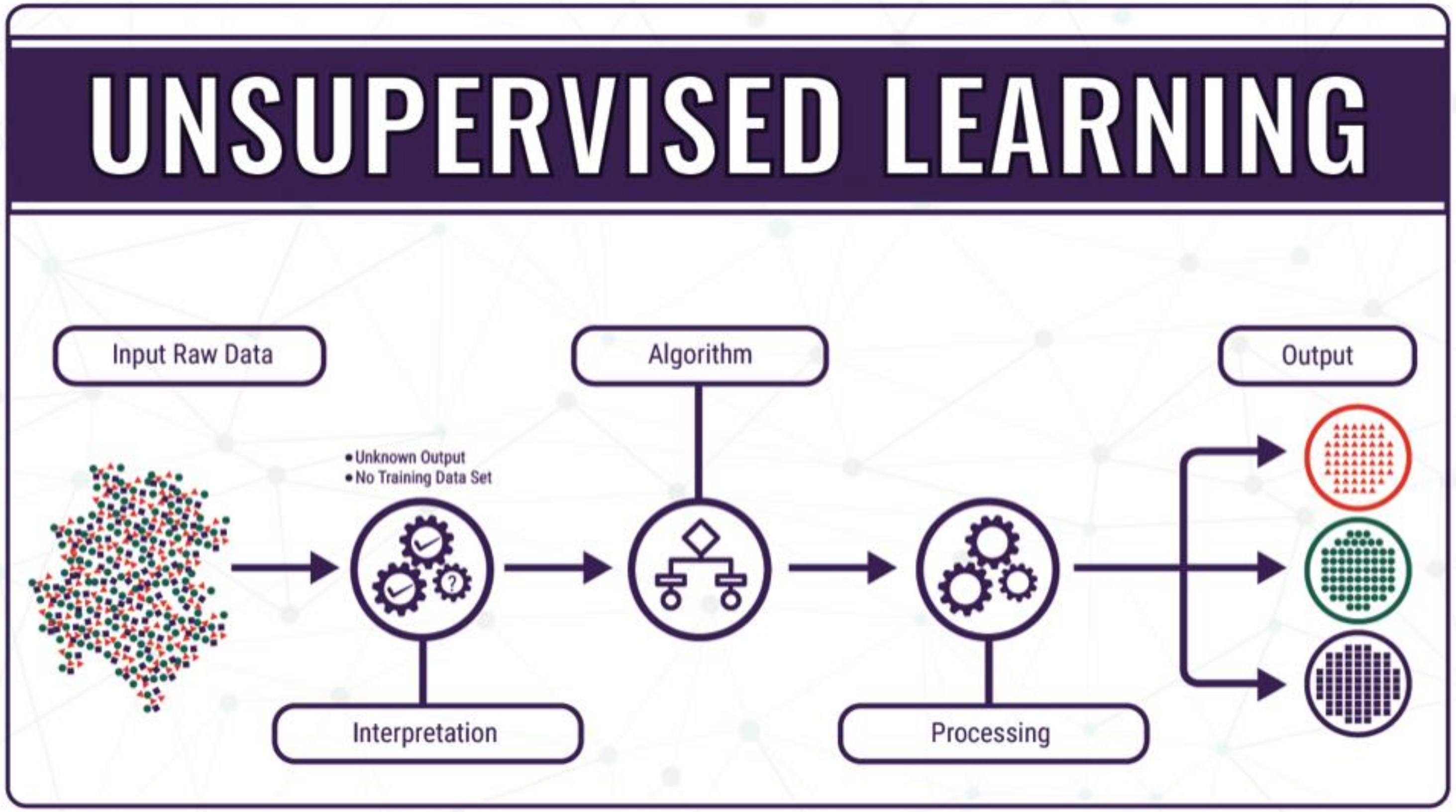
PYTHON FOR DATA ANALYSIS PROJECT

–

Antoine PHISSAMAY Théo PHILIPPE



1. Learning from the dataset

First, in order to face a data analysis problem, we need to understand on what we work. Obviously, we need to open the dataset and see what we are facing. We observe 10 columns, and 5472 rows.

The dataset is about a project of making a decision tree more efficient. In fact, the dataset is composed by block charactristics, there are five different type of blocks :

- Vertical line
- Horrizontal line
- Text
- Pictures
- Graphics

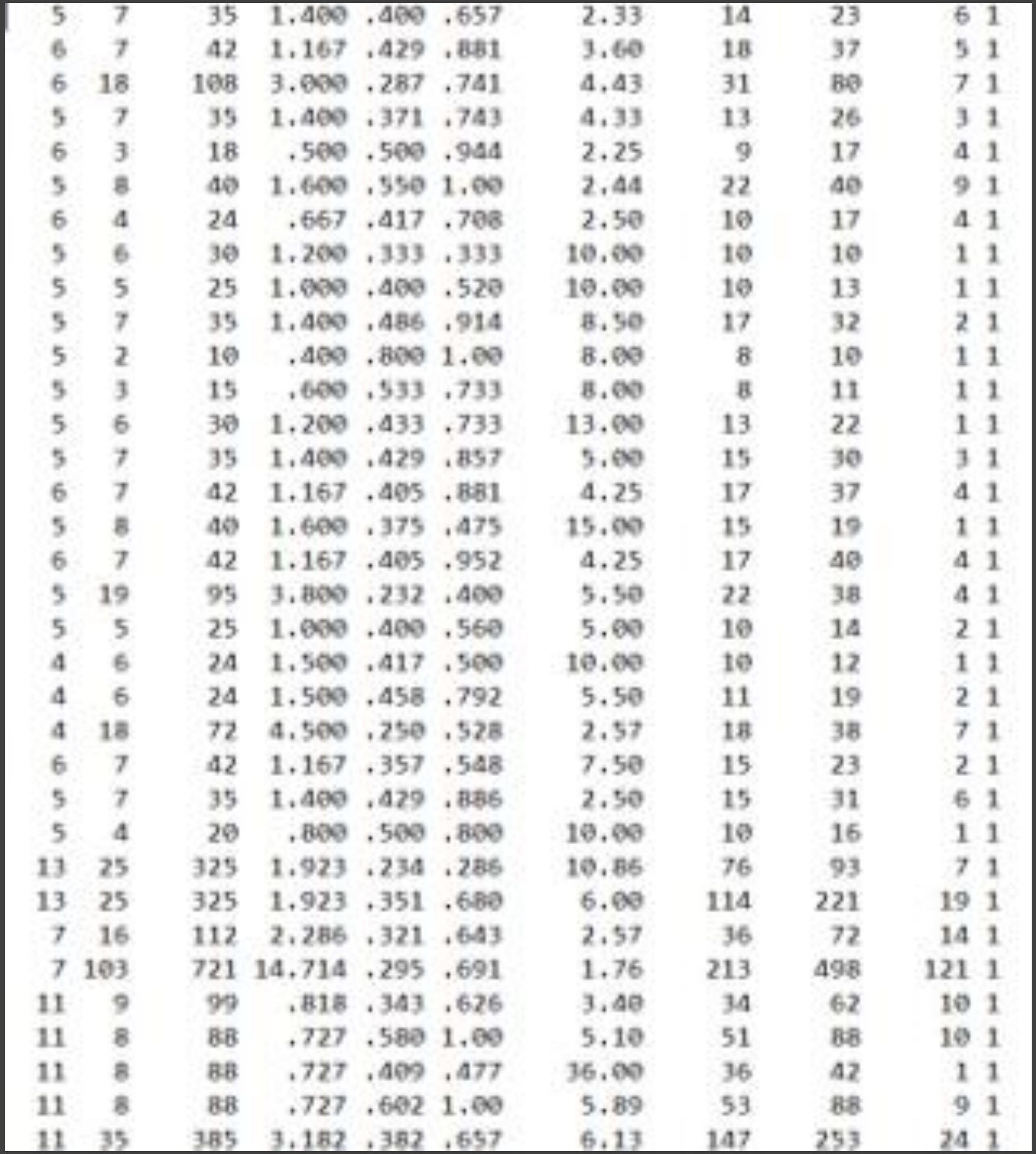
Thus, the aim of the study is to categorize whether if rows in the dataset are data from one type of block or another. To deal with that we need to develop a machin learning model, here we don't have a training set with target values so we have to do an unsupervised learning.

2. Data Cleaning and Preparation

The dataset is filled with columns aligned numbers, hence we can’t use any function with python for data analysis. Before cleaning the data we need to make the file readable by cleaning and other functions.

We create a program which delete all the spaces and put each data in lists which separate each data by a coma. Then we also make a csv file with the data.

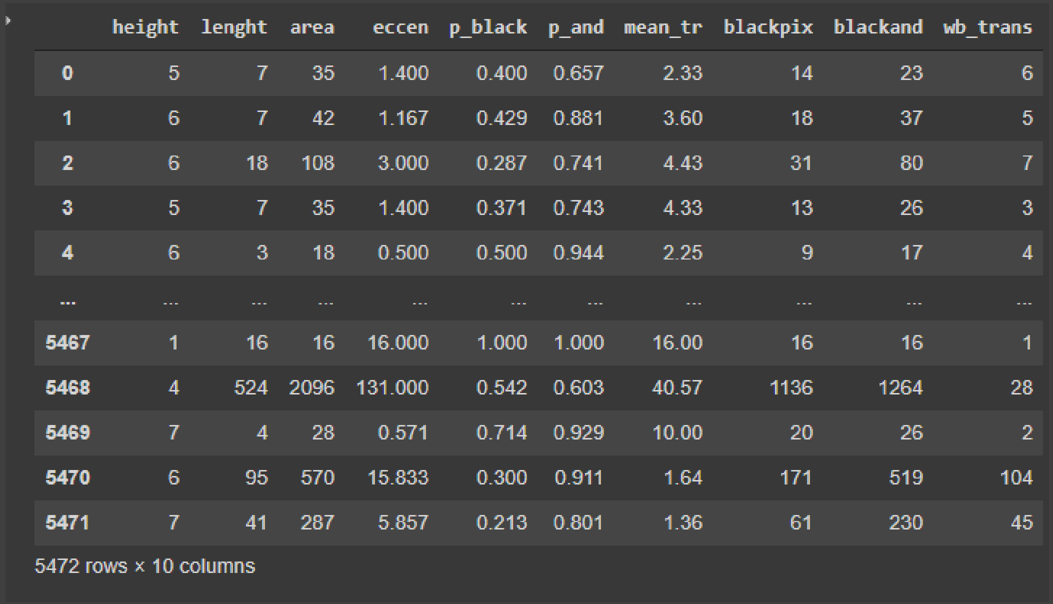
However, like it is specified in the information file attached with the dataset, there is no missing value, so we don’t need to clean the dataset by deleting rows where values are missing. We can begin to work on the data.



```
  5   7     35  1.400 .400 .657      2.33      14       23        6 1
  6   7     42  1.167 .429 .881      3.60      18       37        5 1
  6  18    108  3.000 .287 .741      4.43      31       80        7 1
  5   7     35  1.400 .371 .743      4.33      13       26        3 1
  6   3     18   .500 .500 .944      2.25       9       17        4 1
  5   8     40  1.600 .550 1.00      2.44      22       40        9 1
  6   4     24   .667 .417 .708      2.50      10       17        4 1
  5   6     30  1.200 .333 .333     10.00      10       10        1 1
  5   5     25  1.000 .400 .520     10.00      10       13        1 1
  5   7     35  1.400 .486 .914      8.50      17       32        2 1
  5   2     10   .400 .800 1.00      8.00       8       10        1 1
  5   3     15   .600 .533 .733      8.00       8       11        1 1
  5   6     30  1.200 .433 .733     13.00      13       22        1 1
  5   7     35  1.400 .429 .857      5.00      15       30        3 1
  6   7     42  1.167 .405 .881      4.25      17       37        4 1
  5   8     40  1.600 .375 .475     15.00      15       19        1 1
  6   7     42  1.167 .405 .952      4.25      17       40        4 1
  5  19     95  3.800 .232 .400      5.50      22       38        4 1
  5   5     25  1.000 .400 .560      5.00      10       14        2 1
  4   6     24  1.500 .417 .500     10.00      10       12        1 1
  4   6     24  1.500 .458 .792      5.50      11       19        2 1
  4  18     72  4.500 .250 .528      2.57      18       38        7 1
  6   7     42  1.167 .357 .548      7.50      15       23        2 1
  5   7     35  1.400 .429 .886      2.50      15       31        6 1
  5   4     20   .800 .500 .800     10.00      10       16        1 1
 13  25    325  1.923 .234 .286     10.86      76       93        7 1
 13  25    325  1.923 .351 .680      6.00     114      221       19 1
  7  16    112  2.286 .321 .643      2.57      36       72       14 1
  7 103    721 14.714 .295 .691      1.76     213      498      121 1
 11   9     99   .818 .343 .626      3.40      34       62       10 1
 11   8     88   .727 .580 1.00      5.10      51       88       10 1
 11   8     88   .727 .409 .477     36.00      36       42        1 1
 11   8     88   .727 .602 1.00      5.89      53       88        9 1
 11  35    385  3.182 .382 .657      6.13     147      253       24 1
```

Dataset before preparation

Dataset after preparation

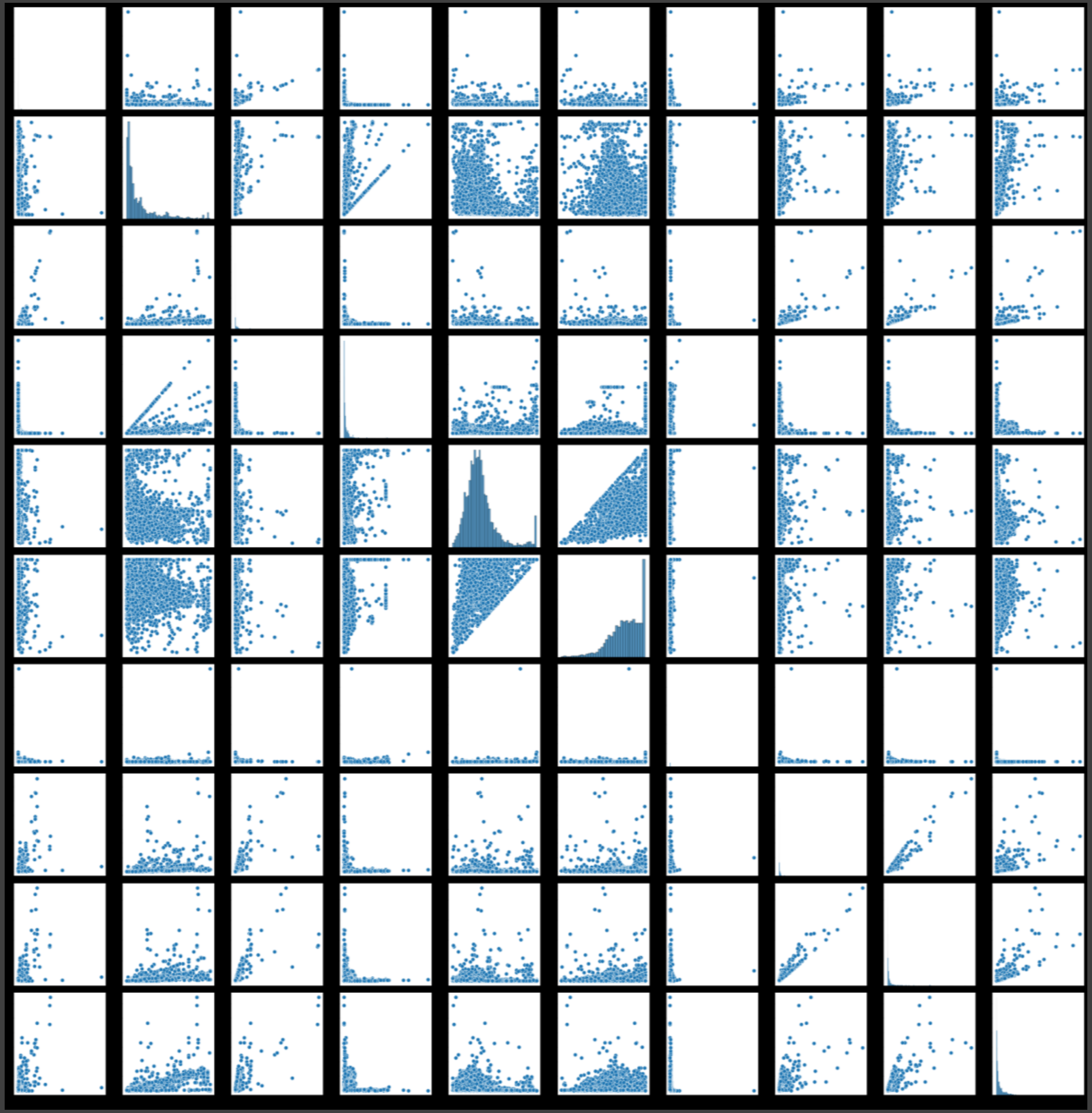


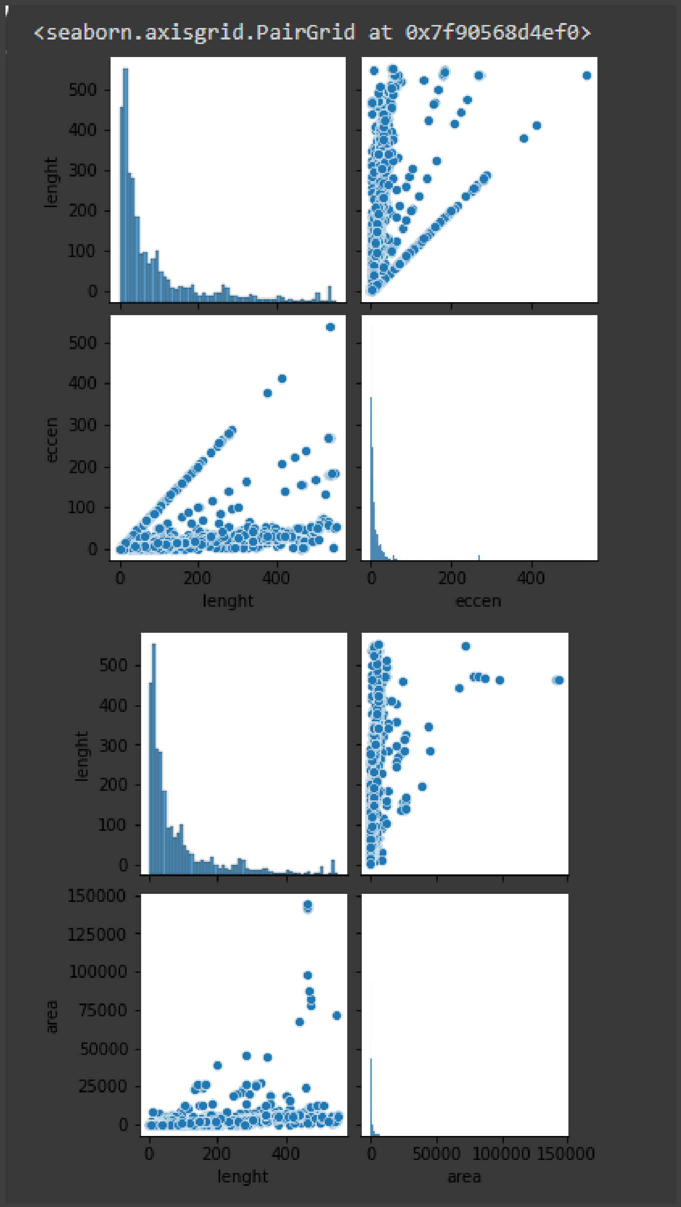
	height	lenght	area	eccen	p_black	p_and	mean_tr	blackpix	blackand	wb_trans
0	5	7	35	1.400	0.400	0.657	2.33	14	23	6
1	6	7	42	1.167	0.429	0.881	3.60	18	37	5
2	6	18	108	3.000	0.287	0.741	4.43	31	80	7
3	5	7	35	1.400	0.371	0.743	4.33	13	26	3
4	6	3	18	0.500	0.500	0.944	2.25	9	17	4
...	...	...	...	...	...	...	...	...	...	...
5467	1	16	16	16.000	1.000	1.000	16.00	16	16	1
5468	4	524	2096	131.000	0.542	0.603	40.57	1136	1264	28
5469	7	4	28	0.571	0.714	0.929	10.00	20	26	2
5470	6	95	570	15.833	0.300	0.911	1.64	171	519	104
5471	7	41	287	5.857	0.213	0.801	1.36	61	230	45

5472 rows × 10 columns

>> ‘Panda.read_csv()’

3. Data Visualization





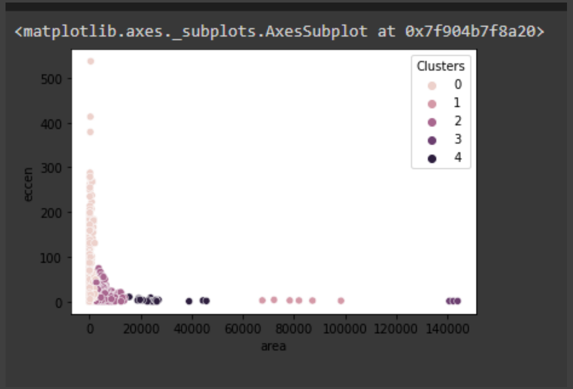
```
<seaborn.axisgrid.PairGrid at 0x7f90568d4ef0>
```

Before we make a model, we need to see data repartition and correlations between parameters.

Displays of correlations shows that a few parameters are not much correlate.

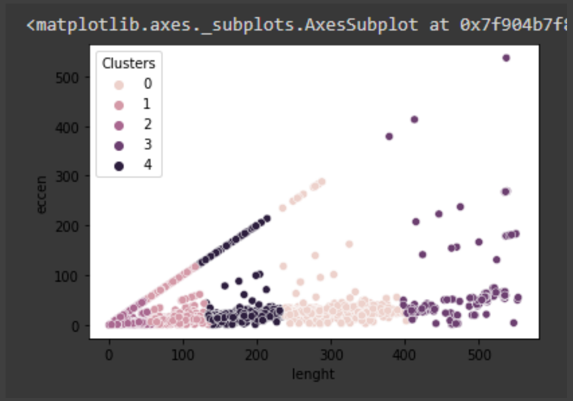
It seems like some features are more correlate than others. We observe relations in order to build our model.

Correlation between area and eccen

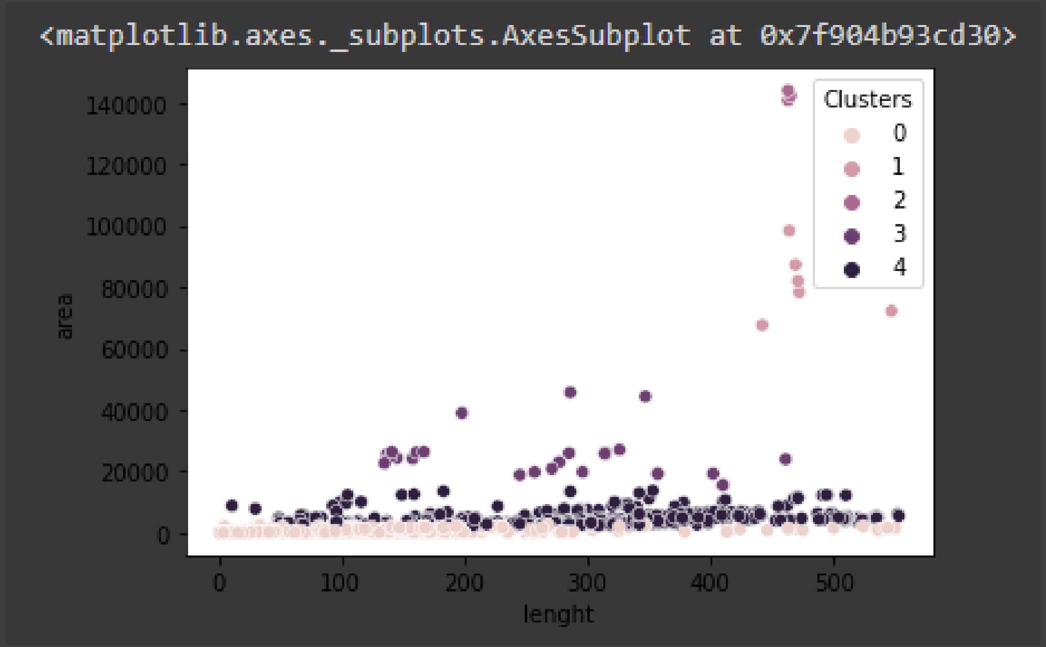


```
<matplotlib.axes._subplots.AxesSubplot at 0x7f904b7f8a20>
```

Correlation between lenght and eccen



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f904b7f
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f904b93cd30>
```

Correlation between area and lenght

- ## 4. Building of the model

As said before, this problem is about predicting a category. Moreover, the data is considered as labeled data as fa as we need to categorize it. The Machin Learning algorithm to use here is a clustering algorithm.

The number of categories is known so we can operate a K-Means algorithm.

Using SciKit Learn we build a model with the K-Means program.

Furthermore, correlation graphs before, expose which pair of features are the less correlate.

For a clustering model we need to maximize the difference between the center of each cluster, so working on the variance, we need to use a pair of feature the less corelate to have a spread model.

Clusters are clearly appearing on graphs.