NitrogenPandas

– Hack4Good –

submitted by

Agustina La Greca, Lluìs Pastor Pérez, Phillip Trummer, Lilian Bonnet

November 29, 2022

Agustina La Greca, Lluìs Pastor Pérez, Phillip Trummer, Lilian Bonnet
asaintest@student.ethz.ch, lpastor@student.ethz.ch, pstrummer@outlook.com, lbonnet@student.ethz.ch

Acknowledgements

We would like to thank our mentor, Stephan !! For the great help and guidance, as well as the whole organizing team of *Hack4Good* for their amazing work. Also special thanks to our mentor Nikolai, who always was ready to give a hand.

Contents

List of Figures

1 Introduction

Nitrogen emissions damage various sensitive ecosystems such as forests, dry meadows, or peatlands. Resulting from these emissions, eutrophication and soil acidification threaten biodiversity and weaken the resilience of the ecosystems to biotic and abiotic threats, such as droughts and insect pests. Agriculture is the main source of these emissions. In particular, ammonia emissions from animal husbandry negatively impact nearby ecosystems due to the limited dispersal. The reduction of ammonia emissions is usually only possible when there is a construction project on the farm. Thus, WWF and other environmental NGOs regularly control whether these projects respect the environmental law. If not, they ask through a legal complaint to do so. Due to the large number of corresponding applications, the various publication sources and the limited personnel resources, a systematic preliminary examination of building applications is not possible.
To help WWF looking through all applications, we created a pipeline, easily usable, to identify building permits with potentially adverse impact on biodiversity, which thus require further investigation.

2 Expected impact

With the implementation of the Graphic User Interface, we aim at helping WWF to look in an easily fashion for building projects potentially non-complying with the law. Indeed, the interface is very intuitive to use and the user can choose over many inputs. Once launched, it automatically reviews all the interesting canton documentations (Amsblatts) and outputs in an Excel sheet all the building projects related with the search (see following sections for the details). Thanks to this, WWF is able to have a quick overview of the main biodiversity endangering projects. Hence, they can focus their attention on them, instead of going through every building projects. It enables a stronger analysis of them and a time saving (for instance, only in the ZH canton, during one week there are generally more than 150 new building projects). We can also imagine that in the long term, if many of their appeals are successful, cantonal authorities will be stricter in the regulations of attribution of building permits.

3 Approach/Deliverables

Our approach to answer the given problem is composed of a Graphic User Interface that makes use of several intermediate steps to output in an Excel sheet the list containing all building projects in relation with the search with some useful information about them. We detail all our solution in the following sections.

3.1 Graphic User Interface

The Graphic User Interface (GUI) can be found in the script *gui.py* and it is the main backbone of the code. We provide an executable called *gui.exe* located in the *exec* folder, which wraps this function without the need for Python installation.

In the GUI, there are several options for the user to specify the search criteria. Firstly, the user must select the starting and end dates (Figure 1).

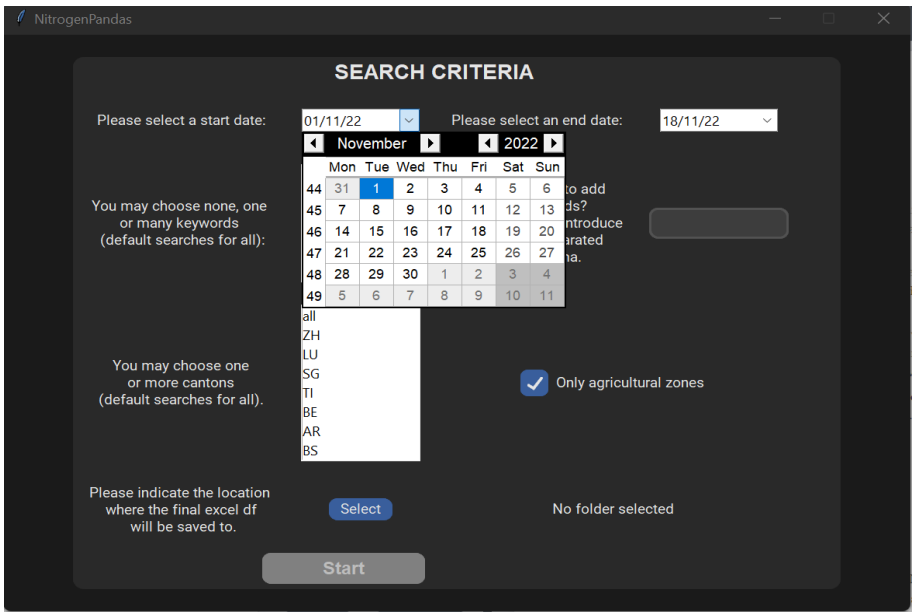

Figure 1: Starting date selection.

Secondly, the user may select none, one or several keywords from the scrollable list provided and/or manually input the desired keywords. Keywords may be words or numbers and should be entered separated by a comma (Figure 2). If no keywords are selected or introduced, the program will search for the provided list (Stall, Bauernhof, etc.) as default.

Next, the user may select all, one or several cantons from the provided list (Figure 3). The default will search for all of the cantons in the list.

The option for only searching for agricultural zones is checked by default. The user may uncheck it if other additional areas are of interest.

Finally, the program will only start once a folder has been selected where the final excel file will be saved. Afterwards, the user may click on the "Start" button and a progress bar will appear (Figure 4). It is important to note that the program cannot be stopped while it is performing the search. If one may wish so, then the program needs to be closed and restarted.

Once the program has finished, the button will say "Done" and the final excel will appear on the previously specified location (Figure 5). If there was a previous Excel named

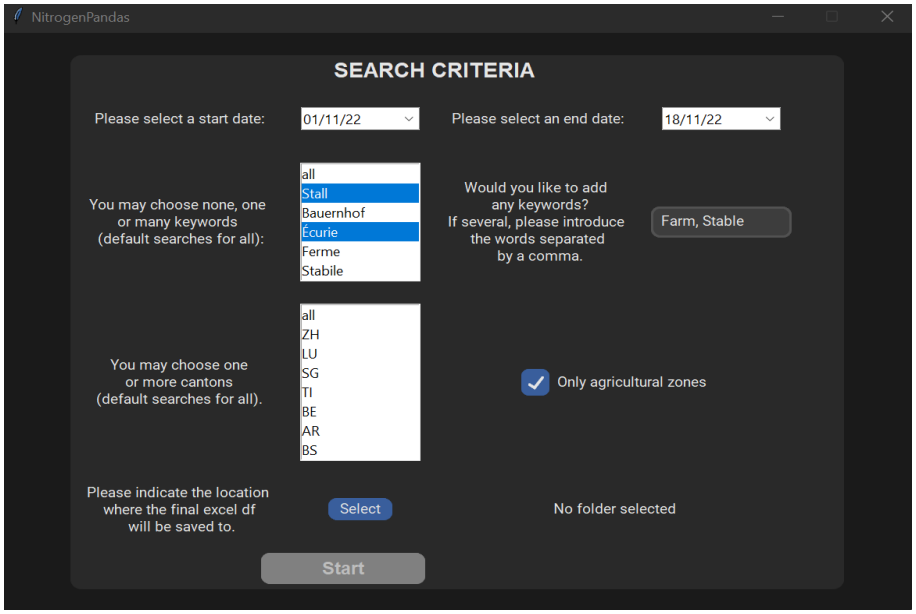

Figure 2: Keyword selection.

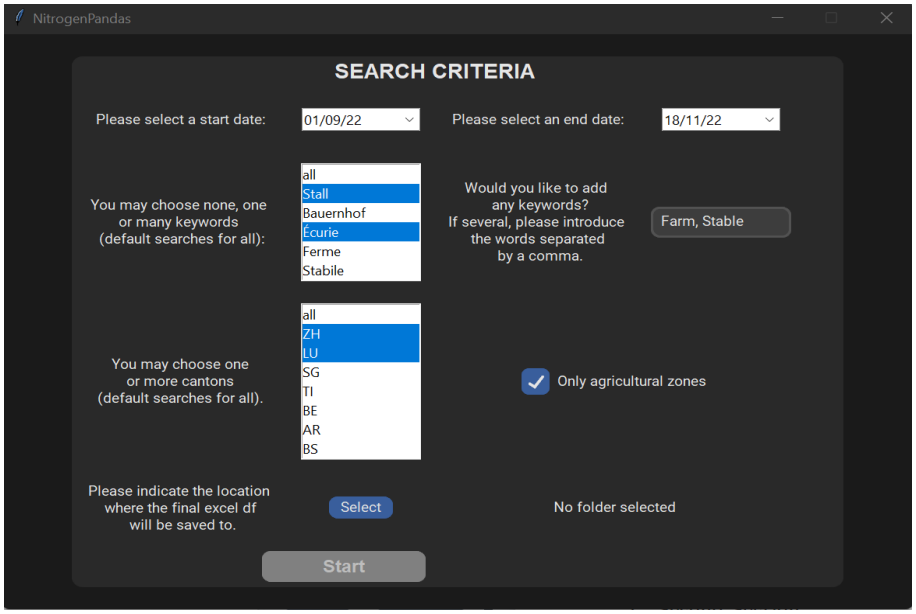

Figure 3: Canton selection.

"output.xlsx" with data in that folder, the program will read it, concatenate the new file to the previous file and eliminate any duplicates.

3.2 Web Scraping

To find the corresponding building projects, the GUI makes use of web scraping to go through the different Amsblatts that we have access to. We explain a bit more how in this

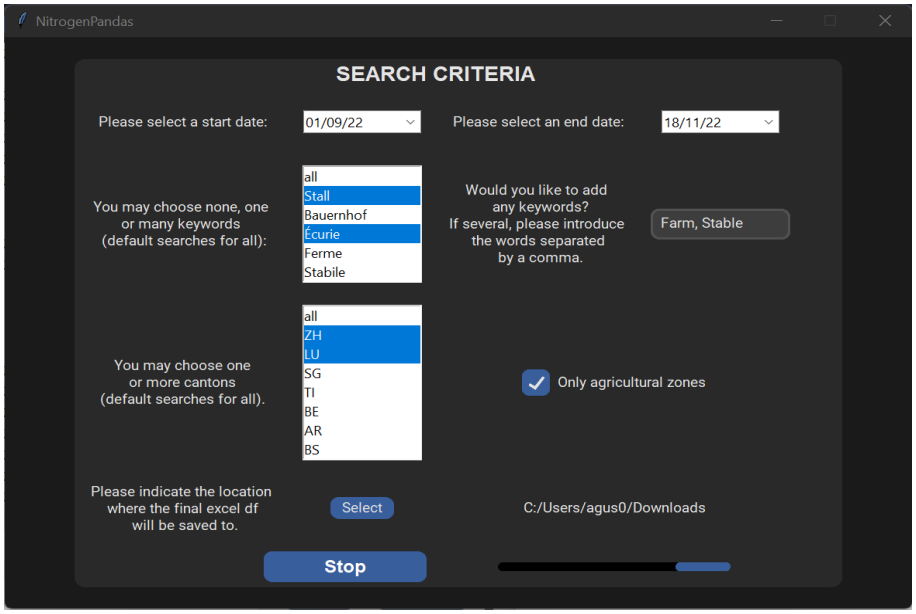

Figure 4: Start of the search after folder has been selected to save the output Excel file.

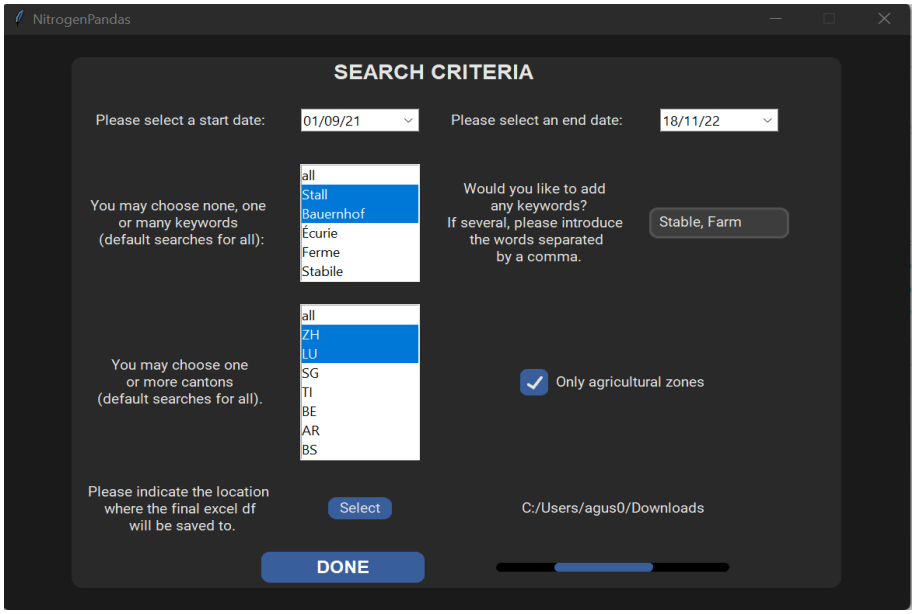

Figure 5: Interface after program has ended.

section. At the moment, the program searches through one API (which can be used for cantons Zurich, Bern, Basel, and Appenzell Rodelbahn) and the official website of Luzern.

3.2.1 API Scraper

The code scrapes the website https://amtsblattportal.ch/api/v1/publications/xml? by modifying the URL based on the search criteria (start and end date, cantons, keywords,

etc.). First, all XML files corresponding to the permits that fulfill the search criteria are collected. Then, the text contained on each XML file is evaluated, from which the following relevant keys are extracted: publication ID, registration office ID and name, publication number, date and expiration date, canton, building contractor, project description, project location, and cadastre. Some of this information may be absent, but it is important that the address of the project is known, as the last step of the code is to calculate the latitude and longitude coordinates based on the address.

3.2.2 Luzern's website Scraper

The code scrapes the website https://www.luzernerkantonsblatt.ch/Public/Kantonsblatt?isOutsideIFrame=True by modifying the URL based on the search criteria (start and end date). First, all the "Amtsblätter" are cropped to only include the section "öffentliche Planauflagen". The distinct building projects are divided by roman numerals. A large part of the code is allocated to finding the different variants of these numerals as the "PyPDF2" python library is not consistent in converting pdfs to strings. The building projects are then filtered by the given search criteria and the "Gemeinde", project description and location are extracted. The last step of the code is to calculate the latitude and longitude coordinates based on the address.

3.3 Distance Calculation

Once all previous information has been found about the projects of interest, we then compute the distance to the nearest sensitive ecosystems, thanks to geospatial data obtained through cantonal sources (where the point for the computation is taken at the address of the project). After careful consideration of the available data we considered two main types of ecosystems that we consider they will not be updated in the short/medium-term: raised bogs and forests.

3.3.1 Method used

First of all, both datasets come with the Swiss coordinates. As the data from the web-scrapping (section 3.2) comes in (Latitude, Longitude) format, there are some functions that transform Swiss points into the standard format.

The distance is computed as follows: given a point and a multipolygon (or an area that is the result of multiple polygons), the distance is the Haversine geodesic (i.e., the route simulated on the Earth that joins two points with minimal distance) between the point and any of the points of the boundary. Another (much faster) method could be consider the distance from the point to the centroid, but this is not optimal as most of the surfaces are sparse and the results could not be reliable. Our choice of the distance could be

computationally expensive, so we have created a function that reduces the number of points of the boundaries so that, with high probability, the results can be computed much faster. In every test we have done, the computation is not too long, so the default method is not to use this helper function.

We have created two functions to look for the closest forests and raised bogs respectively. The default value is to return the closest forest/raised bog given a point. There is a possibility to output the n desired closest ones (say, $2, 3, 25$), but the rest of the code should be adapted if more than 1 output is wanted.

3.4 Emissions Calculation

After having computed the distance to the closest forest and raised bog for each building projects of the output Excel sheet, we use a given model to compute the maximum possible emission of ammonia for the corresponding project to not endanger these two closest environments. To do this, we use the information from the closest forest and raised bog and the model from Meteotest and especially their Excel sheet. Recall that here we take the source as the geographical point of the address.

The maximum emissions are computed by a backward computation. We explain what it means: normally the model above takes an emission in input and outputs the ammonia concentration and deposition due to these emissions at several given distances from the source. The difference is that here, we have access to the distance to the nearest environments but not to the emission. As the emissions of a specific land could be accessed by further investigation, it is still interesting to know the maximum emissions that the land is authorised to have to not endanger its surroundings. To do this, we reversed the formula used in the Meteotest model to be able to compute from a critical level/load, this maximum emission possible for the land of interest (see A and the model for more details about the exact computation). Finally a maximum emission allowed is computed by taking the minimum of NH3 concentration for forest and raised bog and added to the output.

Going more in depth in the model, different profiles are given that suppose different dispersion of the ammonia. Each time we compute the maximum emissions for each profile (because we do not know the corresponding correct profile) and take the minimum result for each (because it is a maximum emission to not be exceeded). Moreover, different types of environments are proposed in the model. For the closest forest distance, we used the minimum of the computed emissions for the two forests environments of the model. For the closest raised bog distance, we used the minimum of the computed emissions for the two raised bogs environments of the model.

We copied all necessary constants for these computations in an Excel sheet (model.xlsx). There, they can be modified them if the user wants to adapt depending on the exact

location or more information about the local environment (see Meteotest report to see exactly to what they correspond).

3.5 Final Output

Finally, all the information found are put in an Excel sheet called "output.xlsx". For each building project respecting the needs of the search, this output contains the following information (maybe empty depending on what is available on the websites): publication ID, registration office ID and name, publication number, date and expiration date, canton, building contractor, project description, project location (full address), and cadastre, Latitude, Longitude, closest forest geographical points, closest forest distance, closest raised bog geographical points, closest raised bog distance, maximum emissions with respect to closest to the closest forest (for NH3 concentration and deposition), maximum emissions with respect to the closest raised bog (for NH3 concentration and deposition) and maximum emission allowed.
All the projects are ranked by maximum emissions allowed in decreasing order. The user has to be attentive to the fact that a lower maximum emission does not necessarily mean a higher risk as the closest forest or raised bog could further away from the source. We chose it like this because generally projects with low maximum emissions requires more attention (as it is easy to attain the maximum emission).

4 Limitations

We present quickly in these section some of the drawbacks of our solution and potential improvements that could be added.

Firstly, due to time restrictions we could not extend it to other cantons, but we wish someone continues the work and includes more of them.

Then, for the Luzern webscrape, it is important to note that this code is hard-coded to work for the current format of Luzern "Amtsblätter". The code searches for the first appearance of a word and then extracts the details that follow. For example, project details were found to follow after the word "Bauvorhaben:".f in the future the canton of Luzern changes some of the standards the code would have to be adapted to that end.

Also, as precised in section 3.3, to speed up the computation, we use a reduction of the number of points in the boundaries of ecosystems. This means that it could be less precise for some address and ecosystem.

Finally, we do not look automatically to constants for each specific region, we use fixed constants that have to be updated by the user. One remedy could be, for instance, to first put general constants (means over Switzerland for example, as they are now) and then if

one project is interesting, compute it again after changing the constants (that depend on the exact location of the land).
Due to time restrictions, we did not have the time neither to approximate the emission of each project but instead only compute a maximum emission. If the exact emissions of a land could be found, they have to be compared to this maximum emission result.

To conclude, the user of the interface always has to be careful with the outputs and see if they make sense before going further in the analysis. This user interface is only here to help to try to find the main projects to focus on but when one wants to actually go further, it requires further investigation and verification of the outputs.

5 Code Modification

In case the user needs to modify the Python code directly, it will be available in the GitLab repository.

Afterwards, an executable can be generated using the script *generate_executable.py*.

6 Conclusion

To put it in a nutshell, we designed a user interface that aims to simplify the search on the different Amsblatts of building projects related with keywords chosen by the user and in the desired cantons. To the basic information characterizing the project (in order to be able to find it again in the Amsblatt of interest), we also computed distances to the closest forest and raised bog and from them the maximum allowed emission for the land. At the end of the process, we output everything in an Excel sheet that is available to the user. The process of running the user interface also allows to overwrite a former output of the interface that was made before and only add the new projects to the ones already in the output file.
Thanks to many parameters in the user interface or in the choices of the constants to compute the maximum emission, the user is free to adapt the use of the interface to its needs.

Next, we hope that WWF will build on this interface to identify potentially dangerous projects and then have further investigation into each of them to appeal if necessary. The code is a good basis to start working but it is still open to some new contributions to have a broader impact and to automate even more the process.

A Appendix

A.1 Precision on the datasets used for the distance computation

A.1.1 Raised bogs

The data can be found here. The data, as it is the case in the next section, comes in a datastructure called "geodataframe" that relies on the library "geopandas". The only difference with standard pandas dataframes is that they have a new columns indicating the shape of a polygon, representing the area that row is refering to. The main features of the data are the "RefObjBlat", which outputs the link of the protected area. Each protected area has and index "ObjNummer". There is also the output "Type" which, in this case, will differentiate between being a "Primäre Hochmoorfläche" or a "Sekundäre Hochmoorfläche".

A.1.2 Forests

In this case the data comes in a json format. Each element comes with two main subfamilies: geometry and attributes. Attributes return the metadata of the area, being "ObjNummer" the code to localize the area. The geometry part behaves exactly as the first one indicated in the Raised bogs part.

A.2 Precision on how the maximum emissions are computed

We did a backward calculation by taking the formula and inverting it to obtain from the critical level/load the maximum emission that the land could have (formulas taken in the Meteotest model/Excel sheet). The formulas are the following:

Profile i ($i = 1, 2, 3$)

$$\text{NH}_3 \text{ concentration:} \quad E^i_{\text{max,conc}} = \frac{C^{\text{conc}}_{\text{crit}}}{K_1 f_i(d)}$$

$$\text{NH}_3 \text{ deposition:} \quad E^i_{\text{max,dep}} = \frac{C^{\text{dep}}_{\text{crit}}}{K_2 f_i(d)}$$