ESILV ENGINEERING SCHOOL DE VINCI PARIS

Estimation of obesity levels based on eating habits and physical condition Data Set

Python for Data Analysis

Adam Dahmoul

Ferdinand Valancogne

DIA 1

Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CAL
Female	21.000000	1.620000	64.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	n
Female	21.000000	1.520000	56.000000	yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometime
Male	23.000000	1.800000	77.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequent
Male	27.000000	1.800000	87.000000	no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequent
Male	22.000000	1.780000	89.800000	no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometime
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Female	20.976842	1.710730	131.408528	yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometime
Female	21.982942	1.748584	133.742943	yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometime
Female	22.524036	1.752206	133.689352	yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometime
Female	24.361936	1.739450	133.346641	yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometime
Female	23.664709	1.738836	133.472641	yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometime

- Nous avons tout d'abord commencé par analyser le dataset. On a ici 17 colonnes et 2111 lignes. Il n'y a pas de valeurs manquantes.

```
df.describe(include = 'object')
```

	Gender	family_history_with_overweight	FAVC	CAEC	SMOKE	SCC	CALC	MTRANS	NObeyesdad
count	2111	2111	2111	2111	2111	2111	2111	2111	2111
unique	2	2	2	4	2	2	4	5	7
top	Male	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_I
freq	1068	1726	1866	1765	2067	2015	1401	1580	351

```
df.describe()
```

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000
mean	24.312600	1.701677	86.586058	2.419043	2.685628	2.008011	1.010298	0.657866
std	6.345968	0.093305	26.191172	0.533927	0.778039	0.612953	0.850592	0.608927
min	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	19.947192	1.630000	65.473343	2.000000	2.658738	1.584812	0.124505	0.000000
50%	22.777890	1.700499	83.000000	2.385502	3.000000	2.000000	1.000000	0.625350
75%	26.000000	1.768464	107.430682	3.000000	3.000000	2.477420	1.666678	1.000000
max	61.000000	1.980000	173.000000	3.000000	4.000000	3.000000	3.000000	2.000000

```
print(df.isnull().sum())
```

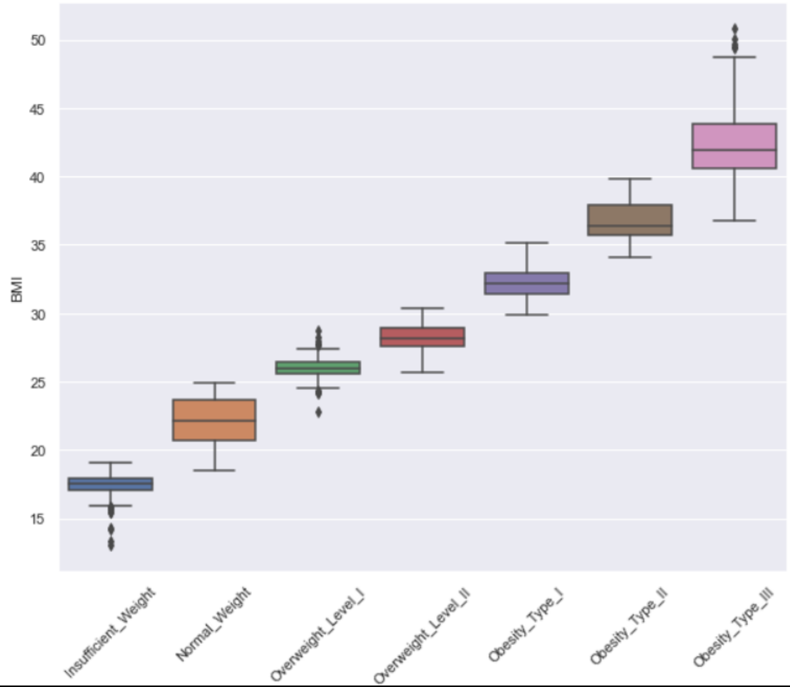
```
Gender                            0
Age                               0
Height                            0
Weight                            0
family_history_with_overweight    0
FAVC                              0
FCVC                              0
NCP                               0
CAEC                              0
SMOKE                             0
CH2O                              0
SCC                               0
FAF                               0
TUE                               0
CALC                              0
MTRANS                            0
NObeyesdad                        0
dtype: int64
```

```python
df['BMI'] = df['Weight']/(df['Height']**2)
```

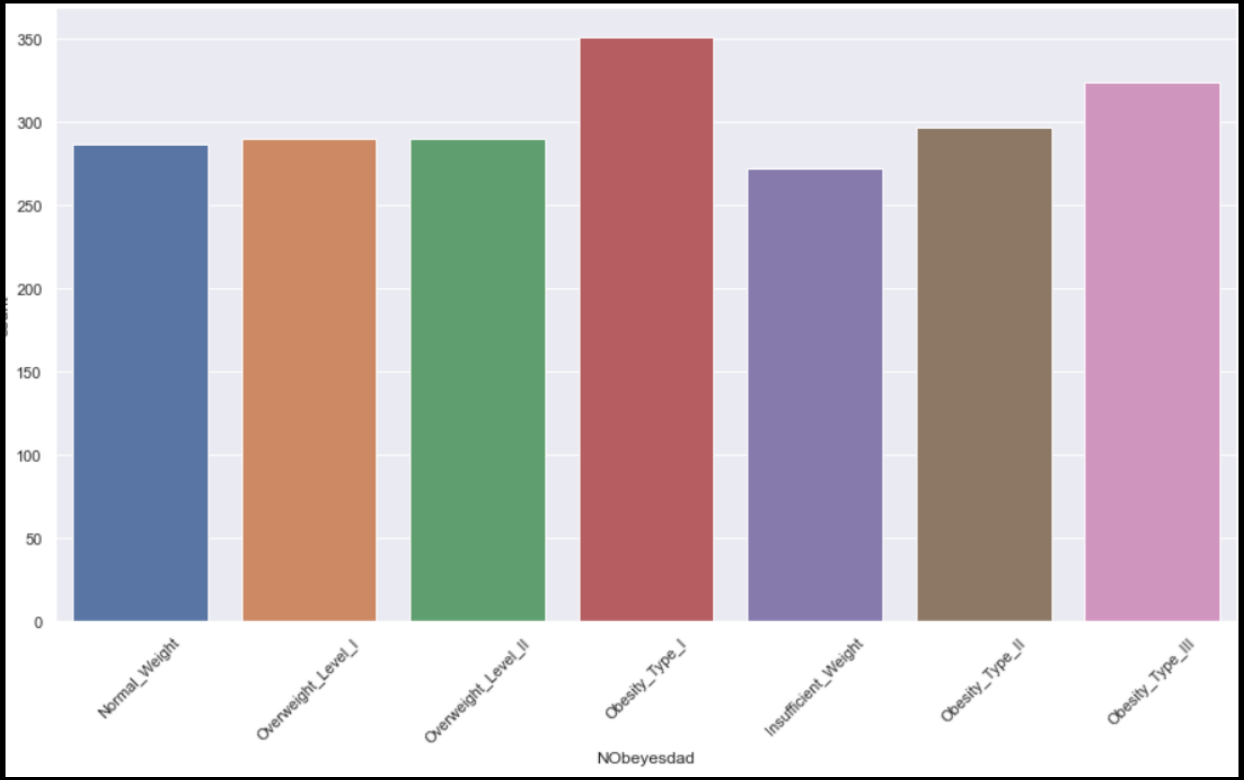
- After studying the dataset, we decided to add a column containing the BMI of each individual. This will help us visualize some information.
- This choice is due to the fact that NObeyesdad is given according to the BMI. We still wanted to check that.

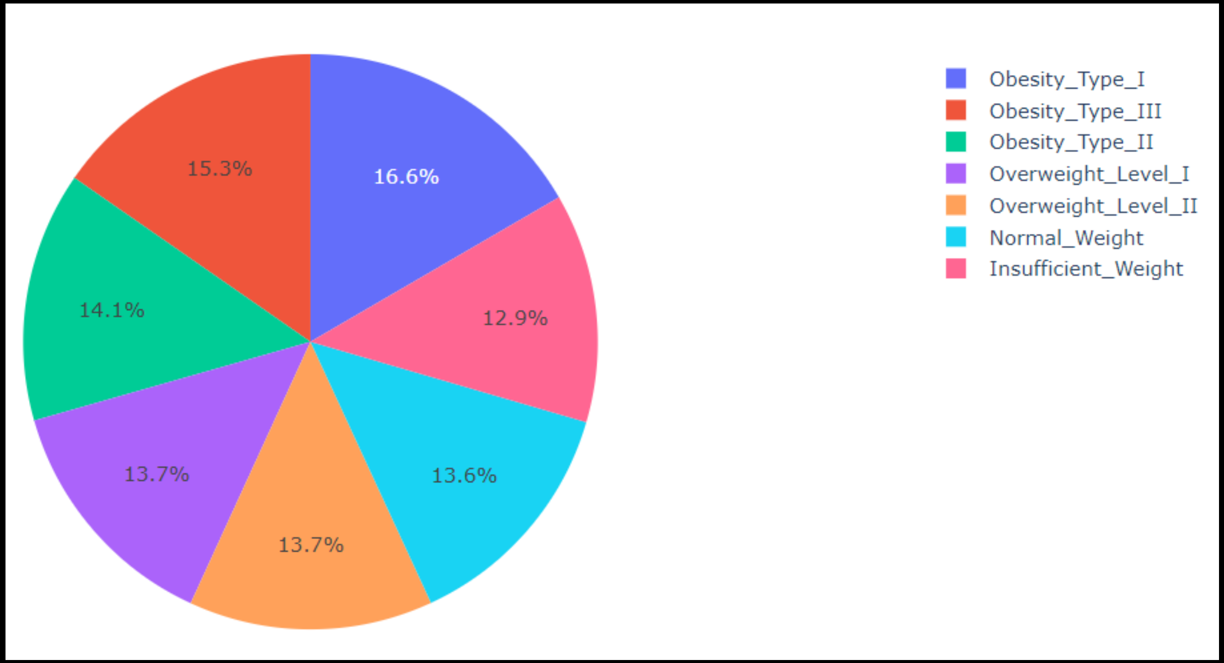
```python
sns.boxplot(x="NObeyesdad", y="BMI", data=df, order=["Insufficient_Weight","Normal_Weight","Overweight_Level_I","Overweight_Leve
plt.xticks(rotation = 45)

sns.set(rc={'figure.figsize':(15,8)})
```



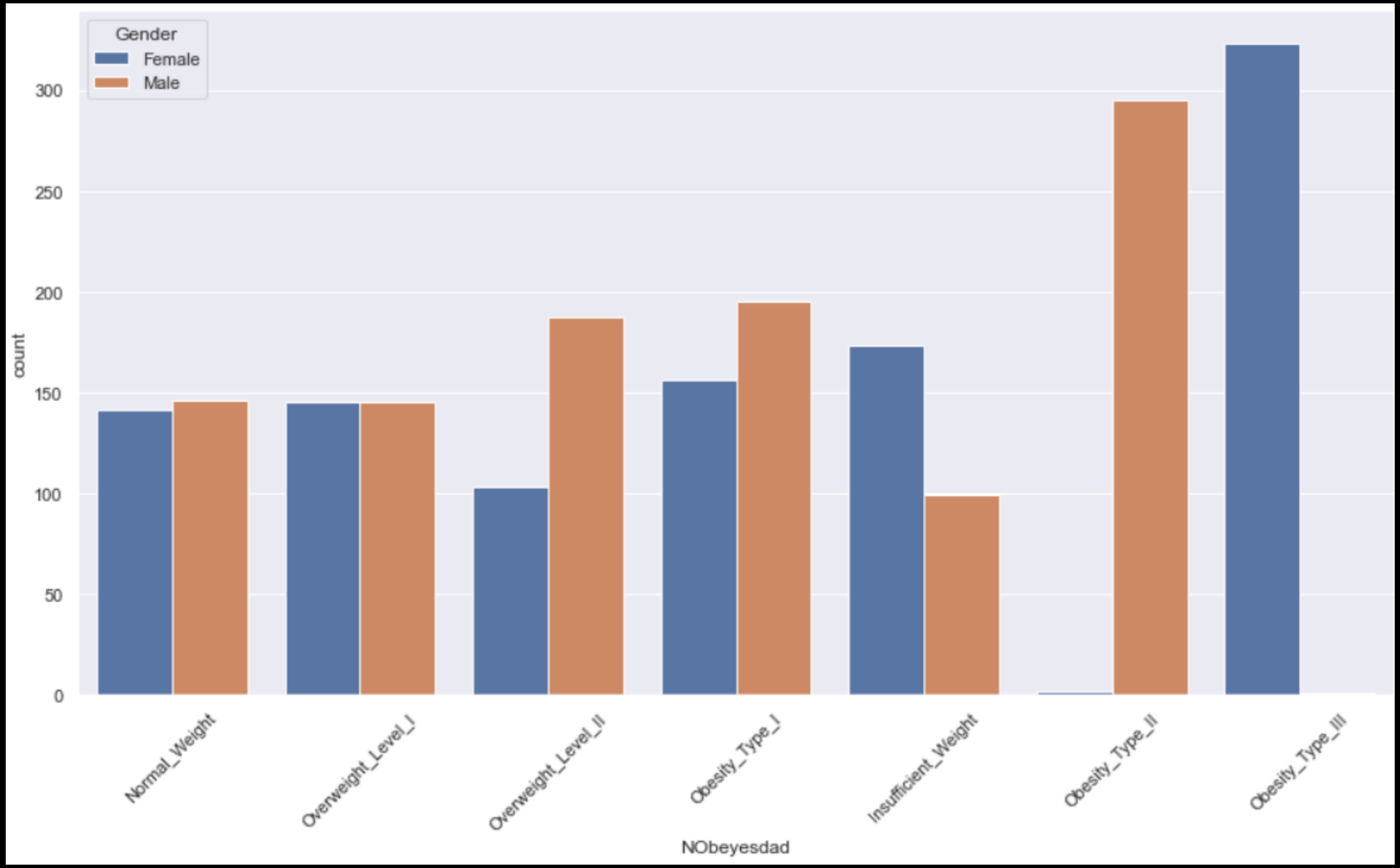
Here we see that individuals are tidy in the different classes in relation to their BMI.

- We also looked at the distribution of different individuals into the different categories.
- 

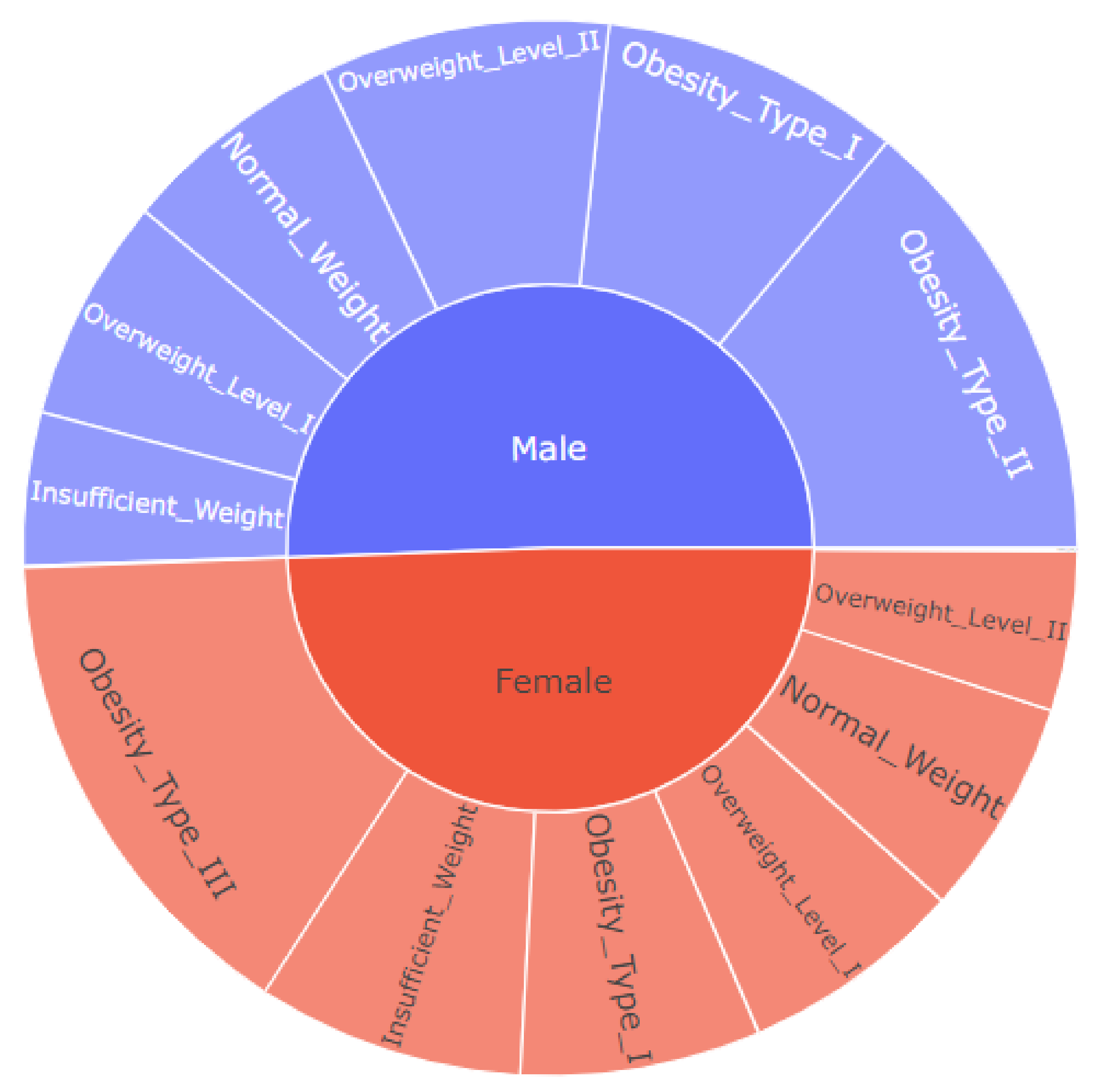


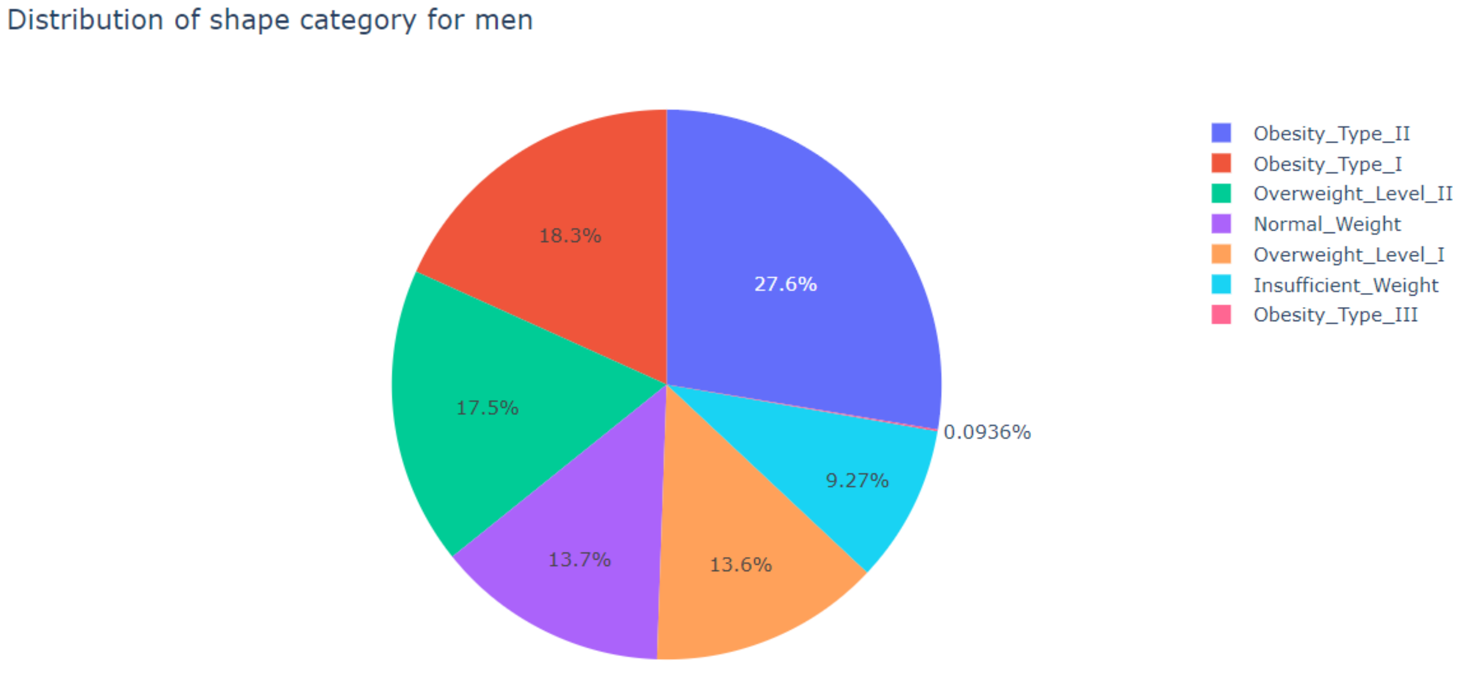
- It is noted that individuals are fairly evenly distributed into different categories.
- Let's not forget that 77% of the data was generated synthetically.

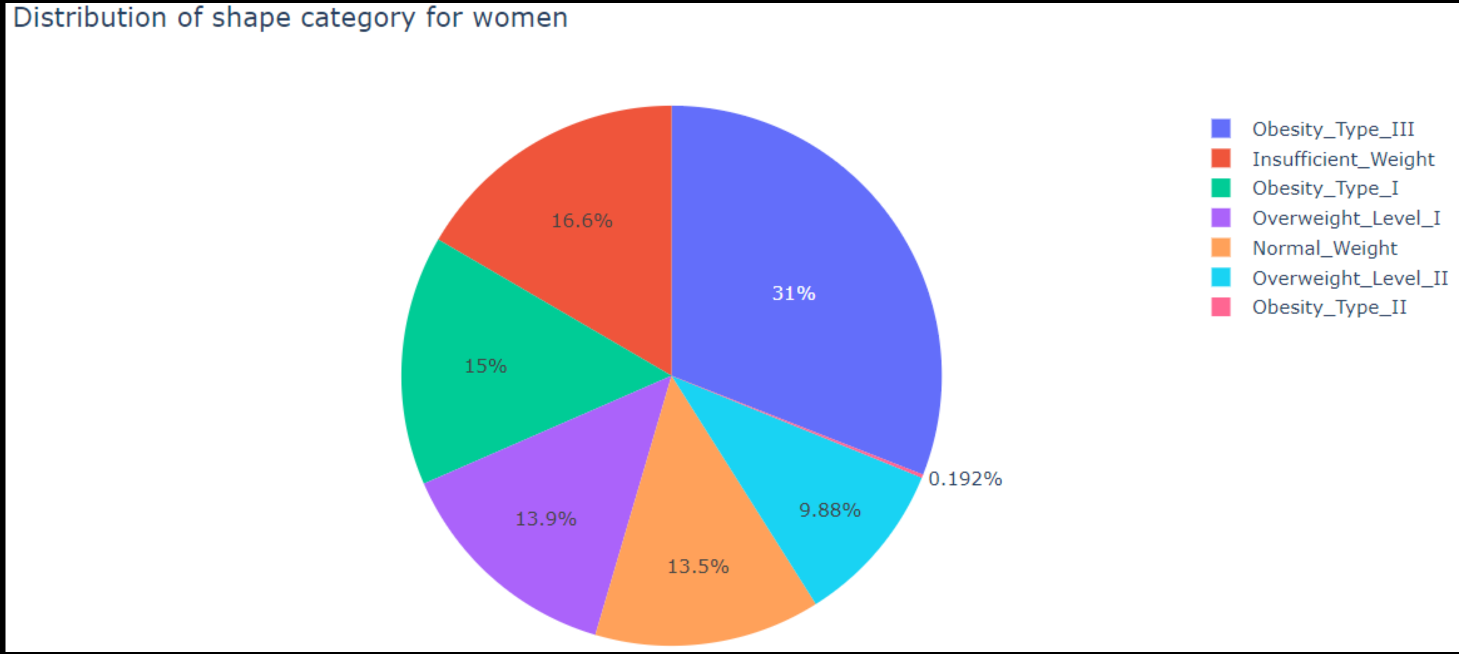
- We now look at the distribution of individuals according to their gender.
- We almost notice that men in obesity_2 and almost women in obesity_3.



- Another way to visualize the distribution by gender.

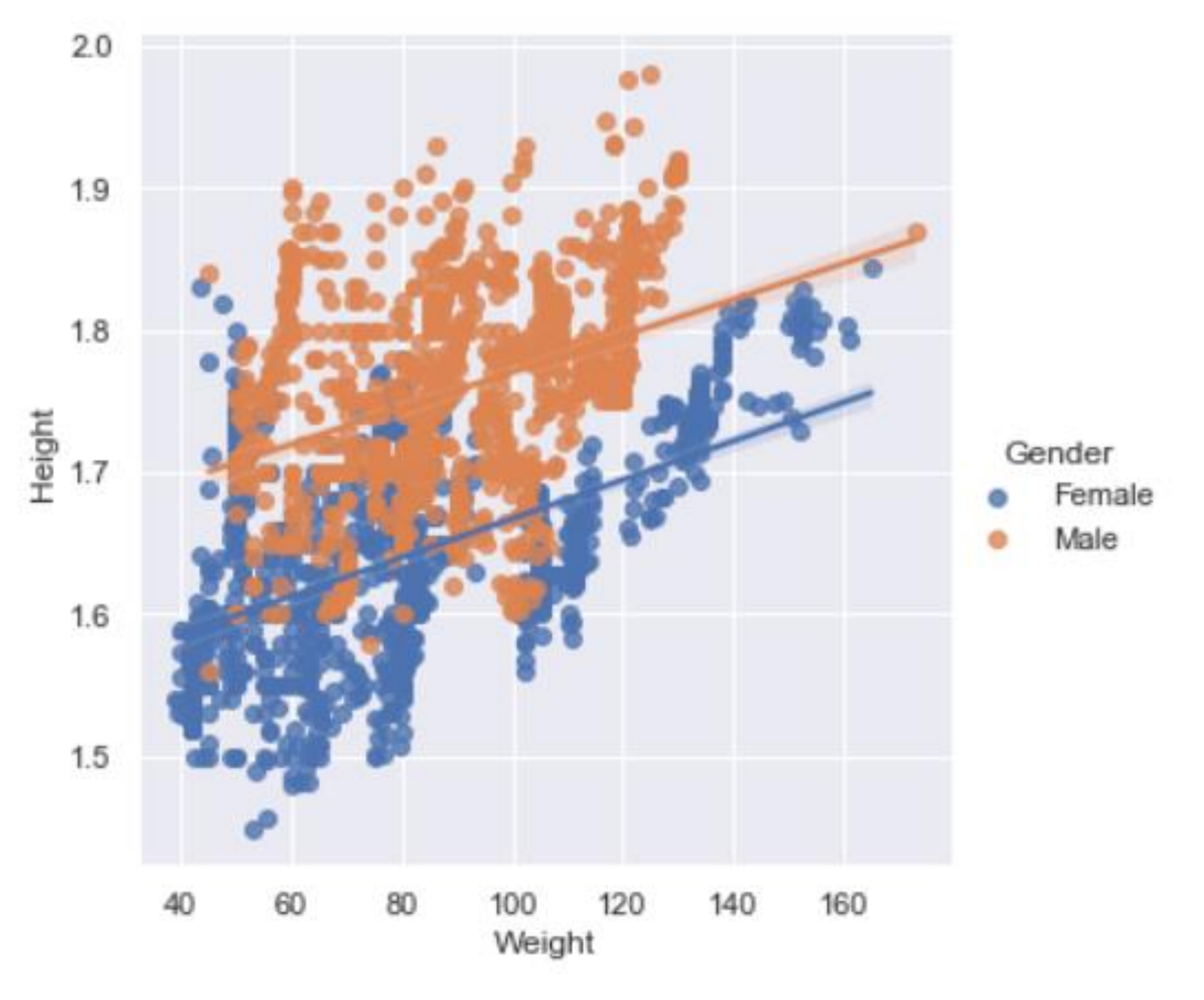




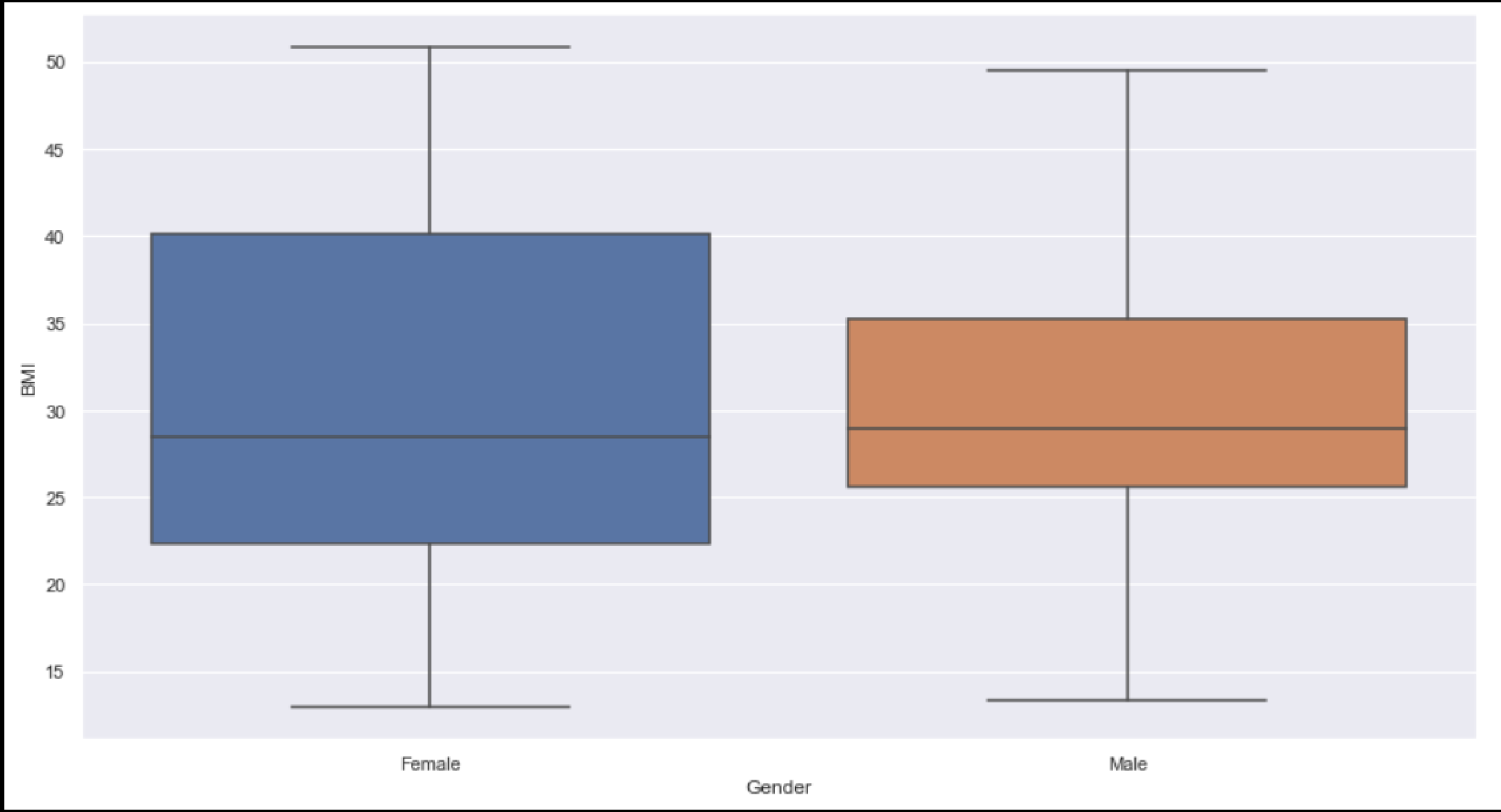


There are fairly equivalent distributions in the different categories.

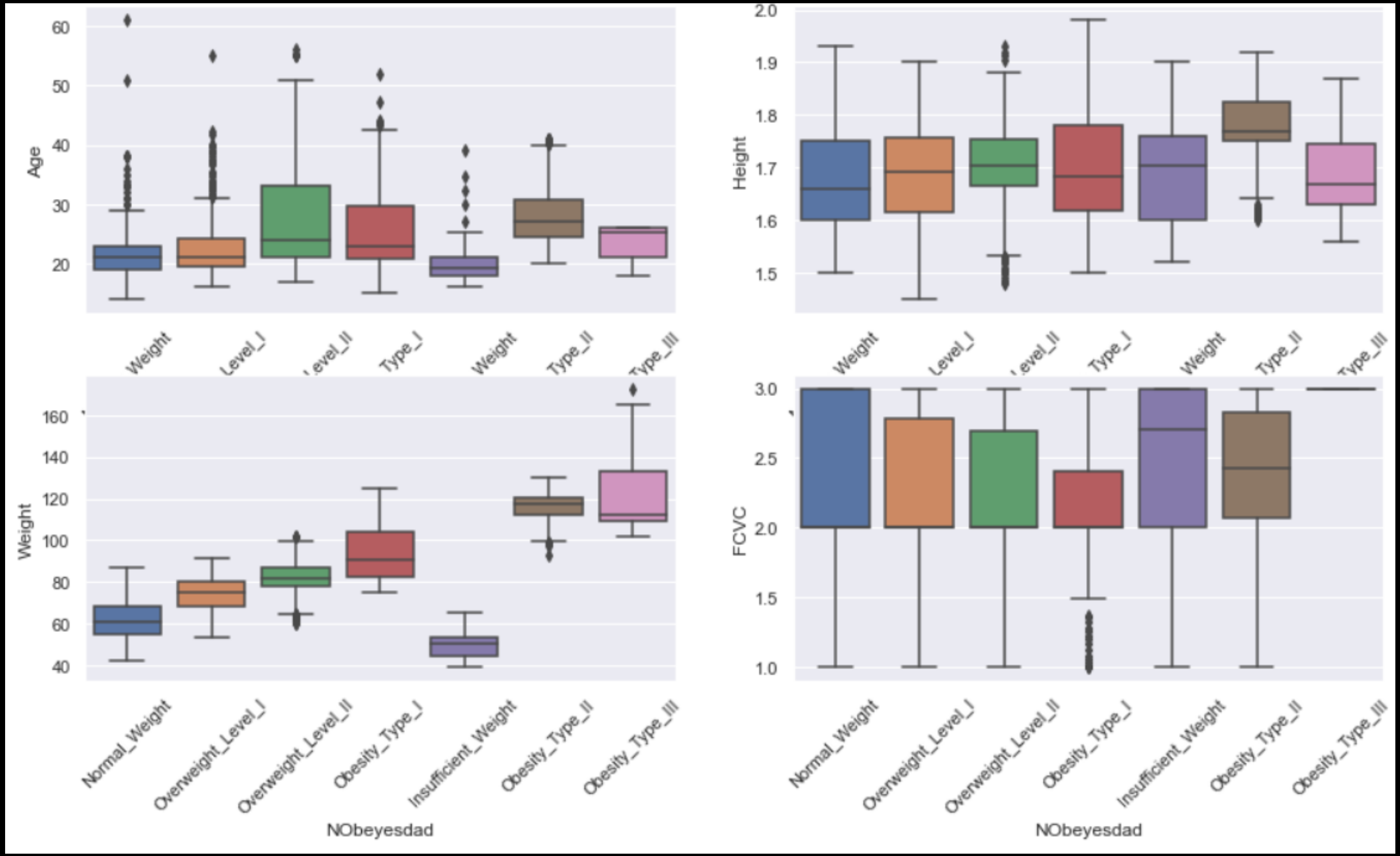
- We see here that men are generally heavier and taller than women.



- On the other hand, it can be said here that gender does not really have an influence on the category of the individual.



- Analysons maintenant les donnés numériques du dataset en fonction de la catégorie.



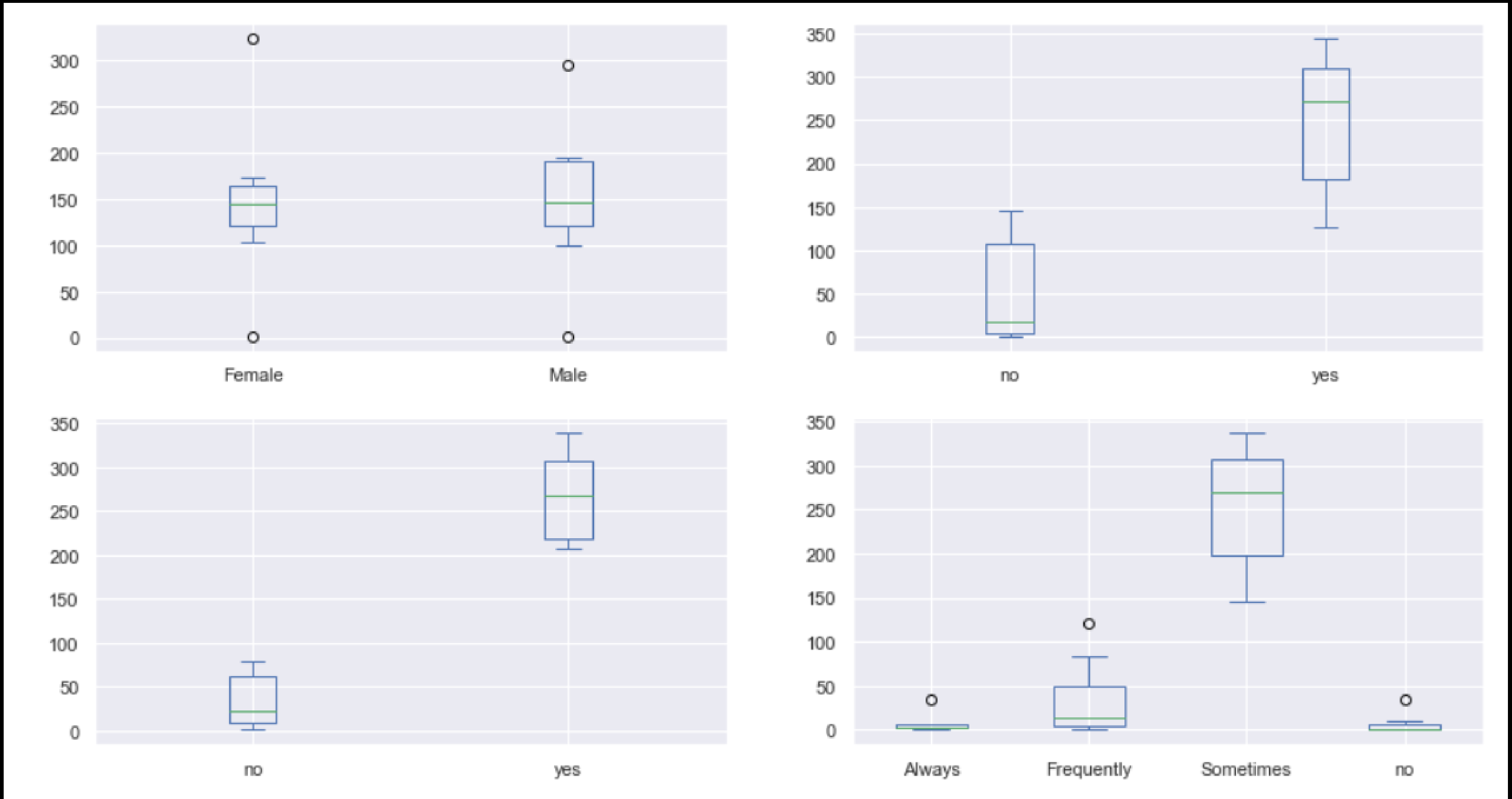
- On voit ici comme on s’en doutait que le poids est un indice important pour définir la catégorie.

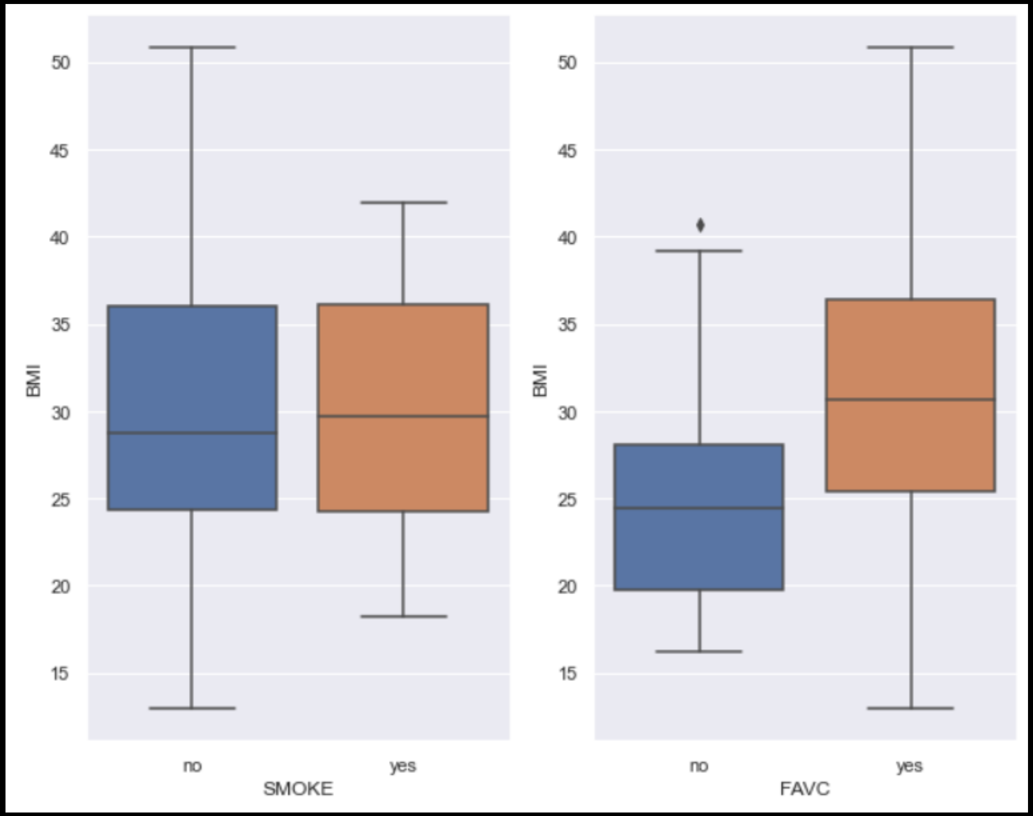
- Now we wanna boxplot the categorical variables, in order to do that we're gonna build a frequency table

	Gender	family_history_with_overweight	FAVC	CAEC	SMOKE	SCC	CALC	MTRANS	NObeyesdad
0	Female	yes	no	Sometimes	no	no	no	Public_Transportation	Normal_Weight
1	Female	yes	no	Sometimes	yes	yes	Sometimes	Public_Transportation	Normal_Weight
2	Male	yes	no	Sometimes	no	no	Frequently	Public_Transportation	Normal_Weight
3	Male	no	no	Sometimes	no	no	Frequently	Walking	Overweight_Level_I
4	Male	no	no	Sometimes	no	no	Sometimes	Public_Transportation	Overweight_Level_II
...	...	...	...	...	...	...	...	...	...
2106	Female	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_III
2107	Female	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_III
2108	Female	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_III
2109	Female	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_III
2110	Female	yes	yes	Sometimes	no	no	Sometimes	Public_Transportation	Obesity_Type_III

2111 rows × 9 columns

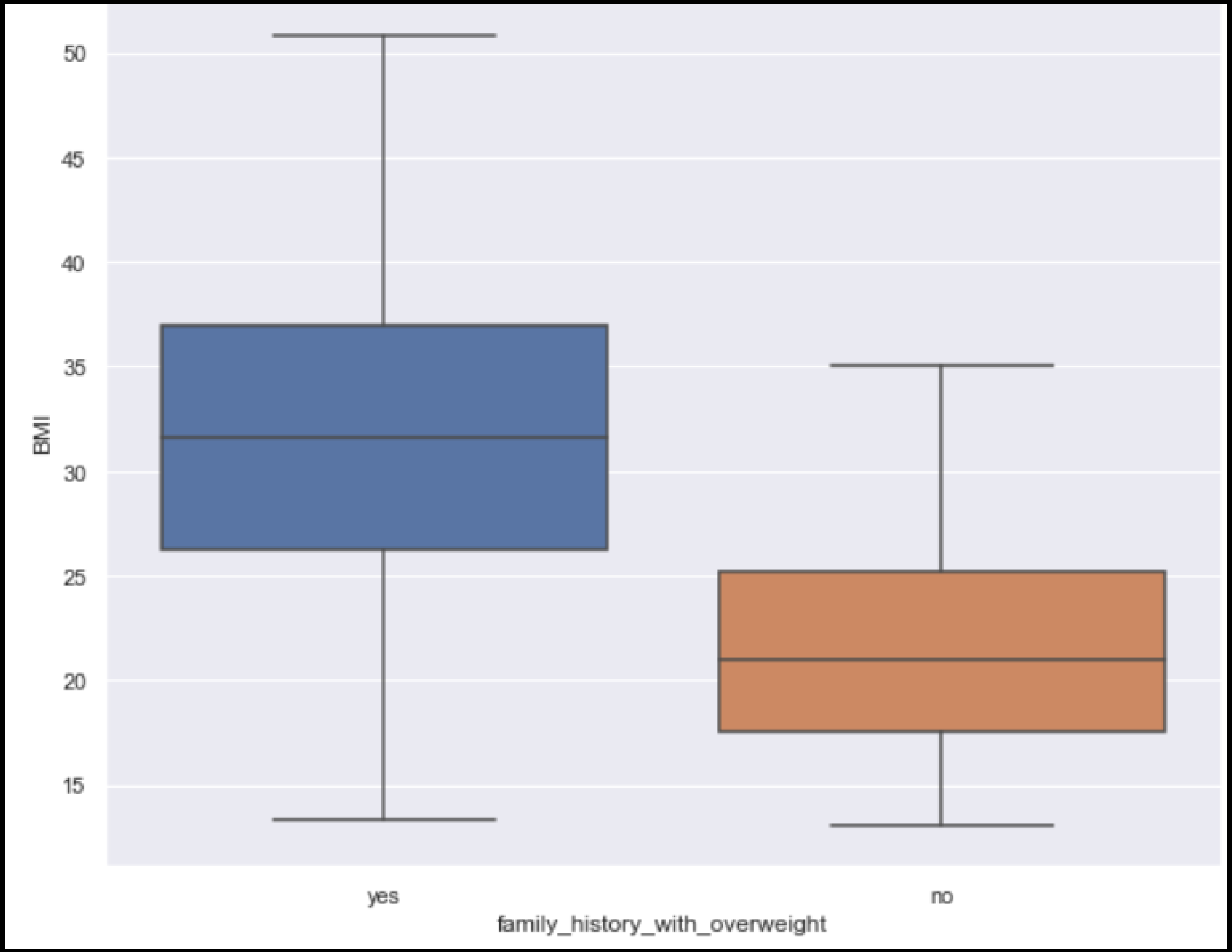
- We can see here that gender does not really have an impact on the category.



- Let's study whether smoking or eating fat often has an impact on the category.
- 

- We notice that smoking has no real impact unlike eating caloric food often. But let's continue the analysis to be sure.

- We also see here that if there are weight problems in the family, it promotes obesity.



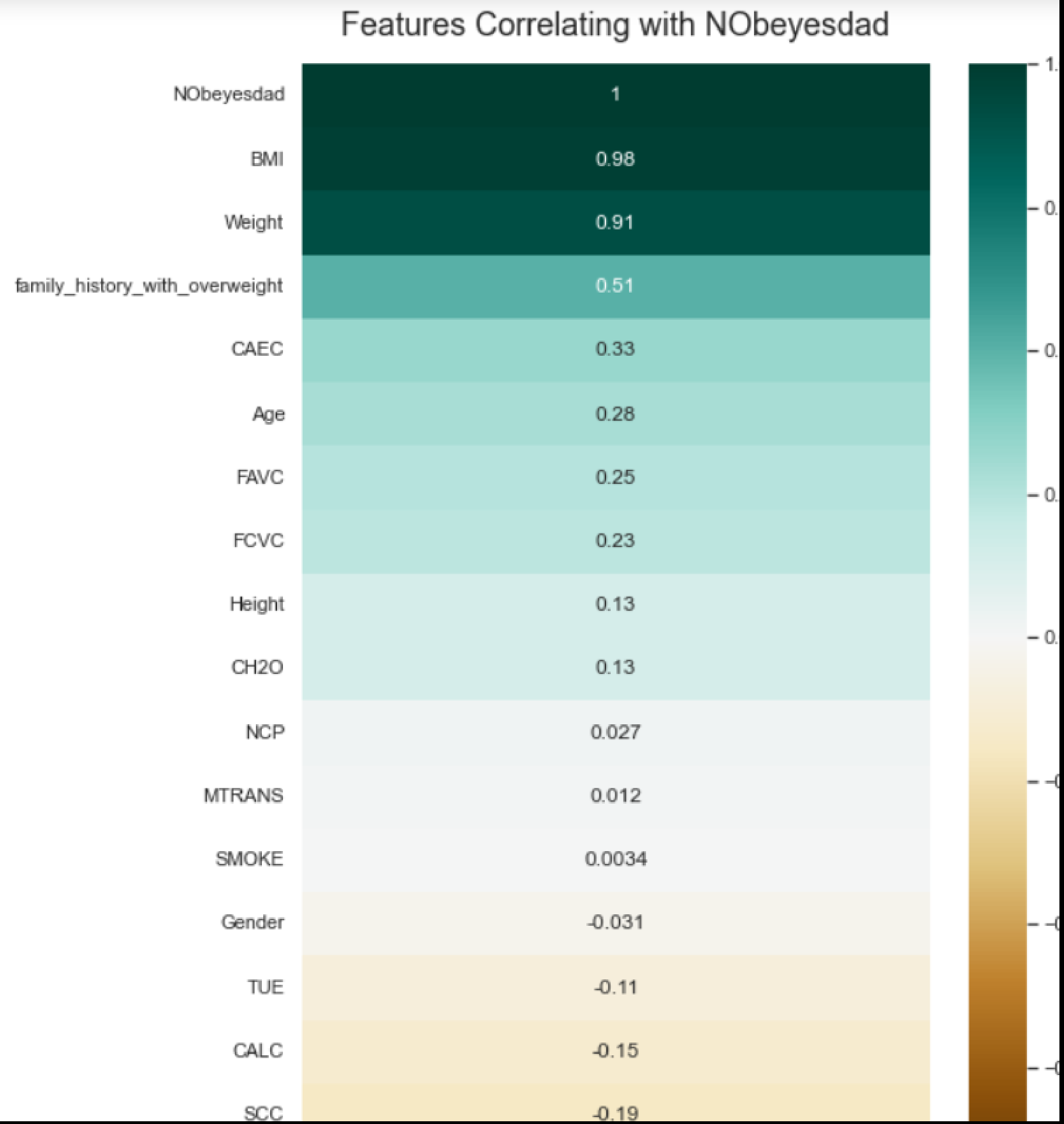
- Etudions la correlation entre les variables numérique.

- These plots show us that some categorical features have an impact on Obesity, let's now check the correlation between these variables and the target :

- Let's first turn all features into numerocal :

ıt	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad	BMI
0	64.000000	1	0	2.0	3.0	2	0	2.000000	0	0.000000	1.000000	3	3	2	24.386526
0	56.000000	1	0	3.0	3.0	2	1	3.000000	1	3.000000	0.000000	2	3	2	24.238227
0	77.000000	1	0	2.0	3.0	2	0	2.000000	0	2.000000	1.000000	1	3	2	23.765432
0	87.000000	0	0	3.0	3.0	2	0	2.000000	0	2.000000	0.000000	1	4	3	26.851852
0	89.800000	0	0	2.0	1.0	2	0	2.000000	0	0.000000	0.000000	2	3	4	28.342381
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	131.408528	1	1	3.0	3.0	2	0	1.728139	0	1.676269	0.906247	2	3	7	44.901475
4	133.742943	1	1	3.0	3.0	2	0	2.005130	0	1.341390	0.599270	2	3	7	43.741923
6	133.689352	1	1	3.0	3.0	2	0	2.054193	0	1.414209	0.646288	2	3	7	43.543817
0	133.346641	1	1	3.0	3.0	2	0	2.852339	0	1.139107	0.586035	2	3	7	44.071535
6	133.472641	1	1	3.0	3.0	2	0	2.863513	0	1.026452	0.714137	2	3	7	44.144338

- We can now see the correlation with NObeyesdad



Modelisation

- First, let's split our data
- We chose the variables to put in our model: Weight, CAEC, family history with overweight, age, CH2O, FAVC, CALC, FAF
- We did not select the BMI because it is directly related to the category.
-

	Weight	family_history_with_overweight	FAVC	CAEC	CH2O	FAF	CALC
0	64.000000	1	0	2	2.000000	0.000000	3
1	56.000000	1	0	2	3.000000	3.000000	2
2	77.000000	1	0	2	2.000000	2.000000	1
3	87.000000	0	0	2	2.000000	2.000000	1
4	89.800000	0	0	2	2.000000	0.000000	2
...	...	...	...	...	...	...	...
2106	131.408528	1	1	2	1.728139	1.676269	2
2107	133.742943	1	1	2	2.005130	1.341390	2
2108	133.689352	1	1	2	2.054193	1.414209	2
2109	133.346641	1	1	2	2.852339	1.139107	2
2110	133.472641	1	1	2	2.863513	1.026452	2

1st Model : SVM (Support Vector Machine) mode

```
from sklearn import metrics
print("Accuracy = ", metrics.accuracy_score(y_test, y_pred_SVM))
```

```
Accuracy =  0.6585365853658537
```

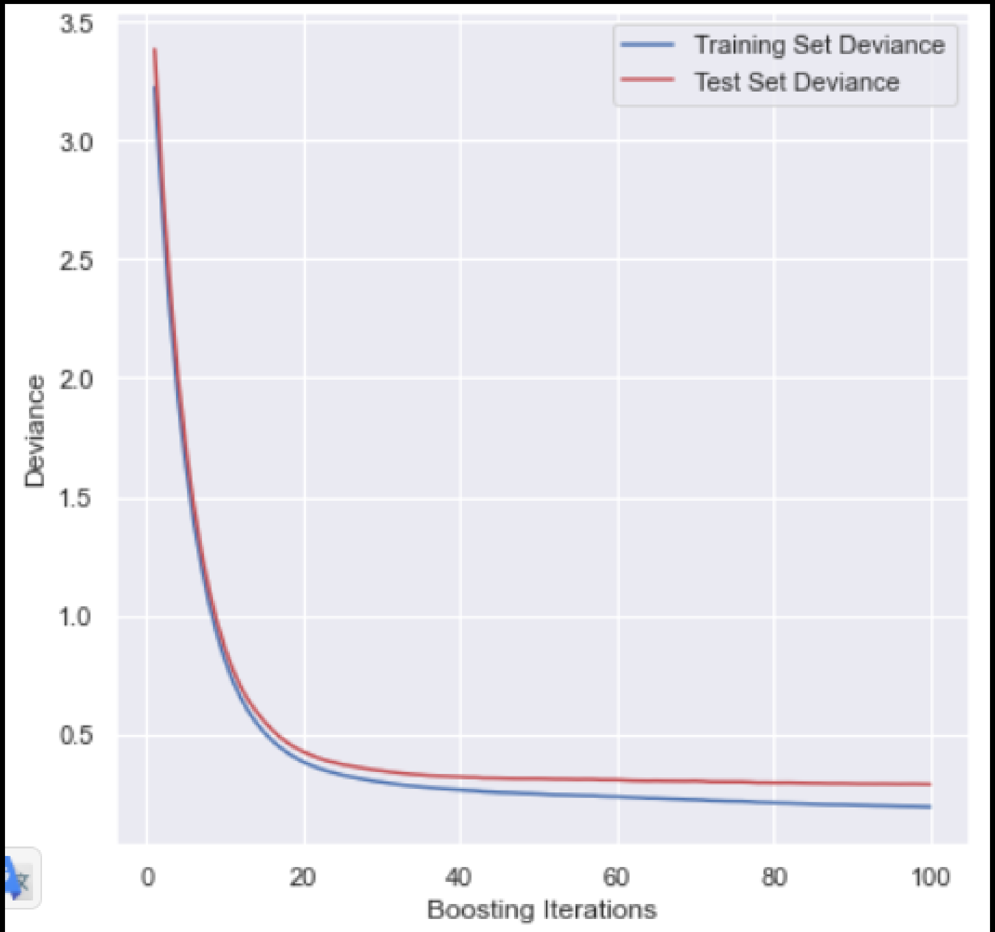
```
print(classification_report(y_test, y_pred_SVM))
```

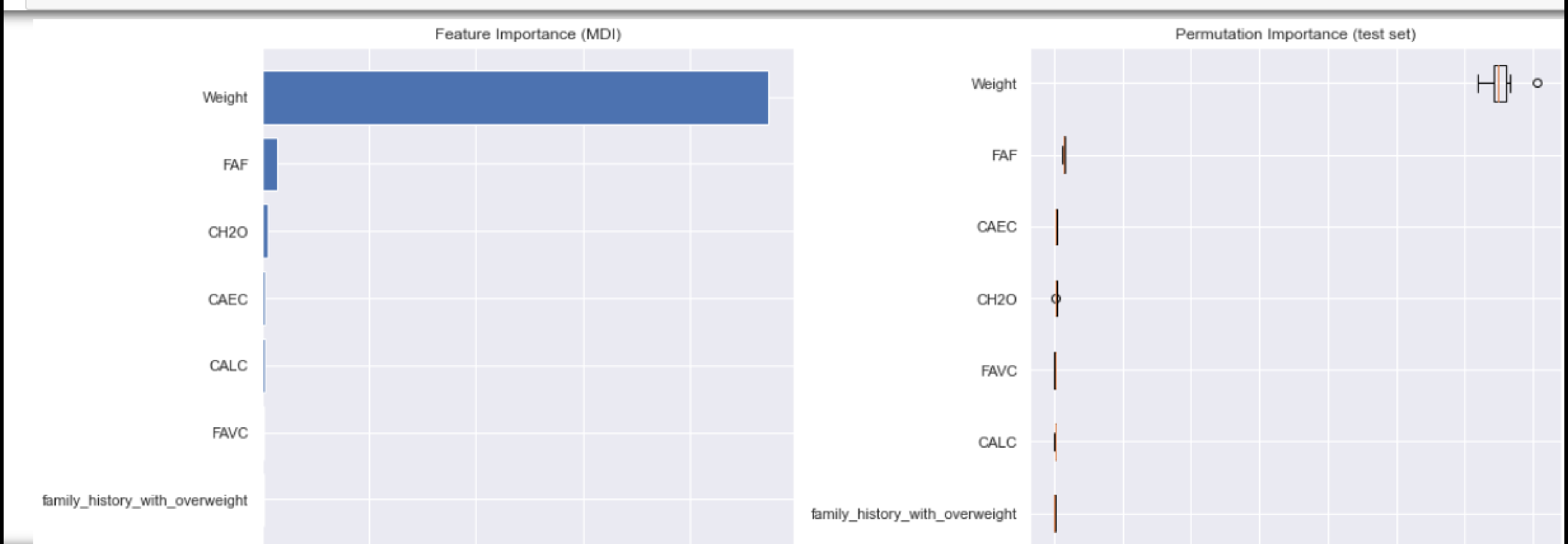
	precision	recall	f1-score	support
1	0.75	0.92	0.83	90
2	0.65	0.49	0.56	103
3	0.55	0.61	0.58	94
4	0.53	0.57	0.55	87
5	0.73	0.58	0.64	114
6	0.68	0.61	0.64	94
7	0.70	0.83	0.76	115
accuracy			0.66	697
macro avg	0.66	0.66	0.65	697
weighted avg	0.66	0.66	0.65	697

- An accuracy of 66% is obtained.

2nd Model : boosting model

- The mean squared error (MSE on test set : 0,2891
- After plot the training deviance :




Feature Importance (MDI)
Weight
FAF
CH2O
CAEC
CALC
FAVC
family_history_with_overweight
Permutation Importance (test set)
Weight
FAF
CAEC
CH2O
FAVC
CALC
family_history_with_overweight

3rd model : Random forest classifier

- We have an accuracy is 82%

	precision	recall	f1-score	support
1	0.85	0.91	0.88	90
2	0.71	0.74	0.72	103
3	0.74	0.74	0.74	94
4	0.65	0.63	0.64	87
5	0.83	0.75	0.79	114
6	0.93	0.96	0.94	94
7	0.96	0.96	0.96	115
accuracy			0.82	697
macro avg	0.81	0.81	0.81	697
weighted avg	0.82	0.82	0.82	697

- The score obtained on the observations not seen by the trees during their construction is 0.8175.

```
model.oob_score_
```

```
0.8175388967468176
```

- We therefore notice that our models are reliable but that the Random Forest is more reliable than the SVM with 82% accuracy against 64%.

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Thank you for your attention !