LEPL1109 - Statistics and Data Sciences
HACKATHON 4 - Clustering: What is it all about?

December 9, 2022

Lastname	Firstname	Noma
Pirot	Alexandre	53811900
Defrère	Sacha	51621900
Ketels	Mathéo	23782000
Hautier	Thomas	80162000
Goffinet	Dylan	08471900

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

Context and objectives **3**

Questions and Answers **4**

1 Data Preprocessing . . . 4
- 1.1 Removing unnecessary features . . . 4
- 1.2 Handling missing data . . . 4
- 1.3 New features . . . 4

2 Data Visualization . . . 5
- 2.1 Features visualization . . . 5
- 2.2 Spatial features visualization . . . 6
- 2.3 Spatial clustering . . . 6
- 2.4 Feature importance visualization . . . 7

3 Clustering . . . 8
- 3.1 Number of clusters . . . 8
- 3.2 Cluster composition . . . 8
- 3.3 Your clustering solution . . . 9

[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.

3.4 Comparing models - BONUS 9

References **10**

Appendix A Demo **11**
A.1 Interesting questions 11
Demo 11

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 synthtic dataset (available on **Moodle**) inspired from a real dataset from Kaggle. Given a couple of features, you should be able to **create Pokémon clusters based on spatial coordinates, temporal informations and other prodived features. Then exploit the content of these different clusters to determine the likeliness of capturing a given Pokémon for some input requests such as time and position**.

Figure 1: Context illustration.

Nowadays, mobile games are gaining more and more attention, sometimes maybe too much. This is especially true for the well known Pokémon GO.

For those who do not know: Pokémon GO is a mobile-game in which players have to capture as many Pokémon as possible. Pokémon are creatures that randomly spawn (i.e., appear) at different positions and times, but some locations are more likely to have Pokémon appearing: shopping malls, city centers, parks, and so on. Once a Pokémon has spawned (i.e. appeared), the players have to physically go to the same place as the Pokémon to hopefully capture it.

As a casual Pokémon GO player and a proficient data scientist, you would like to increase your level by leveraging some data-related techniques. To this end, you found a Pokémon GO spawning versus localization dataset that you will use to, hopefully, achieve your goals (see above).

Questions and Answers

1 Data Preprocessing

Question 1.1: Removing unnecessary features

Can you already, a priori, detect that some features are useless?

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: 2-4 lines.

Answer to 1.1:

It is very often better not to delete any feature *a priori*, as we could very well delete some features that appear trivially useless to us, but that could in fact give a lot of information on our data.

The only feature we could safely prune is either the name or the number in the pokédex feature, as they are complete duplicates of each other, thus giving no additional information. However, since numerical values are always useful AND the name feature was used throughout the hackathon, we pruned neither.

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: 4-8 lines.

Answer to 1.2:

For each NaN appearance, given that NaNs only appear in the *appear_duration* feature, one possible strategy that seems to fit our dataset is to compute the average value of *appear_duration* for this particular pokemon species and put it instead of the NaN, thus still benefiting the non-NaN data.

However, as the level of NaN is quite low (around 2% of total *appear_duration* data contains a NaN), we simply decided to delete the samples that contain any as the loss of information is neglectable. This will result in a smaller dataset, but the fact that we didn't assume some *appear_duration* means that there is no chance of inducing errors in our dataset with our possibly wrong interpolation.

Question 1.3: New features

What features have you added? If a particular manipulation has been applied, please explain.
Expected answer length: 2-4 lines.

Answer to 1.3:

Similarly to the previous hackathon, we added new time-related features to make the date more usable, using a 24h cycle indicating the hour of spawn, and an integer indicating the day of spawn. We also added a third feature that represents the pokémon's type as an integer.

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, i.e., which features are likely to be more important for the clustering than others.
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.
Expected answer length: 2-8 lines and 0-2 image(s).

Answer to 2.1:

Via the 4 graphs below, we evaluated the relevance of 4 features for our clustering, which are the datetime, the type, the name and the appearance duration. Although the latitude and longitude features also seemed important to evaluate, we did not do it in this question as the questions 2.2 and 2.3 are already tackling that feature.
As we can see below on the figure 2, the appearance_duration feature seems to be uniformly distributed (except for the extreme values), so we might conclude that this feature does not give any discriminative information and will not be helpful for the clustering.
From the wordcloud on figure 3, we can see that the pokemon distribution frequency is clearly dicriminative and gives important information. This feature allows us to find the probability to encounter each pokemon, thus it is an important one for our clustering.
From the spawn per datetime on figure 4, we can see that the preferential times for spawns are from 11pm to 1am and from 11am to 1pm, around which the spawn rate is distributed like a normal distribution. This information is very valuable as it allows us to predict the time at which there is the better chance for a pokemon to spawn, and also allows us to differentiate pokemons that spawn at night and during the day, which is useful for our clustering.
Finally, we can see on the figure five that the types of the pokemon is also a feature that differentiates them heavily, which would be beneficial for our clustering.

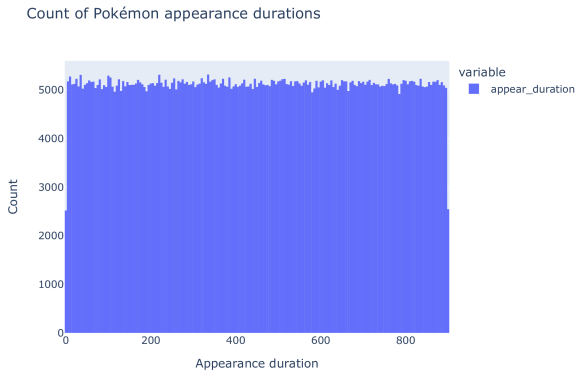

Figure 2: Pokémon appearance durations

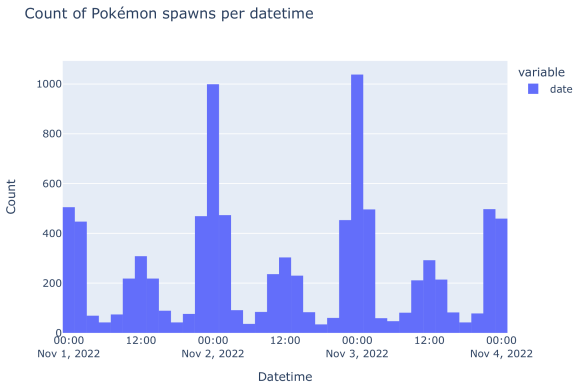

Figure 3: Pokémon spawns over time

Figure 4: Pokémon frequency

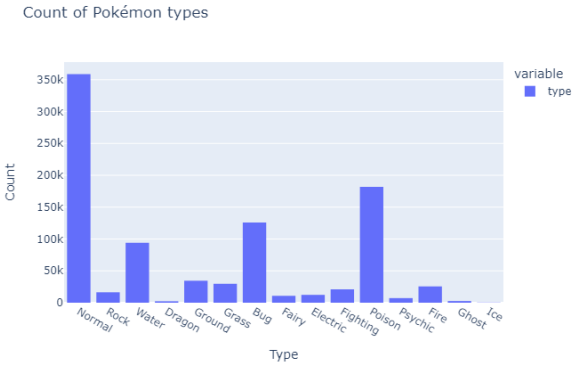

Figure 5: Pokémon types

Question 2.2: Spatial features visualization

Based on the maps from your notebook, what can you infer about the spawn locations of the Pokémon? Is there a link between their types and the land affectation? If yes, explain.
Expected answer length: 2-4 lines.

Answer to 2.2:

Based on the map, we directly see a clear and heavy correlation between the type of the pokemon and the type of the environment it has spawned in (*e.g.* water type pokemon almost exclusively spawn near bodies of water, which are labeled as water type environments, and almost exclusively water type pokemon spawn on terrain labeled as water type).

Question 2.3: Spatial clustering

Based on the maps above, i.e., on the spatial features only, how do you think a clustering will perform according to the number of clusters? In other words, what do you think will

happen with 1, 15 (= number of types), 100+ clusters?
Expected answer length: 2-4 lines.

Answer to 2.3:

In the case of a too small number of cluster (≤ 5), we estimate that the clustering will not be efficient, as our geographical distribution is too variate and complicated to be split in only a few groups. On the other hand, a too big number of cluster (≥ 100) will also be unefficient as the clustering will attempt to divide coherent groups and will be too specific. This is the problem of underfitting and overfitting.
Our approach was to take a number of clusters close or equal to the number of big cities nearby, so that the clusters might be associated with each city, which would create a coherent and close to the game clustering. We deemed 20 to be an adequate potential number of clusters.

Question 2.4: Feature importance visualization

Based on the biplot graph you generated in your Jupyter Notebook, do all features have the same importance? If no, which features are less important and why? You can use all other graphs from the visualization part to justify your answer.
Expected answer length: 2-6 lines + 0-2 image(s).

Answer to 2.4:

Although all the features that we kept have a non-neglectable impact on our principal components, we can still distinguish two that have a strong impact on the first component (and almost null on the second), namely the latitude and longitude features, and four that have a moderate impact on the second component, namely the hour, day, number in the pokedex and appearance duration features. Those features and their impacts are represented on the two graphs below, where we can see that trying to guess the name or the type of the pokemons doesn't change the loadings that much.
It is important to note that those results might be a bit biased, since the number in the pokedex has a correlation of 1 with the name of the pokemon, probably inducing some unwanted effects.

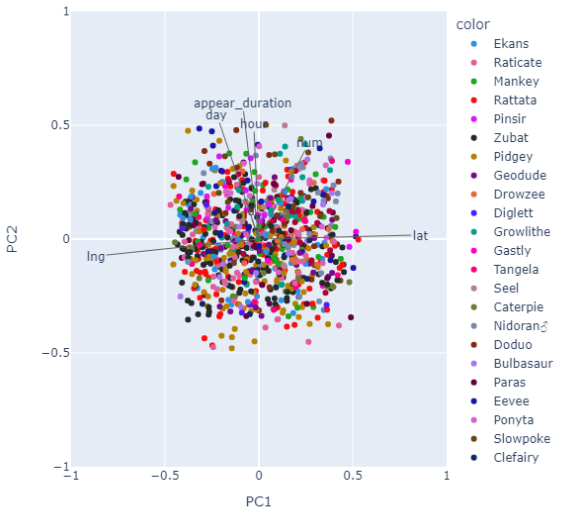

Figure 6: 2D PCA using name as output and associated loadings

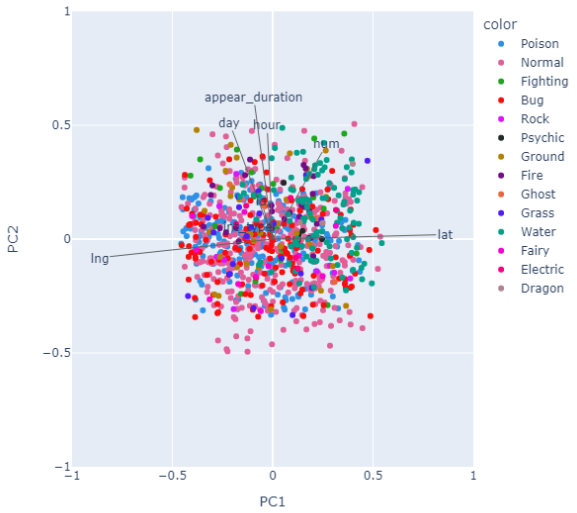

Figure 7: 2D PCA using type as output and associated loadings

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 or even too few clusters are chosen?
Expected answer length: 2-6 lines + 0-2 image(s).

Answer to 3.1:

If we use the temporal coordinates to evaluate the ideal number of clusters, we see that the number of spawns per datetime reaches a peak at 12pm and 12am. Following this logic, we can double the number of clusters by differentiating each spacial coordinate in 2 other clusters, the first at 12pm and the second at 12am. According to our answer to the question 2.3, that would double our estimation, so around 40 clusters.

Question 3.2: Cluster composition

Do you think the naive approach will give the best results? Justify briefly.
What do you think would be the best way to estimate the Pokémon you encounter? Explain.
Expected answer length: 2-4 lines.

Answer to 3.2:

This approach certainly won't give the best results. Predicting a single pokemon species or type while our dataset is way more complex than that is guaranteed to be inaccurate. In fact, its accuracy will be bounded by the occurence percentage of the top pokemon into the dataset, being Pidgey making approximatively 38% of the dataset.
The best way we found to estimate the Pokémon encounters is the following : each cluster we created is assigned a presence percentage for each Pokémon in it (if there is one Pidgey among 19 other Pokémons in a given cluster, that cluster will have a 5% Pidgey presence percentage). When attempting to predict an encounter in any cluster, we will return an array containing each Pokémon listed a number of times proportional to its presence percentage in the cluster. This will ensure that, similarly to the naive solution, we have the top appearing pokemon, but also the 2nd, 3rd etc most appearing to drive the accuracy of our prediction up.

Question 3.3: Your clustering solution

Describe here your clustering solution (how many clusters, which method of sampling the Pokémon, other important choices that have been made, etc.). Justify your choices with the help of the metric. *Expected answer length: 4-8 lines + 0-2 image(s).*

Answer to 3.3:

Our solution has been to modify the naive proposed solution to make it better. Instead of filling the list for each cluster with the name of the most represented pokemon in our prediction, we decided to fill each list proportionally with the name or type of the pokemon based on its appearance percentage in the said cluster (*e.g.* for $n = 100$, let's say that Pidgey form 38% of the cluster, Rattata 22%, Walruss 20% and Pikachu 20%, the returned list for this cluster will consist of 38 entries named 'Pidgey', 22 Rattata, 20 Walruss, and 20 Pikachu). To choose the best k for the K-Means, we simply made tests with different values of k until finding the one giving the best accuracy, which turned out to be $k = 15$.

Question 3.4: Comparing models - BONUS

Compare how your model performs when predicting the Pokémon types, based on `output="type"` versus `output="name"`. I.e., does predicting Pokémon types based on the Pokémon name performs better than directly predicting the types?
Expected answer length: 2-6 lines + 0-4 image(s).

Answer to 3.4:

As can be seen on the figure 8 below, our kmeans algorithm is able to consistently predict the Pokémon name with an accuracy above 47%, and the Pokémon type with an accuracy above 62%. It is then clear than predicting the type based on the name performs significantly better.

With 0,1% of the dataset

	K = 1		K = 5		K = 15		K = 30		K = 50		K = 100	
Naive approach	11,457	38,859	10,859	35,576	10,424	33,641	10,13	30,109	9,435	32,011	7,424	29,717
Better approach	0	84,239	57,304	77,771	50,565	73,272	45	69,152	38,913	60,793	27,924	53,272
Name-type mean for naive	25,158		23,2175		22,0325		20,1195		20,723		18,5705	
Name-type mean for better	42,1195		67,5375		61,9185		57,076		49,853		40,598	
	Name	Type										

With 1% of the dataset

	K = 5		K = 7		K = 9		K = 11		K = 13		K = 15		K = 20	
Better approach	47,793	62,807	47,536	63,046	48,013	62,704	48,826	63,294	47,61	63,585	48,71	63,537	47,756	63,334
Name-type mean	55,3		55,291		55,3585		56,06		55,5975		56,1235		55,545	
	Name	Type												

Figure 8: Kmeans accuracy with various k values and output choices

References

[1] Laurent Jacques and Thomas Feuillen. The importance of phase in complex compressive sensing, 2020.

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 [1] content that I use or refer to A. I can also reference images such as in Figure 9, or equations with (1) or Equation 1.

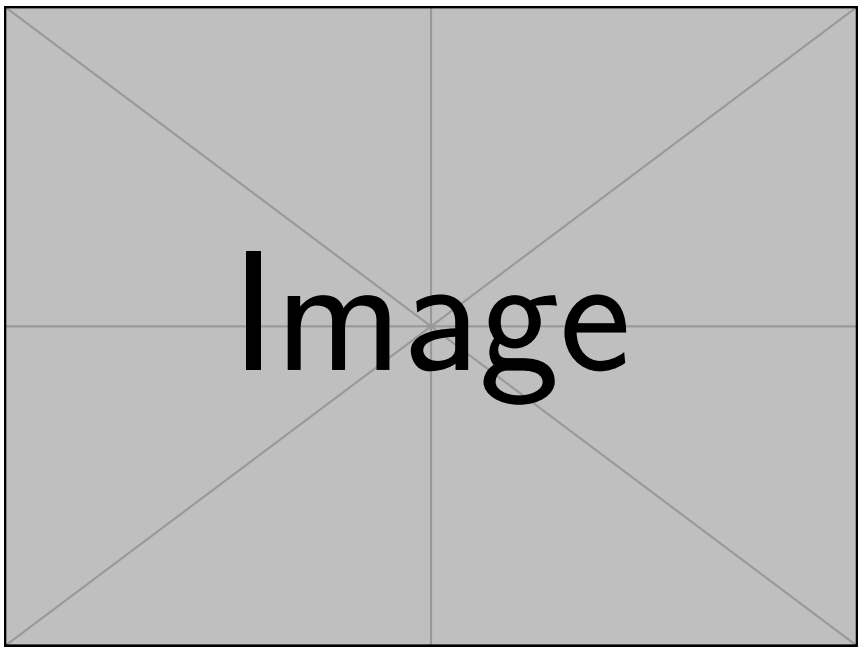

Figure 9: Demo caption.

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

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.