age, Height, Weight, Family_history_with_overweight, NObeyesdad
- **Addictions naucify:** Smoke, Consumption of alcohol (CALC)
- **Good eating habits:** Frequency of consumption of vegetables (FCVC) Consumption of water daily (CH20)
- **Bad eating habits:** Frequent consumption of high caloric food (FAVC), Consumption of food between meals (CALC)
- **Quantification of food consomption :** Number of main meals (NCP), Calorie's consumption monitoring (SCC)
- **Lifestyle habits:** Physical activity frequency (FAF), Time using technology devices (TUE), Transportation used (MTRANS)

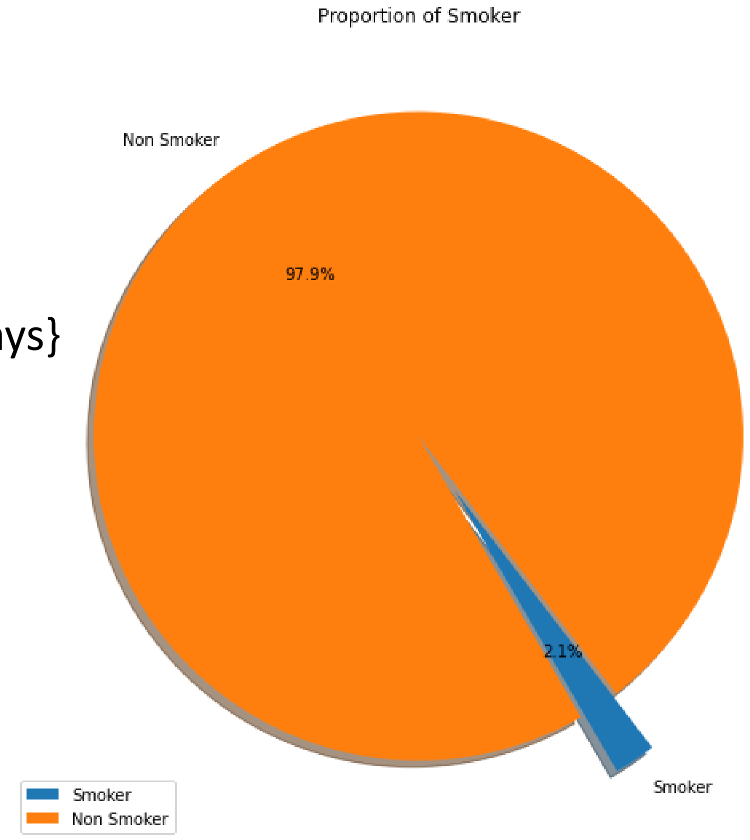
III- Analysis of Data, global analysis Harmful addictions

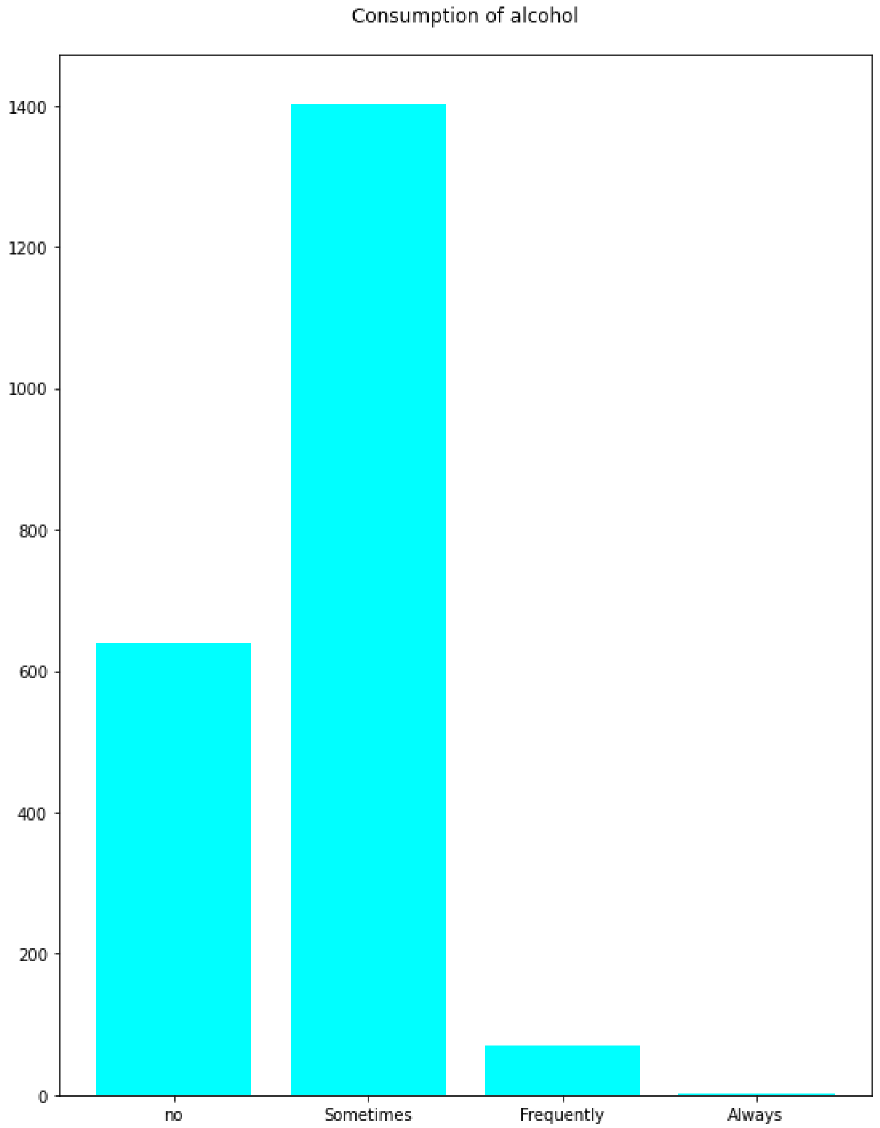
SMOKE {yes,no}

Answering to the question:
"Do you smoke?"

CALC {no,Sometimes,Frequently,Always}

Answering to the question:
"How often do you drink alcohol?"

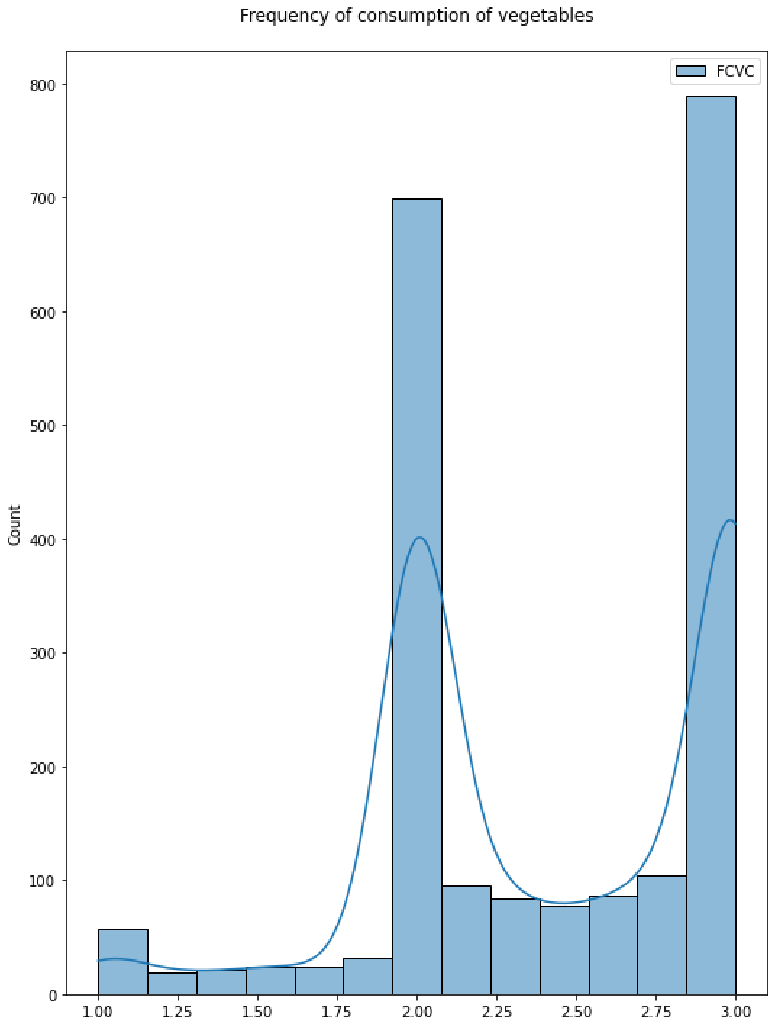


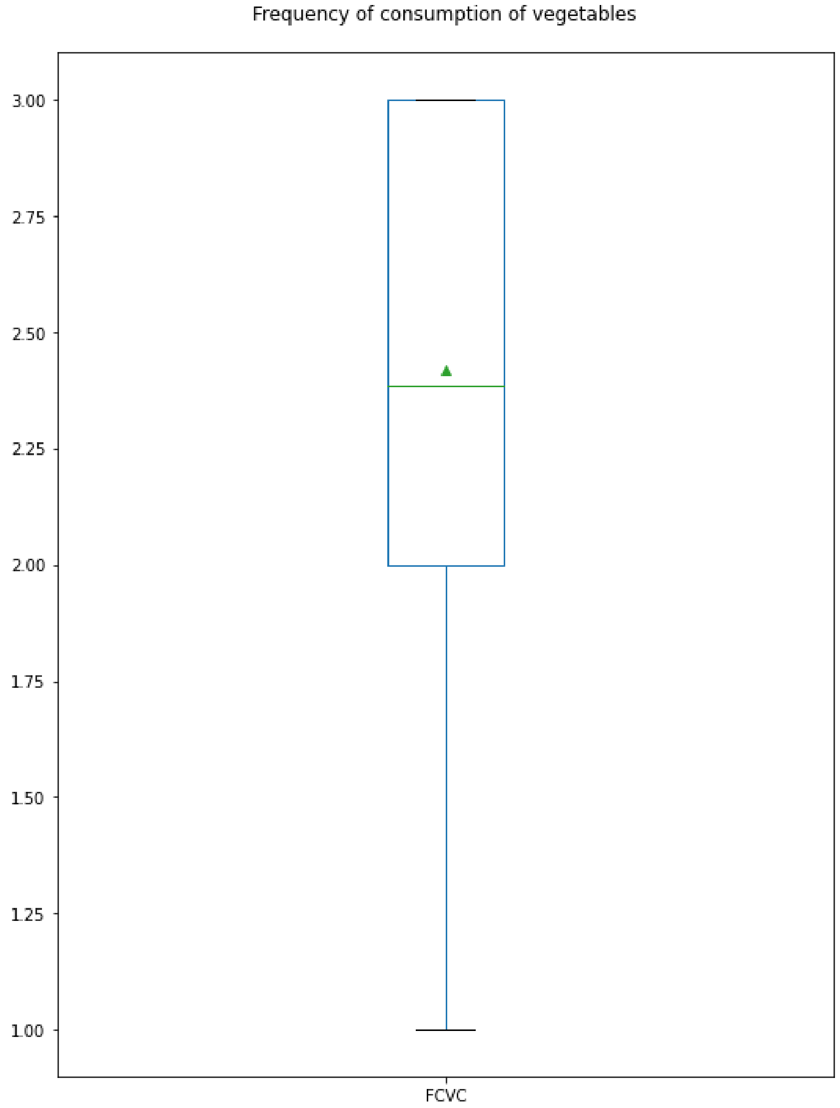


III- Analysis of Data, global analysis of good eating habits

FCVC {numeric value from 1 to 3}
1= Never
2= Sometimes
3= Always

Answering to the question:
“Do you usually eat vegetables in your meals?”





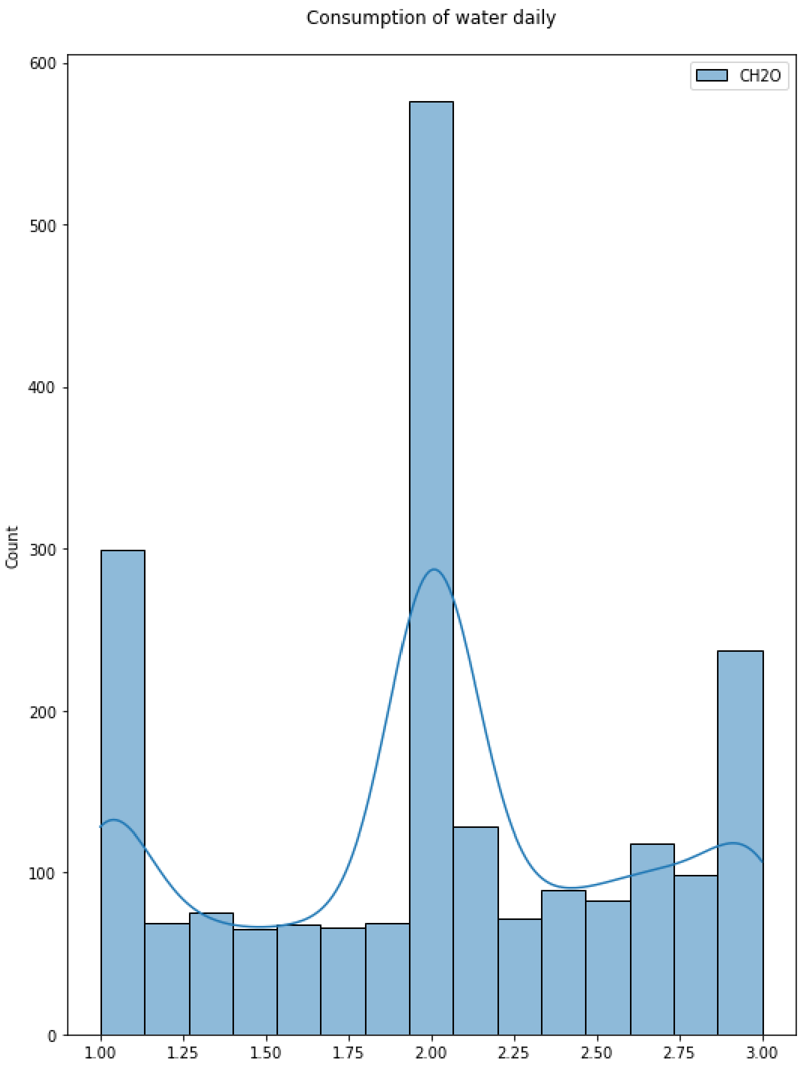
III- Analysis of Data, global analysis of good eating habits

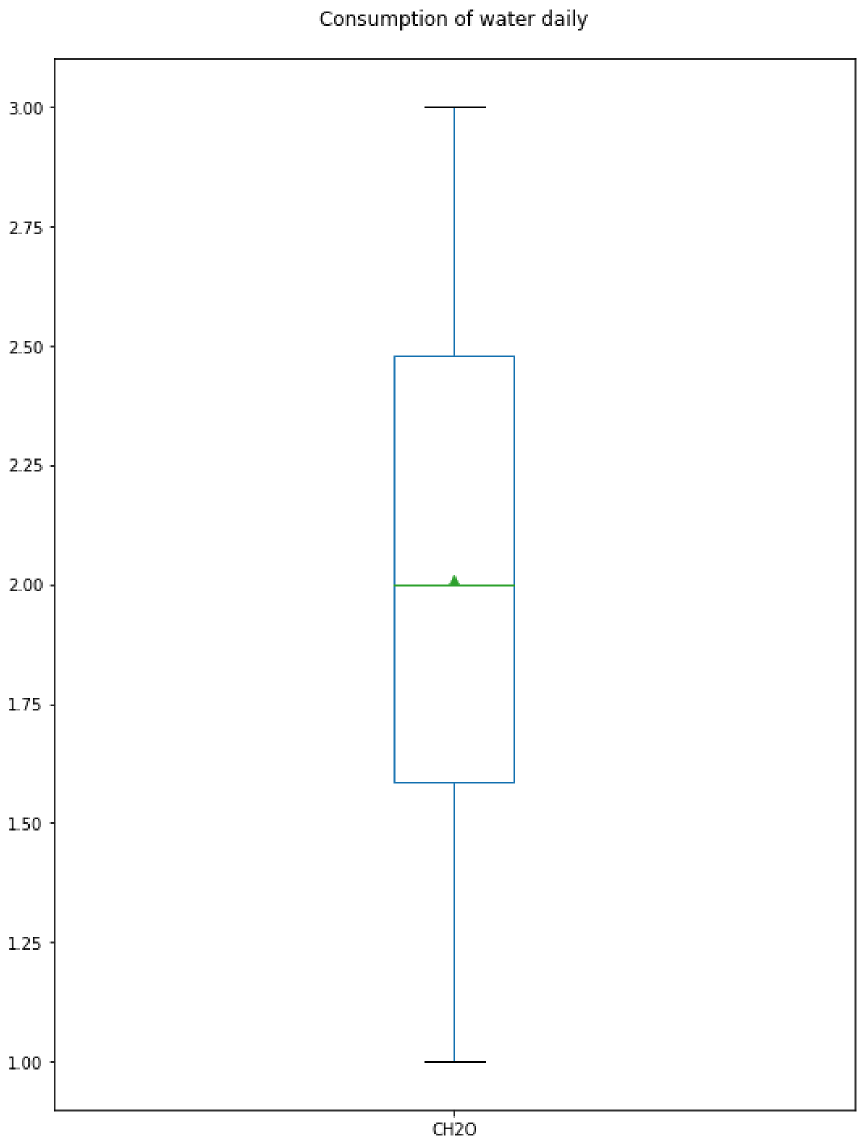
CH2O {numeric value from 1 to 3}
First interpretation:
- 1= Less than a liter
-]1,3[= Between 1 and 2 L
- 3= More than 2 L

Second Interpretation:
- The measuring unit is the liter.

Answering to the question:
"How much water do you drink daily?"

This feature wasn't significant enough. We didn't keep it in our final model.





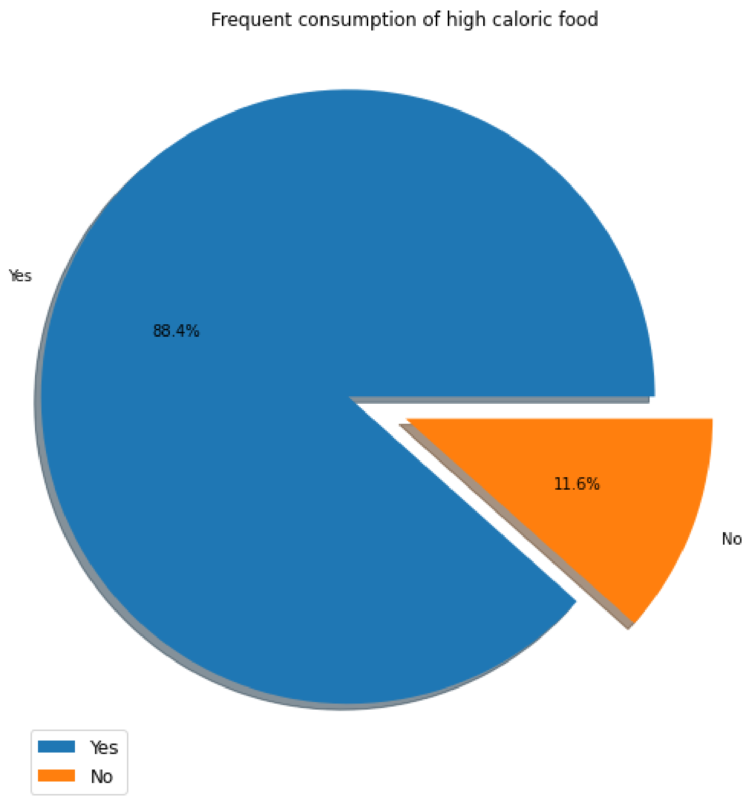
III- Analysis of Data, global analysis of bad eating habits

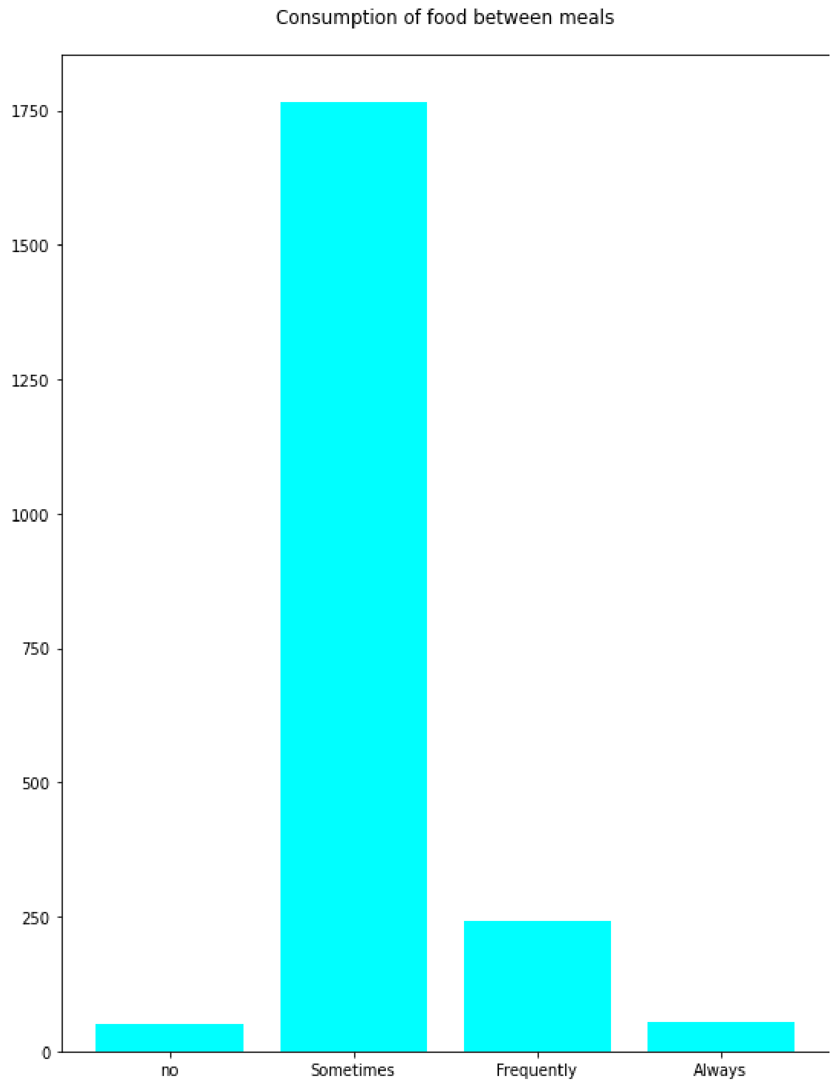
FAVC {yes,no}

Answering to the question:
“Do you eat high caloric food frequently?”

CALC {no,Sometimes,Frequently,Always}

Answering to the question:
“Do you eat any food between meals?”





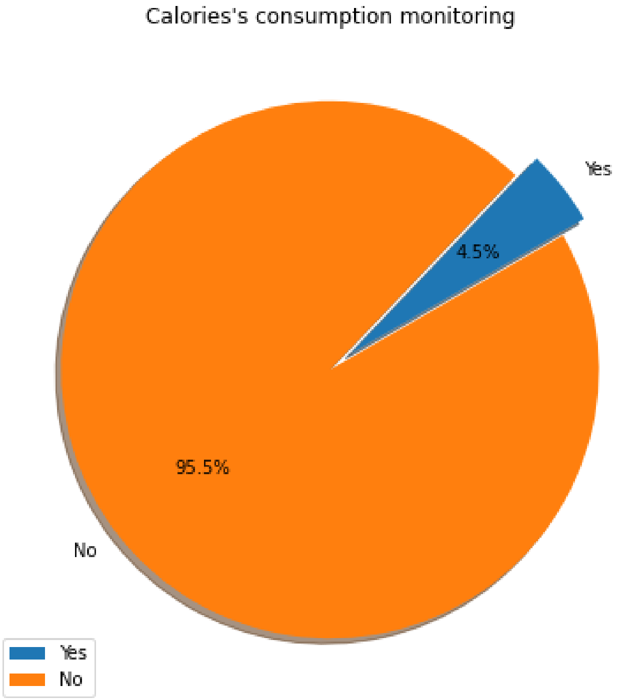
III- Analysis of Data, global analysis of Quantification of food consumption

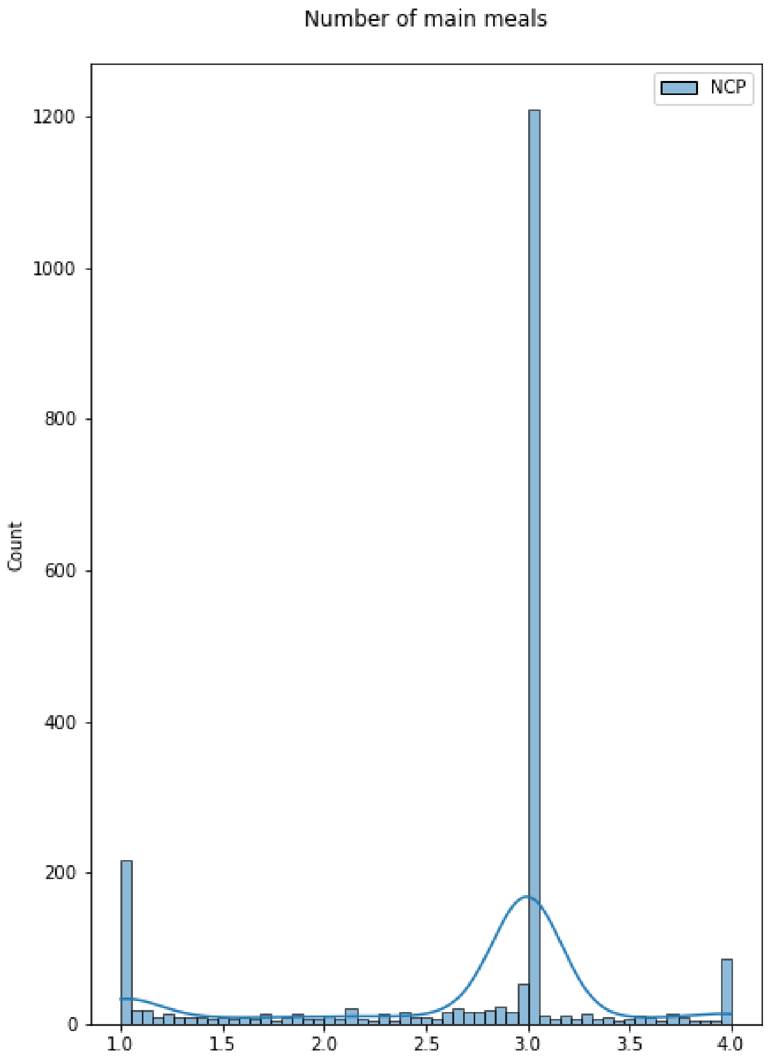
SCC {yes,no}

Answering to the question:
“Do you monitor the calories you eat daily?”

NCP {numeric value from 1 to 3}

“How many main meals do you have daily?”

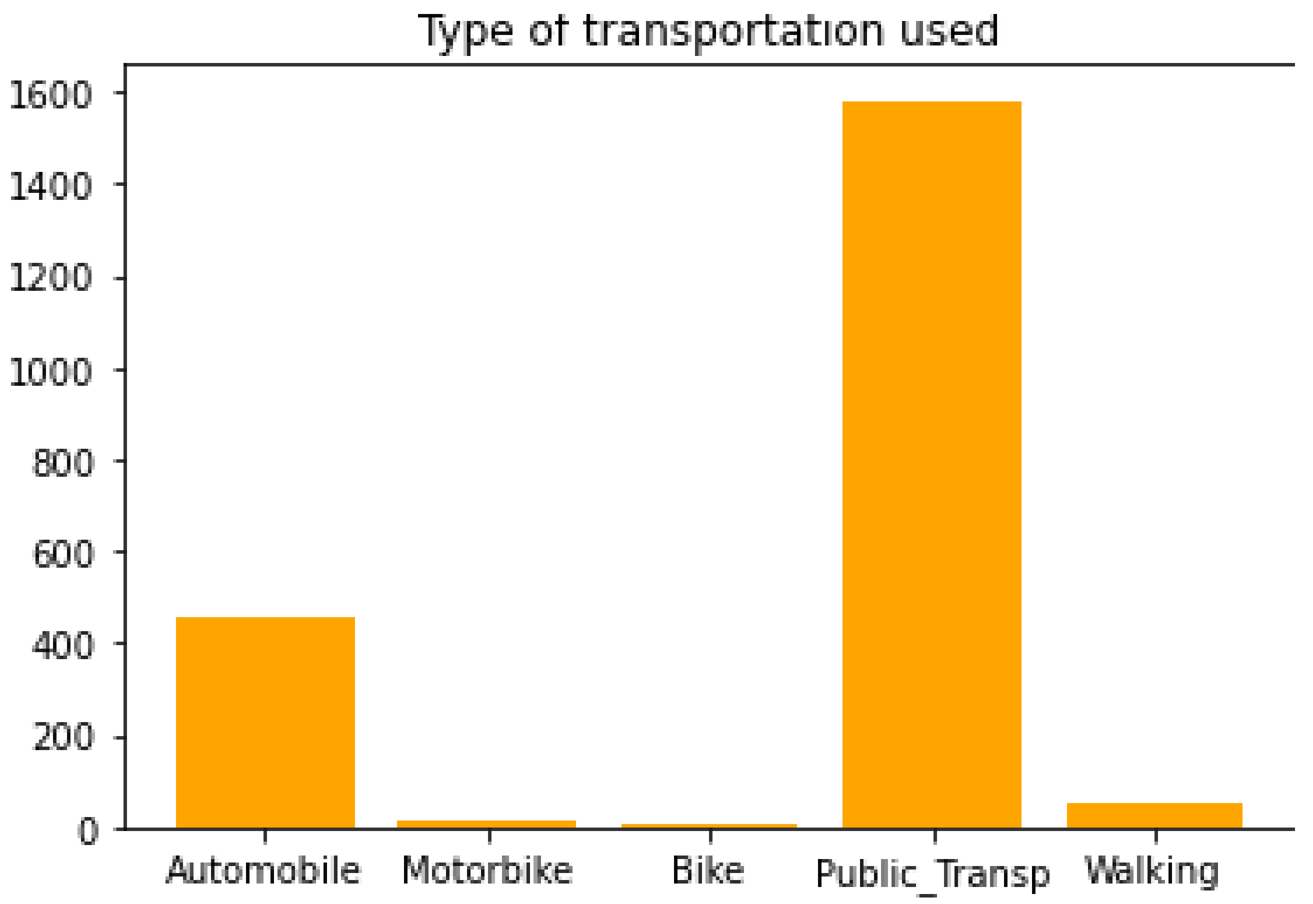




III- Analysis of Data, global analysis of Lifestyle habits

MTRANS
{Automobile,Motorbike,Bike,Public_Transportation, Walking}

Answering to the question:
“Which transportation do you usually use?”

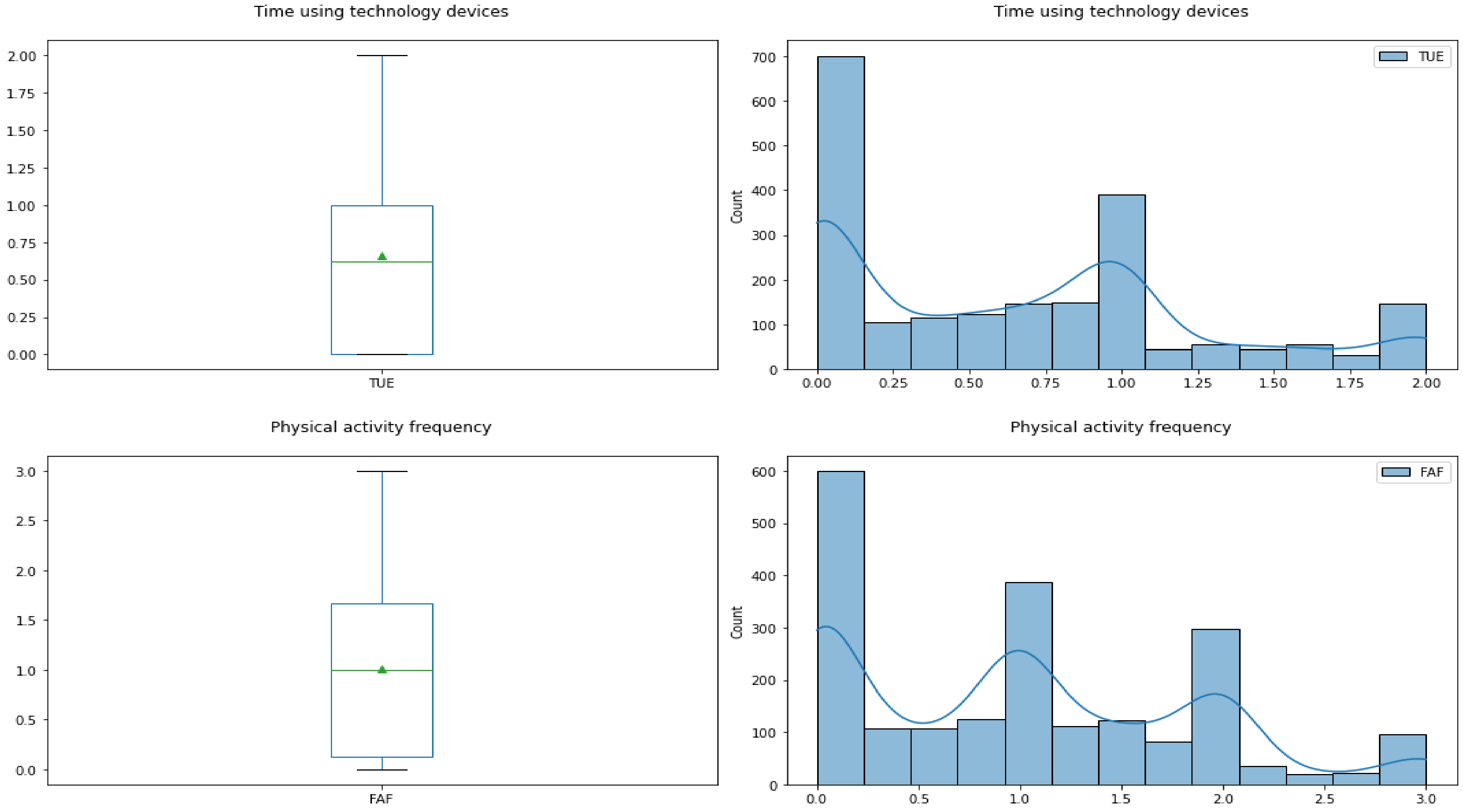


III- Analysis of Data, global analysis of Lifestyle habits

TUE numeric{from 0 to 2}
[0,1[= 0–2 hours
[1,2[=3–5 hours
2=More than 5 hoursAnswering to the question:
“How much time do you use technological devices such as cell phone, videogames, television, computer and others?”

FAF numeric{from 0 to 3}
0=I do not have
]0,1]=1 or 2 days
]1,2]= 2 or 4 days
]2,3]= 4 or 5 days

Answering to the question:
“How often do you have physical activity?”



III- Analysis of Data, local analysis

TRY IT YOURSELF !

If you go to the Notebook section **“Local Data-visualization: studies of the different obesity type groups”**

You will be able to Try Two Functions :

```
def visualization_ID_Variable(obesity_variable,df):
def visualization_ID_Variable_Table(obesity_variable,df):
```

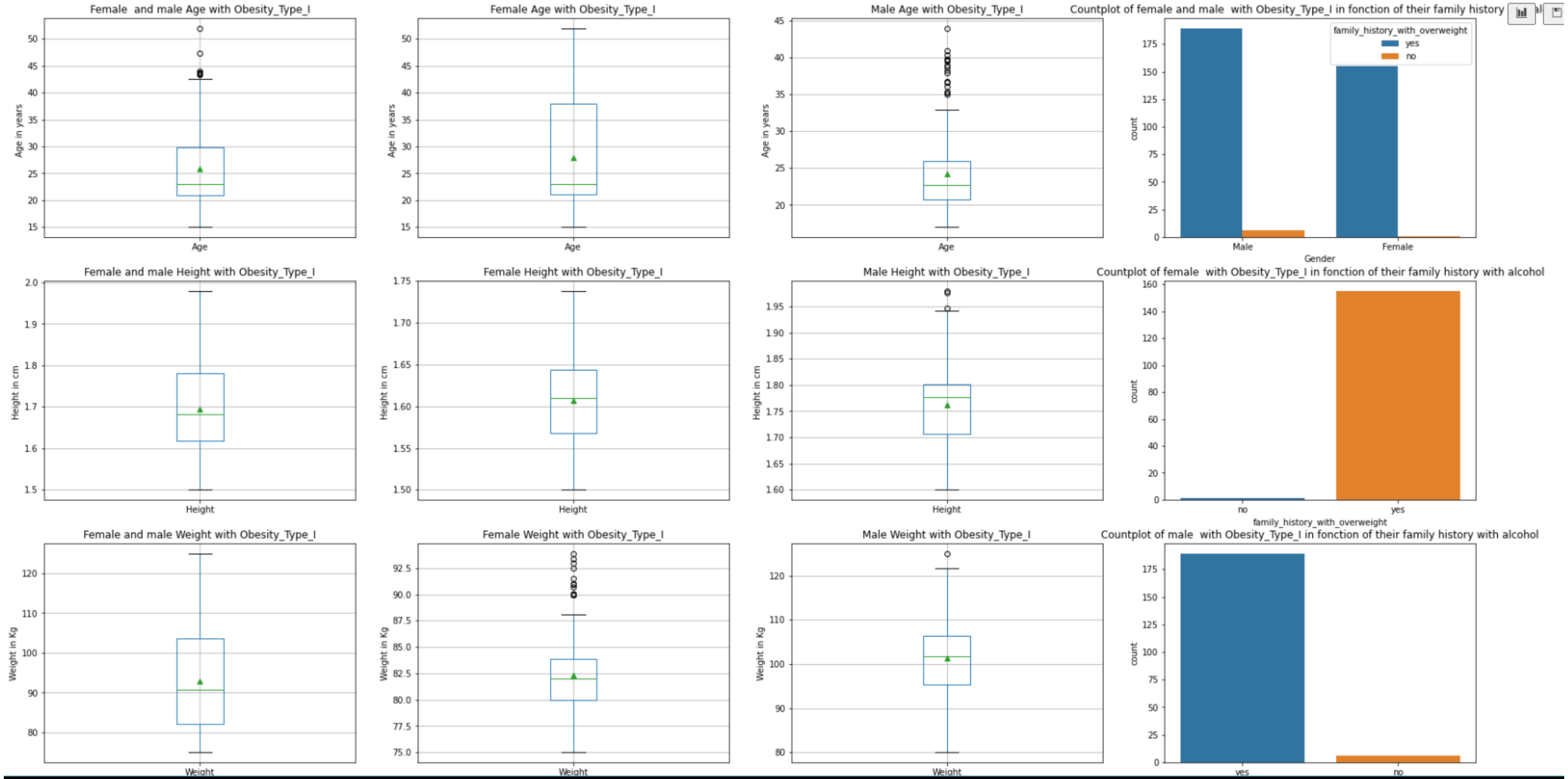
`These functions will allow you to visualize and understand the following variable:`

“ Gender, Age, Height, Weight, Family_history_with_overweight, Nobeyesdad”

III- Analysis of Data, local analysis

```
def visualization_ID_Variable(obesity_variable,df):
```

The visualisation you will see there is for the type of obesity Nobeyesdad=“Obesity_Type_I”



III- Analysis of Data, local analysis

```
def visualization_ID_Variable_Table(obesity_variable,df):
```

The visualisation you will see there is for the type of obesity Nobeyesdad=“Obesity_Type_I”

Informations about Males and Females in Insufficient_Weight:

	Age	Height	Weight
count	351.000000	351.000000	351.000000
mean	25.884941	1.693804	92.870198
std	7.755700	0.098414	11.485987
min	15.000000	1.500000	75.000000
25%	20.875385	1.617939	82.140613
50%	22.975526	1.681855	90.744965
75%	29.781305	1.780758	103.738394
max	52.000000	1.980000	125.000000

Informations about Females in Insufficient_Weight:

	Age	Height	Weight
count	156.000000	156.000000	156.000000
mean	27.894942	1.607389	82.293181
std	9.241103	0.049282	4.072846
min	15.000000	1.500000	75.000000
25%	21.017493	1.567600	80.000000
50%	23.000000	1.610070	82.000000
75%	37.957886	1.644261	83.872662
max	52.000000	1.738397	93.890682

Informations about Males in Insufficient_Weight:

	Age	Height	Weight
count	195.000000	195.000000	195.000000
mean	24.276940	1.762936	101.331813
std	5.868691	0.068735	7.926722
min	17.000000	1.600000	80.000000
25%	20.698872	1.706761	95.288163
50%	22.720449	1.777251	101.780099
75%	26.023932	1.801536	106.325128
max	44.000000	1.980000	125.000000

IV- Modeling *(manuel encoding)*

Variable sous forme qualitative	Variable Encodée	
Gender	male = 0	female = 1

Family_history_with_overweight **FAVC (Frequent consumption of high caloric food)** **SCC (Calorie's consumption monitoring)** **SMOKE**	**no = 0**	**yes = 1**

CAEC (Consumption of food between meals) **CALC (Consumption of alcohol)**	**no = 0**	**Sometimes = 1**	**Frequently = 2**	**Always = 3**

MTRANS (Transportation used)	**Automobile = 0**	**Motorbike = 1**	**Bike = 2**	**Public Transportation = 3**	**Walking = 4**

NObeyesdad (target variable)	**Insufficient Weight = 0**	**Normal Weight = 1**	**Overweight level I = 2**	**Overweight level II = 3**	**Obesity Type I = 4**	**Obesity type II = 5**	**Obesity Type III = 6.**

IV- Modeling *(encoding)*

Encoding by labels

We simply transform the type of the column to ‘category’, this assigns each variable of the column to a category. Example for the gender column: the categories ‘Male’ and ’Female’ are generated.
Once this is done, we just have to ad the code part ‘.cat.codes’ which transforms the qualitative categories into numerical categories.

```
df["Gender"] = df["Gender"].astype('category').cat.codes
```

Ordinal encoding

This type of encoding enables us to transform the qualitative data into numeric data in only one line of code. Indeed, we create a model OrdinalEncoder() than is directly applied to the columns of the dataframe we chose.

IV- Modeling *(test set and train set)*

Splitting the data into a training set and a test set :

```
X_train, X_test,Y_train,Y_test=train_test_split(X, Y, test_size= 0.33,random_state=4)
```

The shapes of the train and test set :

```
train set (1414, 16) test set (697, 16)
```

Once the data split, we scale the data, to get a more precise result

IV- Modeling *(Creation of the models)*

Creation of the Models

We decided to test different models to see which one was the more precise.

1- Neighbors Classifiers

K-Neighbors

2- SVM Classifiers

SVC (Support Vector Classification)

3- Grid-search on SVC model

For this model we chose parameters adapted to our SVC Mode

and obtained : `SVC(C=200, gamma=0.01)`

IV- Modeling*(Creation of the models)*

4- Bagging Classifiers

Bagging, RandomForest

5-Boosting Classifiers

AdaBoost, GradientBoosting, HistGradientBoosting

6-Voting Classifiiers

HardVoting, SoftVoting

IV- Modeling *(Testing of the models)*

Testing of the models
To test the model, we created a list of prediction of our test set that we compered to its already known results.
That is how we obtained the accuracy of each model.
We also created a confusion matrix for each model, to illustrate our results.

Example for the SVC Classification model :

Prediction thanks to the function ".predict()"

```
Y_pred_svc = model_svc.predict(X_test)
```

Final accuracy of the model :

```
Accuracy : 0.8723098995695839
```

Confusion matrix :

```
[[ 82   9   0   0   0   0   0]
 [  8  78  11   5   1   0   0]
 [  1   9  79   1   1   0   0]
 [  0  10  15  73   4   0   0]
 [  0   4   1   0  87   3   0]
 [  0   3   0   0   3  99   0]
 [  0   0   0   0   0   0 110]]
```

IV- Modeling *(Testing of the models)*

```
Accuracy with K-neighbors model : 81.205 %

Accuracy with SVC model: 87.231 %

Accuracy with Grid SVC model : 94.261 %

Accuracy with Bagging model : 93.974 %

Accuracy with RandomForest model : 95.122 %

Accuracy with AdaBoost model : 27.834 %

Accuracy with GradientBoosting model : 94.978 %

Accuracy with HistGradientBoosting model : 96.844 %
```

Using VotingClassifier we selected only the classifier with a score above 0,50 (We deleted the AdaBoostClassifier).
We tried both Hard and soft VotingClassifier.

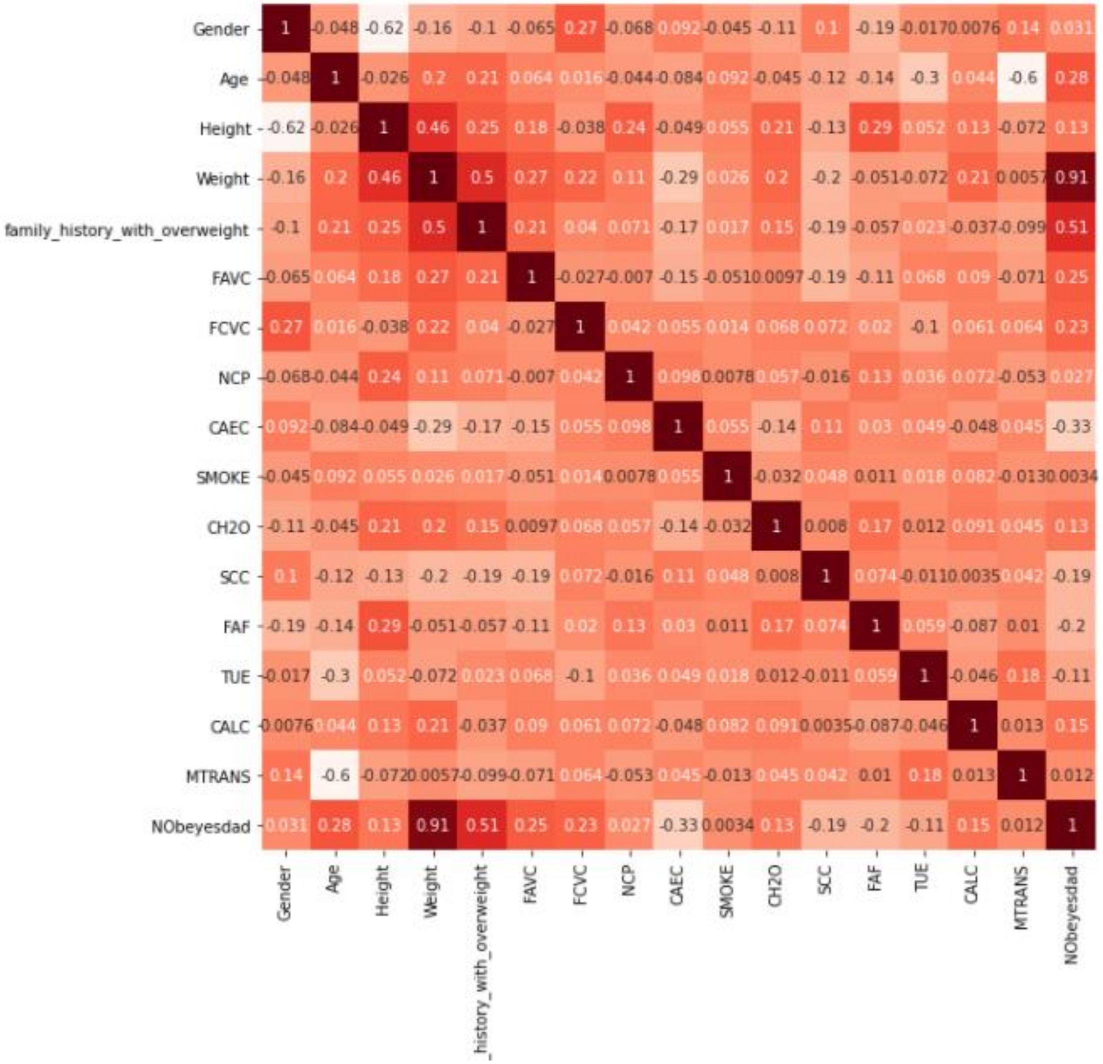
```
Accuracy with VotingHard model : 96.413 %

Accuracy with VotingSoft model : 96.844 %
```

V- Modeling with reduced features

(To go futher)

We can look at the correlation matrix of the encoded data to see which columns/variables are the more correlated to our target variable.

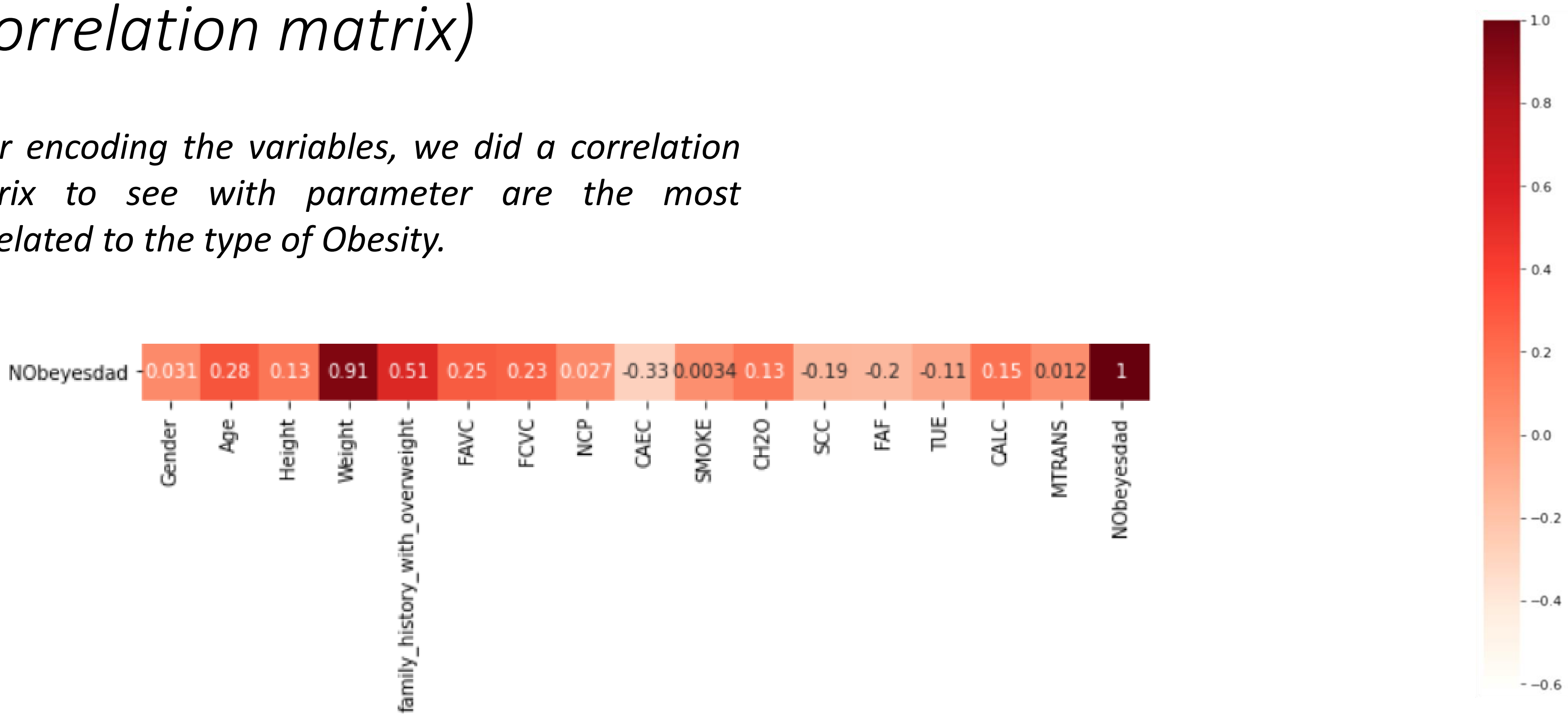
It will allow us to see what variable are really useful for our classification.



V- Modeling with reduced features

(Correlation matrix)

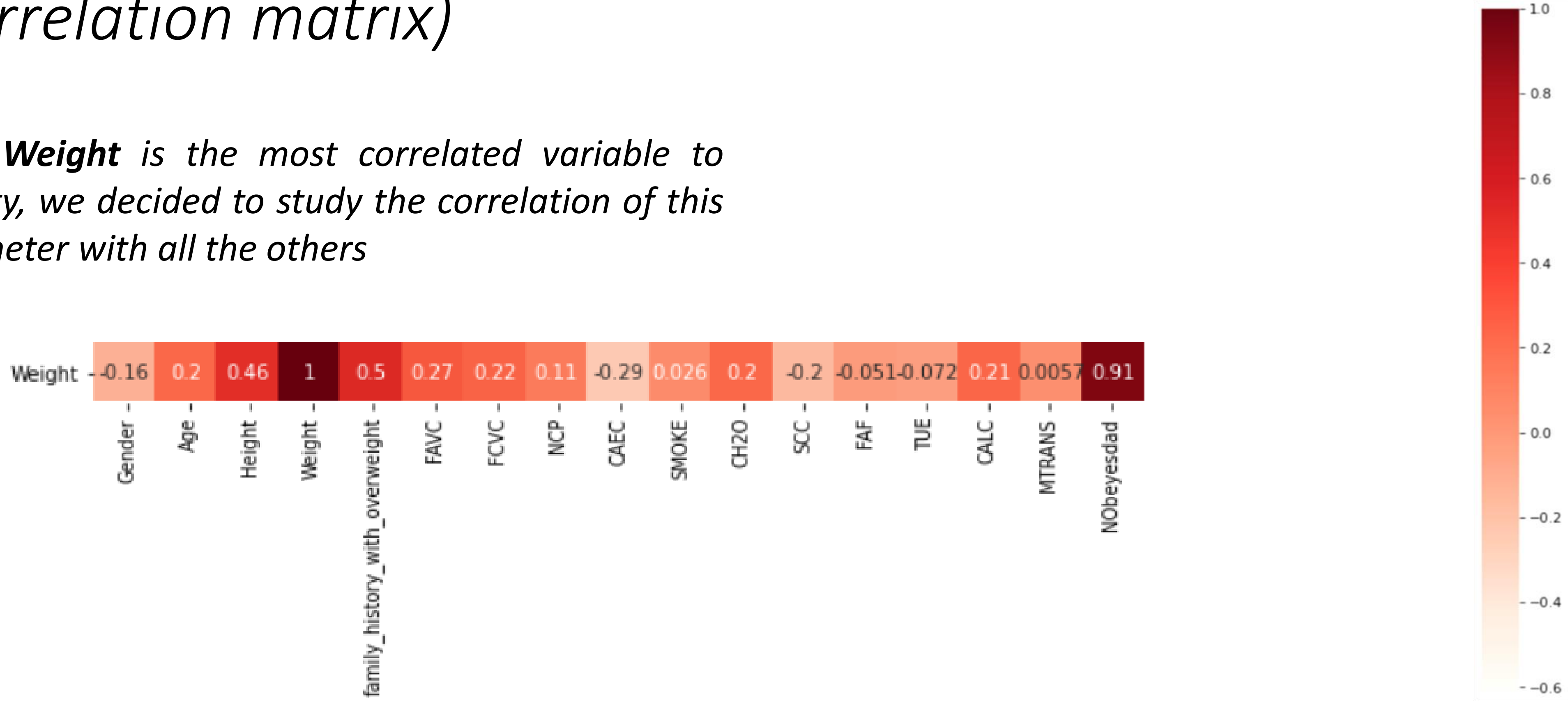
After encoding the variables, we did a correlation matrix to see with parameter are the most correlated to the type of Obesity.



We can see that the most correlated variable is, unsurprisingly, *the Weight. But we can also see that the family history with overweight is highly correlated to the type of obesity.*
The Age, Frequency of consumption of vegetables (FCVC) and if the person has a high consumption of high caloric food (FAVC) are also highly correlated parameters.

V- Modeling with reduced features
(Correlation matrix)

*Since **Weight** is the most correlated variable to Obesity, we decided to study the correlation of this parameter with all the others*



We can see that the Weight has also a high correlation with family history with overweight, but also with the Height.
As for Obesity, the correlation between Weight and the parameters Age, FAVC and FCVC is high, but we can also see a high correlation with new parameters as daily water consumption CH2O and consumption of alcohol (CALC)

V- Modeling with reduced features

(Drop of the unusfull data)

Looking at the correlation matrix, we decided to drop the variables with a correlation with our target variable under 0,15. This would enable us to use only very correlated variables and be more precise in our prediction.

Those are the columns we dropped :

```
column = ["Gender","NCP","CAEC","SMOKE","SCC","FAF","TUE","MTRANS","CH2O"]
```

V- Modeling with reduced features
(Testing of the reduced models)

Example for the SVC Classification model :

Prediction thanks to the function ".predict()"

```
Y_pred_grid_svc_reduiced = grid_svc_reduiced.predict(X_test_reduiced)
```

Final accuracy of the model :

```
Accuracy : 0.8823529411764706
```

Confusion matrix :

```
[[ 86   5   0   0   0   0   0]
 [ 11  71  18   3   0   0   0]
 [  0   4  85   2   0   0   0]
 [  0   6  20  71   5   0   0]
 [  0   1   2   0  91   0   1]
 [  0   1   0   1   1 101   1]
 [  0   0   0   0   0   0 110]]
```

V- Modeling with reduced features

(Results)

```
Accuracy with reduiced SVC model: 96.27 %

Accuracy with reduiced Grid SVC model : 96.27 %
```

```
Accuracy with VotingHard model : 96.413 %

Accuracy with VotingSoft model : 96.7 %
```

We can see that the reduced models (with only significant features) have globally a higher accuracy than the models with all features.

VI- How to use our FLASK application ?

- ON our GitHub You will find a READ_ME_FLASK.docx
- You will find all the explanation here.

127.0.0.1:5000/?Age=22&Height=1.80&Weight=70&family_history_with_overweight=0&FAVC=1&FCVC=1&CALC=1

PREDICTION: Normal_Weight

Accuracy model of: 96.7 %

Age=22.0 on good Format: True
Height=1.8 on good Format: True
Weight=70.0 on good Format True
family_history_with_overweight=0 on good Format: True
FAVC=1 on good Format: True
FCVC=1 on good Format: True
CALC=1 on good Format: True

VI- Conclusion

Our objective: get a Classifier with at least an **accuracy of 90%.**

Our result: Classifier with an **accuracy of 97%.**

We obtain our a **SoftVotingClassifier**

- using the the following classifier : Bagging, RandomForest, GradientBoosting, HistGradientBoosting, SVC (optimize with Grid-search)

- Using the features : Age, Weight, Height, Family History With Overweight, FAVC, FCVC and CALC

VI- Conclusion
(If we had to start the Project again)

What we did Well ?

The Analysis of the dataset.

Realization of a flask App (using few html)

Our collaboration (use of github, division of tasks)

Selection of the most significant features to improve our models.

How to proceeded to find a model with a good accuracy ?

Or what we should have done if we didn’t found conclusive first results.

For a list of features:

- Select a list of classifier.
- For each classifier :
  - applied a grid Search to find the best parameters and create new optimize classifier.
  - Keep only optimize classifier having an accuracy above 50%.
- Proceed to a Voting Classifier.