LEPL1109 - Statistics and Data Sciences
HACKATHON 3 - Clustering: What is it all about?

Deadline: December 22, 2023

Lastname	Firstname	Noma
Antonutti	Adrien	31202100
Arnaud	Batiste	89492100
Delsart	Mathis	31302100
Lebrun	Léa	42072100
Michaux	Bastien	83982100
Remacle	Louis	82162100

Please, read carefully the following guidelines:

- Answer in English, with complete sentences and correct grammar. Feel free to use grammar checker tools such as LanguageTools free and open-source plugin;
- Do not modify questions, and input all answers inside `\begin{answer}...\end{answer}` environments;
- Each question should be followed by an answer;
- Clearly cite every source of information (even for pictures!);
- For bonus material (additional figures, code, very long equations, etc.), use Appendices;
- Whenever possible, use the `.pdf` format when you export your images: this usually makes your report look prettier[1];
- Do not forget to also submit your completed notebook on Moodle.

Contents

[1] This is because `.pdf` is a vector format, meaning that it keeps a perfect description of your image, while `.png` and other standard formats use compression. In other words, this means you can zoom as much as you want on your figure without decreasing image resolution. For simple plots, vector formats can also save a lot of memory space. On the other hand, we recommend using `.png` when you are plotting many data points: large scatter plots, heatmap, etc.

Context and objectives

The objective of this hackathon is threefold: (1) extract meaningful information from a dataset, (2) observe relationship(s) (if any) between features and eventual underlying groups (clusters), and (3) develop an unsupervised clustering tool and exploit the associated data.
To this end, you will use a real dataset (available on **Moodle**) fetched from a observation bank. Given a couple of features, you should be able to **create insects' clusters based on spatial coordinates, temporal informations and other prodived features. Then exploit the content of these different clusters to determine the likeliness of encountering a specific insect for some input requests such as time and position**.

Figure 1: Website Logo.

Nowadays, it is easier than ever to record observations about the world that surrounds us. `iNaturalist` stems as a platform that collects pictures, timestamps and locations of fauna & flora observations made by any volunteer willing to improve our knowledge about current nature's status. The principle is pretty straightforward. You, as an user, wander around and suddenly a rare species appears. Your goal then is to take as much as possible[2] measurements, including an eventual picture, that will then be stored on the platform. Some data can be added afterwards, i.e. the *taxonomy hiearchy* as it is explained in details in your Jupyter Notebook.

[2] As long as the tranquility of the observed is preserved.

Questions and Answers

1 Data Preprocessing

Question 1.1: Removing unnecessary features

Can you already, a priori, detect that some features are useless? **Pay attention** to eventual data restrictions already performed at this stage.

1. if yes, list those (useless) features and explain your choice;
2. if not, then explain why it is better to wait.

Generally speaking, is it a good idea to remove a feature based on a priori knowledge, or it doesn't alter the final outcome?

Expected answer length: 5-8 lines.

Answer to 1.1

In this section, we focus on a specific subset of the training dataset, which is exclusively concerned with the "Insect" class. The objective here is to identify the order of the observed insect. Based on this focus, we can eliminate the features `kingdom`, `phylum`, and `class` as they each hold only a single unique value, given that the taxon level of these features is higher than our target taxon (`order`). Similarly, we can also remove the features `family`, `genus`, and `species`, as these represent lower taxonomic levels compared to our target taxon. Predicting a higher taxonomic level is only meaningful if we lack information on the lower levels. Furthermore, we decide to discard the `stateProvince` feature since we have access to more relevant data through `decimalLatitude` and `decimalLongitude`, which provide precise geographical coordinates. As a general rule, it's prudent to immediately remove features that are redundant or singularly valued. For the remaining features, it's advisable to conduct further tests before making a final decision on their inclusion or exclusion. In our analysis, we retain the features `decimalLatitude` and `decimalLongitude` (indicating the origin of the insect), `eventDate` (the time of insect observation), and `order` (our target taxon).

Question 1.2: Handling missing data

Given the dataset and the amount / type of missing information, what strategy do you propose to follow regarding missing data (NaNs)? You can choose one or many of the following:

1. drop features (column) with missing information;
2. drop samples (row) with missing information;
3. replace missing information with interpolation / extrapolation / simple substitution / ...

Expected answer length: 4-6 lines.

Answer to 1.2

The count of features containing NaN values in the analyzed section of the dataset is zero. Hence, we can assume that the potential number of NaN values in the remaining dataset (*test_set*) is extremely low, particularly considering that the size of the test set is smaller than that of the training set. Therefore, it can be considered negligible. Consequently, we can opt for a data cleaning approach that entails simply eliminating rows containing at least one NaN value.

Question 1.3: New features

What features have you added? Please explain any particular applied manipulation.
Expected answer length: 3-6 lines.

Answer to 1.3

We have introduced five new features into our dataset:

1. `year` (integer representing years)
2. `time_of_day_in_hours` (floating-point number representing the hour/minute of the day)
3. `day_of_year` (integer representing days of the year, ranging from 0 to 365)
4. `day_night` (integer representing the period of activity, where day belongs to [6, 18] and night belongs to [18, 24] U [0, 6])
5. `season` (integer representing the season of the year)

Given that `day_of_year` and `time_of_day_in_hours` are cyclical features, we have also performed the necessary transformations to make these values cyclical.
$feature\_sin = \sin\left(\frac{2\pi \times feature}{T}\right)$ and $feature\_cos = \cos\left(\frac{2\pi \times feature}{T}\right)$ where T is the period of the respective features.

2 Data Visualization

Question 2.1: Features visualization

Based on what you have seen in your notebook and whatever other visualization you will try, you can already get an idea of which features seem to contain discriminative information. Justify which features you think would be interesting or not to keep in order to realize the required task. Feel free to try and add your own data visualization to highlight or not their importance. Nevertheless, we ask you to avoid removing `decimalLongitude` and `decimalLatitude`, both extensively required in **Question 2.2**.
*Expected answer length: 6 lines depending on answer in **Question 1.1** and 0-2 image(s).*

Answer to 2.1

Based on the various plots created, we observe that some features are not relevant in this analysis.
In the first graph, we can see that the `years` feature lacks relevance because there are only four years of study.(low temporal variability)
Furthermore, the features `day_night` and `seasons` are too categorical to be used in subsequent analyses, such as PCA.
Additionally, after normalizing the number of observations for each insect order during the day and night and across seasons, we realize that they are almost all observed in nearly the same proportions. Therefore, we cannot extract additional information from these two features due to there low variability over the target feature.

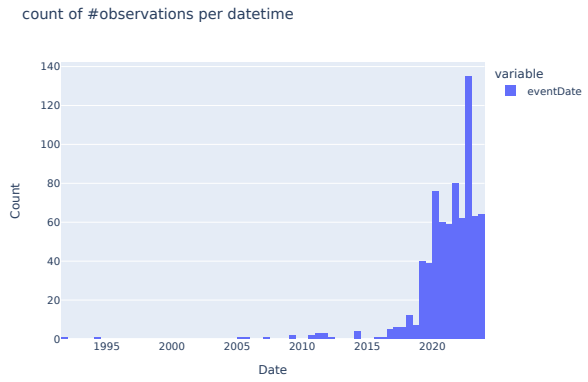


Figure 2: Number of observations per datetime

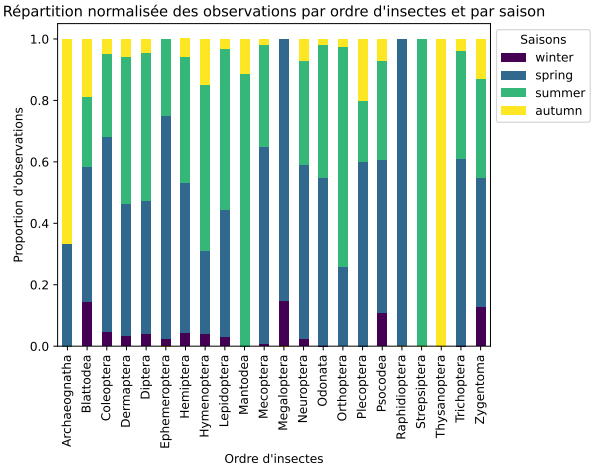


Figure 3: Normalized distribution of observations by insect order and by season

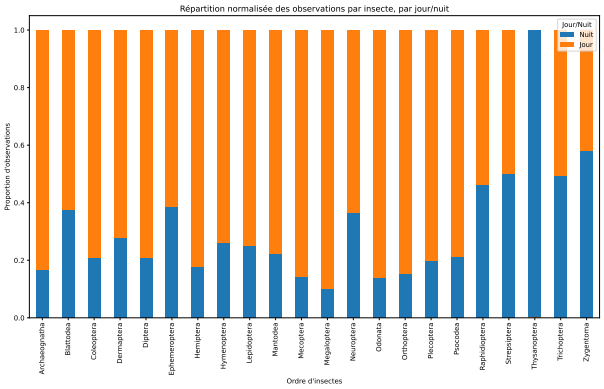


Figure 4: Normalized distribution of observations by insect order and their activity period (day/night)

Question 2.2: Spatial features visualization

Based on the maps from your notebook, what can you infer about the spawn locations of (some of the) insects? Is there a link/pattern between their `order` and where they are observed? If yes, explain.
Expected answer length: 3-5 lines.

Answer to 2.2

The majority of insect orders observed in Belgium exhibit a uniform distribution across the entire territory, as shown on the map of insect observations in Belgium. However, a notable exception is observed for the order Hymenoptera, which is significantly more prevalent in the northern part of the country, in Flanders.

It can be concluded that when observing a specific geographical area, there is no obvious relationship between that location and the insect order that may be observed there. Except for the order Hymenoptera, which stands out from the others by a higher concentration of observations in the northern part of Belgium compared to the southern part.

Question 2.3: Feature importance visualization

Looking at the biplot graph(s) (that you can include) you generated in your Jupyter Notebook, do all features have the same importance? If no, which features are less important and why? Could you explain based on correlations with spatial coordinates towards which direction (e.g. North-East (NE)) does **PC1** (first principal component) point to ? Your final answers should be based on the whole dataset, not a fraction.
Expected answer length: 3-6 lines + 0-2 image(s).

Answer to 2.3

The features don't all have the same importance.
The features `day_of_year` and `time_of_day_in_hours` can be removed as we have previously created their cyclical copies. Through PCA, it can be observed that the importance of features depends on their length. It is also noticeable that some features are opposed to others, indicating that they vary in opposition. For instance, the geographical coordinates (`decimalLatitude` and `decimalLongitude`) are opposed due to the inclination (approximately 45° with a negative orientation) of Belgium.
It is most relevant to discard features for which PCA returns the smallest segments. There are several of them, including `day`, `decimalLatitude`, and `decimalLongitude`.
The PC1 axis points to the North because there is a higher concentration of Hymenoptera on the right, corresponding to Flanders and thus the northern part of the country

Answer to 2.3

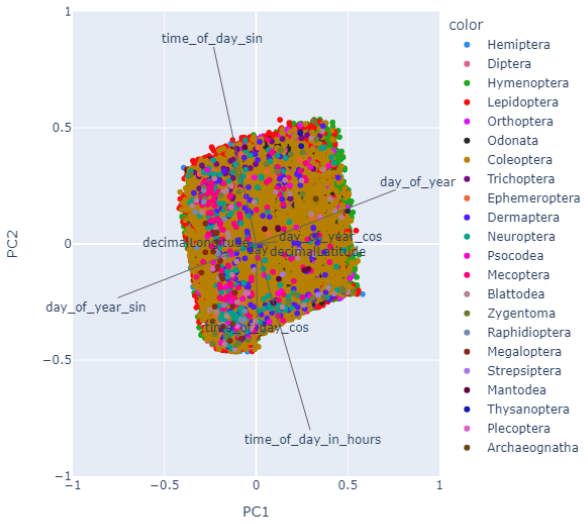


Figure 5: PCA Filled

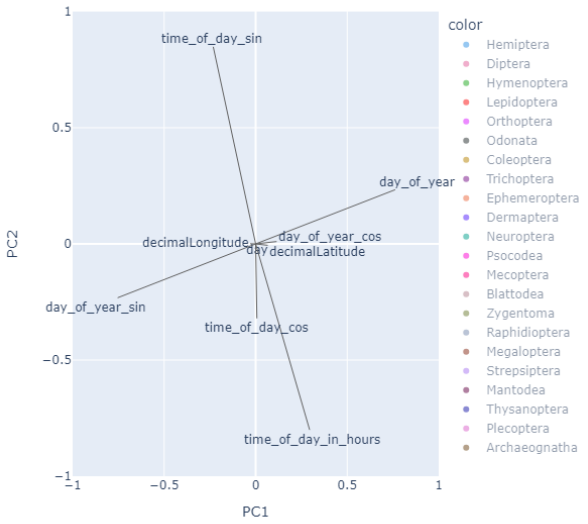


Figure 6: PCA Empty

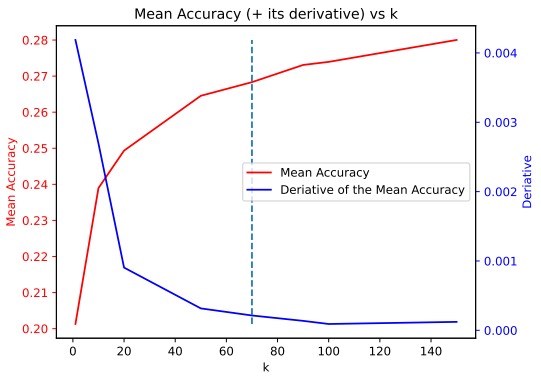
3 Clustering

Question 3.1: Number of clusters

Accounting for all features (i.e. spatial **and** temporal coordinates), what do you think is the ideal number of clusters ? What will happen if too many / few clusters are chosen ? *Expected answer length: 4-8 lines.*

Answer to 3.1

The optimal number of clusters, determined as $k = 70$, was chosen based on visual assessment of K-means accuracy against cluster numbers, identifying the elbow point on the graph.
If **too high** number of clusters: **overfitting** (captures noise, lacks generality) and **difficult interpretation** (too many separations hinder visualization). If **too low** number of clusters: **loss of information** (hides important patterns) and **poor representation** (overly broad clusters yield unreliable models).



Question 3.2: Cluster composition

Currently, we suggest returning a list of probabilities, not a true unique prediction.
How could you return a prediction `y_pred` given an input observation vector `x` ?
Also, what do you think would be the best way to do so (based on a binary score) ?
Expected answer length: 2-6 lines.

Answer to 3.2

Using the observation vector `x`, we can identify the cluster to which the observation belongs. Subsequently, we can choose the class with the highest probability of appearing in that cluster as the prediction.

Question 3.3: Your clustering solution

Describe here your clustering solution (how many clusters, which method of sampling within clusters, ... etc.). Justify your choices with the help of the metric.
Expected answer length: 4-8 lines + 1 image(s).

Answer to 3.3

As mentioned in Section 3.1, we have implemented the K-Means algorithm with 70 clusters for training on the training dataset. Initially, we modified the vector `x` by removing features deemed irrelevant in the previous sections. This allows for the effective application of the K-Means algorithm to `x` to determine its cluster. Then, we predict the probability of each test data with the cluster. Finally, we apply the accuracy metric to our test data. As a result, the mean accuracy is 0.0913 and the standard deviation is 0.1314. The mean accuracy is significantly smaller than in question 3.1. We can therefore deduce that the model works much less well with a new dataset.

Question 3.4: Comparing models - BONUS

Compare how your model performs when predicting other taxon levels. I.e. does predicting taxonomy at a different level performs better or worse than the default?
Expected answer length: 2-6 lines + 0-4 image(s).

Answer to 3.4

When we try to predict family, we obtain an average accuracy of 0.02878 and a standard deviation of 0.06865 with 70 clusters. We can see that these numbers are lower than when we predict order, because there are much more family than order. And this is the case for all the other taxon levels. They are lower in the hierarchy. Therefore, there are more categories in them than in order. So, it is harder to predict their categories with kmeans.

A Demo

A.1 Interesting questions

Question Demo:

Can you show me what I can do?

Answer to Demo

This is how I answer to Demo. I can cite [**?**] content that I use or refer to A. I can also reference images such as in Figure 7, or equations with (1) or Equation 1.

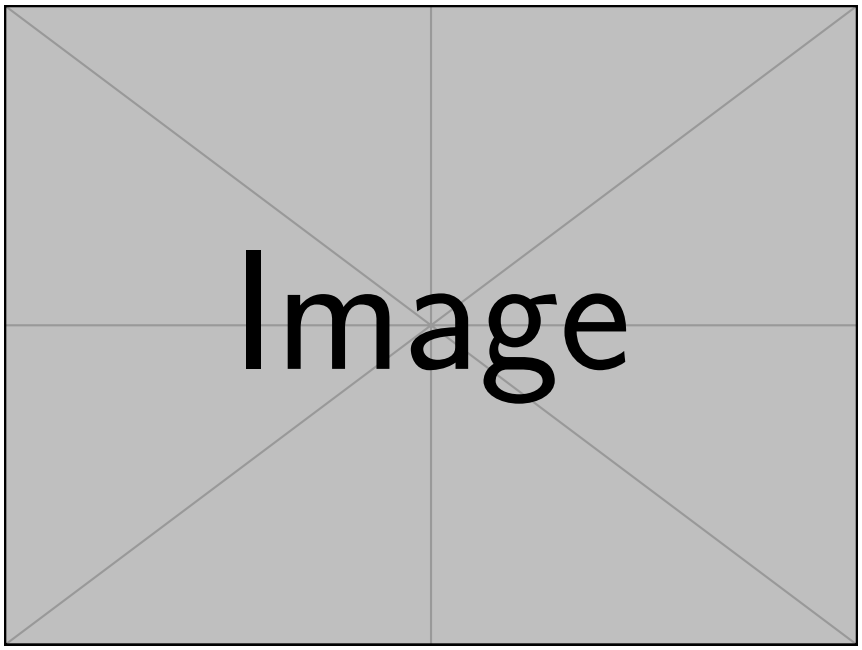


Figure 7: Demo caption.

$$E = mc^2 \tag{1}$$

If you wish to present code samples, you can either use the Listing 1 format or use inline code `import numpy as np; x = np.arange(10)` if this better suits your needs. However, we recommend putting your code in the Appendices.

```
1  import numpy as np
2
3  x = np.arange(10)
```

Listing 1: My super code.

Note: syntax highlighting for code is provided by the `minted` package. If you are not using Overleaf, you might need to install some requirements before it can work.