ESILV ENGINEERING SCHOOL DE VINCI PARIS

DIA : DATA & ARTIFICIAL INTELLIGENCE

PYTHON FOR DATA ANALYSIS

PREDICTION OF OBESITY

01

THE DATASET

Presentation of the dataset
Variables description
First observations and remarks

02

INITIAL APPROACH

Data Visualisation
Data pre-processing

03

MODELS & PREDICTIONS

Models presentation
Results and analyse
Difficulties

04

FINAL PRODUCT & CONCLUSION

ERWAN BOURHIS
DIEGO MATEOS

01) THE DATASET – Presentation of the dataset

The dataset we had to work with is called:

" Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico "

The data contains 17 attributes and 2111 records

The records are labeled with the class variable NObesity (Obesity Level), which allows classification of the data using the values of :

- Insufficient Weight
- Normal Weight
- Overweight Level I
- Overweight Level II
- Obesity Type I
- Obesity Type II
- Obesity Type III

1/4 of the data was collected from a survey
3/4 of the data was generated synthetically using the Weka tool and the SMOTE filter (oversampling)

01) THE DATASET – Dataset description

Category	Feature Name	Description	Variable Type
Target Variable	NObesity	Based on BMI	Categorical
Eating Habits	FAVC	Frequent consumption of high caloric food	Categorical
Eating Habits	FCVC	Frequency of consumption of vegetables	Ordinal
Eating Habits	NCP	Number of main meals	Ordinal
Eating Habits	CAEC	Consumption of food between meals	Ordinal
Eating Habits	CH20	Consumption of water daily	Ordinal
Eating Habits	CALC	Consumption of alcohol	Ordinal
Physical Conditioning	SCC	Calories consumption monitoring	Categorical
Physical Conditioning	FAF	Pysical activity frequency	Ordinal
Physical Conditioning	TUE	Time using technology devices	Ordinal
Physical Conditioning	MTRANS	Transportation used	Categorical

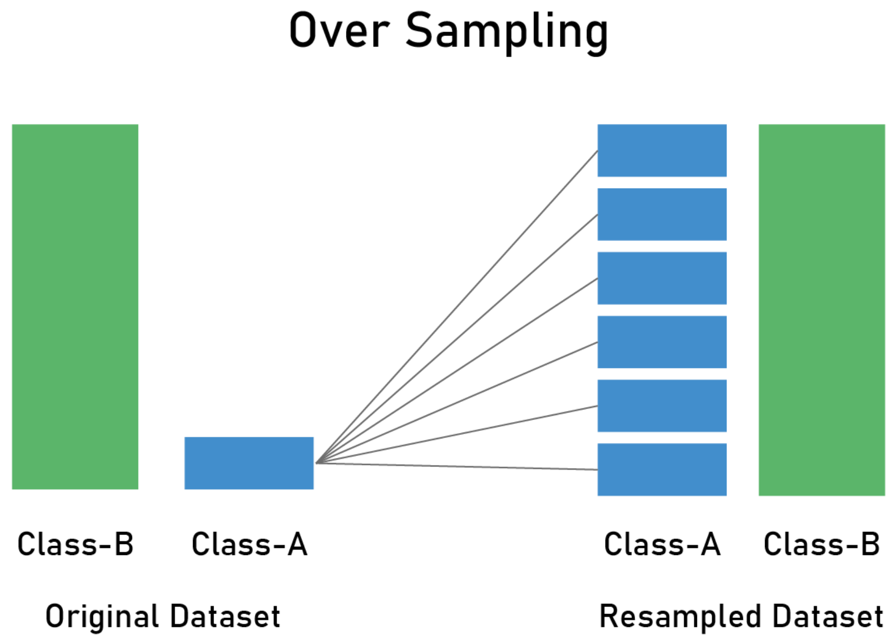
01) THE DATASET – Dataset description

Category	Feature Name	Description	Variable Type
Physical Conditioning	SMOKE	Smokes Yes or No	Categorical
Responder Charateristics	Family History with Overweight	Yes or No	Categorical
Responder Charateristics	Gender	Gender is Male or Female	Categorical
Responder Charateristics	Age	Age in years	Integer
Responder Charateristics	Height	Height in meters	Float
Responder Charateristics	Weight	Weight in kilograms	Float

01) THE DATASET – Presentation of the dataset

The first thing we noticed is that the majority of the data are synthetically generated. This is due to the fact that the class of the target variable were not balanced.

This detail is important because we know by experience that sometimes oversampling can generate overfitting issues. Thus, we have to keep this in mind for the rest of our study.

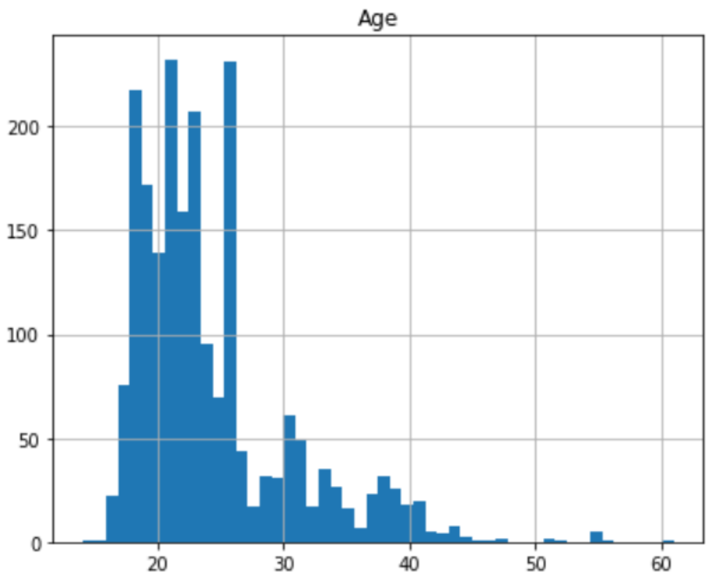


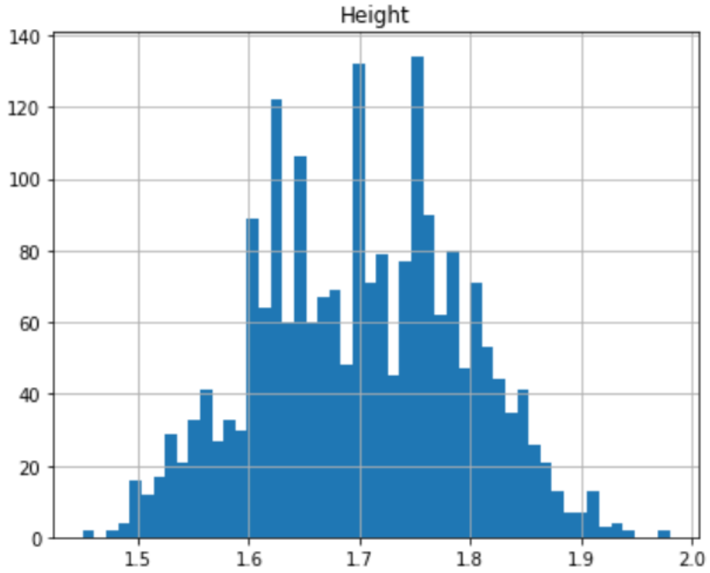
We've also noticed that our data are quite clean, there are neither duplicates nor empty fields

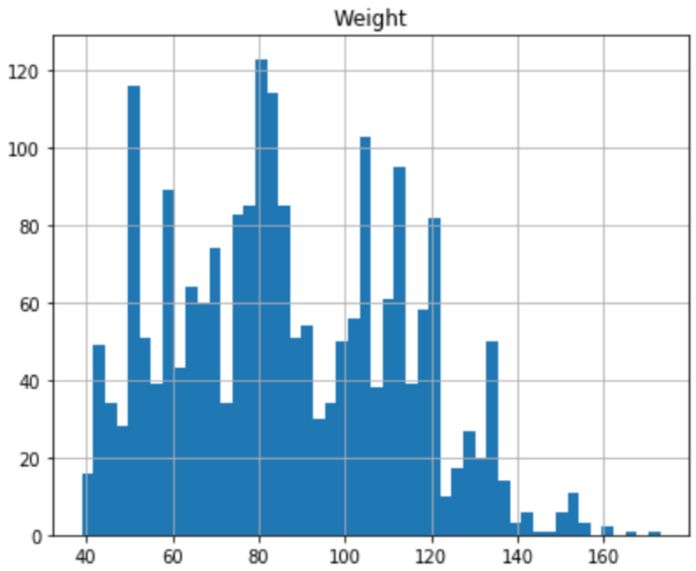
After some pre-processing that will be presented later, we have a dataset composed of:

- 13 categorical features
- 3 numeric features
- 1 categorical target

02) INITIAL APPROACH – Data Vizualisations



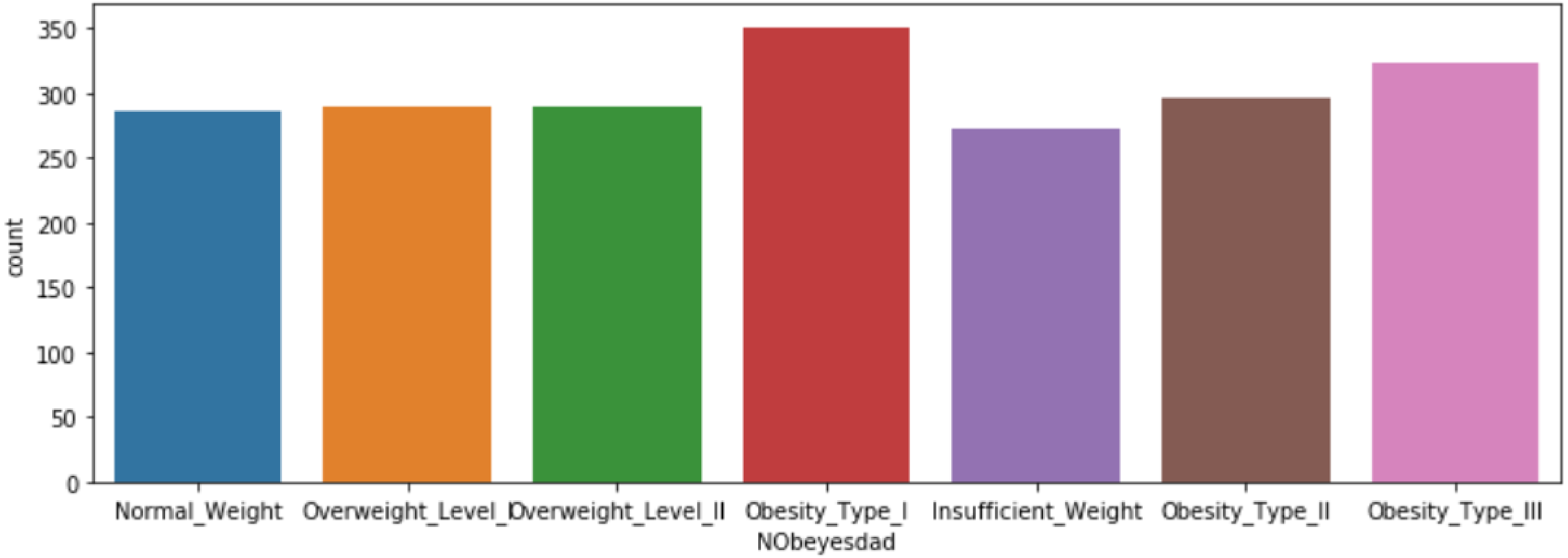




Some Vizualisations to observe the repartition of the numerical features and of the target.

We can observe that the repartition of the height of the studied people is gaussian.

We can also verify that the target variable is balanced

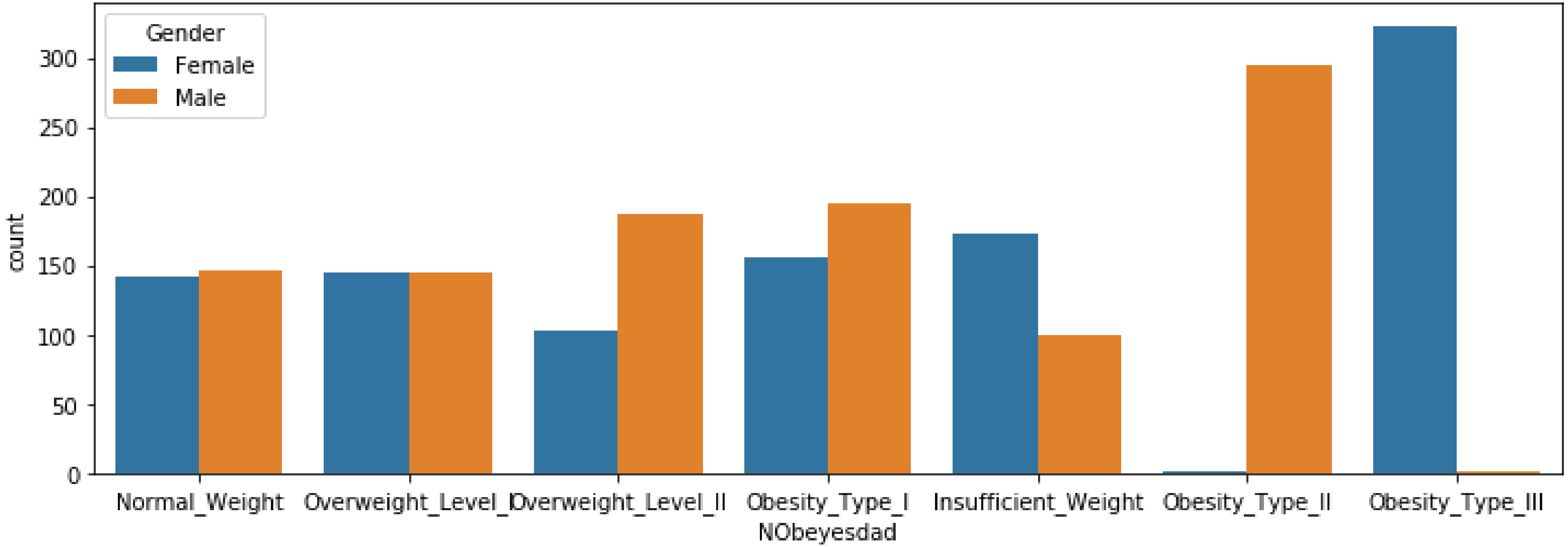


02) INITIAL APPROACH – Data Vizualisations

Starting from the vizualisations of the last slide, we splitted the data by gender. We noticed here something very interesting. There's clearly a problem with the the *Obesity_Type_II* and *Obesity_Type_III* classes.

In our data, all the people who are obese(type II) are males and the obese(type III) are females.
In our opinion this is due to the oversampling done to balance the classes. We think that in the suvey, there was probably only males in type II and females in type III. Thus, when they did the SMOTE oversampling, all the data generated "copied" the existing and produced only males for type II and female for type III.

That's a big issue because it introduces a bias in our study that will surely impact our precision in real cases.



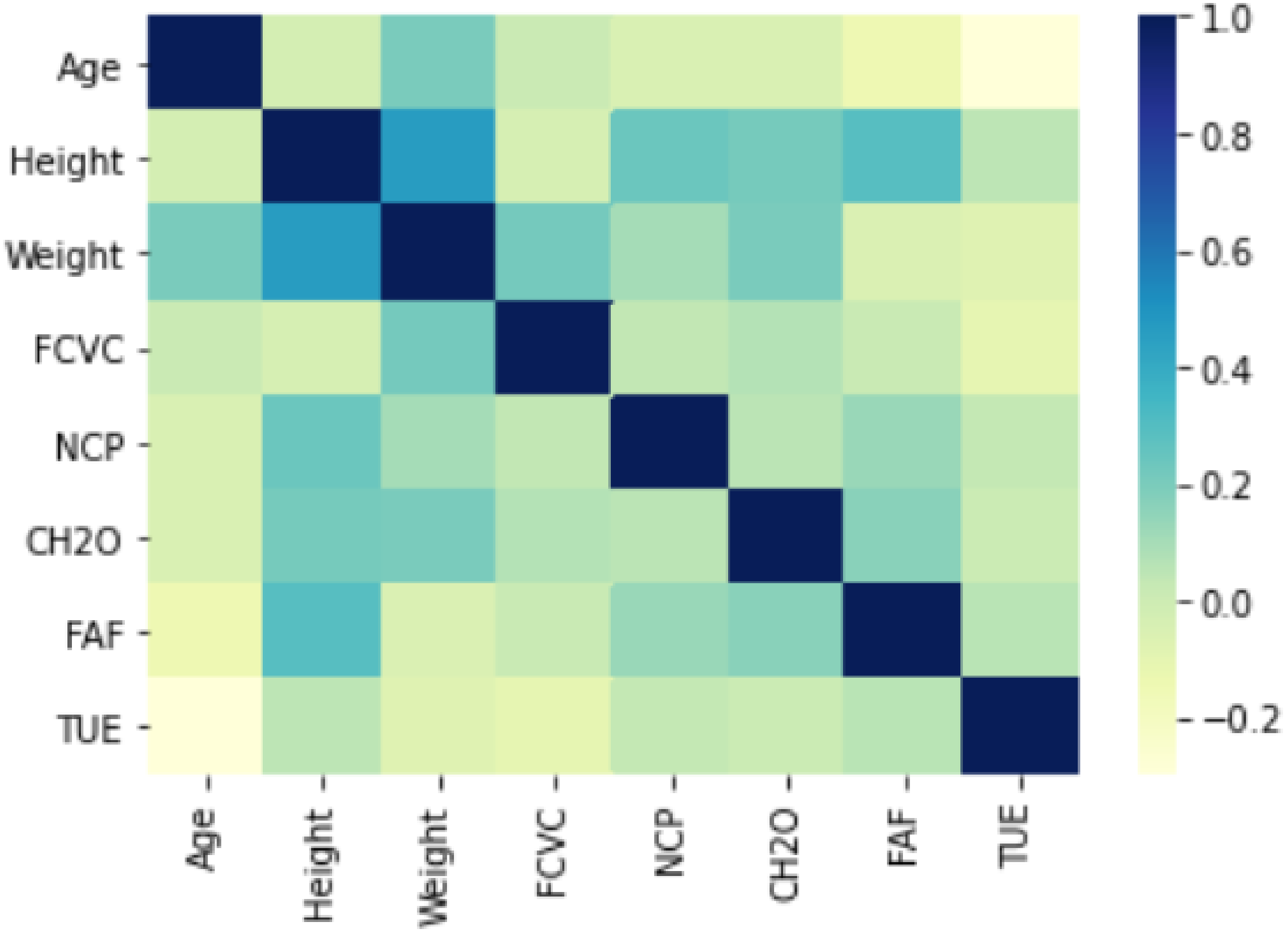
02) INITIAL APPROACH – Data Vizualisations

We tried to plot a correlation matrix of the features and we observed that there are few correlations between variables except for height and weight.
It seems quite logic that these two variables are highly correlated (a tall person is in average heavier than a short person).

This trivial observation made us remember one central issue in our project : The target variable is in fact calculated with a formula using the height and the weight.

$$Mass\ body\ index = \frac{Weight}{height*height}$$

This is why we had to remove these two features from our dataset.



02) INITIAL APPROACH – Data pre-processing

Now that we know a little better our dataset, we have to go deeper into the reflexion :

Before starting to code the differents implemtations of the models, we thought to the data pre-processing work we had to do.

First, we saw that some of the categorical features had float values. We supposed that these values were generated by the SMOTE oversampling so we decided to use the round() function to fix this issue

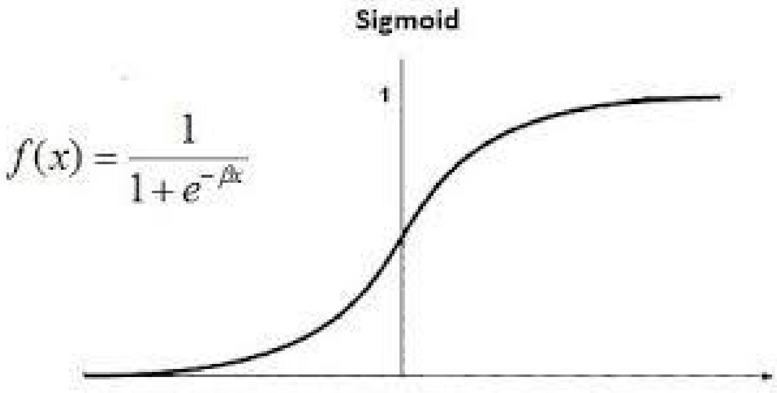
Then, we decided to dummify (also called onehotencoding) the categorical features because machine learning algorithms cannot operate on label data directly

Of course, as usual, we had to scale the features and split into train and test dataset our data before training the models

Finally, before choosing the models to test, it's important to remember that we should find a type of model which is efficient with limited amount of data.

03 – MODELES & PREDICTIONS

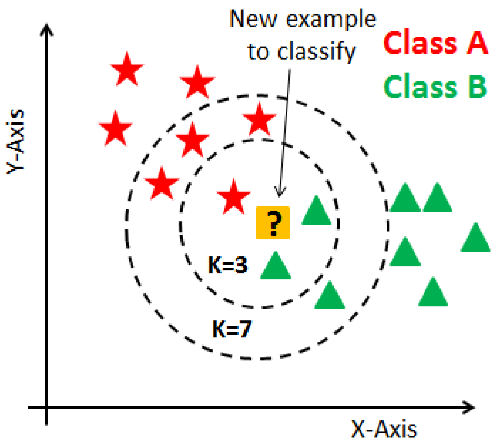
Here are the models we decided to test. First of all, we started by the "simpler" one for classification problems: The logistic regression



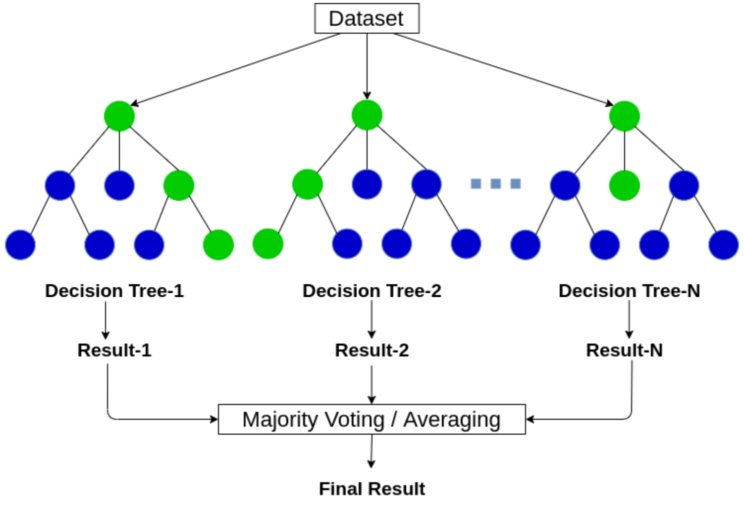
Then, we tried some other models known for their effectiveness in classification :

XGBoost

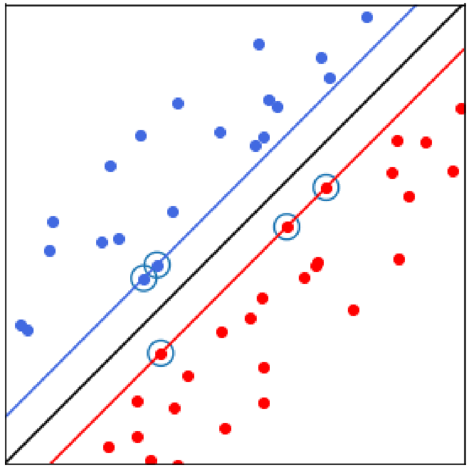
KNN



Random Forest



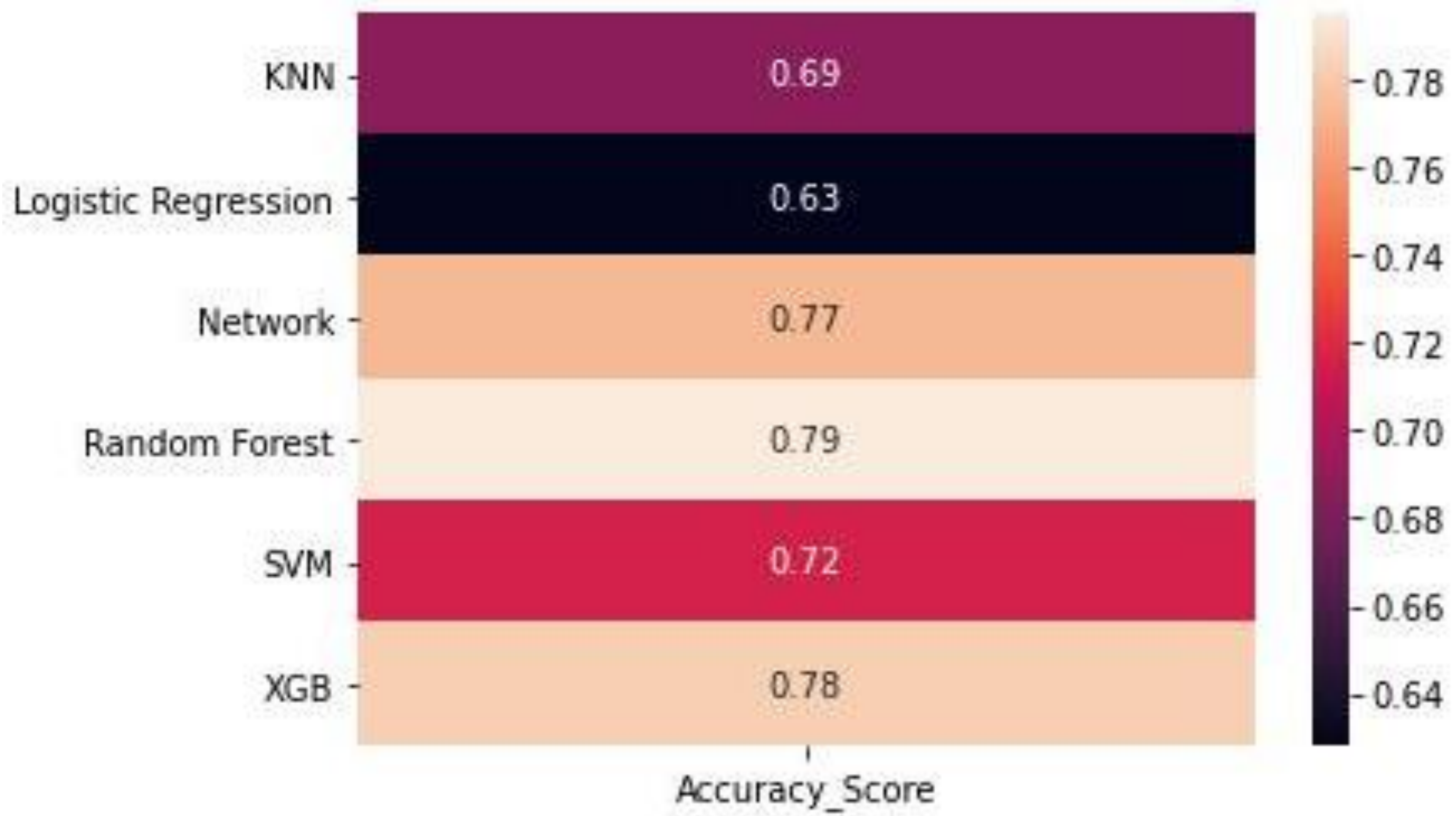
Support Vector Machine



03 – MODELES & PREDICTIONS

These are the results we managed to get with the differents models tested.

The model getting the best accuracy is Random Forest followed closely by XGB.



03 – MODELES & PREDICTIONS

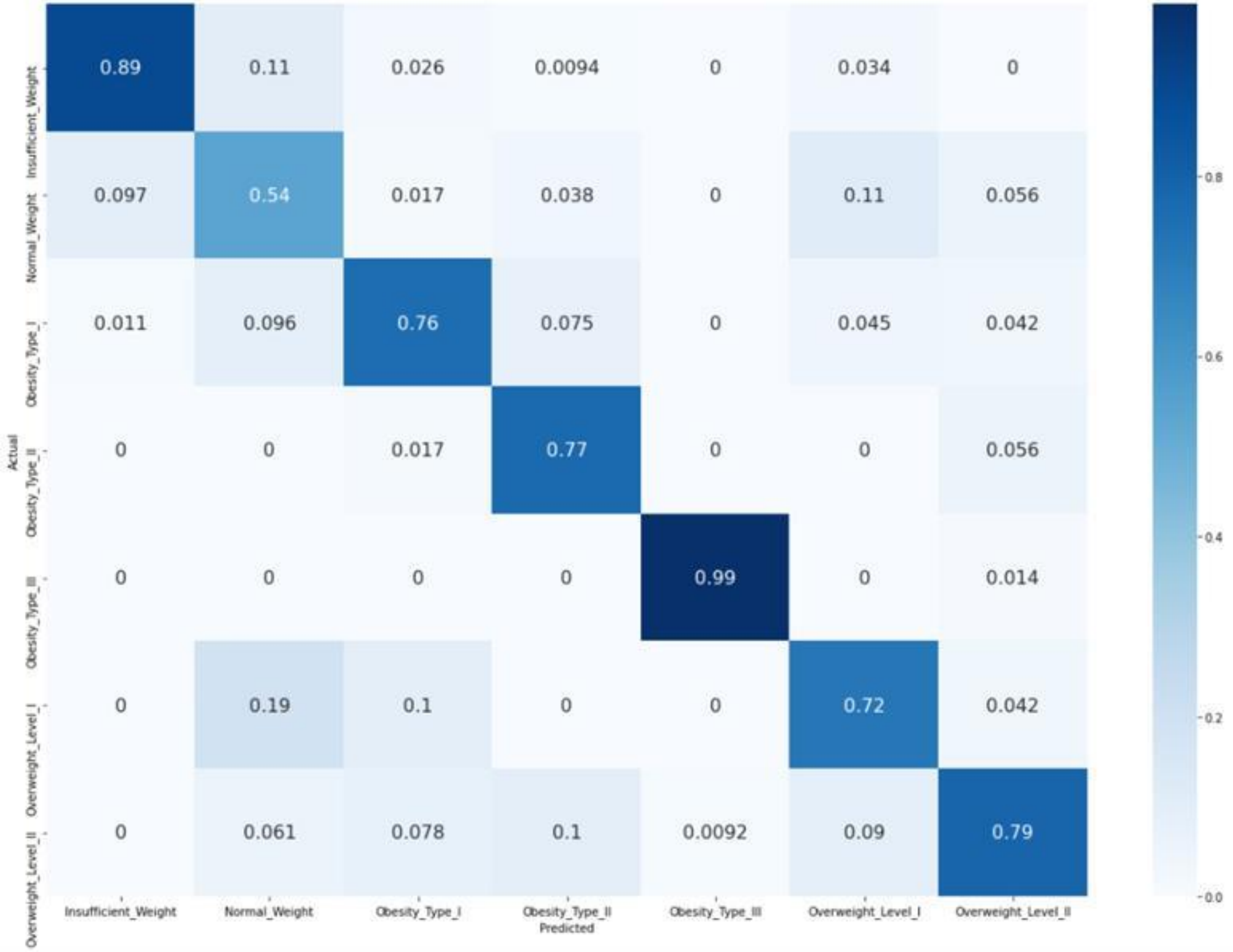
Let's look closely to the results of our best model : Random Forest

There are many thing intersting in this confusion matrix.

First we can see that we are very bad at predicting Normal Weight and, on the contrary, very good at predicting Obesity Type III and insufficient weight.

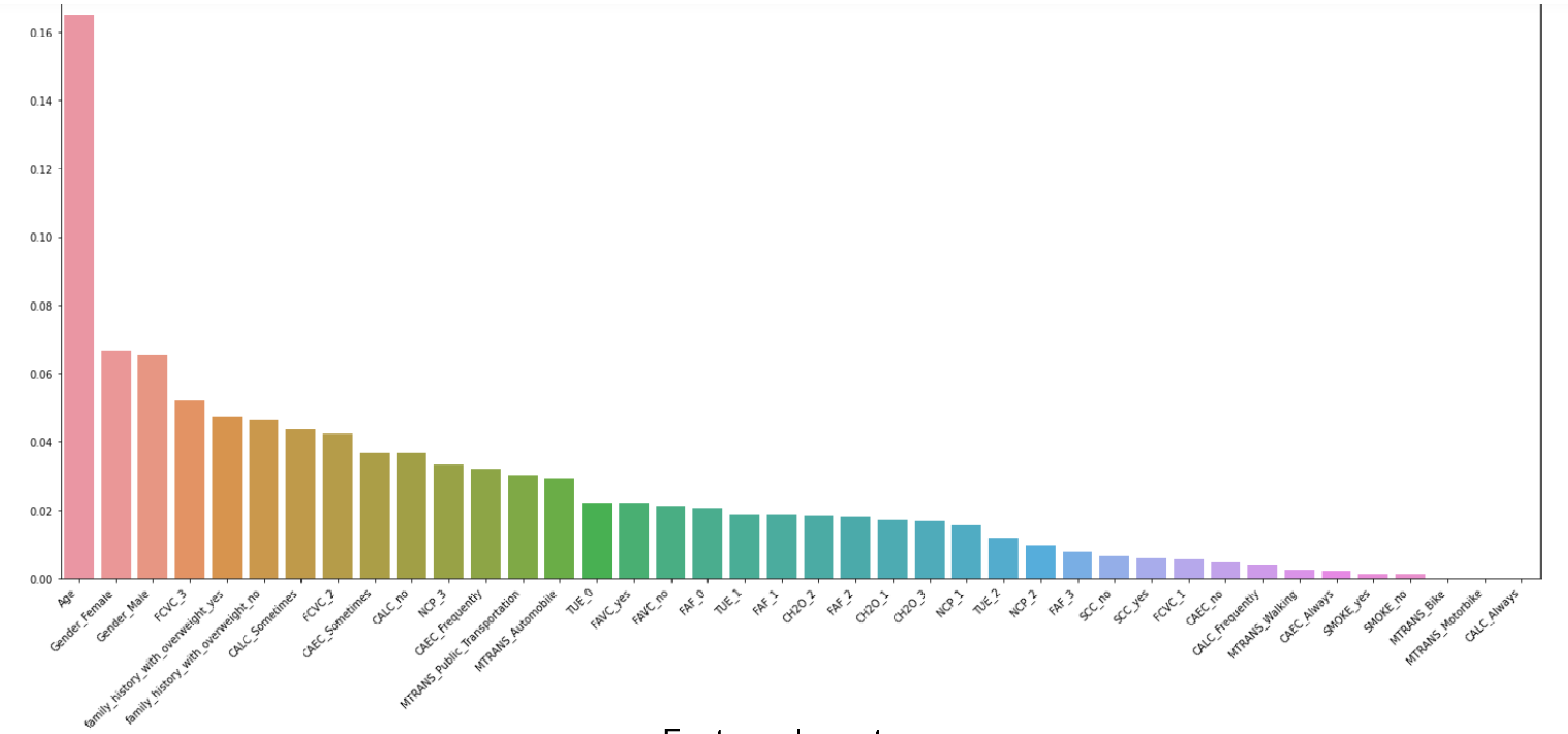
We can maybe explain this by the fact that it is easier to detect extreme values due to extreme behaviors in the people habits.

Indeed, if we look to the false prediction of normal weight people, our model often predict the closer classes (Normal_Weight is close to Overweight I and far from obesity III)



	Insufficient_Weight	Normal_Weight	Obesity_Type_I	Obesity_Type_II	Obesity_Type_III	Overweight_Level_I	Overweight_Level_II
Insufficient_Weight	0.89	0.11	0.026	0.0094	0	0.034	0
Normal_Weight	0.097	0.54	0.017	0.038	0	0.11	0.056
Obesity_Type_I	0.011	0.096	0.76	0.075	0	0.045	0.042
Obesity_Type_II	0	0	0.017	0.77	0	0	0.056
Obesity_Type_III	0	0	0	0	0.99	0	0.014
Overweight_Level_I	0	0.19	0.1	0	0	0.72	0.042
Overweight_Level_II	0	0.061	0.078	0.1	0.0092	0.09	0.79

03 – MODELES & PREDICTIONS



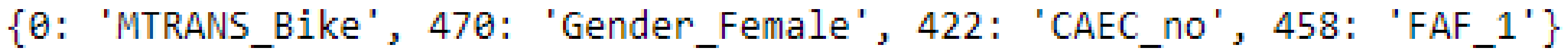
Features Importances

03 – MODELES & PREDICTIONS

Difficulties :

- After getting all the results off the models, we tried to do some tuning. As you will see in our code, the hyperparameters tuning didn't improve our results.

- We also tried to do some features selection. First with the feature_selection tool from scikitlearn librairy the results were very bad because it removes to many features.
Then, we created a function that compute random forest model but each time it removes one variable among all. Doing that 1000 times, we have :



```
{0: 'MTRANS_Bike', 470: 'Gender_Female', 422: 'CAEC_no', 458: 'FAF_1'}
```

- As you can see, removing those 3 values increased our model accuracy but less than 500 time on 1000. That's why we kept all features because removing some features improves our result less than 1 time on 2.

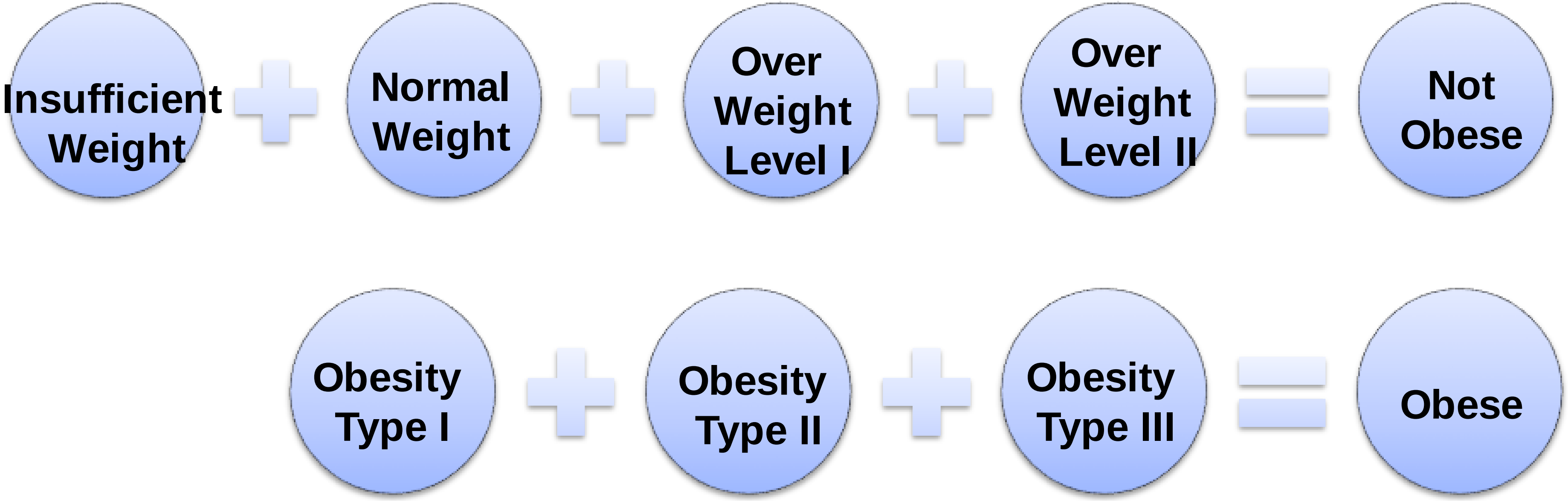
- We didn't use pipeline because we have developped alone our "one_encode_data" function, we couldn't use pipeline.fit() with it. We used the pickle librairy to save the RF model and the scaler and use it in the API.

03 – MODELES & PREDICTIONS

Given the confusion matrix we have and the precision we are able to get, we decided to try something else. Maybe we could get a better score by reducing the number of target classes.

Our idea is to merge some of the existing class to get only two classes (Obese or not Obese)
Of course, the result will be a little less interesting, by doing that we will be much less specific about the obesity level of the user. But at least, we maybe can manage to be more precise in our predictions.

To do that, we splitted the target as follows :



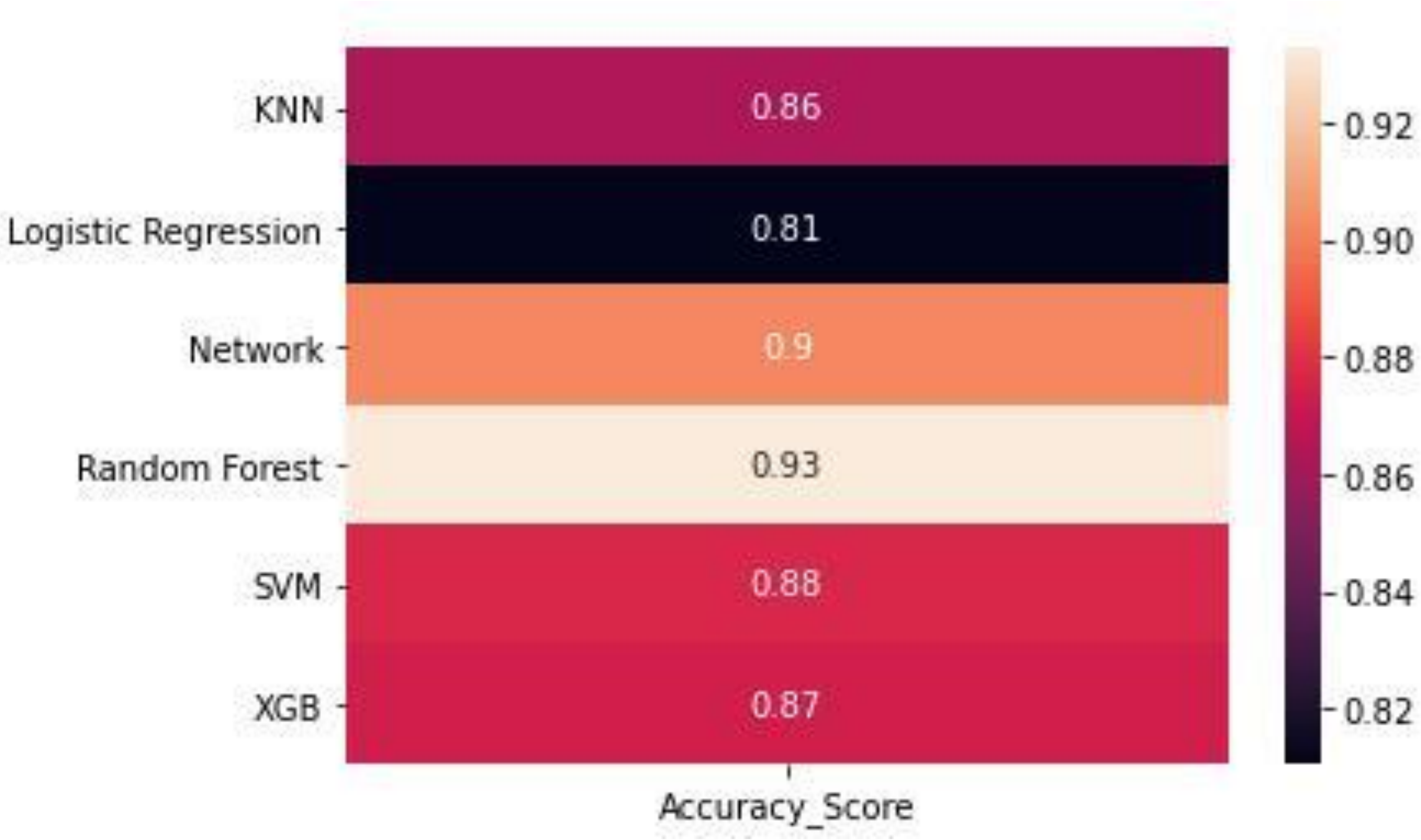
Insufficient Weight + Normal Weight + Over Weight Level I + Over Weight Level II = Not Obese

Obesity Type I + Obesity Type II + Obesity Type III = Obese

03 – MODELES & PREDICTIONS

Here are the results we get once we merged the target classes into Obese / Not obese.

Random Forest is still the more accurate model but this time, our Neural Network performed better than XGB or SVM



04) FINAL PRODUCT & CONCLUSION – flask application

Our final goal is to be able to predict the obesity level of someone on an API.

Even if we have better reults on predicting obesity/not obesity of someone. In the API we will try to predict the obesity level of a person because it’s more significant.

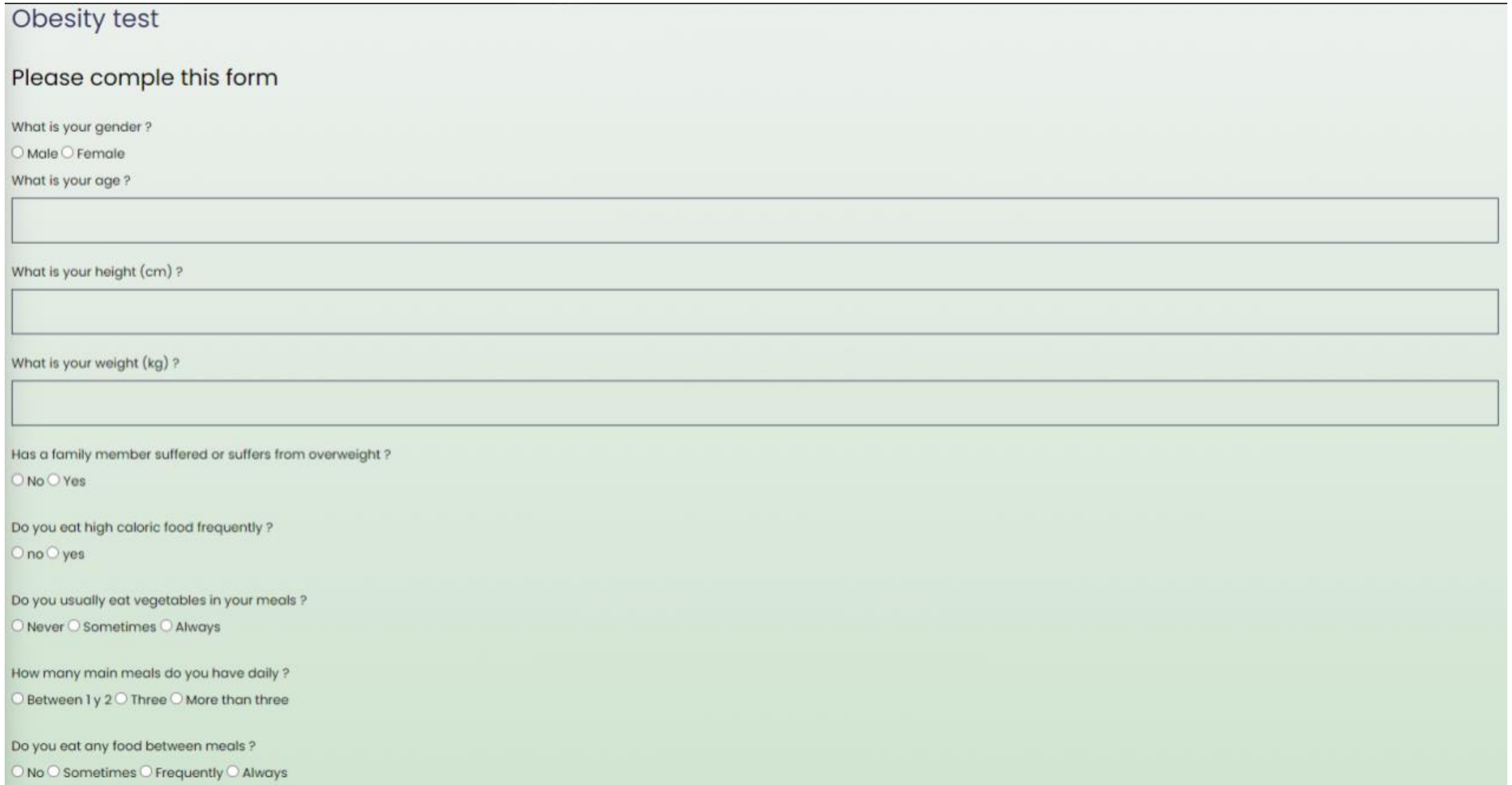
To do so, we developed a Flask Application so the user can fill a form answering to the needed questions. Once the form is filled, we call our model and do our prediction. Moreover, we display the real level of obesity of the person by computing it with his height and weight.

When doing our form, we had some doubts about the order of the responses. In the dataset some fields such as FAF in which the answers are values (0, 1, 2). The problem is that in the dataset documentation there was no information about the mapping between theses values and the corresponding answers. After observing the data, it seems that it is organized by increasing order so that's the logic we choosed.

04) FINAL PRODUCT & CONCLUSION– flask application

We also wanted to have a better web site so we enhanced it with some HTML+CSS.

That way, our final product is a Flask Application composed of an home page presenting the project. A page with some data vizualisations of our dataset and finally the page with the form to be filled by the user.



Obesity test

Please comple this form

What is your gender ?

○ Male ○ Female

What is your age ?

What is your height (cm) ?

What is your weight (kg) ?

Has a family member suffered or suffers from overweight ?

○ No ○ Yes

Do you eat high caloric food frequently ?

○ no ○ yes

Do you usually eat vegetables in your meals ?

○ Never ○ Sometimes ○ Always

How many main meals do you have daily ?

○ Between 1 y 2 ○ Three ○ More than three

Do you eat any food between meals ?

○ No ○ Sometimes ○ Frequently ○ Always

Screen shot of the forms

04) FINAL PRODUCT & CONCLUSION

Conclusion :

To conclude this study, we finally manage to get a 79% of accuracy by predicting the level of obesity and 93% of accuracy by predicting if a person is obese or not.

We could have better results with more data. In fact, the sample is too small (only 500 people) and there are only 10 basic questions on our habbits. To improve the study, there should be more questions with more detailed answers like what is done on *https://etude-nutrinet-sante.fr/* which is a website where you can answer really detailed forms and some Searchers/Statisticians analyse the Data to make more advanced studies.
(One of us made a 2month internship with them, that's why we know this website)

We enjoyed working on these data because it's a concrete use case of data science and about a subject on which everyone has knowledge.