Final project presentation

Python for data analysis

Lucas ENARD – Myriam ESSAFOUI

Summary

Dataset introduction

Dataset exploration and visualization

Modeling

API

DATASET INTRODUCTION

We selected the following dataset [UCI Machine Learning Repository: Estimation of obesity levels based on eating habits and physical condition Data Set] which include data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia, based on their **eating habits** and **physical condition**.

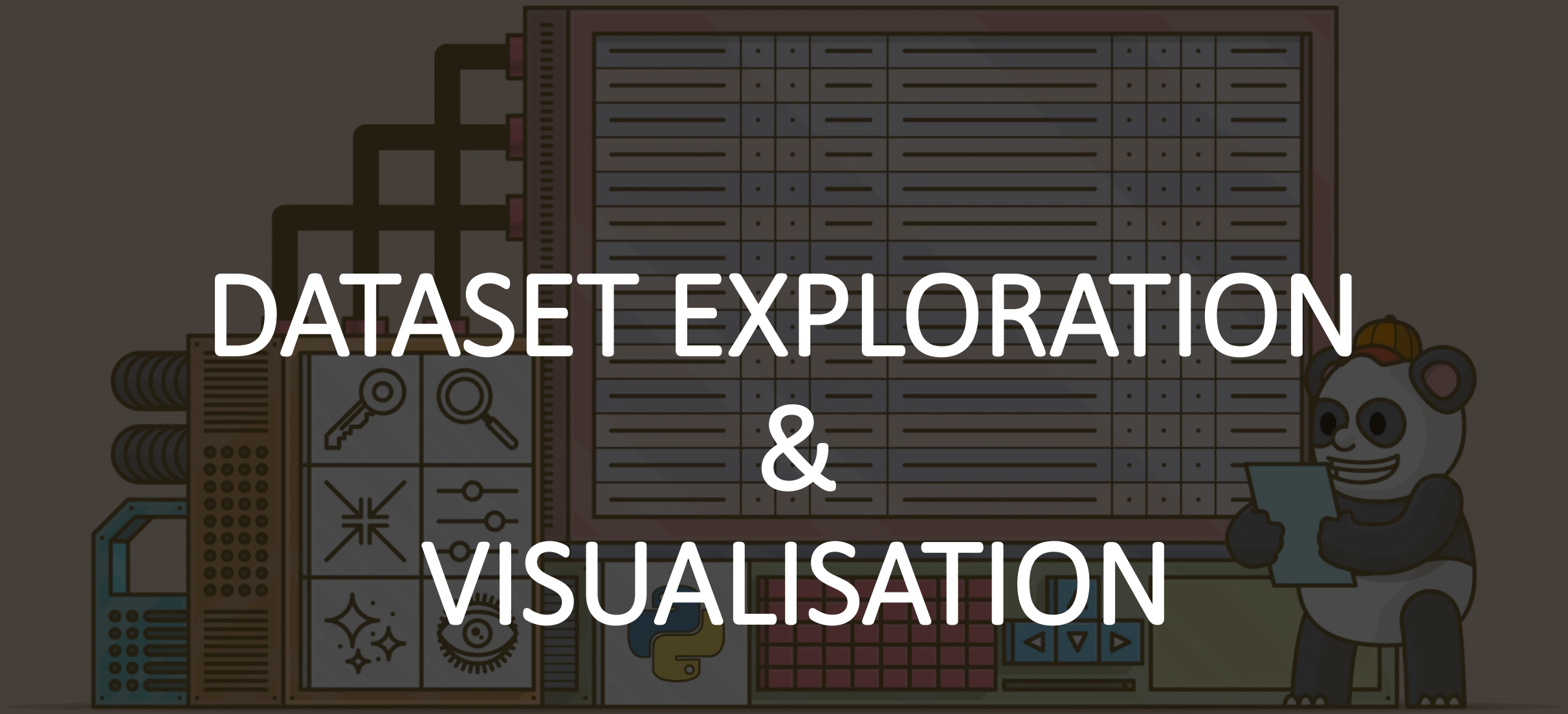
The first thing we did was to try and understand the content of the dataset, the records are labeled with the class variable NObesity (Obesity Level), that allows classification of the data using the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III.

Dataset origin and content

It is to be noted that 77% of the data was generated synthetically using the Weka tool and the SMOTE filter, 23% of the data was collected directly from users through a web platform.

The data content is as follow :

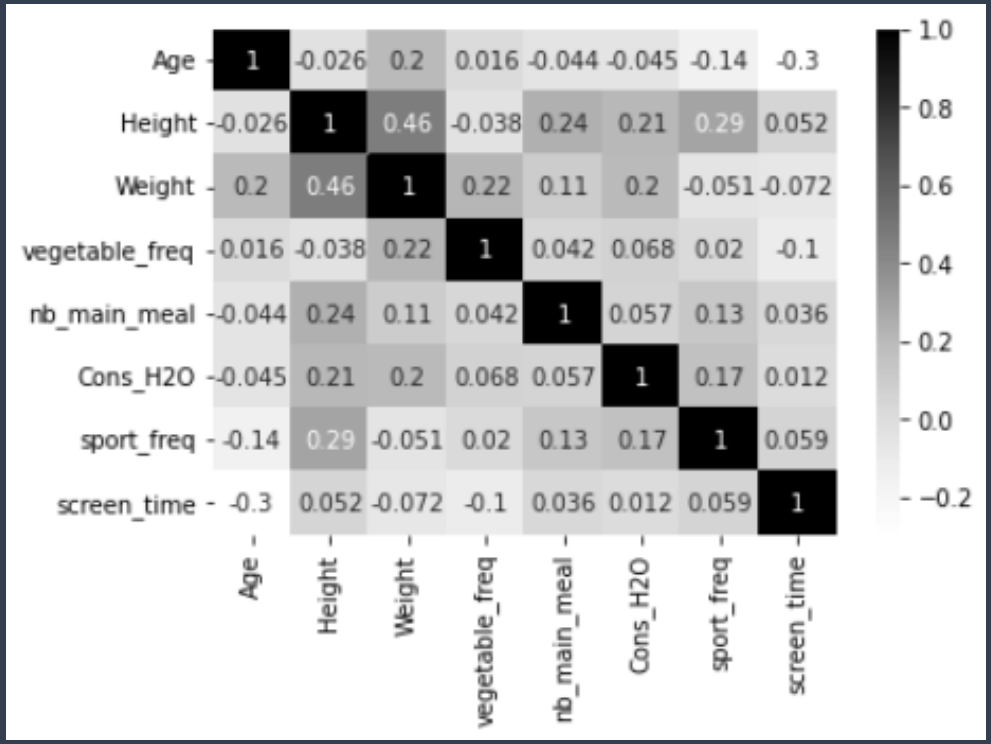
- Frequent consumption of high caloric food (FAVC)
- Frequency of consumption of vegetables (FCVC)
- Number of main meals (NCP)
- Consumption of food between meals (CAEC)
- Consumption of water daily (CH20)
- Consumption of alcohol (CALC)
- Calories consumption monitoring (SCC)
- Physical activity frequency (FAF)
- Time using technology devices(TUE)
- Transportation used (MTRANS)
- Other variables obtained were: Gender, Age,Height andWeight.



DATASET EXPLORATION & VISUALISATION

Real Python

After seeing the dataset, we decided to visualize it by creating a heatmap of the different variables.

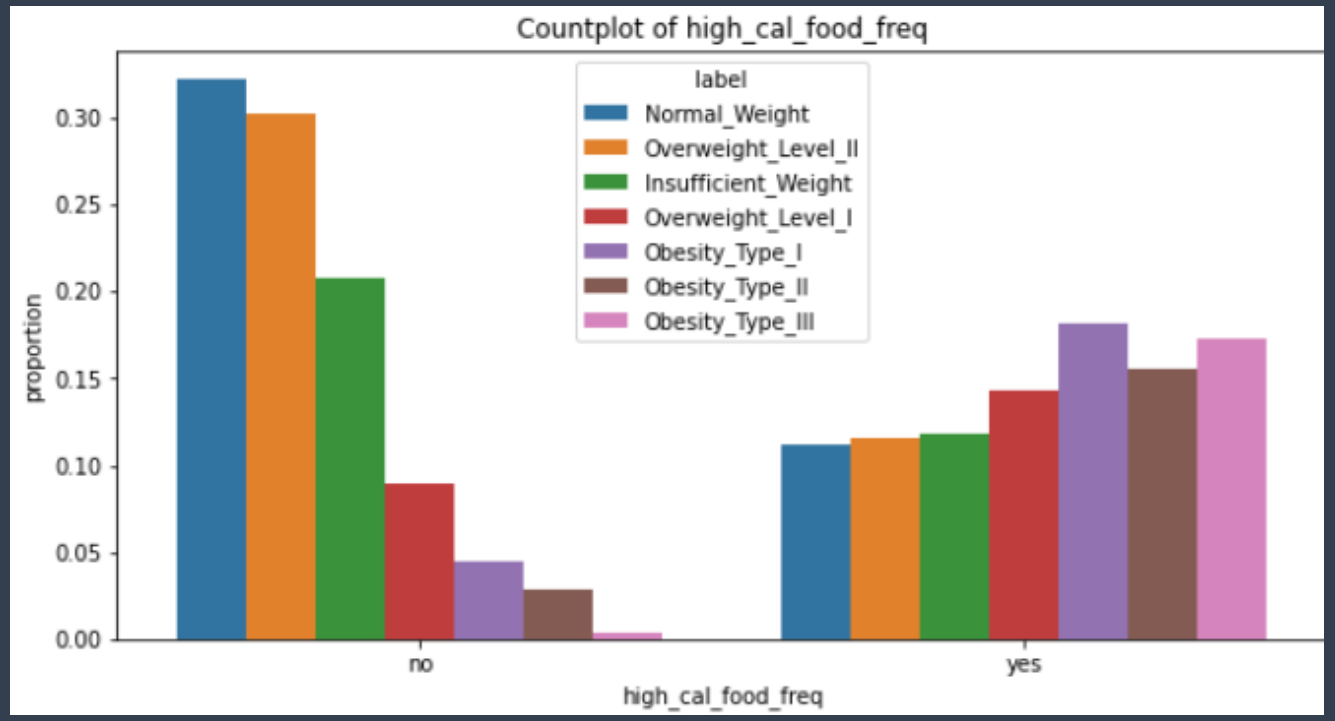


Heat map

The heatmap shows us the high degree of relationship between the height and weight, which is indeed logical, but also the relationship between the sport frequence and height of individuals.

We also observe a high relationship between the height and the number of main meal.

We also plotted, for each variables, a bar plot of the percentage of NObesity



Example of a bar plot that can be found in the code

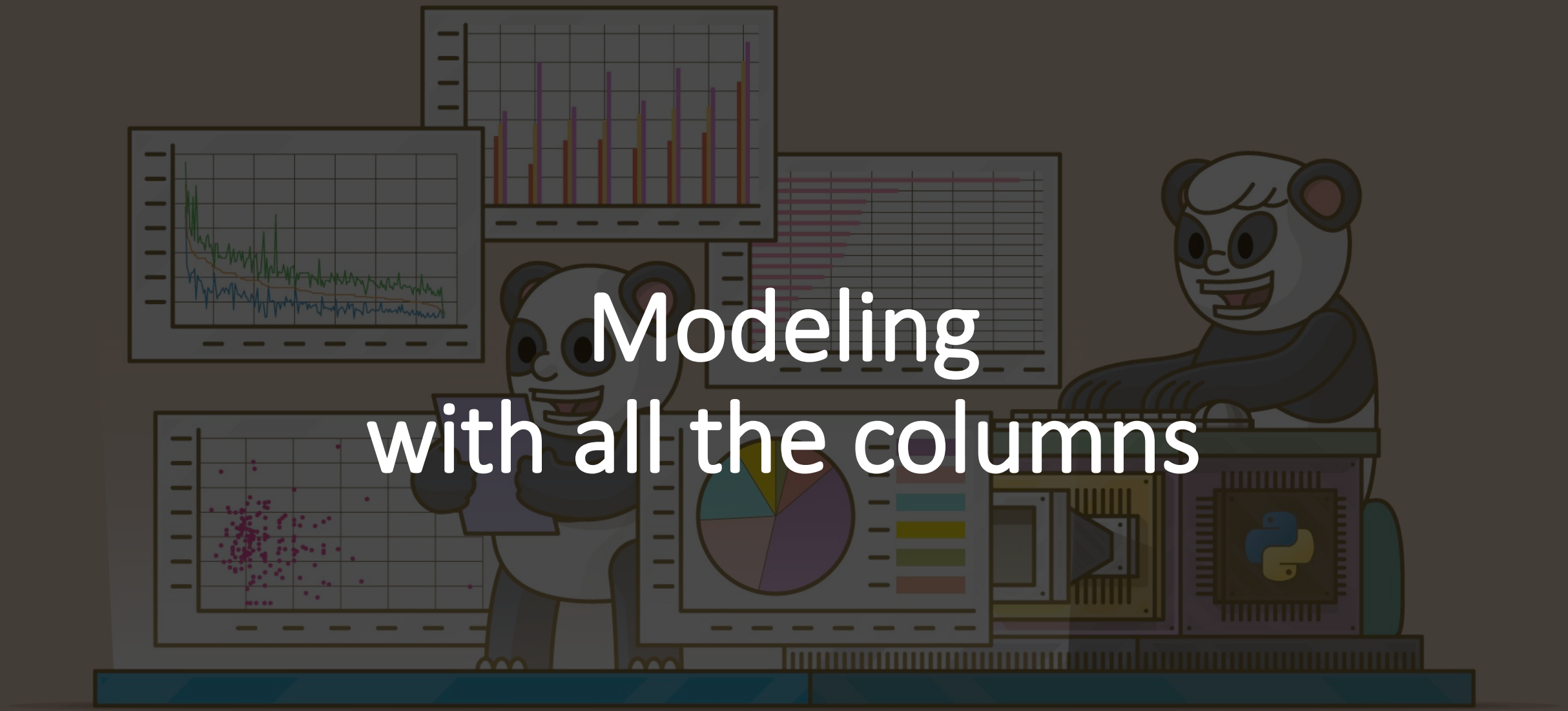
For this example, we are not surprised to see that people who don't have habits of eating high calories food are basically the normal-weighted, overweighted and insufficient weighted people.

Something interesting is the highest rate of overweighted people of level 1 who have habits of eating high calories food compared to the level 2's one, while they are less overweight.

Modification of the columns' names

	Gender	Age	Height	Weight	family_history_with_overweight	high_cal_food_freq	vegetable_freq	nb_main_meal	snack_freq	SMOKE	Cons_H2O	cal_count	sport_freq	screen_time	Cons_ALC	Means_TRANS	label
0	Female	21.000000	1.620000	64.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no	Public_Transportation	Normal_Weight
1	Female	21.000000	1.520000	56.000000	yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes	Public_Transportation	Normal_Weight
2	Male	23.000000	1.800000	77.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequently	Public_Transportation	Normal_Weight
3	Male	27.000000	1.800000	87.000000	no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequently	Walking	Overweight_Level_I
4	Male	22.000000	1.780000	89.800000	no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Public_Transportation	Overweight_Level_II
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2106	Female	20.976842	1.710730	131.408528	yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes	Public_Transportation	Obesity_Type_III
2107	Female	21.982942	1.748584	133.742943	yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes	Public_Transportation	Obesity_Type_III
2108	Female	22.524036	1.752206	133.689352	yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes	Public_Transportation	Obesity_Type_III
2109	Female	24.361936	1.739450	133.346641	yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes	Public_Transportation	Obesity_Type_III
2110	Female	23.664709	1.738836	133.472641	yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes	Public_Transportation	Obesity_Type_III

We first decided to change the columns names in order to have a more practical way of seeing the data in python using Pandas.


Modeling
with all the columns
Real Python

The first thing we had to do was to **encode the qualitative data**.

We decided to go for an encoding that replaced string value by **numeric values**.

This method doesn’t lose any information since there was already an order in the data.

for instance, the frequencies would be no : 0, sometimes : 1 ,frequently : 2 and always : 3.

```
df['Gender'] = df.Gender.replace({'Male':0, 'Female':1})
df['family_history_with_overweight'] = df.family_history_with_overweight.replace({'yes':0, 'no':1})
df['high_cal_food_freq'] = df.high_cal_food_freq.replace({'yes':0, 'no':1})
df['SMOKE'] = df.SMOKE.replace({'yes':0, 'no':1})
df['cal_count'] = df.cal_count.replace({'yes':0, 'no':1})
df['snack_freq'] = df.snack_freq.replace({'Sometimes':1, 'Frequently':2, 'Always':3, 'no':0})
df['Cons_ALC'] = df.Cons_ALC.replace({'Sometimes':1, 'Frequently':2, 'Always':3, 'no':0})
df['Means_TRANS'] = df.Means_TRANS.replace({'Public_Transportation':1, 'Walking':3, 'Automobile':0, 'Motorbike':2, 'Bike':4})
df['label'] = df.label.replace({'Normal_Weight':1, 'Overweight_Level_I':2, 'Overweight_Level_II':3, 'Obesity_Type_I':4, 'Insufficient_Weight':0,'Obesity_Type_II':5, 'Obesity_Type_III':6})
```

At first, the model seemed obvious, we would just predict the label (Nobesity) **using all the columns**.

However, we quickly saw that predicting the BMI using the height and the weight of the subject was of course not that hard and machine learning wasn't needed for that.

```
[65] from sklearn.model_selection import train_test_split

     # Split into validation and training data
     train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=1)
```

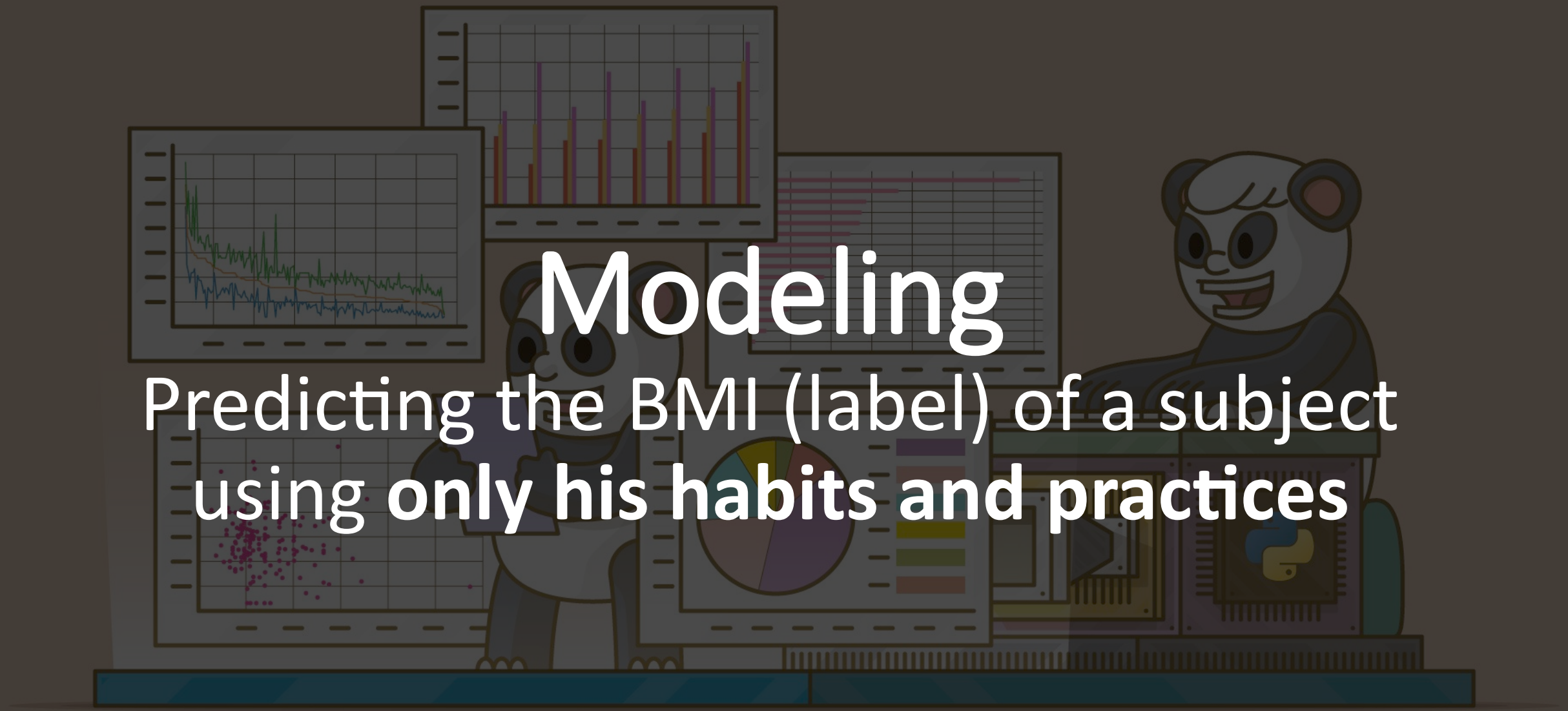
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error

# Define a random forest model
rf_model = RandomForestRegressor(random_state=1)
rf_model.fit(train_X, train_y)
rf_val_predictions = rf_model.predict(val_X)
rf_val_mae = mean_absolute_error(rf_val_predictions, val_y)
rf_accuracy = sk.metrics.accuracy_score(val_y,list(map(round,rf_val_predictions)))

print("Validation MAE for Random Forest Model: {}".format(rf_val_mae))
print("Accuracy by rounding predicted results into classes: {}".format(rf_accuracy))
```

```
Validation MAE for Random Forest Model: 0.07551136363636361
Accuracy by rounding predicted results into classes: 0.9659090909090909
```

Here we can see that a simple random forest without neither grid search nor further modification gives really decent results with a really good MAE and accuracy



Modeling

Predicting the BMI (label) of a subject using **only his habits and practices**

Real Python

This is where our real problematic comes from:

Can we predict the BMI of a subject using ONLY his habits and practices and not his weight nor height ?

The next step was to get rid of the weight and the height in the dataset to then try different algorithms and pick the best

We chose five algorithms to try and predict the class (label) of each individuals BMI without using weight nor height,

-Random forest

-Decision Tree Classifier

-Knn

-SGD

-Logistic regression

There were seven different labels as seen earlier and we decided to measure the performance to show the accuracy of the model but also the MEA (mean absolute error).

The MEA allows use to understand if the models are only a little wrong, which means predicting high obesity type 1 instead of type 2 or if it predicts underweight in an obesity type 2 situation

It was clear that the Knn algorithm was the best performing one

Model	Accuracy	MAE
Random Forest	0.717803	0.431705
Decision Tree CLassifier	0.342803	1.000000
Knn	0.820076	0.354167
SGD classifier	0.399621	1.195076
Logistic regression	0.539773	0.837121

Accuracy and MAE(mean absolute error)
for 5 different models

Because we decided to explore the Knn algorithms for our problem it was time to do a Grid Search in order to have the best knn possible.

```
# pipeline combining transformers and estimator
pipe_knn= make_pipeline(StandardScaler(), KNeighborsClassifier())

param = {'kneighborsclassifier__n_neighbors':[4,5,6,7],
         'kneighborsclassifier__weights':['uniform', 'distance'],
         'kneighborsclassifier__leaf_size':[25,30,35],
         'kneighborsclassifier__p':[1,2],
        kneighborsclassifier__n_jobs':[-1]}
# grid searh to choose the best (combination of) hyperparameters
gs_knn=GridSearchCV(estimator= pipe_knn,
                    param_grid=param,
                    scoring='accuracy',
                    cv=10)
```

Grid Search on the knn parameters

At the end we get an amazing accuracy of 0.82 and a MAE of 0.354

```
Validation MAE for Knn: 0.3541666666666667
Accuracy : 0.8200757575757576
```

CONCLUSION

We saw that 82% of the predictions were correct based only on the habits, and physical conditions and even on the family history !

Surprisingly, these factors have a significant impact on the chances of developing obesity and even without access to the weight or the height of an individual it is possible to predict his BMI.

To avoid and limit the probability of being affected, we should take care of our health, our eating and drinking habits, because like many chronic conditions, obesity is preventable with a healthy lifestyle : staying active, following a healthy diet, and so on.