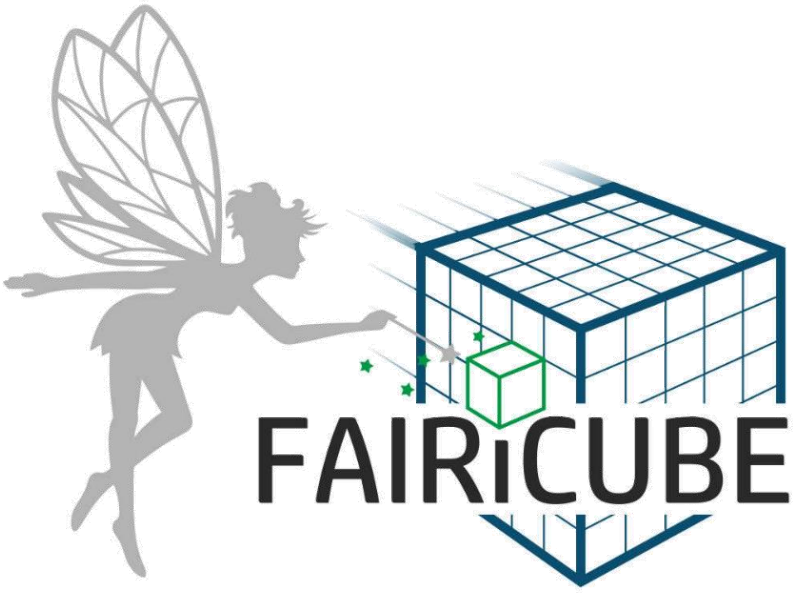


FAIRICUBE – F.A.I.R. INFORMATION CUBES

Project Number: 101059238

WP 3 Process

D3.2 Machine learning strategy specific for each use case

Deliverable Lead: NIL
Deliverable due date: 31/01/2025

Version: 3.2
2025-02-03

Document Control Page

Document Control Page	
Title	D3.2 Machine learning strategy specific for each use case
Creator	NIL
Description	D3.2 Machine learning strategy specific for each use case
Publisher	"FAIRICUBE – F.A.I.R. information cubes" Consortium
Contributors	NIL, WER, NHM, S4E, 4SF
Date of delivery	31/01/2025
Type	Text
Language	EN-GB
Rights	Copyright "FAIRICUBE – F.A.I.R. information cubes"
Audience	☒ Public ☐ Confidential ☐ Classified
Status	☐ In Progress ☐ For Review ☒ For Approval ☐ Approved

Revision History			
Version	Date	Modified by	Comments
0.1	16/05/2023	Stefan Jetschny, NIL	Draft setup, headings and partner/contributor assignments
	27/05/2023	Rob Knapen, WER	Use case 2 contribution
0.2	11/06/2023	Stefan Jetschny	Ready for partial review, only the UC4 contribution missing
1.0	21/06/2023	Stefan Jetschny	Ready for review, minor comments still open
1.1	27/06/2023	Jaume Targa and Stefan Jetschny	Review and minor modifications according to review comments.
2.0	07/11/2023	Stefan Jetschny	Extraordinary update to reflect the progress of the work, read-through version for assigning writing tasks
2.1	26/01/2023	Jaume Targa	Initial review
2.2	07/02/2023	Jaume Targa	Final review
3.0	29/10/2024	Stefan Jetschny	Reopening report and preparing for the final update
3.1	20/01/2025	UC leads and contributors	contributions and updates
3.2	28/01/2025	Stefan Jetschny	Review and format checking

Disclaimer

This document is issued within the frame and for the purpose of the FAIRICUBE project. This project has received funding from the European Union's Horizon research and innovation programme under grant agreement No. 101059238. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

Table of Contents

List of Figures

List of Tables

1 Introduction

WP3 aims to provide guidance, recommendations, technical expertise, and implementation support expertise to all use case efforts in terms of data analysis and processing. While the tasks will be executed by the use case developers, support will be given to assist in all data handling steps after ingestion and provision on both the Rasdaman- and EOxHub services as part of FAIRiCUBE's overall data and model services. Special emphasis is given to the data driven machine learning (ML) model generation.

This deliverable needs to be seen as one item of a classical and logical execution of a machine learning (ML) application. First, the research questions for each of the diverse FAIRiCUBE use cases (UCs) are proposed and described in Deliverable 2.2: Use Cases Analysis Plans with proper linkage to the domain of the UC and the potential application and benefits. D2.2 will also serve as the primary report for describing the output of the use case work, i.e. to which degree and for which target user group the research question has been answered. Formulation of the research question implies the identification of data sources that are needed and given availability/ingestion of this data, we first perform an exploratory data analysis to get familiar with the data, analyse statistical parameters and distribution, check for completeness, outliers and other characteristics which could be relevant for the choice of the machine learning. The in-depth data analysis is covered by deliverable *D3.1 UC exploratory data analysis*.

Subsequently, the raw data might require conversion into features through a data engineering step. This could imply a combination of several input data sources or applying simple mathematical operations to enhance the meaningfulness of the raw data given the relationships that are to be revealed. The more prior information is available, the better the feature engineering process can be performed. Based on the findings from the exploratory data analysis, the formulation of the research question and the relationship between raw data sources/features, machine learning algorithms can be recommended to establish a baseline model if this is not provided by use case owners. Starting from the most efficient machine learning algorithm, more advanced ML methods can be identified to form a machine learning strategy. Several different methods might also be tested to recommend a method based on computational demands and the accuracy of the ML output. Typically, the testing of ML algorithms is performed on a subset of the original input data or on selected cases. The feature engineering process, testing of ML algorithms and the recommendation of a cascade to ML algorithms, as well as analysing the output of ML methods is covered by this deliverable *D3.2 Machine learning strategy specific for each use case*.

As the FAIRiCUBE Hub ultimately aims to also provide resource estimations and guidance for ML applications, we collect and share computational parameters, timings, requirements, concrete implementation details of processing steps and pipelines and give an outlook on the expected scalability of the ML problems defined by the use cases. For each ML algorithm identified and executed as described in D3.2 we collect information on e.g., disk storage, CPU runtime, main memory consumption, describe the hardware and environment where the ML algorithm is executed on and list essential libraries that are needed to exactly replicate the ML application. This technical documentation of the ML execution is covered in the deliverable *D3.3 Processing and ML applications*.

In summary, the exploratory data analysis (D3.1) can be seen as an essential input to the development of a UC specific machine learning strategy (D3.2) whereas the technical description in D3.3 acts as a reference to follow up on the execution and serves as valuable input to estimate the demands for other ML applications. In the following, the machine learning strategy is described for each use case.

2 Machine learning strategies

Developing a sound machine learning strategy can be considered as matching the findings from the data exploratory analysis, the formulation of the data analysis objective, the defined accuracy expectations from use case owners with the available & suitable machine learning algorithms. This can be a cascade of algorithms sorted after computational costs and/or transparency of the results. Usually, the more complex a ML algorithm is, the better the expected performance can be at a cost of higher computational efforts and less possibilities to debug/reproduce the exact numerical operations that lead to the output of the ML algorithm. Weighting available resources, the numerical efforts and the quality of the ML application by analysis of output metrics yields the most optimal ML algorithm that addresses the scientific problem that was formulated upfront. In the following, we will provide details on the machine-learning strategies of all use cases (UCs).

UC1 Urban adaptation to climate change

When developing strategies for urban adaptation to climate change, a common challenge persists at all scales: datasets are complex to integrate and analyse due to their heterogeneity, varying formats, and quality differences. To circumvent this problem, UC1 approach is based on two components: a process to harmonize diverse datasets into structured data cubes, and a comprehensive toolkit for their analysis and presentation. These tools support UC applications at multiple scales. At the European level, they helped identify cities with similar characteristics and analyse how different factors influence cities' adaptation capacity. At the local level, specialized "city cubes" serve specific goals, as demonstrated through collaborations with Luxembourg City on managing invasive plant species and support provided to Vienna city initiatives.

i) European Level

In the first part this use case aims to provide data to cities and other stakeholders (such as European institutions or city networks) to support the decision-making process. Some of the data need to be processed and analysed to make sense for decision makers (e.g., extract useful patterns instead of presenting the whole data). For example, the land use/land cover dataset (the five classes of level-1 i.e., *1. Artificial surfaces 2. Agricultural areas 3. Natural and semi-natural areas 4. Water 5. Wetlands*) can be processed to identify the cities that are similar w.r.t land use and hence provides a clearer picture to decision-makers.

Figure 1 presents the expected processing workflow in UC1. The flowchart describes the data flow and processing steps for the first part of the use case, namely the analysis of cities across the EU. In this part, a preparatory step is the calculation of descriptive indicators for a large number of cities across the EU and for several years. These indicators are partly derived from EU-wide spatial datasets (land-based and climate data) and partly are already available as socio-economic indicators. All the indicators are eventually collected into a data cube where the spatial coordinates are not the traditional geographic coordinates, but rather the city identifiers, which can in turn be linked back to geographic coordinates by using either the city boundaries or the city centre point coordinates.

There are three different types of data sources:

- Land-based data, mostly derived from the Copernicus Land Monitoring Service (CLMS).
- Climate data, mostly originating from the Copernicus Climate Change Service (C3S); and
- Socio-economic data from the Urban Audit database hosted by Eurostat.

Additional spatial datasets already publicly available in the EDC Hub will also be used. The following processing steps are being implemented:

- Land and climate data are harmonised and ingested in a spatial data cube (spatial coordinates are lat/lon). Harmonisation consists for example in ensuring that the dimensions all share the same projection and geographical extent.
  - The spatial data cube is used to compute various city indicators. One indicator has one value per city/timestamp.
- The spatial data cube will also be used in the subsequent spatially aware analysis (e.g. green distribution with cities) and for visualization purposes.
- These land- and climate-based indicators are then fed into a city data cube (technically implemented as a Postgres database for the time being). The city data cube differs from the spatial data cube because the spatial dimension (coordinates) are not the traditional geographic coordinates (e.g. lat/lon) but rather the city identifiers. The spatial dimension in the traditional sense can be at any time recovered by linking the identifiers to the city boundaries or centre point coordinates. This data representation is less memory-intense and is more practical for further analysis.
  - At the same time, socio-economic data is ingested. Socio-economic data do not have spatially explicit coordinates, but rather they are indexed by the city identifier. Therefore, indicators are directly fed into the city data cube, and derived indicators are computed.
- Attempts at ML-based gap filling have been made on the socioeconomic data but with limited success due to the large extent of the gaps. Rather the original dataset has been heavily filtered down to the indicators that have sufficient data.
- Cluster analysis is then carried out on the city data cube. The goal is to generate different clustering scenarios driven by different themes/questions (e.g. urban heat, flood retention, green infrastructure). This can be achieved by weighted clustering, where a weight is assigned to each feature (indicator) to control its influence on the clustering. An advantage of weighted clustering is also that it can be easily tuned to different needs, thereby generating multiple scenarios relatively quickly.
- Ultimately, this framework can be used to run simulations, by tuning the indicators values and measuring their influence on the outcome.
- The city data cube can be linked to visualisation tools to create dashboards and city fact sheets

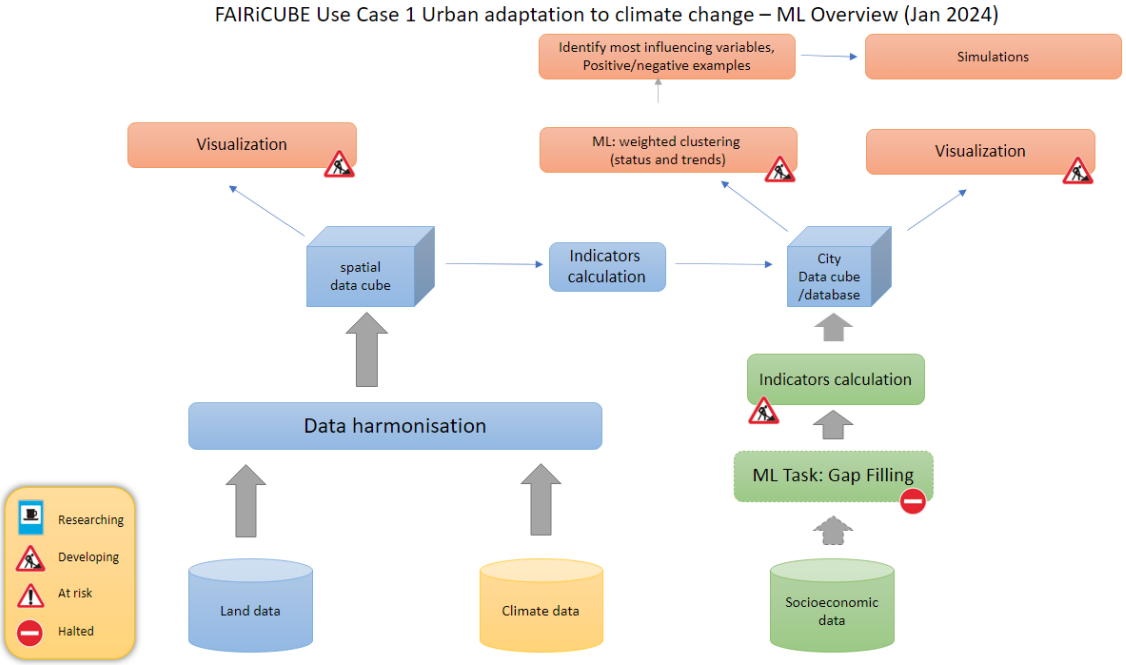


Figure 1 : UC1 data analysis and processing workflow

2.1.1.1 Clustering based on land use

To get a general overview of the cities that are similar with respect to area coverage, clustering can be used. Clustering is a type of Unsupervised Machine Learning tool that aims to put a given data into different clusters, each containing data points that are similar with respect to a set of features.

K-means is one of the most commonly used clustering algorithms due to its simplicity and efficiency in practice. Other algorithms include Mean-Shift, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Expectation–Maximization Clustering and Agglomerative Hierarchical Clustering (AHC) each useful on some type of data/problem. Additionally, Deep Learning can be used to reduce the number of features prior to implementing any clustering. This does not apply to this specific problem, because we only have 5 features (classes' ratios) as input. In addition, the features are already normalized, so there is no need for encoding/normalization steps.

For the task of clustering cities with respect to coverage ratios, we have experimented with three different clustering algorithms: k-means, Mean-Shift and AHC. Below, we will provide a report on the results obtained from each approach:

K-means is a clustering algorithm that starts with randomly selected K mean points and associates every data point to the closest mean. The means are incrementally updated until no significant change is noticed. A limitation of the k-means algorithm is the requirement to specify the number of clusters (K) as a parameter, which can be a drawback in some cases. However, this can be overcome using the Elbow Method. The elbow method runs k-means using multiple K values and selects the minimum K for which the inertia (squared sum of the distance between each point and its mean) starts to converge to a minimum value. For example, in this specific problem, a good choice of K would be 4, see Figure 1. Applying k-means with K = 4, has yielded four clusters of cities distributed as presented in Figure 2.

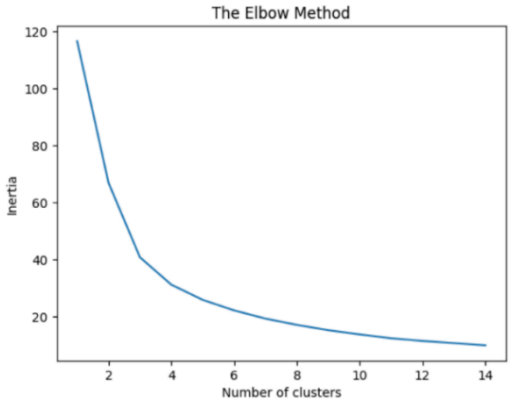


Figure 2 : Elbow metho to select k, the number of clusters.

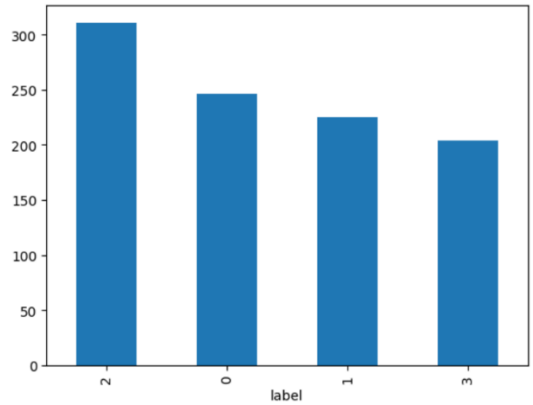


Figure 3 : Distribution of the four clusters using k-means.

Mean-Shift is based on shifting a window (with a pre-defined size) to the mean of the data points that are within the window. When it is no longer possible to shift the window (the centre of the window is the mean itself), the visited points are grouped into a single cluster and start constructing the next cluster by placing the window into a randomly selected data point. The main advantage of this algorithm is that there is no need for defining the number of clusters beforehand. However, we need to define the size of the window. In this study we tried different sizes and discovered that with size = 0.25, the number of clusters is acceptable (4 clusters). With a higher value, we get a lower number of clusters (a single cluster with size = 0.35). On the other hand, we get a higher number of clusters with smaller size value (9 clusters with size = 0.2). The distribution of the four clusters using Mean-Shift is presented in Figure 4. We clearly see that Mean-Shift puts most points (93% of the data) in a single cluster. This shows that this algorithm is not suitable for this specific study.

Agglomerative Hierarchical Clustering (AHC) builds a hierarchy of clusters by constructing a parent cluster from two closest sub-clusters (children). The Algorithm builds a hierarchy until every point is associated to a cluster. The user then can select any level of the hierarchy to be considered as clusters outcome. For this specific problem, we have built the hierarchy and selected 4 as the number of clusters (to be consistent with k-means). The distribution of the clusters is presented in Figure 5.

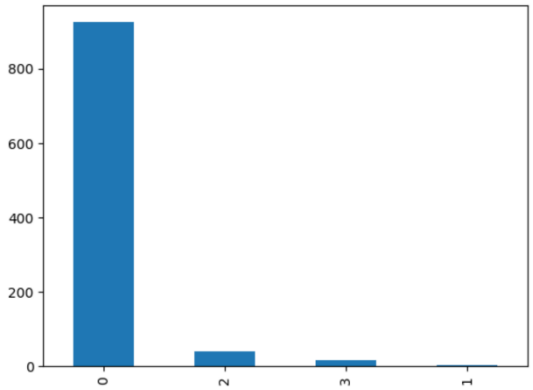


Figure 4 : Distribution of the four clusters using Mean-Shift.

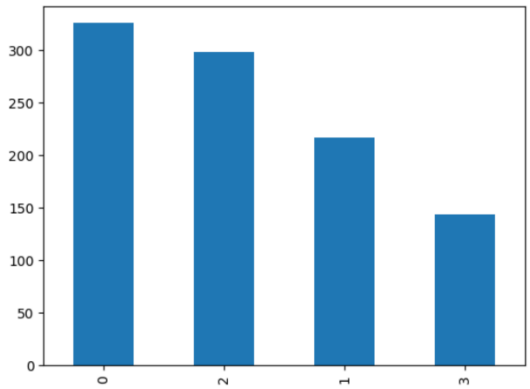


Figure 5 : Distribution of the four clusters using AHC.

In order to obtain an overview of each approach/cluster, we have cross-plotted the data points using each pair of features, as it is impossible to visualize a 5-dimensional clustering. We notice that very few cities have high water and wetland areas (see Figure 5), hence we consider the other three features (Artificial surfaces, Agricultural areas, and Natural and semi-natural areas) for plotting clusters (see Figure 6). The first observation is that Mean-Shift did not cluster the cities in an equitable way. It seems to group most of the cities into a single cluster, which is not meaningful in the context of this study.

On the other hand, k-means and AHC have yielded almost similar clustering. However, we can see some few overlapping using AHC (see Figure 6). Hence, based on the resulting clusters, the k-means algorithm seems to produce more meaningful clusters.
Following the plots in Figure 6, a possible significance of each cluster using k-means is the following:

- Cluster 0 represents cities with high artificial surfaces (e.g., Vienna, Austria).
- Cluster 1 represents cities with high natural and semi-natural areas (e.g., Stara Zagora, Bulgaria).
- Cluster 2 represents cities with a balance between artificial surfaces and agricultural areas (e.g., Linz, Austria).
- Cluster 3 represents cities with high agricultural areas (e.g., Rugby, Great Britain).

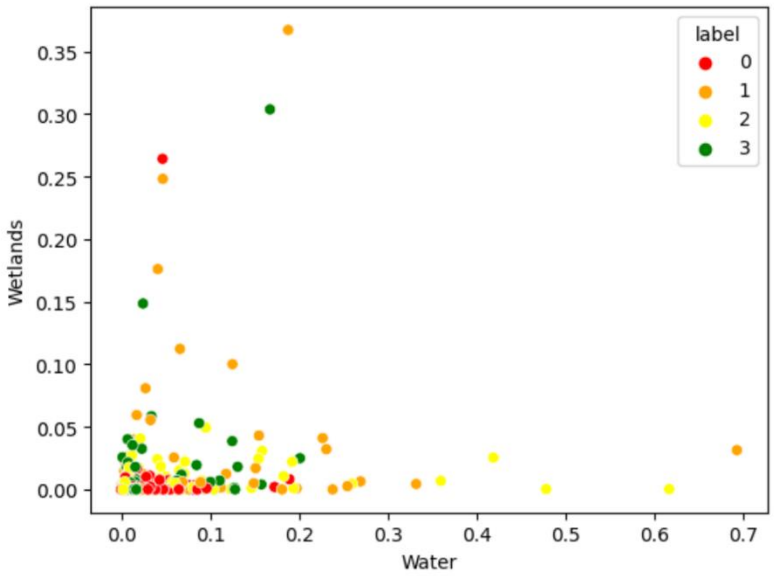


Figure 6 : Clustering with respect to Water and Wetlands

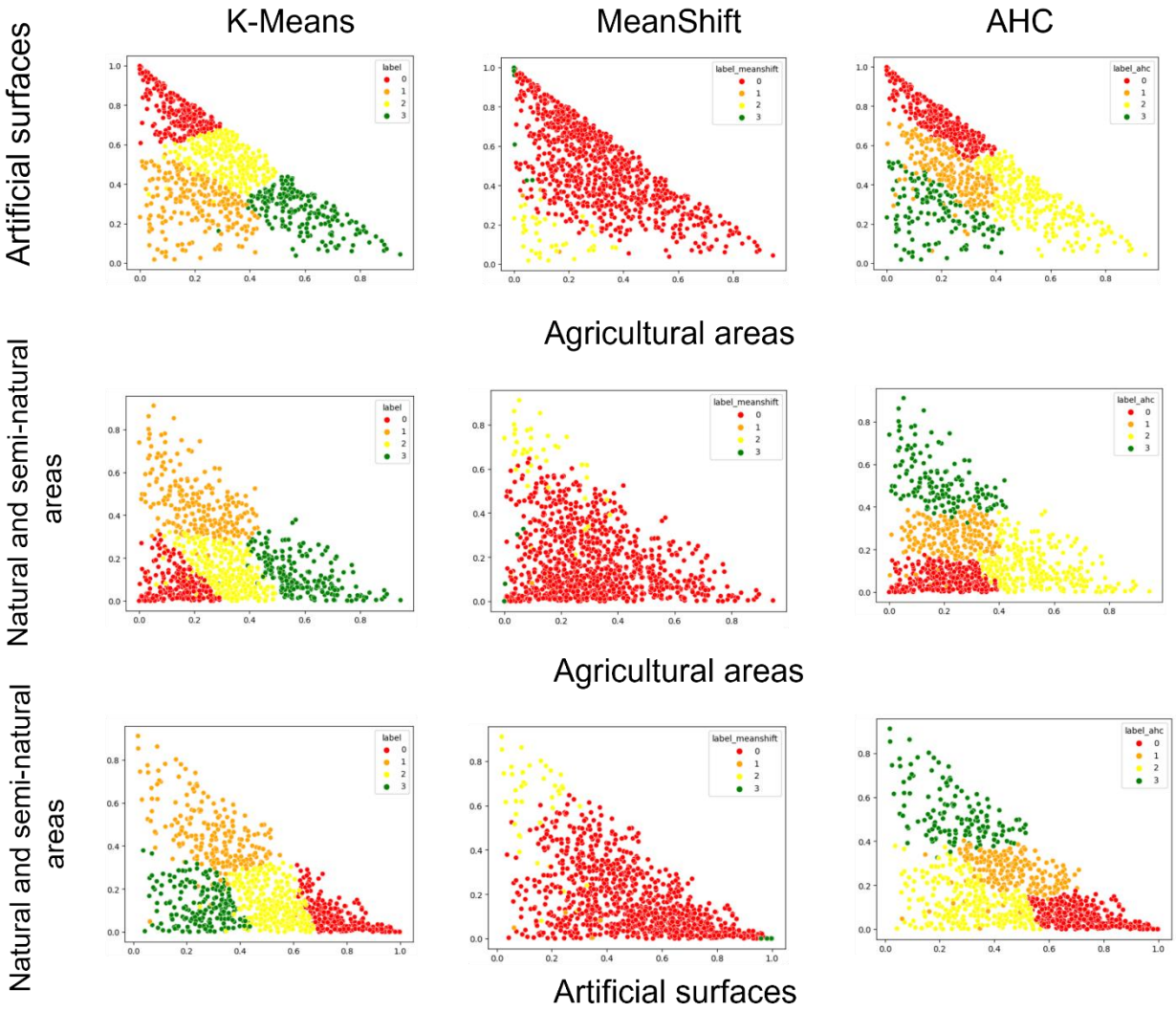


Figure 7 : Plotting clusters with respect to the three main features.

In this Machine Learning application, we have generated clusters of cities based on the input of five level-1 land use classes (*1. Artificial surfaces; 2. Agricultural areas; 3. Natural and semi-natural areas; 4. Water; 5. Wetlands*). We have used three different clustering algorithm (k-means, Mean-shift and AHC), and identified k-means as the baseline giving its consistent output (see clusters significance above). For further analysis, we will consider more features to have a multi-dimensional cluster of cities. For example, socio-economical features (e.g., population density), or climate features (e.g., heights temperature) can give more insight of cities from different perspectives.

2.1.1.2 Socioeconomic data gap filling

Obviously, clustering using only the land cover does not give a general insight into cities. Hence, using indicators from the socioeconomic data in addition will be useful. Unfortunately, the socioeconomic data reported by Eurostat is very limited and contains many gaps. In what follows, we propose several possible Machine Learning (ML) strategies to recover the missing data.

2.1.1.3 Using time series:

A possible direction into gap fills in missing indicator values of some specific years is to use the information of the other years i.e., the time series. For example, if the total population of a given city is increasing in the few last years, one can predict an increase -w.r.t some ratio- for the next year. Clearly, some time series are very complex and hardly predictable. One of the most known ML strategies for learning from time series is LSTM, for Long Short-Term Memory. LSTM is a type of Recurrent Neural Networks (RNNs) that learns from sequences of connected data and retain information over long sequences. Given a sequence size s, the LSTM model is trained to predict a value at time t using the previous s data points in the time series. Furthermore, **Bidirectional LSTM** (BLSTM) is an extension of LSTM, that learns both in forward and backward directions. As an example, we have used a Bidirectional LSTM for 'Total population' prediction. In the Eurostat data, the 'Total population' of some cities is available over 31 years (from 1991 to 2022). We selected two cities, one with simple linear increasing time series (Helsinki in Figure 8), and one with a more complex time series (Bari in Figure 9).

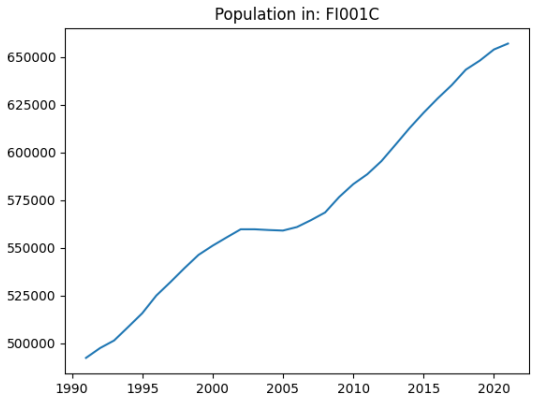


Figure 8 : Total population over the years in Helsinki

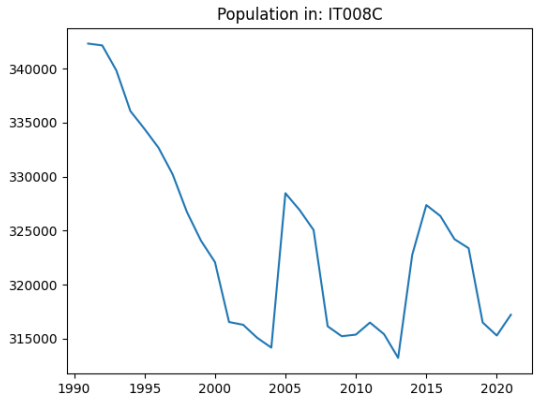


Figure 9 : Total population over the years in Bari

We have trained two BLSTMs, one for each city. We have selected the first 70% years for training, with a sequence length of 3 (e.g., predict 2010 using 2009, 2008 and 2007), and the last years for testing. After training over 100 epochs, the predicted time series vs the actual for Helsinki and Bari are presented in Figure 10 and Figure 11, respectively. We can clearly see that the prediction for Helsinki is close to the real values (with a Mean Absolute Percentage Error of 0.63%), this is thanks to the simple linear trend of the time series. On the other hand, for Bari, the prediction does not follow a linear pattern, but it is far from exactly following the actual trend even with a Mean Absolute Percentage Error of 0.53%.

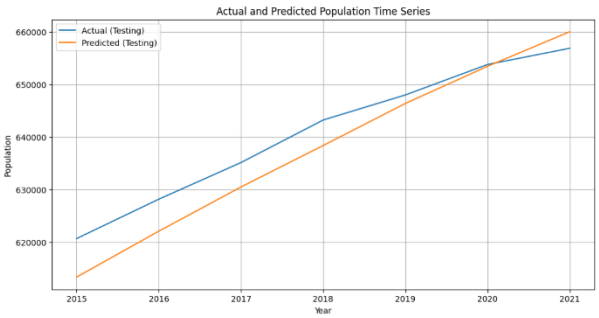


Figure 10 : Predicted vs Actual for Helsinki

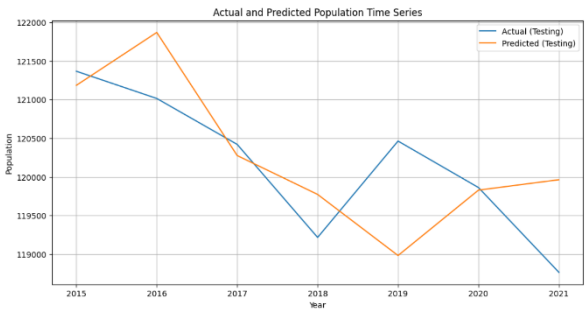


Figure 11 : Predicted vs Actual for Bari

Limitations: This approach is limited in our context. First, the number of data points is small for an effective training (only 31 data points) in case of complex sequences. To have acceptable results, a time series need to contain at least 50 data points for training. Most importantly, we can notice that the time series are city/indicator specific. In this case, we need to train a model for each time series which is computationally expensive. Furthermore, no model can be useful for cities with 0 data points (unless we use a model of some similar city, which is hard to determine).Finally, in Eurostat gap filling, this approach can be useful for cities/indicators with a few missing years (between 1 and 5), that follow regular trends (e.g., linear).

2.1.1.4 Using correlations between indicators:

Some indicators are related to each other. For instance, the *'Proportion of population aged 75 years and over'* is related to *'Total Population'*. Hence, we can use ML regression models to approximate the value of a given indicator using other available inputs. Regression models are approaches that learn a function to estimate the value of a given feature using a set of input features. Regression models range from simple linear regression (that only learns linear functions) to complex deep learning models.

As show case, we have implemented an ML model to estimate: the *'Total number of hours of sunshine per day [EN1002V]'* using two inputs: *'Average temperature of warmest month - degrees [EN1003V]'* and *'Average temperature of coldest month - degrees [EN1004V]'*. To train the model, we need data points that have the three features available i.e., 1,673 data points in Eurostat. Then the trained model can be used to gap-fill data points where the two input features are available, but not the target feature i.e., 299 data points in Eurostat. In Figure 12 we present the correlation matrix (computed using the 1,673 data points) between the three features. Clearly, a positive correlation exists between the target feature (number of sunshine hours) and the input features (temperature of warmest month, and temperature of coldest month). This is especially true for 'temperature of warmest month' where the correlation reaches 74%, this is promising for a regression model to be efficient.

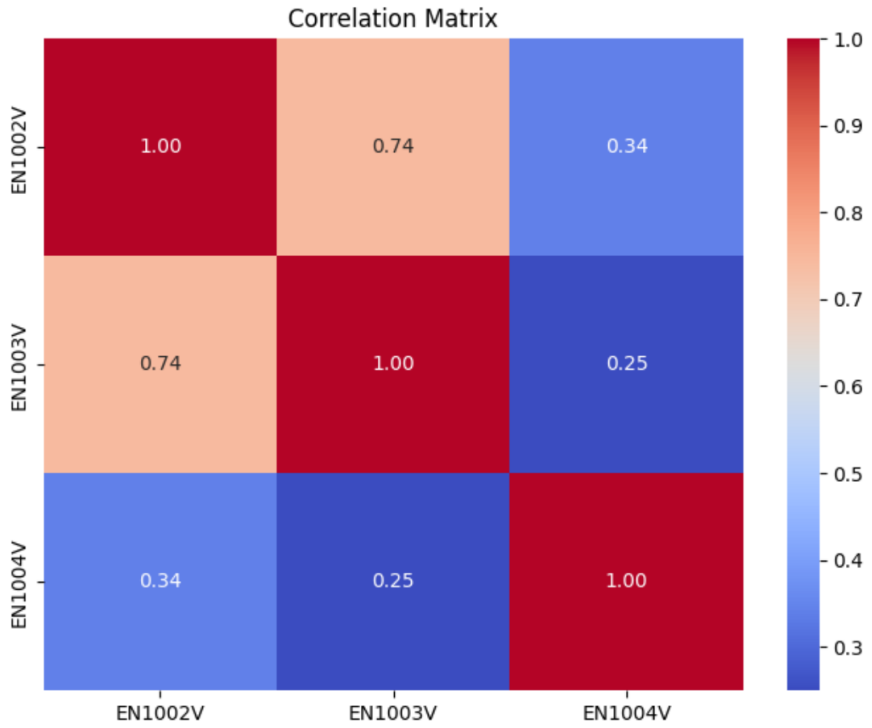


Figure 12 : Correlation matrix between 'Total number of hours of sunshine per day [EN1002V]', 'Average temperature of warmest month - degrees [EN1003V]' and 'Average temperature of coldest month - degrees [EN1004V]'

To predict EN1002V using EN1003V and EN1004V, we have trained several regression models on 70% of the selected data (the 1,673 data points). The **Gradient Boosting Regressor** (that uses an ensemble of decision trees for regression) was the most effective reaching mean squared error of 0.48 (error of less than 30 minutes of sunshine on average). Furthermore, the F1 score reached 0.677 indicating promising results. The distribution of the predicted vs the actual number of sunshine hours is presented inFigure 13. The distributions show close prediction to actual values.

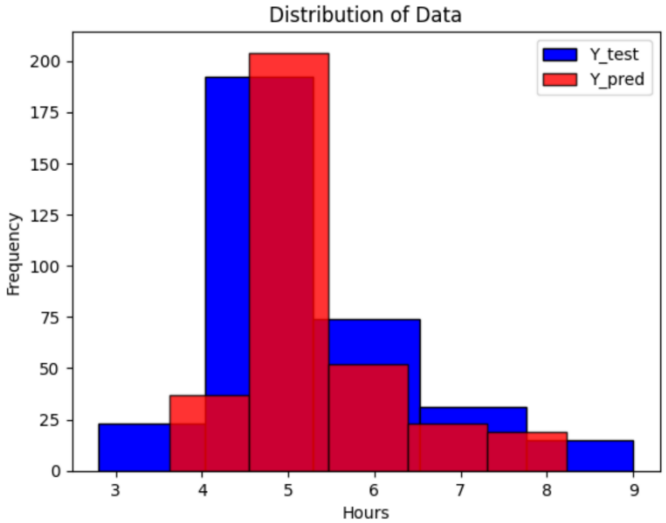


Figure 13 : Sunshine hours distribution (predicted vs actual)

Limitations: The results show that regression models can be effective in recovering missing data. The model above was useful to gap-fill in almost 300 data points. However, this approach requires designing regression models for each indicator and manually defining the potentially related input features. Furthermore, training data is not sufficient for each indicator. For example, it is not possible to predict the *'Proportion of households that are lone-pensioner households [DE3008I]'* using the input features: *'Lone pensioner (above retirement age) households [DE3008V]',* and *'Proportion of population aged 65-74 years [DE1028I]'* because 0 training data points are available (no data point that has all the three indicators available).

2.1.1.5 A more general solution:

Above, we have shown the usefulness of both time series patterns and interdependence between indicators to gap-fill missing data. A more general solution is to take into consideration both the time and indicator dimensions in a single model to predict missing values. Furthermore, due to the absence of data for some cities, one may consider learning from cities with similar characteristics. In Figure 14 we present a potential ML architecture that considers all these information in a single model. Here, we first cluster cities w.r.t some pre-defined features, then we collect all available [indicators, years] matrices and feed it to a bidirectional LSTM architecture that takes into consideration the interdependence between indicators as well as the time series connections to predict values of a specific 'city cluster/indicator/year'. Implementing such a model was not possible due to the limitations mentioned below.

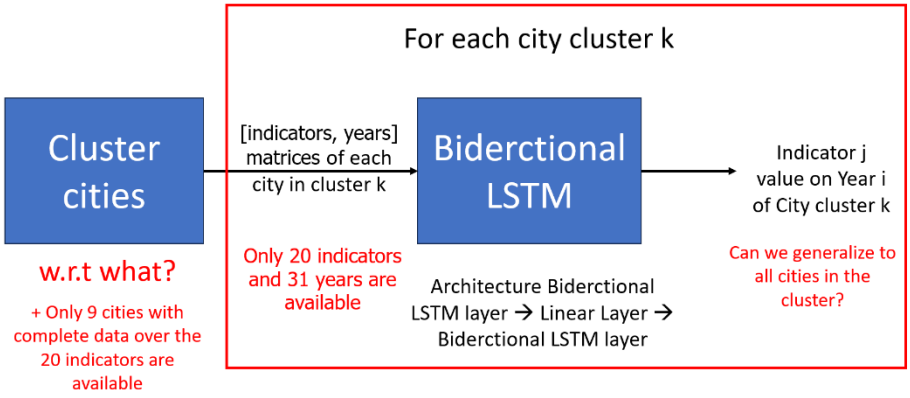


Figure 14 : Possible architecture for a general solution.

Limitations: Obviously, this model requires a large amount of training data that is not available in Eurostat. In Eurostat, we only have 9 cities that have 20 indicators with all available years. In addition, it is hard to pre-define features for which we should cluster cities at the first step, mostly, these features are the ones we want to gap fill. Finally, predicting values of a specific cluster of cities does not necessarily generalize to all cities in the same cluster. Overall, gap-filling the Eurostat data is very challenging due to the absence of enough training data. Here, we have explored some low-hanging fruits (time series analysis, regression models). However, these strategies require some manual effort, and do not recover enough data. Finally, a more general solution requires more training data.

ii) Local level

The second application of this Use Case employed a high-resolution "city-cube" to monitor the spread of invasive alien species in urban areas. The analysis was carried out for the Environmental Department of Luxembourg City, hence used the city administrative boundary as study area. In agreement with the city representatives, it was decided to develop species distribution probability maps based on occurrence data sourced from neobiota.lu, and an habitat data cube combining environmental, ecological and weather data. These maps served as a basis for "priority maps" used by the city administration to better plan monitoring and mitigation measures against invasive plant species. After an initial exploratory phase (see also Deliverable 3.1), and several exchanges with UC2 and UC5 – both working with species distribution models – we selected MaxEnt as the most suitable ML for this task. MaxEnt, short for Maximum Entropy, is a widely used approach for modeling species distributions. One of its advantages is its ability to learn from "presence only" data. This is very useful in real-world scenarios where data often consists only of locations where a species has been observed, while absence data is typically hard to get or unavailable.

MaxEnt works by solving an optimization problem (see the mathematical model in Figure 15). Its goal is to find a probability distribution that maximizes entropy—essentially, the most spread-out, unbiased distribution possible—while still satisfying certain constraints. These constraints are derived from the presence data, ensuring that the average value of the environmental variables in the predicted distribution matches their average in the observed presence points. To allow for some flexibility, the constraints can be slightly relaxed, which prevents overfitting. The output of Maxent is a probability distribution over the study area, indicating the likelihood of species presence. This is then transformed into a habitat suitability score using a logistic function, making the results more interpretable. Then to test the model we use AUC (Area Under the Curve) metric that represents the ability of the model to distinguish presence from absence data. For example, if Maxent achieves an AUC of 93%, it means there is a 93% chance that a randomly selected presence point will have a higher suitability score than a randomly selected background point.

More specifically, the model in Figure 15 maximizes the entropy H, of the distribution P. One interpretation of the entropy is "the minimum number of questions required to reveal a randomly selected sample". By maximizing the entropy, the distribution will be less biassed towards specific points (more questions are required). For example, given 5 points A, B, C, D and E where A and B represent the presence data and C, D and E represent the background data. A possible learned distribution is P(A) = P(B) = 0.5 and P(C) = P(D) = P(E) = 0. This distribution is biased towards A and B and requires 1 question only ('Is the randomly selected point A?' if yes, we know it is A otherwise, we know it is B). In addition, the distribution needs to satisfy a constraint for each feature k i.e., the continuous sum of the distribution times value of feature k needs to equal the mean of the training sample X. Finally, the distribution needs to sum-up to 1.

One widely used approach to solve similar optimization problems is by using 'Lagrange multiplier' where P(x) represent decision variables in this context. This approach finds local optimal points and does not prove global optimality (if the space is not convex), but it shows useful results in practice.

$$\text{Maximize: } H(P) = -\int_X P(x)\log P(x)\,dx$$

$$\text{Subject to: } \int_X P(x)f_k(x)\,dx = \frac{1}{n}\sum_{i=1}^{n} f_k(x_i),\ \forall k$$

$$\text{and: } \int_X P(x)\,dx = 1$$

$$\text{Maximize: } H(P) = -\int_X P(x)\log P(x)\,dx$$

$$\text{Subject to: } \int_X P(x)f_k(x)\,dx = \frac{1}{n}\sum_{i=1}^{n} f_k(x_i),\ \forall k$$

$$\text{and: } \int_X P(x)\,dx = 1$$

Figure 15: The MaxEnt mathematical model

In this task, we have applied Maxent for distribution learning of some selected plant species in Luxembourg. For this, we have used the following features (environmental variables) as input:

- Light & shading (at 10m resolution)
- Wetness (yearly starting from 2019 at 10m resolution)
- Temperature (monthly mean at 10m spatial resolution available since 2017)
- Soil acidity (at 10m resolution)
- Soil nutrient (at 10m resolution)
- Land cover (at 10m resolution)

We first applied the model on predicting occurrence suitability of **Heracleum Mantegazzianum** given only **60 presence points,** and the rest of the region is considered as background data. The model was trained on 80% of the data and tested on the remaining 20%. It achieved promising results with an AUC of 0.79, and this given only 60 presence points.

On **Robinia Pseudoacacia** the model was trained on **380 presence points** and achieved an AUC of 0.74. The occurrence suitability maps of both species are presented in Figure 16. We can see that the model learned that Heracleum Mantegazzianum has potential to grow on the east. On the other hand, the model showed strong suitability for Robinia Pseudoacacia on the southwest area.

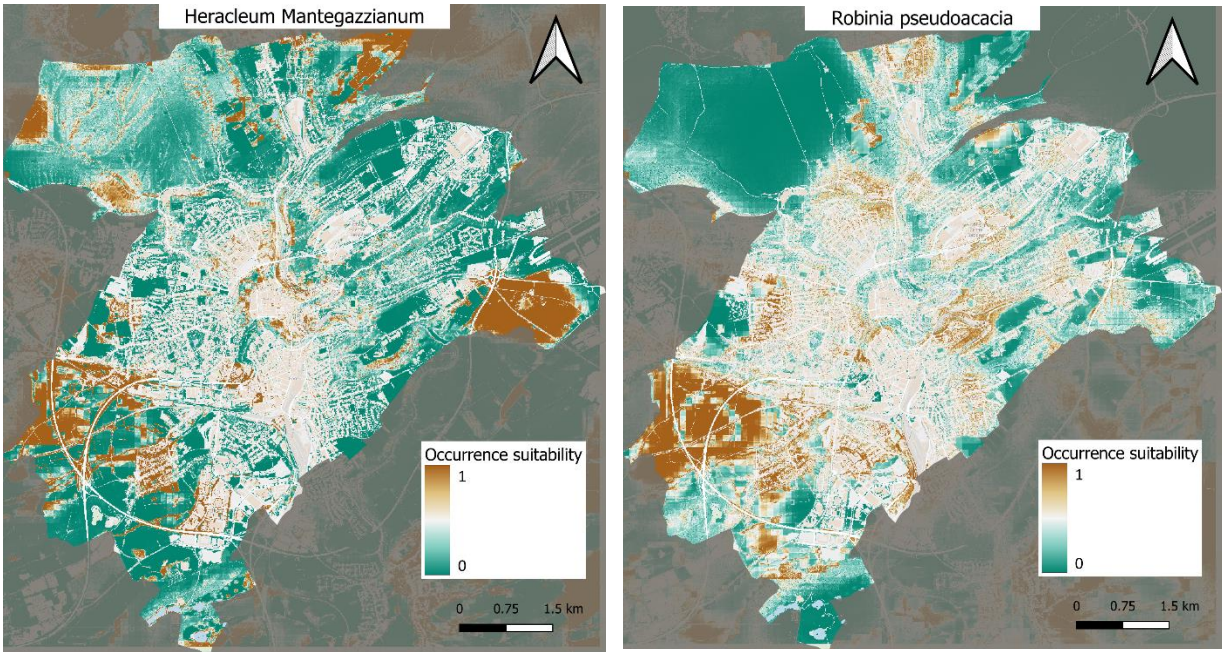


Figure 16: Occurrence suitability maps of Heracleum Mantegazzianum on the left and Robinia Pseudoacacia on the right.

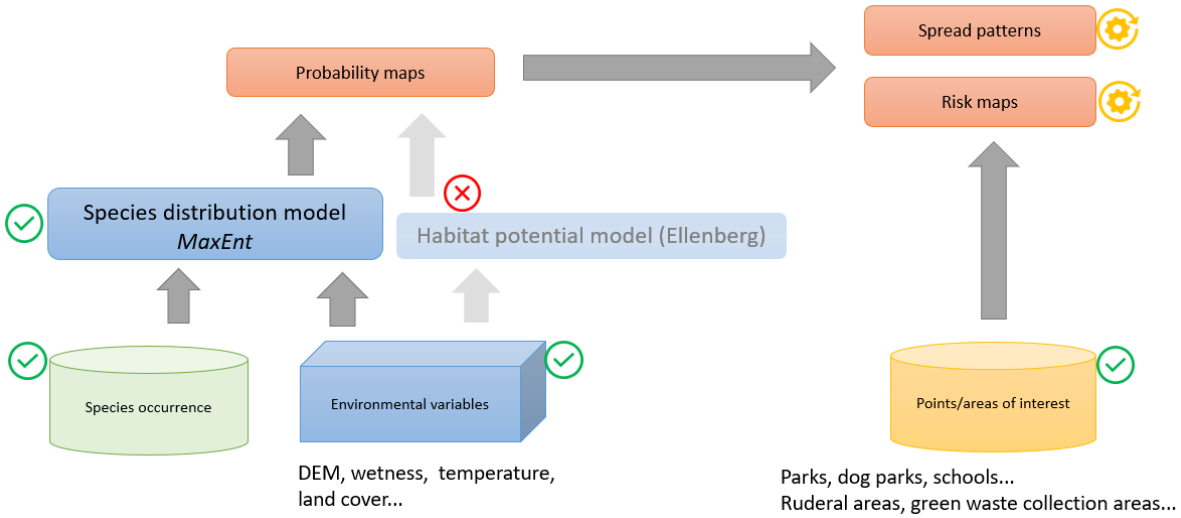


Figure 17: ML workflow for UC1 – monitoring of invasive alien species

UC2 Agriculture and Biodiversity Nexus

i) General concept

Central to this use case are **the effects of agriculture and farming activities on biodiversity**, specifically in agricultural areas. Evaluating and quantifying the correlation between agriculture and biodiversity is a crucial step in understanding the extent and nature of agriculture's impact on the environment. This process involves the systematic collection, analysis, and interpretation of data to reveal relationships and patterns that can inform policy and management decisions. To assess this correlation, various statistical methods and tools enables quantifying the strength and direction of the relationship between agricultural practices and biodiversity indicators. Key steps in this evaluation include defining appropriate biodiversity metrics (such as species richness, occurrence and abundance), selecting relevant agricultural variables (such as crop diversity, crop rotation, grassland mowing intensity), and applying evaluation techniques such as causal reasoning. As an overall guiding principle, we follow **a detection and attribution approach**, based on causal machine learning to relate (attribute) detected changes in biodiversity to farm management practices, using detailed spatial data cubes as primary information storage.

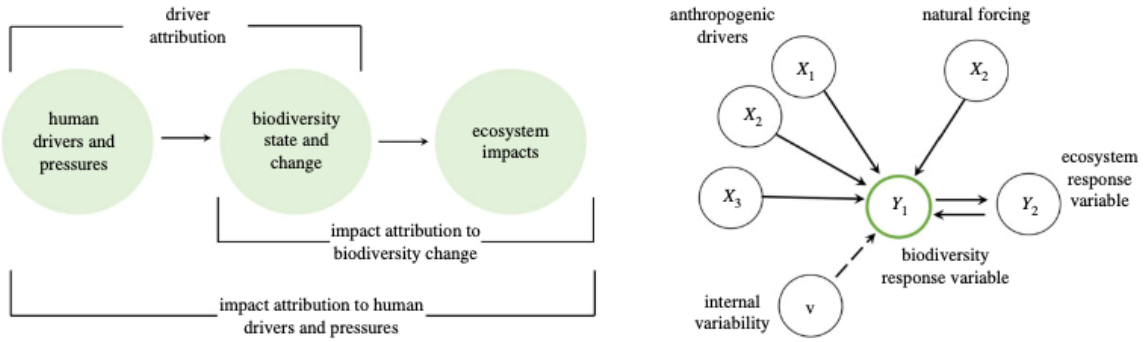


Figure 18 : Attributing biodiversity change to human drivers and pressure

Causal machine learning (CML) combines the strengths of causal inference and machine learning to identify and estimate causal correlations from data. Unlike traditional correlation-based approaches, CML aims to uncover the underlying causal mechanisms, providing more reliable insights for policy and decision-making. Techniques such as causal forests, targeted maximum likelihood estimation, and Bayesian networks are commonly used in CML to address these challenges.

Causal machine learning improves probabilistic approaches by adding causal reasoning, which for humans is a crucial element in learning about and understanding how the world works. Cause-and-effect relationships drive data, but regular statistical analysis alone is insufficient to recover those causal relationships from that data. The causality is part of the data generator, not of the data itself. In that sense, causal inference allows the discovery of the characteristics of the data generator. It is particularly helpful in cases when fully Randomized Controlled Trials cannot be conducted, which clearly applies when working in the environmental / nature domain. Causal machine learning, including causal inference and causal discovery, can leverage provided causal graphs, e.g. for answering counterfactual ("what if?") questions, it can be used to validate and indicate the robustness of causations expected by domain experts, and discover causal graphs from **observational data**. It is a promising field of study in machine learning, but also still a very active research topic.

The concept of a framework for the detection and attribution of biodiversity change has been proposed by Gonzalez et al. (2023)[1]. Where possible for this use case we have implemented elements of this framework.

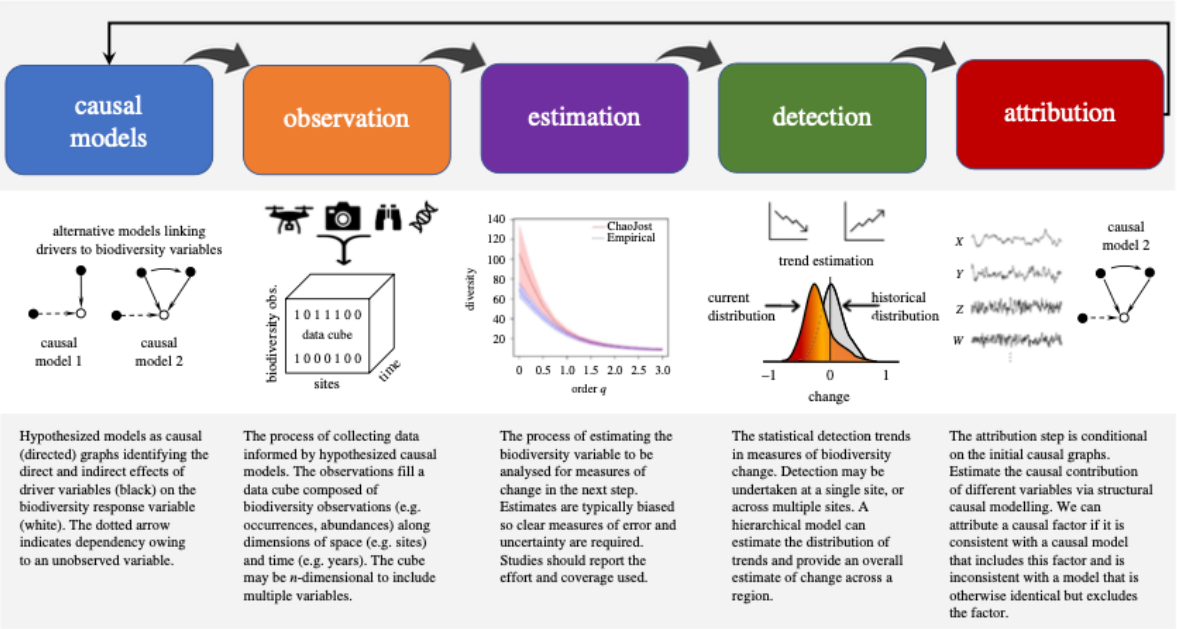


Figure 19 : Steps in the detection and attribution framework for biodiversity change

The causal machine learning approach in this use case is used to validate and assess the robustness of a few expected causal factors contributing to changes in biodiversity within agricultural landscapes. This involves distinguishing between correlation and causation, which is helpful when designing effective conservation and management strategies. Causal models facilitate counterfactual analysis, which in this use case involved comparing observed biodiversity outcomes with what would have happened in terms of biodiversity if a particular agricultural intervention or land management decision had, or had not, been implemented.

ii) Conceptual ML and processing workflow

For the assessment of how agriculture impacts biodiversity, three main categories of data have been used: biodiversity related data, environmental data, and agricultural data. Using these as pillars, an initial conceptual diagram of the proposed ML and processing workflow has been sketched for clarification and to serve as guidance for the various types of data engineering and ML tasks that were thought to be relevant and needed (see Figure 20). Since data science and machine learning are highly experimental and iterative in nature, the initially foreseen workflow has been updated during the project as insights into data availability, provided data cube functionality, relevant research literature, and possible ML algorithms evolved.

[1] Gonzalez A, Chase JM, O'Connor MI. 2023 A framework for the detection and attribution of biodiversity change. Phil. Trans. R. Soc. B 378: 20220182. https://doi.org/10.1098/rstb.2022.0182

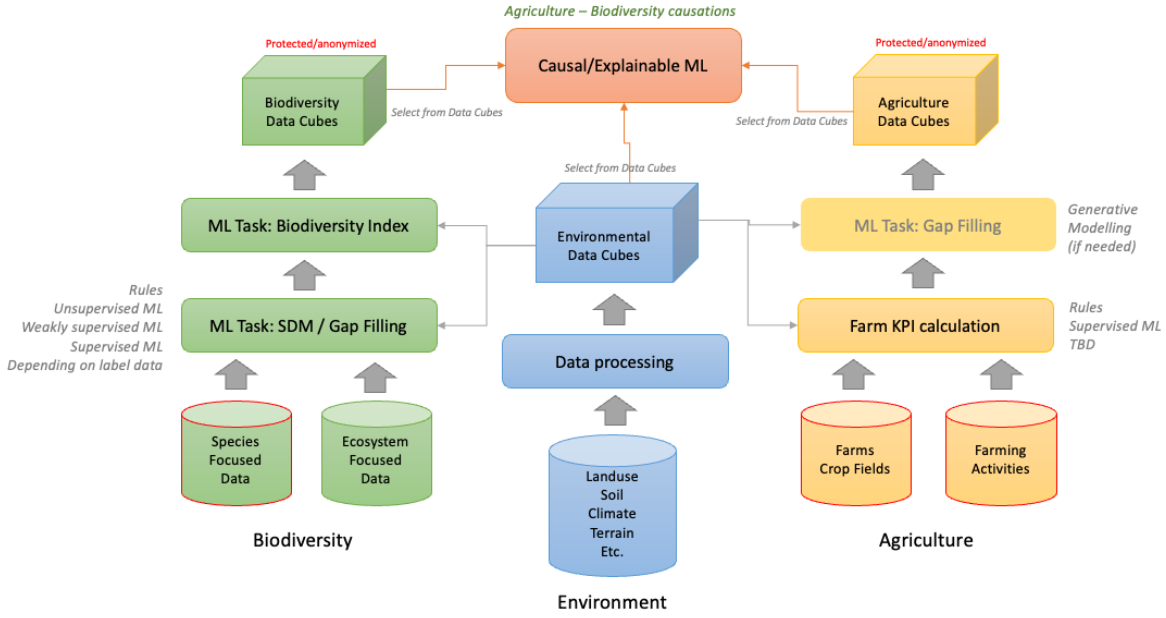


Figure 20 : Initially envisioned UC2 data analysis and processing workflow

The diagram indicates the three pillars – main groups of data analysis and processing workflows (biodiversity, environment, and agriculture) that play a role in the use case. The data engineering and ML tasks flow from the bottom of the diagram to the top, where they merge into the application of causal machine learning. By applying these types of ML algorithms, we are demonstrating that not only associations in the data between agricultural activities and changes in biodiversity can be detected, but also causality can be indicated and explained. For example, an increase in the presence of herb-rich grasslands causing an increased presence of a species under study. The datasets at the bottom marked with a red border note the inputs with additional requirements (such as restricted access and privacy sensitivity).

In Figure 21 the final achieved workflow state for the use case is shown. Some adjustments needed to be made due to data not being available or proved more difficult to work with as was initially expected. Also, some workflow steps were adjusted accordingly and tailored to the data cube and processing functionality as they became available in the FAIRiCUBE Hub. In the end most goals could be implemented, at least in an initial version that has been used to run first causal modelling experiments. The validation and further discussion of these initial experimental results with relevant domain experts will lead to the new insights, essential updates to the workflow, new and additional data processed, and new sets of causal modelling experiments that can be performed.

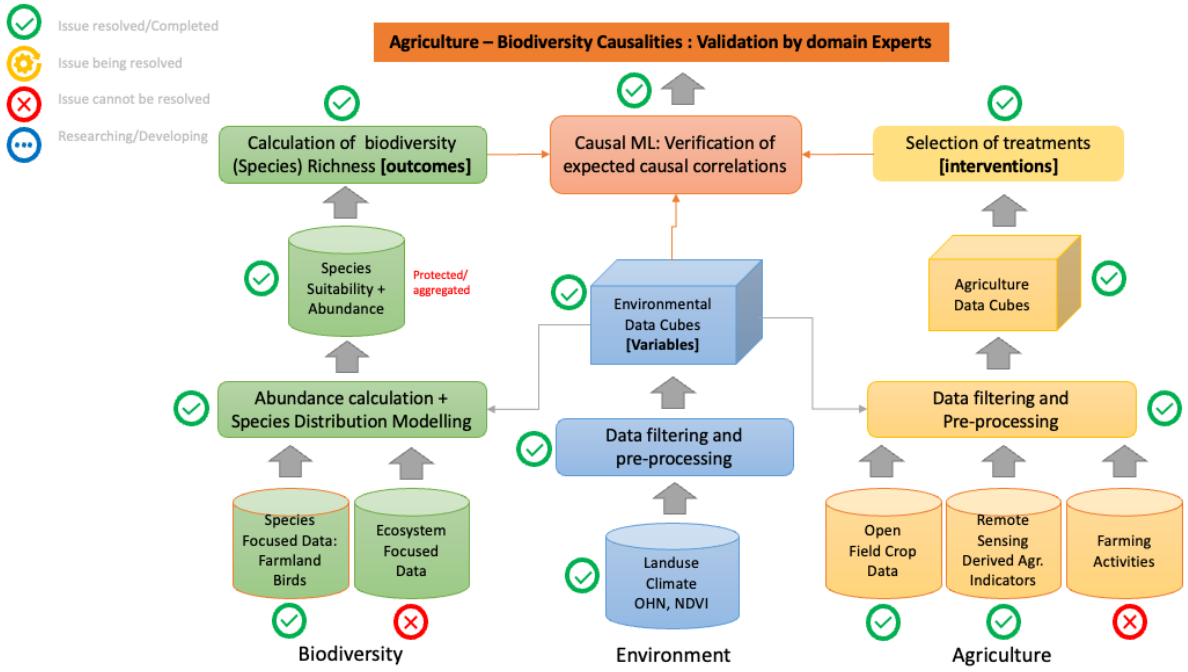


Figure 21 : Achieved UC2 data analysis and processing workflow

Some notable points in the final workflow are as follows, split up into the main data categories that have been used and the final causal machine learning step.

Biodiversity

- For species-focused data, observation data on farmland birds was selected as key species data to be used. In general, bird species are well-monitored following established protocols, resulting in good availability of presence data. The Dutch NDFF database (which requires a license to access) was used to provide species observation data, including geometries that indicate ranges. From this source data, proportional species distribution per grid cell and species abundance were calculated.
- Due to project constraints the biodiversity aspect of the use case has been limited to species-focused data, leaving out ecosystem-focused data. This kind of data can be added in follow-up activities.
- Given the species observation data, the **species distribution modelling** (SDM) has been completed using the common Maximum Entropy (MaxEnt) model. All input data used for SDM was fully ingested into data cubes, the model (in the end we applied the MaxEnt implementation that is part of the *elapid* Python package) has been run on the available processing environment at EOXHub.
- The abundance and suitability data created are considered as intermediate data within the workflow and will not necessarily be all ingested and published as data cube. Most likely only a few suitability maps will be published as part of the results of the use case.
- Using information about the species, the calculated abundance, and the predicted suitability per grid cell, a straightforward formula (initially summing predicted presence probabilities of the considered species at each location) takes these data as input to derive a **biodiversity (species) richness** value. Based on literature[1] we chose to initially work with a mean probability value as relative estimate of the biodiversity. It is calculated using the maxent probabilities of the species where the model meets the following criteria:

[1] *Grenié, M., Violle, C., & Munoz, F. (2020). Is prediction of species richness from stacked species distribution models biased by habitat saturation?. Ecological Indicators, 111, 105970.* https://doi.org/10.1016/j.ecolind.2019.105970

- Number of available data points (presence observations) >= 150
- Omission Rate <= 0.2
- AUC >= 0.7

The calculated mean probability value per grid cell then is used as the "**outcomes"** input for the causal modelling.

Environment

- For causal modelling in the agriculture – biodiversity nexus, **variables** about the environment are the main source of both **confounders** (variables that distort the apparent relationship between an independent variable (intervention) and a dependent variable (the outcome)), and **covariates** (variables that explain variations in dependent variables, but are not related to the independent variable).
- Selection and pre-processing of all required environmental data has been completed. Some of this data (e.g. small woody features) is shared between use cases. Every dataset required discussion on how exactly the data needed to be represented in one or more data cubes. Such specifications had to be provided by the use cases to the data ingestion work package.
- Some of the environmental data needed extensive pre-processing before it was suitable for ingestion into a data cube. Since most of has been a one-time activity, no specific data processing pipelines have been built for it. The most common data operations included reprojecting, resampling, rasterization, and clipping to a specific spatial extent (the study region of the use case), all of which has been performed using traditional GIS.
- Based on the specifications provided by the use case and the pre-processed data (when applicable), rasdaman has taken care of the ingestion of the data into their system, and for making it available as data cubes for further use.

Agriculture

- In the Netherlands, as in many EU countries, farmers need to provide information about their parcels and the crops grown on them every year. Part of these data are publicly available. However, the ownership information of all parcels and data about farming activities in general is not. Also, the available crop parcel data is in vector (polygon) format and thus had to be gridded with a usable spatial resolution first before ingesting into data cubes.
- Detailed (per-farm) data about some relevant farming activities (e.g. pesticide use) cannot easily be obtained, as it turned out and despite some efforts. It basically requires data-sharing consent by each farmer involved, which got too impractical for this project. As alternative approach, instead of working with information directly related to farms we chose to use detailed spatial grid cells (10m x 10m) as **units** for processing and the final causal modelling. We furthermore split the total study area into regions with predominant arable parcels and with grassland parcels (typical for dairy farming), so that the causal modelling could be tailored to these two major types of farming in The Netherlands.
- Given the availability of selected agriculture datasets relevant for our use case in the open-source format, we decided to use AgroDataCube to obtain the source data needed. These includes **crop code, crop rotation index, greenness and the number of mowing events for grassland parcels**. The parcel-oriented data has been extracted from AgroDataCube for multiple years, converted to 10m x 10m raster cells, and ingested by rasdaman into data cubes. In the causal modelling approach, we use these datasets primarily as a treatment but also as confounders due to links they are sharing.

Causal Machine Learning

- The final step in the workflow of the use case relates to the **attribution** part of the framework, in which agricultural activities (interventions) are causally correlated to changes in biodiversity richness (outcomes), considering the environmental variables (confounders and covariates). As part of the project, we investigated the causal modelling method that, given an expected causal correlation provided by domain experts (in discussion or via literature), validates the relation and calculates its robustness (as probability (p) value).

- It is recognized that as part of this project we could only take a first step into applying causal modelling in this domain, and needed to take a lot of assumptions and short-cuts which are open to scientific critique that hopefully can lead to future improvements of the methodology.

iii) Biodiversity Pillar Data Engineering and Machine Learning

Despite decades of biodiversity research, significant information gaps remain, hindering efforts to reduce uncertainties. Diverse data collection and monitoring protocols are used by scientists, public administrations, and organizations, but insufficient coordination and biases (spatial, temporal, and taxonomic) persist. To address these issues, the Essential Biodiversity Variables (EBVs) framework was proposed in 2013[1]. Since then, its acceptance has grown, along with research on integrating in-situ and remote sensing data for EBVs. In this use case we focus on species-related EBV classes[2], as illustrated in Figure 22.

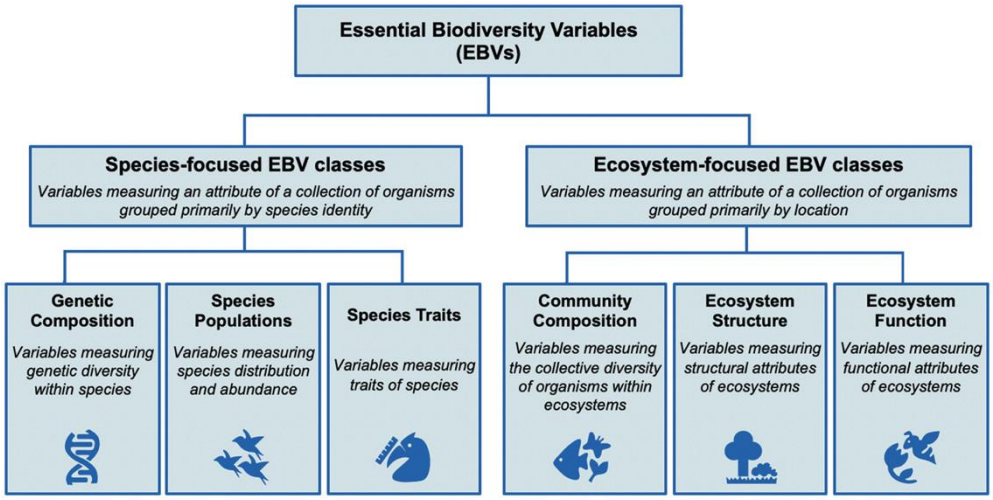


Figure 22 : Essential Biodiversity Variables

Fittingly, the data structure of an EBV is thought of as a space-time-biology hypercube, which should be a good fit for the multidimensional array data cubes used in this project. For this however we need the biology dimension to describe taxonomy and values to inform about presence/absence or population abundance. A key issue then becomes how to transform the in-situ observation data into 'gridded' data that matches the available data cube technology in the FAIRiCUBE project. The two used steps in this transformation are: (1) *primary data aggregation*; and (2) *model-based estimation (gap filling)*.

2.1.1.6 Primary data aggregation

Following the EBV production workflow described in literature the steps to take for the aggregation of the primary data, i.e. the species distribution data acquired from the Dutch NDFF ("Nationale Databank Flora en Fauna") and from GBIF, include: (1) *data harmonization*, (2) *data aggregation*, (3) *uncertainty estimation*, and (4) *metadata annotation*. In this case the required *data harmonization* is minimal, since it is part of the work already performed by the NDFF and GBIF organizations. Sufficient care must be taken though when selecting and filtering the species distribution data from both databases.

[1] Pereira, H. M., Ferrier, S., Walters, M., Geller, G. N., Jongman, R. H., Scholes, R. J., Bruford, M. W., Brummitt, N., Butchart, S. H., Cardoso, A. C., Coops, N. C., Dulloo, E., Faith, D. P., Freyhof, J., Gregory, R. D., Heip, C., Höft, R., Hurtt, G., Jetz, W., . . . Wegmann, M. (2013). Ecology. Essential biodiversity variables. *Science*, *339*(6117), 277-278. https://doi.org/10.1126/science.1229931

[2] Fernández, N., Ferrier, S., Navarro, L. M., & Pereira, H. M. (2020). Essential Biodiversity Variables: Integrating In-Situ Observations and Remote Sensing Through Modeling. In *Remote Sensing of Plant Biodiversity* (pp. 485-501). Springer International Publishing. https://doi.org/10.1007/978-3-030-33157-3_18

Initially, we only had three NDFF datasets from 2016 to work with (nesting birds, other species of interest, and plants). In the second phase we got available extended dataset covering the same spatial extent of study area but including all available species records for the period of years 2014 – 2022 as described in deliverable D3.1. In the third phase, we received extended dataset covering full extent of study area consisting of bird species between years 2014 – 2022.

Figure 23 shows a combination of multiple datasets created during the aggregation. Polygons with red boundaries represents individual observations. These get spatially intersected with the grid cells after which the intersections are used to calculate the shares of the abundance to assign to the cells. All shares are then summed per cell to get the final aggregated value. In the background the agricultural fields in the area are shown. The empty (not coloured) cells indicate 'unknown' abundance, since no species absence data is included in the processing.

Figure 23: A view of multiple datasets from the aggregation process

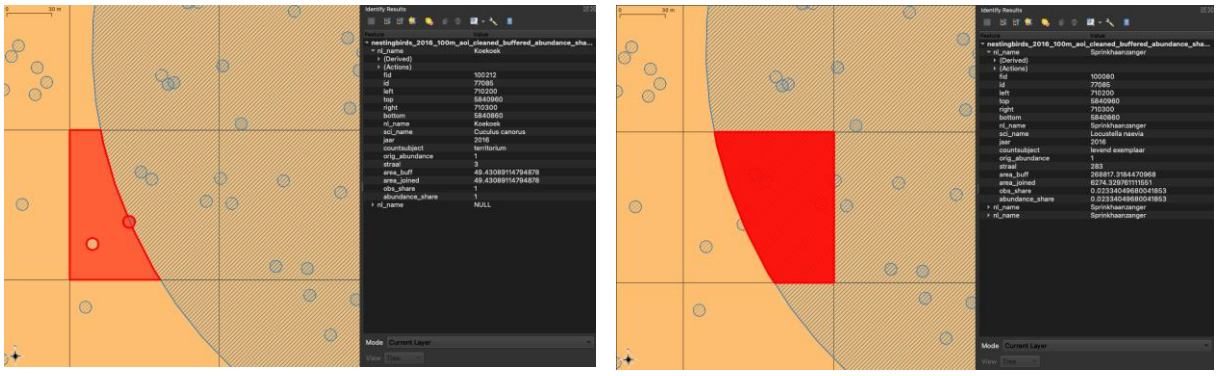


Figure 24: Examples of intersections generated during processing

In the second phase of data aggregation effort, we focused on breeding bird species of year 2018 and specifically on selection of Farmland Bird species (please see further explanation in section 2.2.4.2 of this report). Outcome of this aggregation procedure serves as an input to species distribution models (section 2.2.4.2). Following steps were taken to reach aggregated data layer:

- Load the Geopackage file with the species distribution data into a GeoPandas DataFrame (A).
  I. Select only the observations of birds.
  II. Select only the observations from year 2018.
  III. Select only the observations inside the study area of the use case.
  IV. Select only the observations of Farmland Birds and breeding birds.
- Create GeoDataFrame for each farmland bird species.
- Load a prepared grid (e.g. with 100m x 100m grid cells) into another GeoDataFrame (B).
  V. Calculate the spatial overlap (by union) (C) between A and B for each farmland bird species.
  VI. In C, calculate the species abundance proportion for the overlapping geometries (by area) for each farmland bird species.
- Merge (using a spatial join) the grid B with C, by covered geometries, creating dataset D.
  VII. Dissolve (aggregate) D by either counting the observations or summing the abundance shares.
  VIII. Assign the aggregated values to the cells of grid B and remove records with 0 value.
  IX. Create point dataset as centroids of grid cells of B and copy attribute values from b, giving the final output.

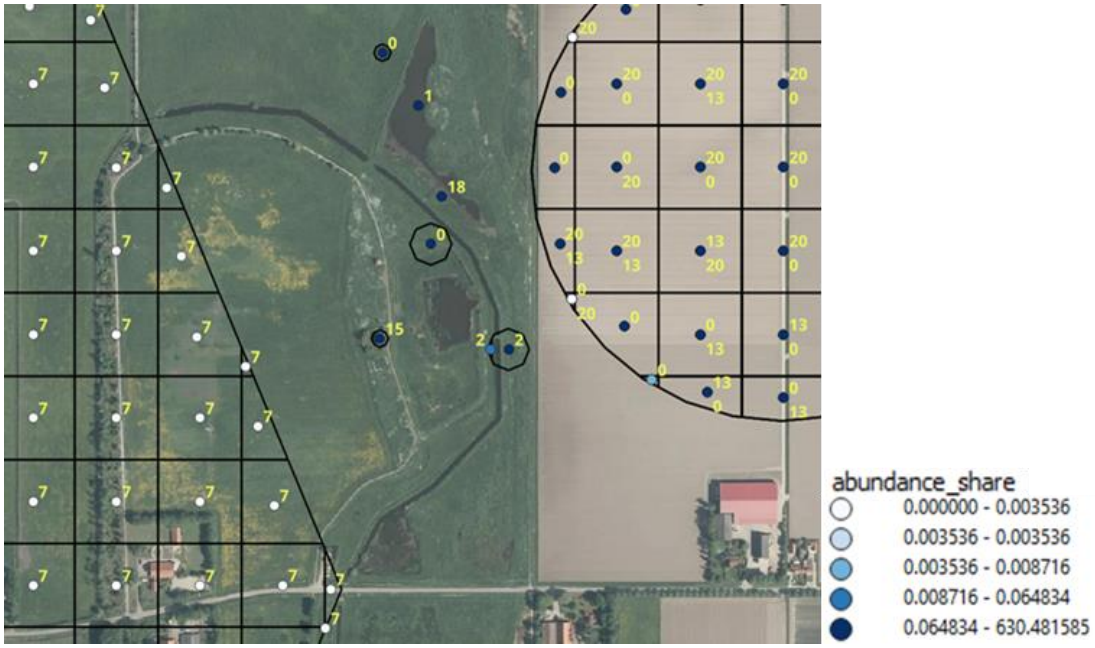


Figure 25 : Example of grid with abundance share and points generated per individual species

The approach described above needs some expert input to judge its validity, how it can be improved, and how eventually the *uncertainty estimation* can be done. When available, we might use existing species suitability maps for that, although they typically are not more detailed than 1km grid cells.

It is important to note that the spatial processing steps required in this algorithm can be time-consuming. The species distribution data contains numerous records, and the decision to work with small grid cells (e.g., 10–100 meters) results in a large number of potential spatial intersections that need to be checked and calculated geometrically. When processing times become excessive, even with spatial indexing methods like R-trees, distributed processing can provide a solution. Alternatives as *Dask*[1] and *PySpark*[2] were tested, but not fully implemented.

[1] https://www.dask.org

2.1.1.7 Model-based estimation

After aggregating primary data, the next step in creating biodiversity variables often involves species distribution modeling (SDM)[1]. SDMs are quantitative models that describe relationships between species distributions and environmental variables, using species location data (e.g., abundance or occurrence) and relevant factors like vegetation indices, land use, elevation, and climate data (e.g. Worldclim[2]). MaxEnt[3] a popular tool for SDM, uses presence-only data and environmental variables to create habitat suitability maps by predicting species occurrence based on maximum entropy. While MaxEnt is robust for presence-only data, it relies on many assumptions. When presence-absence data is available, alternative models or ensemble approaches may perform better. Additionally, advancements in deep learning, including generative modeling, offer promising alternatives[4] by accounting for factors like species co-occurrence[5].

Following are the steps currently applied within described modelling approach: (1) *Identification of covariates* (the environmental variables) from existing data cubes or ingesting new data when needed; (2) *Model learning and evaluation* using a few selected approaches (MaxEnt modelling results might already exist but perhaps only at a coarser resolution); (3) *Uncertainty estimation* of (the best possible) model; and (4) *metadata annotation*, which should again be covered by the FAIRiCUBE data ingestion process.

A first set of experiments is currently being run using the MaxEnt[6] model on observation data for 23 selected farmland bird species in our study area. These selected species are commonly used to calculate a 'Farmland Bird Indicator', that can serve as a proxy for assessing the biodiversity status of agricultural landscapes in Europe[7]. It includes birds that are dependent on farmland for feeding and nesting and are not able to thrive in other habitats. For these species the observation data from NDFF for years 2014, 2018 and 2022 is used.

As environmental variables the following datasets are used (as in first step all from 2018):

- LGN: Land use classes of the Netherlands, grouped into monitoring classes.
- OHN: Object Heights (in the Netherlands) on the land surface, derived from the AHN elevation dataset.
- NDVI: Normalized Difference Vegetation Index, 4 seasonal images (February, April, June, September) derived from Sentinel 2 data.

In the in first step due to the used MaxEnt model software requiring a Java runtime, for which installation was on that time under discussion at EOX, all processing was don done locally. In total 32306 species observation points derived from observation records have been used (0.8 MiB input file), and each species has been modelled individually (10 cross validation runs and 500 iterations of MaxEnt, taking about 17 hours on a M2 MacBook Pro with 16 GiB memory).To evaluate the performance of the model on the classification task per species the ROC (Receiver Operating Characteristics) curve can be used. It shows the true positive rate against the false positive rate at classification thresholds. The closer the curve is to the top left corner the better the model performs. The AUC (Area Under the Curve) indicates how well the model can distinguish between classes (Figure 21,22).

---

[2] https://spark.apache.org/docs/latest/api/python/

[1] Elith, J., & Franklin, J. (2013). Species Distribution Modeling. Encyclopedia of Biodiversity: Second Edition, 692–705. https://doi.org/10.1016/B978-0-12-384719-5.00318-X

[2] https://worldclim.org

[3] Steven J. Phillips, Robert P. Anderson, Robert E. Schapire. 2006. Maximum entropy modeling of species geographic distributions. Ecological Modelling, 190:231-259, 2006.

[4] Estopinan, J., Servajean, M., Bonnet, P., Munoz, F., & Joly, A. (2022). Deep Species Distribution Modeling From Sentinel-2 Image Time-Series: A Global Scale Analysis on the Orchid Family. *Front Plant Sci*, *13*, 839327. https://doi.org/10.3389/fpls.2022.839327

[5] Rademaker, M., Hogeweg, L., & Vos, R. (2019). Modelling the niches of wild and domesticated Ungulate species using deep learning. *bioRxiv*. https://doi.org/10.1101/744441

[6] Steven J. Phillips, Miroslav Dudík, Robert E. Schapire. [Internet] Maxent software for modeling species niches and distributions (Version 3.4.1). Available from url: http://biodiversityinformatics.amnh.org/open_source/maxent/.

[7] https://agridata.ec.europa.eu/Qlik_Downloads/InfoSheetEnvironmental/infoC35.html#_ftn1

In this study the (10 run) average AUC value per species varied between 0.71 and 0.93, with an outlier of 0.56 which can probably be related to faulty/noisy input data. The relative contribution of each of the selected environmental variables also differs per species and needs to be further analysed. OHN always seems to be one of the least contributing variables. Below are examples of results of two species Gallinago gallinago (Common snipe) and Carduelis carduelis (European goldfinch) as obtained from MaxEnt.

MaxEnt results for Common Snipe

Species: Gallinago gallinago (Common snipe)
https://en.wikipedia.org/wiki/Common_snipe

Mean AUC: 0.93
AUC stddev: 0.02
Training samples: 98
Test samples: 11
Datapoints: 10098

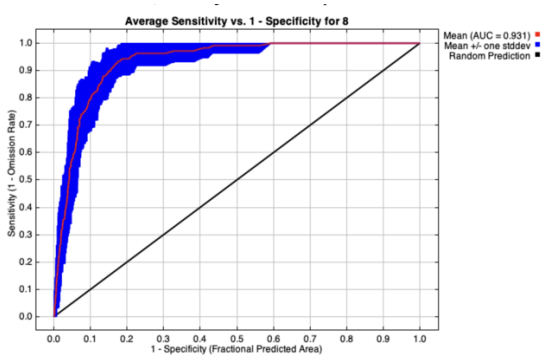


Figure 26 : 'Snipe' ROC and mean AUC

MaxEnt results for European Goldfinch

Species: Carduelis carduelis (European goldfinch)
https://en.wikipedia.org/wiki/European_goldfinch

Mean AUC: 0.79
AUC stddev: 0.03
Training samples: 418
Test samples: 47
Datapoints: 10418

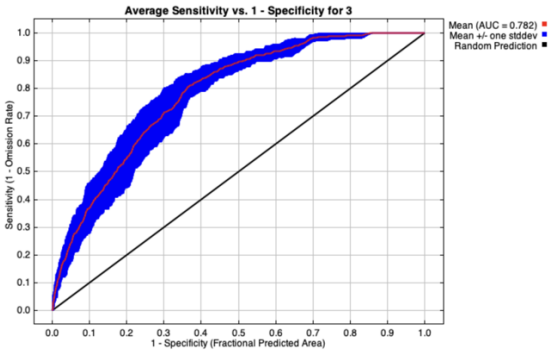


Figure 27 : 'Goldfinch' ROC and mean AUC

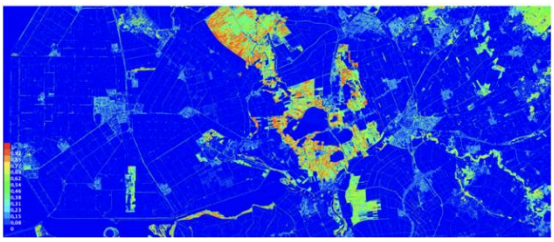


Figure 28 : 'Snipe' occurrence probabilities

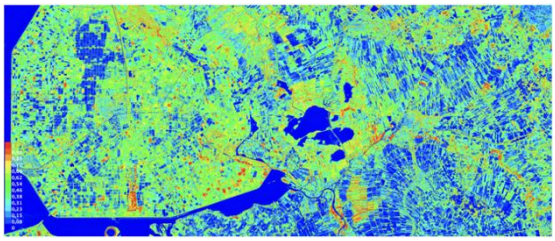


Figure 29 : 'Goldfinch' occurrence probabilities

Variable contribution (and permutation importance):
LGN: 69.3% [74.7%]
NDVI August: 10.2% [9.2%]
NDVI February: 8.8% [8.3%]
NDVI May: 8.6% [5.1%]
OHN: 1.7% [1.6%]
NDVI September: 1.4% [1.2%]

Variable contribution (and permutation importance):
NDVI February: 57.3% [40.3%]
NDVI May: 19.8% [26.2%]
NDVI September: 9.4% [15.1%]
NDVI August: 6.8% [15.1%]
LGN: 5.9% [6.6%]
OHN: 0.8% [1.2%]

In the second phase was MaxEnt modelling conducted by using the *elapid* [10] library in python, running at EOXHub. Due to computational constraints associated with the high resolution of the environmental layers (10 × 10 meters), the modelling was initially performed on a subset of the total study area. This subset, located in the northern part of the overall study area and known as the Noordoostpolder, corresponds to selected region of our use case and primarily consist of arable land. Consequently, MaxEnt modeling focused on a selection of bird species typically associated with such landscapes. Initially, in addition to the environmental variables described above, climatic variables (three-month average temperature and three-month precipitation sum) were included in the models.

However, due to the relatively small size of the subregion, these variables exhibited minimal variation between grid cells, leading to model overfitting on specific combinations. This resulted in unrealistic patterns in the predicted occurrence maps (Figure 30).

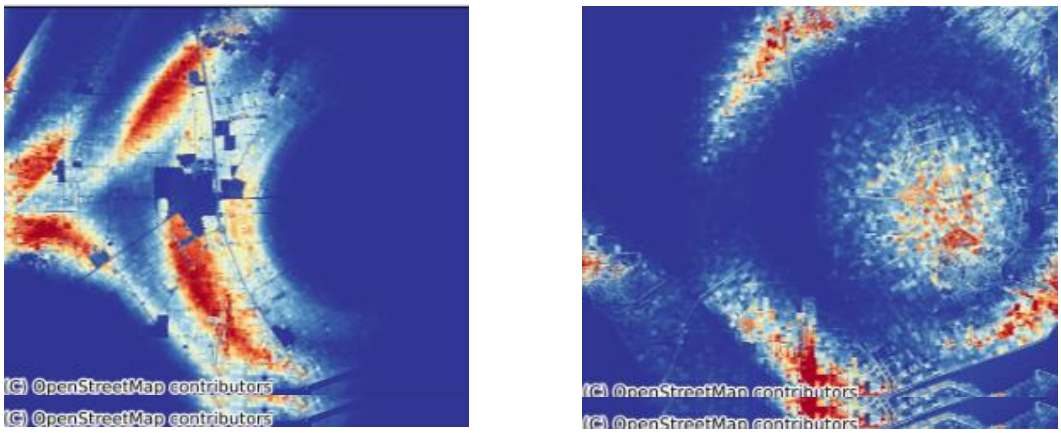
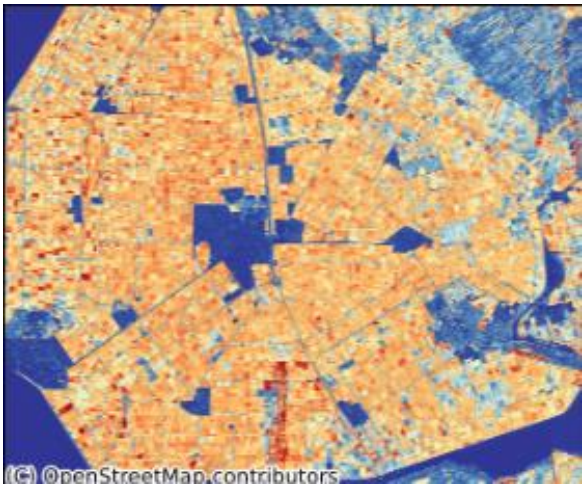


Figure 30 Example of MaxEnt output maps when climate variables were included

Table 1 shows the summary per bird species for the MaxEnt models with some evaluation parameters such as the AUC value and the omission rate. There is ongoing development on the modelling during the time of writing, which will result in including other metrics such as individual variable contribution and partial dependence scores/plots. These metrics will also be further considered in order to identify the influence of individual environmental variables, which will be of importance for the causal modelling phase.

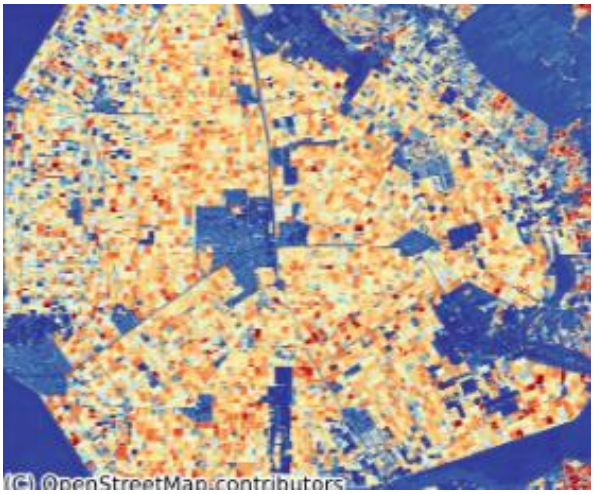
Table 1: Summary metrics for the MaxEnt modelling of a selection of bird species occurring in the Noord Oost polder, arable land region

Species	Number of samples	AUC	Omission rate
Skylark	730	0.705	0.104
Meadow pipit	1094	0.691	0.152
Goldfinch	316	0.798	0.345
Oystercatcher	712	0.737	0.169
Icterine warbler	129	0.851	0.279
Yellow Wagtail	1676	0.692	0.141
Tree sparrow	377	0.771	0.119
Starling	375	0.738	0.173
Whitethroat	315	0.860	0.365
Redshank	117	0.876	0.231

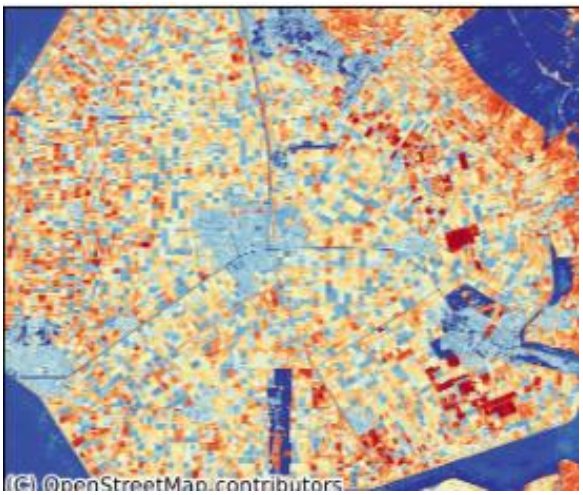


A) Skylark

B) Meadow pipit



C) Goldfinch



D) Oystercatcher

Figure 31 : MaxEnt probability of occurrence predictions for selected farmland bird species

From Table 1 it seems that for the models with a relatively low number of samples, both AUC values and the omission rates are higher. This can indicate that these models have a higher risk of overfitting, which can be caused when the variation between environmental variables is low. In the next steps this will be investigated further, with comparing more heterogeneous areas and fine tuning the models.

The maps above show some differences which can be explained by the ecology of the different bird species. For instance, species like the skylark and yellow wagtail, which are characteristic species for open arable landscapes, show high probabilities of occurrence in the open arable fields within the study region while they show a low probability of occurrence in the Northeastern corner of the study region, an area which is more wet and has rough vegetation and forested parts. This type of landscape would be more typical for a species such as the whitethroat, which is indeed predicted to have a high probability of occurrence here. Another noticeable difference is the fact that some bird species seem to avoid build up areas, while for example the gold finch, which has a more generalist habitat use which also includes villages and towns, shows less avoidance for build-up areas.

iv) Agricultural Pillar Data Engineering and Machine Learning

Regarding the agriculture aspect of the use case the intention was to make use of data from existing initiatives to reward farmers for more nature friendly agricultural management. The most challenging aspect of this data as it turns out is that it is privacy sensitive and not easily shared by farmers since it directly involves their business operation. Even when permission to use the data is given, proper pseudonymization / anonymization must be considered. Including any accidentally 'leaking' of such privacy sensitive data, e.g. by overfitted ML models from which data points can then still be extracted. While attempts were made to obtain (data usage) consents from a sufficient large number of farmers, among others by cooperating with internal projects that also needed them, in practice this has turned out to be too time-consuming and thus impractical for this project.

Finally, we decided to work with freely available agriculture data available at AgroDataCube, which is an open-data platform developed by WENR and provides access to diverse geospatial and agricultural datasets in the Netherlands. It offers detailed information on agricultural parcels, including field boundaries and annual crop types, as well as supporting datasets such as soil properties, weather data, vegetation indices (e.g., NDVI), and high-resolution elevation models. Additionally, administrative region boundaries and related metadata are available for regional analyses. These datasets are available per agricultural parcel, accessible via a REST API in GeoJSON format. Relevant for our use case we found following datasets:

- Crop Rotation Index (CRI) is a metric developed to assess and reward sustainable agricultural practices by evaluating crop diversity and management on fields over a 6-year window. It incorporates eight indicators, such as crop diversity, frequency of certain crops, inclusion of green manure, and intervals between planting the same crop, to quantify impacts on soil health, biodiversity, and ecosystem services. Each indicator is scored using response functions, combined into a single score ranging from 0.0 (least favourable) to 1.0 (most favourable).
- Greenness – indicates what are the field conditions on spring and autumn i.e. presence or absence of any vegetation
- Number of Mowing Events - refers to the count of mowing activities recorded for a given agricultural field per year, which gives information on how intensive management of the grassland is.

As an additional data layer relevant to agriculture, we use Small Woody Features (SWF). This dataset is available through Copernicus Land Monitoring Services and maps small woody elements within a landscape. These features typically include hedgerows, tree lines, small clusters of trees or shrubs. Acting as ecological corridors and stepping stones, they connect fragmented habitats, facilitating species movement and pollination across landscapes. SWFs support diverse flora and fauna, create stable microclimates, and contribute to soil and wind protection, indirectly benefiting ecosystems. By enhancing landscape heterogeneity, they promote greater species richness and resilience, making them indispensable for sustainable biodiversity conservation and ecosystem health.

Considering spatial units when looking at farming activities as interventions, the logical unit to choose is a (spatial) agro-ecosystem. Initially we wanted to look at farms, using direct farm data as input. However, in this would require consent for the data use by every farmer involved, which was too impractical. The other choices considered are: (i) a farming region of sorts; or (ii) spatial grid cells. Important things to keep in mind are required computation performance (the solution needs to be tractable), and geospatial biases (e.g. land from a single farmer will not receive randomised treatments, and farmers are affected by neighbours, communities, and local and national government). To avoid getting lost in the process of defining farming regions and having to pre-process data once more, we will start by working with grid cells as unit and handle spatial biases and data imbalance by pre- or post-processing of the grid cell-based inputs and outputs. The size of the grid cells should make sense given the characteristics of the data used, the (local) effects studied, and the computational resources available. Either 50m x 50m or 100m x 100m can be a good size for initial experiments. Based on first results and remaining time available the size can be increased or decreased as needed.

v) Environmental Pillar Data Engineering and Machine Learning

This pillar concerns the more 'classical' Earth Observation data storage and processing, which should be most native to the available FAIRiCUBE platform since they are already gridded. No specific need for machine learning is expected. Rasdaman offers extensive data processing functionality via its query language (*rasql*), and if needed it can be extended with *User Defined Functions* (UDFs).

Most challenging in process to ensure compatibility among different datasets is that the environmental data are usually available as raster data, while the other two kinds of data used (biodiversity and agriculture) are vector data (or 'plain' tabular data with columns for coordinates or a location indication). These need to be matched by choosing an adequate grid (including cell size), one (or more) suitable coordinate reference systems (CRS), and proper transformations. Specifically for machine learning, mis-aligning grid cells or deforming the data too much (e.g. by accumulating transformation errors) can easily lead to a garbage-in, garbage-out situation, that might go undetected or take a long time to figure out. For deep learning a possible way to counter can be to focus on models that generalize better, for example by including (or simulating) coordinate transformations as part of the regular data augmentations.

Ingestion of data into data cubes is completed, while procedures and validation approaches are being evaluated and tested. Common datasets, such as remote sensing-based data (e.g. Sentinel imagery) and products derived by Copernicus, are generally available though in rather 'raw' form. Therefore, to obtain cloud free image mosaics of the Sentinel imagery further processing was performed at rasdaman characterized by following steps:

1. Collect the relevant Sentinel-2 bands for the months Feb, Apr, Jun, Sep for 2018 and 2022, intersecting the central Netherlands bbox (including the SCL classification band and CLDPRB cloud probability band)
2. Remove scenes with more than 50% cloud cover based on the CLOUDY_PIXEL_PERCENTAGE from the scene metadata
3. Create a cloud / shadow and invalid pixels mask, which marks any pixels with CLDPRB > 20% and SCL values different from 2, 4, 5, 6, 7, 11 (these are documented in the Sentinel-2 product guide)
4. The SCL mask contains a lot of bogus small patches of a few pixels: these are removed with an erosion step with a 5x5 kernel
5. The remaining cloudy patches are dilated by 5 pixels to avoid halos or other artifacts as much as possible
6. The mask is applied to the bands
7. The masked bands are reprojected to EPSG: 28992 and clipped to the bbox
8. The multiple scenes per day are mosaiced into one file per day
9. Tor a whole month, the daily files are stacked by taking the *median* pixel value. In first attempt we used method starting from 15th of the month and filling in the masked gaps, but this produces a lot of artifacts. We decided to use median method as it seems to be standard for producing cloud-free mosaics.

Access coverage datasets at rasdaman has been tested with a Python Notebook (using the WCPS API and running on EOXHub). A required next step to use the data for machine learning / deep learning is to bridge between this WCPS API (common in the EO domain) and the Datasets and DataLoaders used by the frequently applied Python ML frameworks, such as PyTorch and Tensorflow (Keras). No existing implementations (e.g. software libraries or packages) that handle this have been found so far.

vi) Causal Machine Learning

As mentioned before in this use case Causal Machine Learning has been applied to validate causal correlations between agricultural activities and biodiversity richness, taking as starting point the expected relations indicated by domain experts and/or found in relevant scientific literature. Early in the project we envisioned that the input data, model explanations, and results could be presented in a dashboard (see Figure 32) like application that would be easy to use by interested researchers and possibly stakeholders.

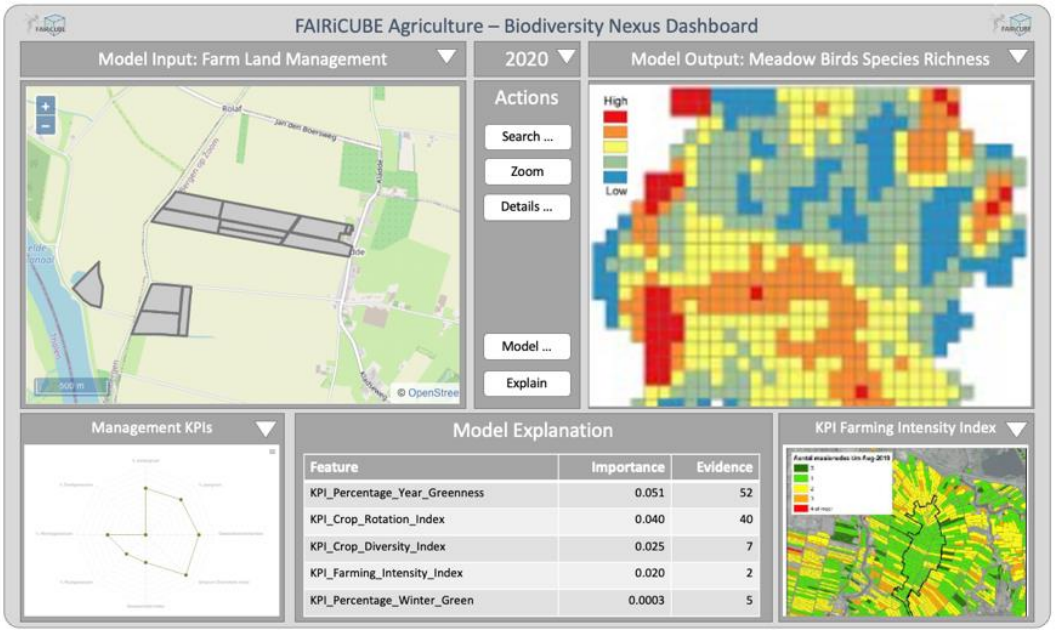


Figure 32 : Initially envisioned UC2 Model serving - Dashboard application mockup

While this goal could not be fully realized, many of the underlying steps have been performed as indicated in the realized conceptual workflow overview (in Figure 21), and most of this work could successfully be performed using the FAIRiCUBE Hub and all provided resources. Unfortunately, within this use case we ran out of time to be able to fully test and apply all the functionality that we requested, however as they are very generic machine learning tools (such as TensorBoard, MLFlow, headless notebook execution with GPU access), we trust they are still valuable extensions to the Hub.

Specifical to causal modelling, in the exploration phase we considered different available Python packages that can provide comprehensive analysis using provided causal graphs (directed acyclic graphs, or DAGs). Besides causal inference based on provided (expected) DAGs or equations, there are also methods for causal discovery that try to extract a graph from observational data. We will however not be able to investigate these within this project.

After careful consideration the PyWhy[1] library (containing the DoWhy Python package) was selected for use in this use case. With this package we constructed causal graphs based on the information obtained from domain experts, and using the prepared data for outcomes, interventions, and possible covariates and confounding variables. Figure 33 shows a possible causal graph that can be evaluated for validity and robustness. Next, the causal models were trained on these assumed hypotheses. The outcomes of the causal models are evaluated using the following three steps:

[1] https://pywhy.org

- **Validation of the Causal Graph:** In the first step, it is checked (by consulting domain experts) if the casual graph and the connections between different variables make sense. The main point of this step is to ensure that the causal effect we want to estimate is valid under the assumptions of our graph.

- **Estimation of the Causal Effect:** In the second step the numerical casual effect of our treatment variable (e.g. number of grasslands mowing events) on the outcome (farmland birds biodiversity richness) is estimated, using 10m x 10m grid cells as units. A positive value indicates a positive relationship between treatment variable and outcome, i.e. a higher value in treatment variable is associated with an increase in outcome, on average.

- **Robustness Evaluation:** With this final step the robustness of the estimated causal effect is evaluated by applying a placebo test. This test will replace the treatment variable with a randomly generated placebo variable. Ideally, the result this should be close to 0, as the placebo variable should have no causal relationship with the outcome. Besides, this test also computes the probability (p) value to check if the estimated casual effect of the treatment from step two is *significantly* important or not. A high p-value (typically above 0.05) suggests that the placebo effect is not significantly different from the original causal estimate, which weakens confidence in the robustness of the causal estimate (and thus the examined hypothesis).

- When results are not satisfactory, i.e. they do not align well with expert judgement or real-world findings from experimental studies, adjustments to the causal graph as well as to the input data used for model training can be made and tried in additional experiments to improve the model's accuracy and reliability, and the insights gained from this use case.

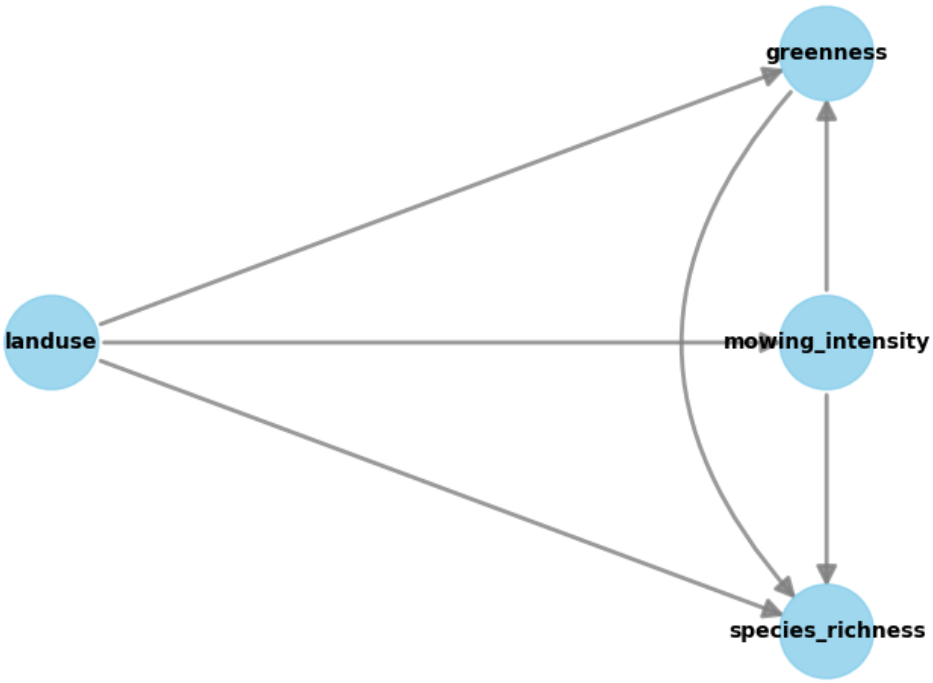


Figure 33 : Example causal graph for "mowing_intensity" and "species_richness", with confounders and covariates

UC3 Biodiversity occurrence cubes – *Drosophila* landscape genomics

UC3 exploits the massive collection of DNA sequenced data of natural populations of the fruit fly *Drosophila melanogaster*. Apart from the challenge to adapt the data for storage as data cubes, the WP3 supports the processing and potentially application of machine learning algorithm to enrich the data set and reveal further insights while making advantage of the scalability and accessibility of data storage and processing capabilities of the FAIRiCUBE Hub.

i) ML based gap filling

As a first demonstrator task, gap filling of missing information at single nucleotide polymorphisms (SNPs) in the genome-wide sequencing data based on pools of individuals that were sequenced jointly (Pool-Seq) was identified. The provided data set DESTv2[1] comprises of more than 730 samples of *Drosophila melanogaster* populations from all across the world. In a first step, we focus on populations from North America, which are predominantly collected along the East Coast. Many of these samples are densely and repeatedly collected across multiple seasons and years. This dataset, available in the "Variant Call Format" (VCF) format, contains data entries for several million polymorphic positions along the whole genome for each population. However, these positions do not necessarily contain information for the entire number of populations. Additionally, missing data in one genomic position does not necessarily reflect which populations is missing data in another position. To perform valid statistical analysis, positions with missing data cannot be included and the data used for various studies only represents a subset of the data available.

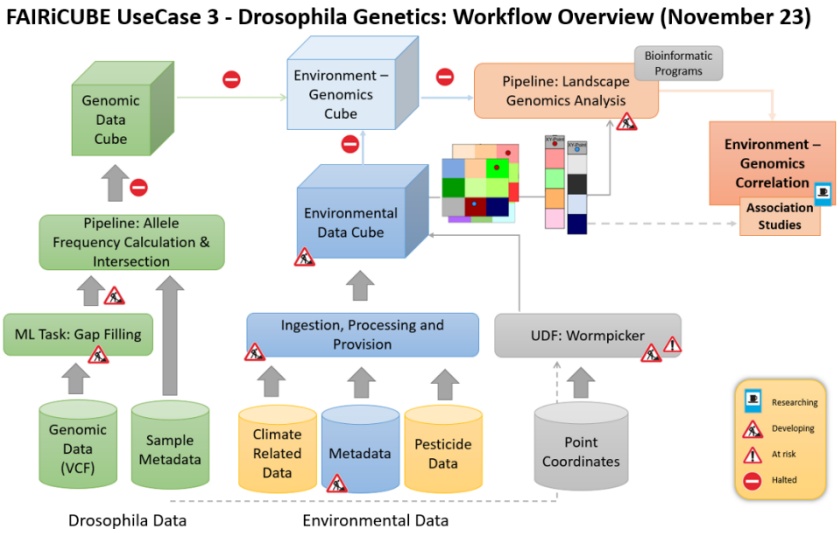


Figure 34 : UC3 data analysis and processing workflow. Machine Learning based gap filling methods can be applied to genomic data (lower left green cylinder: "VCF") to avoid non-usability of valid information and data on samples due to lack of data in other samples.

As described in the exploratory data analysis in deliverable D3.1, the provided test dataset that was generated to develop this method excludes all SNP positions with missing data in at least one sample, in contrary to the original DESTv2 dataset which characterized by different levels of missing allelic information in the population samples (D2.2). To evaluate the accuracy of an applied gap-filling method, artificial gaps, i.e., missing data will therefore be introduced that mimic actual gap

[1] https://dest.bio/

characteristics. The number of missing data sums up to about 5% of the total amount of allele frequencies and the distribution of the number of subsequent SNP positions follows an exponential function, that we observed for the relationship between number of gap-occurrence and gap-size. Accordingly, most of the gaps that have been introduced consist of single SNP locations.

A previous attempt has been made to gap-fill the data set by imputing missing allele frequency data based on information from close-by populations using inverse-distance-weighting (IDW) based on geographic and temporal distance of neighbouring populations. The DESTv2 dataset is particularly well suited for this, as many populations are closely sampled across space and time. To consider isolation-by-distance, due to limited dispersal among populations which reduces the relatedness (and potentially the similarity in allele frequencies), we restrict the inference to neighbours, which are within a user-defined geographic distance (in kilometres) and temporal distance (days between sampling dates). Moreover, we used IDW to average allele frequencies of neighbouring populations as a function of their proximity to the focal population with the missing allele frequency. This approach has been tested against a dataset who's artificially introduced gaps are distributed similar to the real dataset and provided satisfying results. However, this approach assumes that all neighbouring populations have an equal relation regardless of geological boundaries, which may deviate from realistic patterns. In the following, this gap filling approach is addressed as an *empirical baseline*.

An alternative approach will be provided through clustering of the existing populations regardless of their origin in time and space. This is justified since mutation pattern can occur independently and recurrently. In preparation of the clustering data using machine learning techniques, the allele counts are converted from the VCF file format to allele frequency data (based on the reference allele) in TSV format. The corresponding files are then loaded, stripped off the header information, and converted to a NumPy array. In the context of the clustering approach, the populations are considered data points that are to be clustered, and allele frequencies are the features by which the clustering is performed. However, as there are 50,024 allele frequencies per population in the test dataset, we attempt ML clustering in 50,024 dimensions. Accordingly, choosing a clustering algorithm by reviewing and visualizing cross plots of data is impossible. As an outcome of exploratory data analysis in deliverable D3.1, we have identified the position of allele frequencies with the highest variance across the populations which widens the spectrum of differences in allele frequencies. After all, we do not want to cluster based on similarities of data but by different characteristics. By selecting only 99.5 % of the most variable allele frequencies, we reduce the dimensionality of our clustering to 251. Based on this, we can still not truly identify patterns of the data but can attempt a blind run of a *k*-means algorithm with application of the elbow method to find an optimal number of clusters *k* as input for the actual clustering.

k-means algorithms are usually the first choice in clustering attempts due to their simplicity and computational efficiency (see also chapter 0 for description of other clustering methods). After initialisation with random *k* centres of clusters, each data point is assigned to the closest cluster centre and an iterative process, the centres of the clusters are updated by re-calculating the means (centroids) of each cluster and re-assigning data points. The performance of *k*-means clustering depends on the number of chosen clusters *k*, the initialization of cluster centres and, most importantly, on the nature of data. Specifically, the dataset needs to exhibit higher-dimensional spheres as imposed by definition of the distance metrics to the nearest cluster centre and the calculation of means to update the new cluster centre. If data clusters shape differently, alternatives such as spherical k-means algorithms or DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can be explored. Data with high dimensions can be further reduced in dimensionality using auto-encoder deep neural networks (DNN) which try to extract relevant characteristics instead of all dimensions of the data points – similar to our attempt to identify the most variable allele frequency positions. A subsequent application of e.g., k-means is then performing the actual clustering of lower-dimension data.

To find the optimal number of clusters *k* for our k-means clustering we apply the elbow method and perform several k-means clustering with a range of *k* from 1-15 and calculate the WCSS (Within-Cluster Sum of Square) for each run. When analysing Figure 35 and the plot of WCSS as a function of number of clusters *k*, we can observe a typical "elbow" shape of the curve which recommends k=5 for an optimal k-means clustering. A subsequent test with the silhouette method confirms this choice.

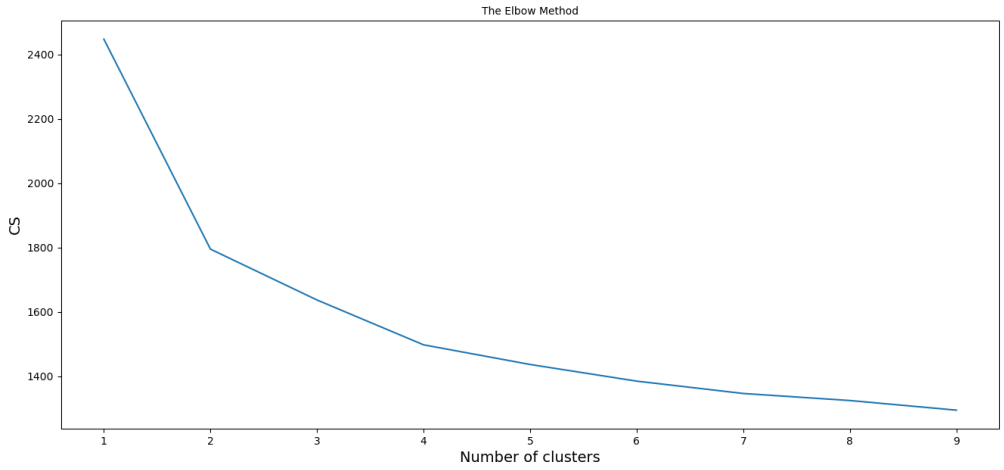


Figure 35 : Applying the elbow method to determine the optimal number of classes for the k-means clustering.

We can therefore perform a *k*-means clustering with *k*=5 and assign each population to a cluster number. Arguably, the recommended number of clusters is low which indicates large similarities in the allele frequency information across the populations and reflects the evolutionary history of world-wide *D. melanogaster* population very well (Kapun et al. 2020). The number and distribution of the clusters potentially thus reflects the geographic sampling and the evolutionary history of the investigated populations.

In the subsequent gap-filling step we can replace missing allele frequency data in one population with a mean value from populations belonging to the same cluster. The gap filling has been tested for selections of populations in Europe and North America separately. When assuming the introduced gaps to be of zero values, we can calculate the summed difference and root mean square (RMS) error of data with and without gap. After gap-filling, we can compare the summed difference and RMS and calculate the ratio that indicates how well the gap-filling performed. The results of both the k-means clustering with *k*=5 and the previous inverse-distance weighting method are shown in Table 2. In the following the *k*-means gap filling approach is addressed as *machine learning (ML) baseline*.

Table 2: Statistics on the gap filling methods applied to selected populations.

	North American populations	European populations
Number of populations	121	332
Number of samples	6 152 952	16 708 016
Number of gaps	125 000	338 000
Total number of NaN	309 753	837 327
Percentage of gaps	5.03	5.01
Summed difference of gap data (rounded)	33794	90059
Summed difference after k-mean gap filling (rounded)	8216	256
RMS error of gap data	0.05189	0.05508

In order to improve the accuracy of the gap-filling we now employ sequential deep learning models. There are two types of models that we use: generative models and non-generative models. Generative models try to learn a probability distribution over the data and calculate missing values by sampling from the learned distribution. On the other hand, non-generative models try to impute the missing values directly from the data by exploiting the relationship between the known data points.

Table 3 shows the RMSE scores for the methods used to impute missing allele frequencies. The deep learning methods are benchmarked against the empirical and ML baseline. The best performing methods are the Masked Autoencoder, Variational Autoencoder and Generative Adversarial Network. A VAE operates within the framework of unsupervised learning, compressing data while maintaining the structure of input data. It introduces stochasticity in its encoding process, resulting in a structured, continuous latent space ideal for generating new samples. Similarly, Masked Autoencoders function in unsupervised settings, primarily focusing on reconstructing incomplete input data. They operate by masking a portion of the input data and learning to predict these masked values, effectively training the model to identify and fill in missing information. This approach is particularly useful in handling incomplete datasets, as it enables the model to learn the underlying structure of the data and accurately impute the missing values based on learned patterns.

GANs, on the other hand, comprise two neural networks: a generator and a discriminator, functioning in a competitive setup. The generator fabricates data aiming to pass as genuine, while the discriminator improves its ability to differentiate between real and synthetic data. Over time, the generator becomes increasingly proficient at producing realistic data. Apart from MAE, VAE and GAN, we tried several other well-known machine learning and deep learning architectures such as: the KNN imputer that imputes missing values with K-nearest neighbours, Gaussian Process Regression (GPR) that imputes values by sampling from a learned distribution from the available data, Temporal CNN (TCN) is tailored towards sequential data based on supervised learning, LSTM/Bidirectional LSTM (LSTM/BiLSTM) are the go-to methods for sequential tasks and has the ability to learn sequential dependencies. These methods outperformed the inverse distance method and performed admirably.

Table 3 : RMSE on missing allele frequency imputation

Methods	RMSE NA	RMSE EU
No gap filling	0.05189	0.05508
Inverse distance (empirical baseline)	0.04330 = 19%	0.04511 = 18%
k-means clustering (ML baseline)	0.01137 = 79%	0.00131 = 98%
Masked Autoencoder (MAE)	0.00142 = 97%	0.00231 = 96%
Variational Autoencoder (VAE)	0.00467 = 92%	0.00093 = 98%
Generative Adversarial Networks (GAN)	0.00019 = 99%	0.00134 = 98%

VAE and GAN stand out in gap filling for genomic sequences due to their capability to generate plausible data points, crucial for handling the high-dimensional nature of genetic data. VAEs adeptly learn data distributions, enabling accurate imputations of missing values, while GANs, with their distinctive network architecture, create realistic sequences, enhancing data authenticity. However, the performance of GANs is hard to replicate as these models are notorious to train. It requires a lot of fine-tuning and balancing training between the generator and the discriminator. On the other hand, MAE also show comparable effectiveness in reconstructing incomplete data and predicting missing information, a key aspect in genomic studies where accurate and biologically plausible imputations are essential. All these models surpass traditional methods by innovatively maintaining genomic sequence integrity, however the family of autoencoder models are simpler, faster and easier to train.

Generally, even the fairly simple k-means ML clustering produce a highly accurate reconstruction of the missing data. Deep learning methods beyond the ML baseline model increase the accuracy as they have more layers and parameters which are capable of learning high dimensional interactions and correlations, especially in this case, where the input is a long sequence of 50,024 allele frequencies. The models defined above are capable of extracting features in such high dimensions. Also, another factor that contributes to DL methods' superior performance over ML baselines is that the DL models

have been defined and implemented in a such a manner that the sequence of allele frequencies is preserved. This is imperative in this case as we are dealing with sequential data.

As a next step, we will focus on the actual DESTv2 with gaps that comprises of more than 730 population-based samples of Drosophila melanogaster distributed over the globe and more than 3 million SNP positions. This data will have gaps where we don't know the true values to compare against. Only the general distribution of Allele frequencies and a-priori information can be used to validate the results. It will further be numerically challenging to handle the whole records of more than 3 million SNP positions at once, we will therefore consider a window approach to input missing data piece by piece.

ii) Landscape Genomics (Environmental Association Analysis)

We conducted a landscape genomics analysis using the *Drosophila* genetic data from the DEST dataset for European populations and selected environmental variables to identify the key factors driving genetic variation. Our approach included performing and comparing a multivariate method, namely redundancy analysis (RDA) using the vegan package in R, latent factor mixed models (LFMM), which can account for hidden factors like population structure, using the LEA package in R and linear models, also performed in R, to evaluate the relationship between genetic data and environmental data. We compared the outcomes of these methods to identify which environmental variables were consistently influential across approaches and which single nucleotide polymorphisms (SNPs) were identified as important in each analysis.

Linear models build a baseline of association and are performed for each environmental variable individually, therefore underlying structures in the dataset cannot be captured and relationships between variables remain undescribed when looking at pure correlation statistics. Latent factor mixed models offer the possibility to account for such underlying or hidden factors such as population structure in the case of genomic data, from which it is known to exists in the DEST dataset. Multivariate approaches are generally considered more restrictive and are applied in the field of landscape genomics quite often, since their power to test under different scenarios can help avoid jumping into fast conclusions.

For our Use Case, we applied RDA, a multivariate approach combining regression analysis and principal component analysis by taking an environmental data matrix as explanatory variables and the genetic data (allele frequences) as response variables. Further, we performed partial RDA to test different conditioning scenarios hypothesized, determining explanatory power for each scenario by doing a variance partition analysis on their respective R-squared measures. The scenarios tested were a full (unconstrained) model, a climatic model, a population genetic structure model and a geographic model.

Table 4: Variance partitioning of different RDA models.

Partial RDA models	Inertia	R2	p (>F)	Explainable Variance	Total Variance
Full model: F ~ clim. + geog. + struct.	1.108,92	0,27	0,001	100,00 %	55,86 %
Pure climate: F ~ clim. | (geog. + struct.)	375,84	0,06	0,001	33,89 %	18,93 %
Pure structure: F ~ struct. | (clim. + geog.)	189,38	0,05	0,001	17,08 %	9,54 %
Pure geography: F ~ geog. | (clim. + struct.)	41,70	0,01	0,001	3,76 %	2,10 %
Confounded climate/structure/geography	**501,99**			45,27 %	25,29 %
Total unexplained	**876,12**				44,14 %
Total inertia	**1.985,04**				100,00 %

Associations study is a popular tool to detect genomic loci that putatively are strongly influenced by environmental factors. The outcomes of the linear model as well as the LFMM in our analysis approach are p-values calculated for the association of each SNP with a single environmental variable. We therefore obtain a p-value matrix with n number of SNPs used and m number of environmental factors. From there, a significance threshold can be set to determine the SNPs (loci) showing association with an environmental variable. In the case of our EAA we consistently use a Bonferroni corrected significance threshold of 0.05, divided by the number of SNPs, to account for multiple testing.

For redundancy analysis (RDA), we followed two complementary approaches to identify genomic outliers in the climatic scenario in R. We first used a function called outliers (popgen.nescent.org), considering the loading scores of individual SNPs on individual RDA axes as indicator of association. To filter for significant associations, outliers on both ends of the score distribution are selected by setting the threshold the standard deviation matching our desired p-value threshold (two-tailed).

Second, we used a function called *rdadapt* to calculate and define outlier loci. This approach includes calculation of a multivariate mean and is testing the distances to this mean, instead of identifying outliers, based on their single axis mean. Both outlier detection strategies are based on association of the environmental predictor with RDA axes of the data instead of the single environmental variables. Therefore, direct comparison of significant loci of multivariate methods with significant loci from univariate methods is limited.

iii) Species distribution modelling of urban *Drosophila* flies from Vienna/Austria

Complementary to the analyses of a genomics datasets based on wild-caught natural *D. melanogaster* populations across Europe, which aim at investigating genomic signals of adaptation to Europe-wide environmental variation in a focal species, we investigated biodiversity in fruit flies within the city of Vienna to assess how urban environments influence the ecology and community composition of *Drosophila* species. Many of which are often found as human commensals and may therefore even benefit from a strong anthropogenic influence.

By initiating a citizen science initiative, we were able to collect approximately 20,000 flies in 290 traps that were placed within the volunteers' homes during the sampling period from June to Dezember. We additionally obtained detailed administrative, land use and climatic data for the sampling region as described in deliverable 3.1, by collaborating with UC1 and by collecting and processing climatic data from GeoSphere Austria.

In a first step we employed redundancy analysis (RDA) by fitting scaled environmental data to the abundance data of the 13 *Drosophila* species that we detected and identified. To avoid overparameterization of the model, we employed forward stepwise regression as implemented in the ordi2step function of the vegan package in R and only retained the optimal combination of parameters that maximizes the variance explained. The final model explained more than 20% of the variance in the species community data (R^2 =23.5%) and was based on 13 significant environmental variables, among which monthly sunshine hours, building volumes and street had the strongest influence (see Table 4).

Table 4: ANOVA table showing the significance of the effects of 13 environmental variables on species abundance based on redundancy analysis.

Effect	Df	Variance	F	Pr(>F)	
Monthly_SA	1	0.026255	18.0303	0.001	***
Wien_build_volume_i_grid	1	0.024514	16.8346	0.001	***
greenstreets	1	0.015491	10.6385	0.001	***
looseresidential	1	0.013638	9.3661	0.001	***
Monthly_RR	1	0.008443	5.7981	0.003	**
vineyards	1	0.00568	3.9007	0.007	**
INCAL_WindSpeedEast_daily	1	0.006256	4.2961	0.005	**
INCAL_RelativeHumidity_daily	1	0.005031	3.4551	0.012	*
Daily_TN	1	0.006448	4.4284	0.006	**
forest	1	0.00416	2.8569	0.034	*
Yearly_SA	1	0.00436	2.9945	0.017	*
imp2018	1	0.0038	2.6094	0.034	*
parking	1	0.004237	2.9096	0.019	*

As shown in Figure 36, we observe strong differences in species-specific patterns, particularly in the two most common species, *D. mercatorum* and *D. melanogaster. D. mercatroum* abundance appears to be associated strongest with building volume, which may suggest that this species prefers urban environments compared to *D. melanogaster*.

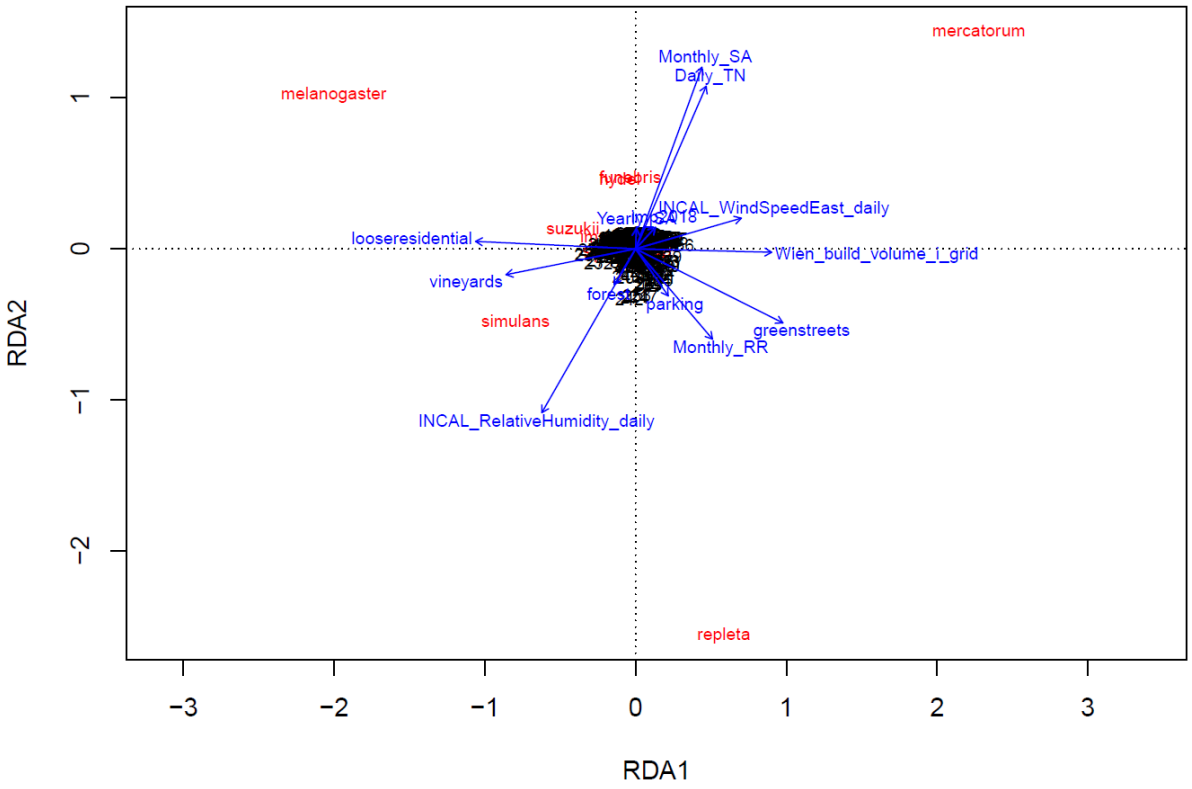


Figure 36 : Scatterplot showing the distribution of Drosophila species (in red) along RDA axes 1 and 2 based on their abundance. Blue arrows indicate the correlation of environmental variables with the RDA axes, with arrow lengths representing the strength of the correlations.

Complementary to this, we employed species distribution modelling for each species separately using a random forest approach. The count data in all 290 samples were first transformed using the Hellinger transformation implemented in the vegan R package, followed by splitting the dataset into training (80%) and testing (20%) subsets. Random forest species distribution models (SDM) with 500 trees were fit using the ranger function from the ranger R package to analyse the effects of yearly environmental, administrative, and land-use variables (44 variables in total) on the abundance of the focal species. To evaluate model performance, we employed 5-fold cross-validation and calculated metrics such as R^2 and RMSE. We then used the predict function from ranger to estimate expected abundances for each grid cell in the sampling area. Finally, we visualized the spatial abundance predictions using ggplot2 in R (see Figure YYY).

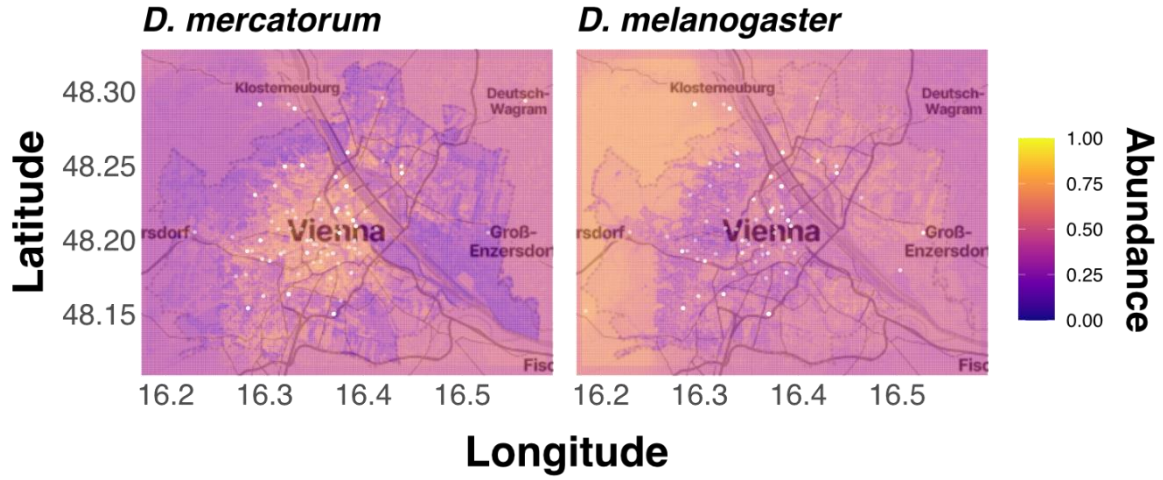


Figure 37 : Heatmaps showing the predicted abundance of the two most common *Drosophila* species in Vienna, based on random forest species distribution models using 44 predictor variables.

These analyses confirmed the previous results from the RDA as the random forest models for *D. mercatorum* indicate that this species is supposed to be more common within the city of Vienna. *D. melanogaster*, in contrast, appears to be much less common in the city and more likely to be found in the suburban, more rural regions surrounding the inner city. However, for *D. melanogaster*, the test R-squared (0.149) is low, indicating the model has limited predictive power for unseen data. Conversely, for *D. mercatorum*, the test R-squared (0.547) is reasonably good and shows that the model performs much better for this species (see Table Table 5).

Table 5: Summary of test statistics to evaluate the performance of the random forest SDM models for the two focal species.

Performance Metrics for random forest models	*D. melanogaster*	*D. mercatorum*
Training R-squared	0.415	0.5
Training RMSE	0.293	0.254
Test R-squared	0.149	0.547
Test RMSE	0.354	0.25
Test MAE	0.311	0.213
Cross-Validation R-squared	0.098	0.297
Cross-Validation RMSE	0.367	0.3

In summary, the two complementary analyse highlight previously unknown differences in ecological niches occupied by different *Drosophila* species in urban environments, which represent highly dynamic landscapes that are particularly affected by the accelerated climate change. Combining high resolution in combination with long-term monitoring of urban *Drosophila* communities will thus prove helpful to further depict the influence of climatic and anthropogenic effects on biodiversity in the wake of ongoing climate and biodiversity crises.

UC4 Spatial and temporal assessment of neighbourhood building stock

To create a building stock model that can be used to estimate the building energy performance, climate gas emissions and the building's material composition, we take the join outcome of the IEE Project EPISCOPE and TABULA[1] as a starting point (see also the final EPISCOPE report[2]). Country wise, a building classification is presented and a straightforward energy performance and building composition estimation is published there. Given the availability of the input parameters (construction year, building type and total floor area) we can thereby directly link public city data to our desired output parameters.

First, we will focus on the parameter *total floor area* which can be derived from the building volume being – in the simplest definition of the building Level of Detail 1(LoD) - the product of the ground area and the building height. However, finding complete height buildings datasets is challenging. Gap filling is therefore a must, and different methods need to be considered. Even though we strictly only need the number of floors and the ground area of a building to relate to the EPISCOPE building classification, we see the building height as a more widely available data source which can be further improved using gap filling.

Many approaches exist in literature for estimating building heights from data. Those include a simple multiplication of the number of floors by a constant ceiling height, machine learning tree-based approaches, estimation from digital elevation models and more advanced as, for example, estimations based on satellite data. In the context of our FAIRiCUBE use case, three approaches were used to estimate building heights. The city of Halle, Germany, was selected as a test case as we have available the ground truth building height data, the input data to all methods in focus (see also deliverable D3.1) and the published machine learning model could be applied without problems. For larger cities, like Vienna, Austria or Oslo, Norway which were identified originally as target cities, we ran into numerical issue of the published ML method most likely due to the size of the cities. Once we conclude from the city of Halle test case, we revert to the original selected European test cities to allow for synergies with other FAIRiCUBE use case, e.g. UC 1.

We now focus on the first method which is using Open Street Map (OSM) to extract the *number of building floors* and multiplying them by a *constant ceiling height*. Given the limited availability of ceiling height data for Halle, a region around Halle was selected to get a better representation of the ceiling height constant. The dataset contained information of 230,255 buildings of which only 20% (18,837) had the number of levels and just 850 both heights and number of levels. The average level height for the dataset was 2.5 m. In a brute force attempt, we tested different constants ranging from 2.4 m to 4.3 m as potential ceiling heights and calculated the mean absolute error (MAE) and root mean squared error (RMSE). The optimal ceiling height was given with 3.0 m with a MAE of 2.7 m. Later, the dataset was cropped to the outline of the city of Halle and building heights were computed. The results are shown in Figure 38.

[1] https://episcope.eu
[2] https://episcope.eu/fileadmin/episcope/public/docs/reports/EPISCOPE_FinalReport.pdf

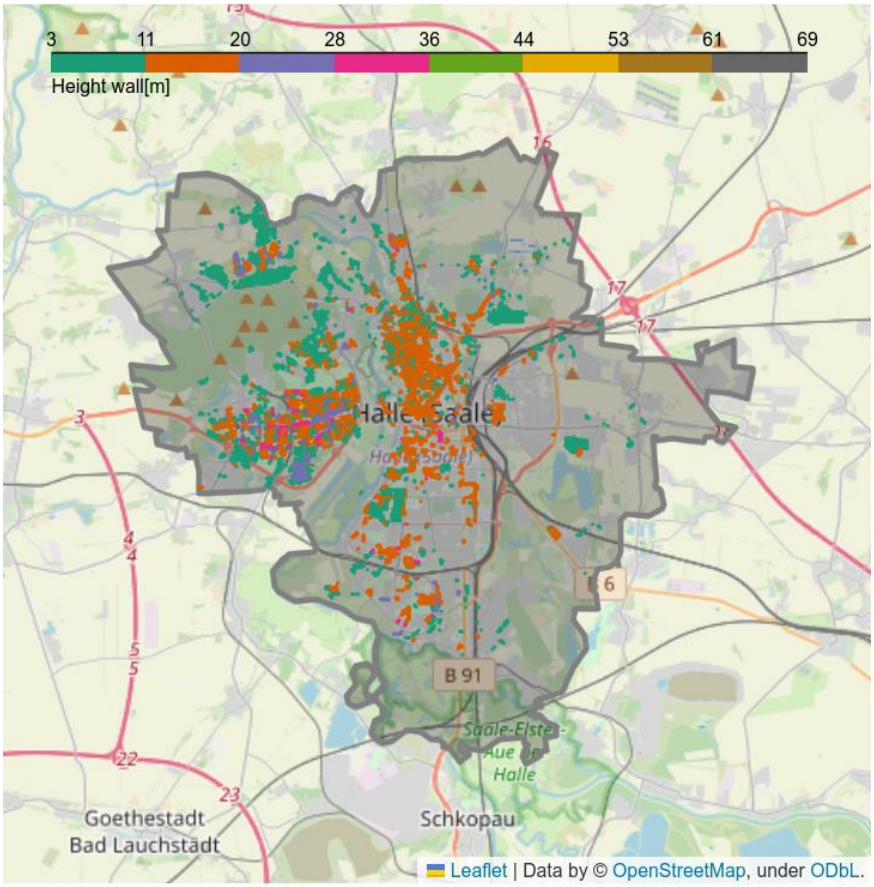


Figure 38 : Building heights [m] estimated by the multiplication of the number of levels by a constant.

Despite its easy implementation, the number of floors times a constant ceiling height method is not always applicable as the number of floors for a whole city is rarely available. Moreover, the method can introduce bias, as the height of the levels is generally heterogeneous and can vary significatively between buildings, city neighborhoods and cities. Regarding machine learning (ML), tree-based algorithms are the most used for this kind of application. In this work we have used a random forest ML algorithm, implemented in the ready-to-use Geoclimate[1] software, a geospatial processing toolbox for environmental and climate studies.

A decision tree is a kind of flow-chart with tree structure with nodes, leaves and branches. On each node, a test is carried out on an attribute of a dataset. The outcome of the test represents a new branch on the tree, and it is stored in a terminal node known as leaf. Then recursively introduce new tests on each of the terminal nodes (leaves) hence splitting the tree until a stopping criterion is met. The last output is compared to a known true value and the loop repeats. A metrics is used to select the paths maximizing information gain. What the random forest does is to combine the output of multiple decision trees to reach a single result with lowest bias and variance in the results.

The random forest model for the estimation of building heights implemented in Geoclimate was trained on a dataset of 14 French communes with reference heights provided by BDTopo IGN. For inference (prediction), the software first retrieves the building layer of Open Street Map (OSM) and to compute 62 geographical indicators from a building's closest environment. These features are fed as features (predictors) into the random forest algorithm to predict building heights. Geoclimate is open

[1] Jérémy Bernard et al., Estimated height of the OpenStreetMap buildings of 24 French communes using the GeoClimate Software (version 0.0.1) (2021), , doi:10.5281/zenodo.5746612.

source. As the machine learning model is available through the paper publication, we can directly apply it to other cities, like our city of Halle, Germany. For a wider application, to especially larger cities, a re-write or at least debugging of the existing ML inference is however necessary. Figure 39 shows the prediction of building heights for Halle, Germany.

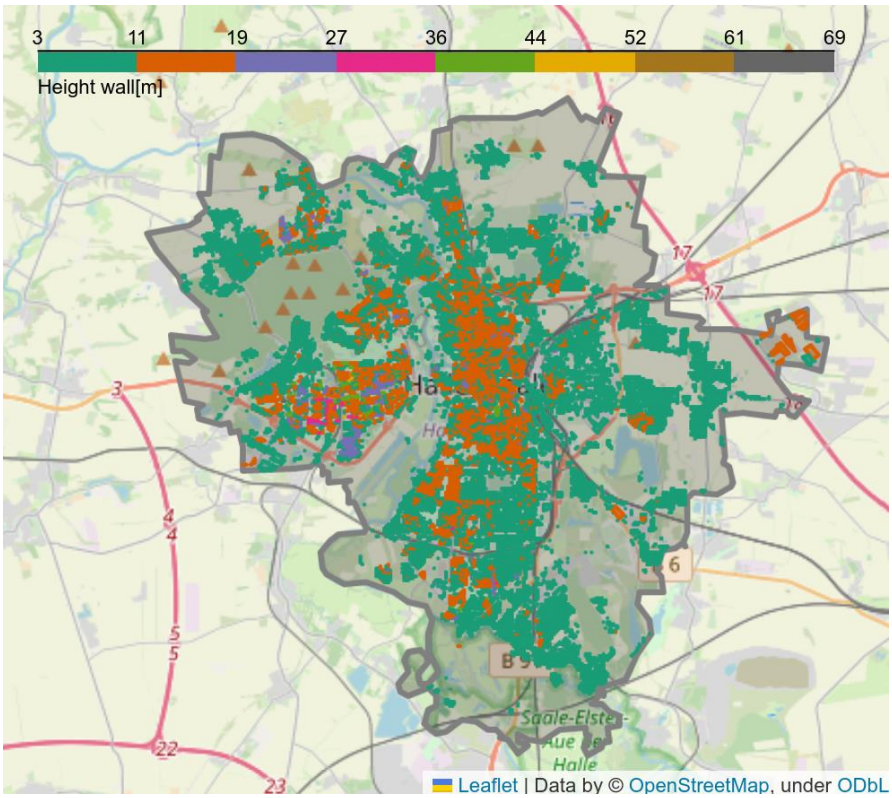


Figure 39 : Building heights estimated by random forest algorithm using the Geoclimate software for the city of Halle, Germany.

Despite the capability of Geoclimate at the current development state, the software still needs fixing some stability issues. Debugging and improving the code can be challenging as it requires knowledge on Apache Groovy, a java syntax-compatible object-oriented programming language for the Java platform not so known in the data science community.

A third approach is using estimating building heights form the difference between a Digital Surface Model (DSM) and Digital Terrain Model (DTM) data as illustrated in Figure 40. Only building heights greater than 3.0m were included in the results shown in Figure 41. Of all the three methods described here, this method is the most stable and least computationally demanding one as it only requires the subtraction of two data layers. However, high resolution DSM data is not widely available in Europe whereas DTM data is available through the digital elevation model (DEM) provided by the Copernicus land monitoring service[1].

[1] https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1

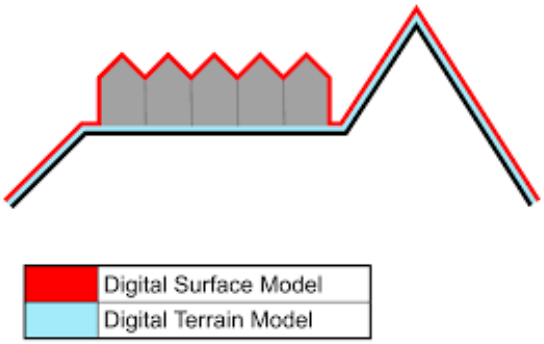


Figure 40 : Illustration of Digital Surface Model (DSM) and Digital Terrain Model (DTM).

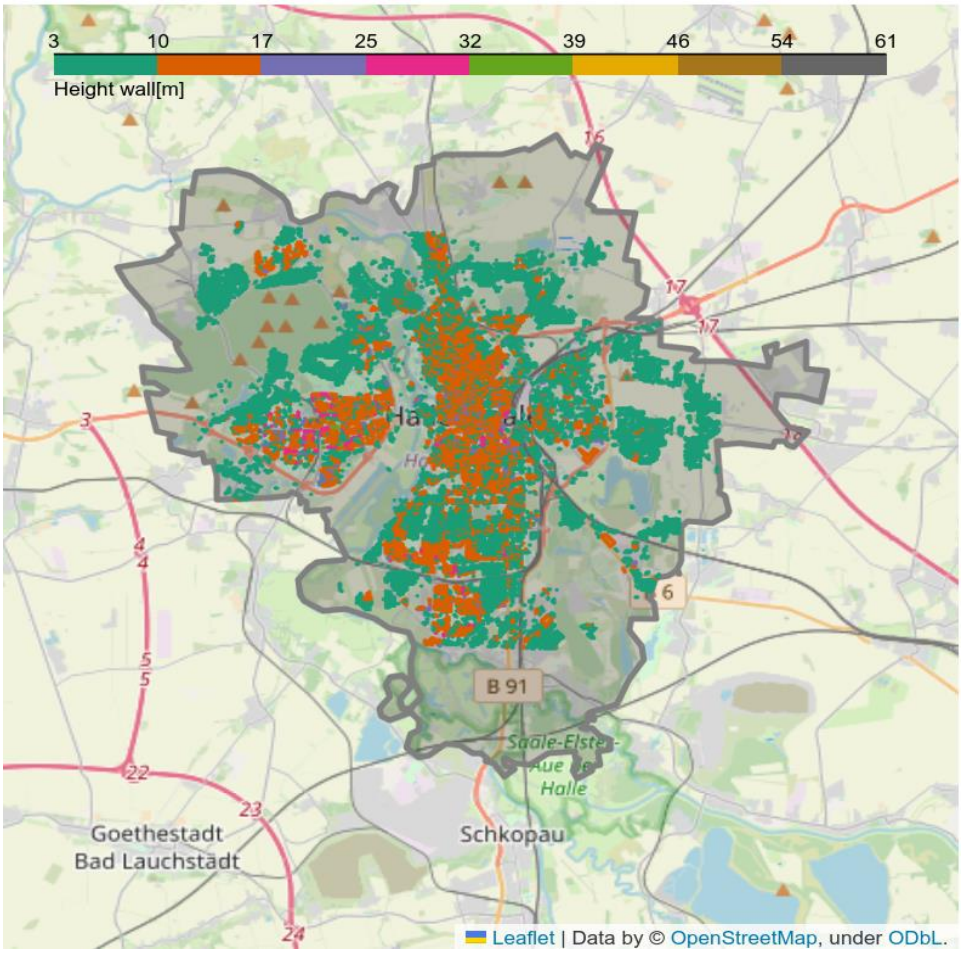


Figure 41 : Building heights estimated by the difference of DSM and DTM for the city of Halle, Germany.

Table 6: Descriptive statistics of building heights in all the GeoTiff layers.

Parameter	Ground truth	Number of levels x factor	Geoclimate	DSM-DTM	Copernicus building height dataset
Max	62.3	45	66.0	30.4	45.0
Mean	8.12	11.9	10.3	9.6	7.7
Min	0.06	6.0	3.4	3.0	3.0

As a final step of the building height estimation task, the output of the three different methods and the Copernicus building height dataset is compared against the ground truth. The ground truth data was provided as 53 separated files in cityGML format which is an extension of the GML standard that handles 3D data such as varying building height information complying to level-of-detail 2 (LoD 2). Note that the ground truth is still not 100% accurate as it is also combines the processed input from several data sources as described in deliverable D3.1. A QGis script was created to convert each GML file into GeoJson and posterior merge into a single GeoJson files. Finally, the merged GeoJson was rasterized and exported as GeoTiff to make it directly comparable to the output of our building height estimation layers. Binary overlap layers between the GeoTiff layer of the ground truth and the results in all the three estimation methods were generated. The different overlap layers were used as a binary mask to extract data from the different estimation results and the ground truth. Table 7 shows a descriptive statistic of the layers overlapping with the ground truth.

Table 7: Descriptive statistics of building heights in all the GeoTiff layers.

Parameter	Ground truth	Number of levels x factor	Geoclimate	DSM-DTM	Copernicus building height
Max	62.3	45	66.0	30.4	45.0
Mean	8.12	11.9	10.3	9.6	7.7
Min	0.06	6.0	3.4	3.0	3.0

To evaluate the accuracy of all estimation methods, the root mean square error (RMSE) with respect to the ground truth data was calculated. The result is presented in Table 8.

Table 8: RMSE in the estimation of building heights by different methods.

Method	RMSE [m]
Number of levels x constant	2.41
Geoclimate-random forest	3.18
DSM - DTM	2.46
Copernicus building height dataset	3.38

Originally, the city of Halle, Germany, was a testcase to perform the building height estimation using several methods. The data availability was good and the provided ML algorithm for the random forest regression was running stable. However, depending on the data availability and completeness for other cities we are not able to recommend a method that will likely succeed in all cases. Naturally, the preferred and recommended approach is to obtain local data from municipalities, similar to the Halle ground truth data. This kind of data seems to be very fragmented within European cities and is not available as a European data layer. The Copernicus urban atlas building height layer[1] is a good

[1] https://land.copernicus.eu/local/urban-atlas/building-height-2012

starting point but is as well an advanced version of the *DST-DTM* method. If local building height data is not available, we can refer to the methods tested in this work. According to Table 4 the lowest RMSE is given by the method *number of levels x constant*. However, given the sparsity of the OSM data *number of building levels*, building heights cannot be estimated for a majority of buildings. A more complete estimation can be provided via subtracting *DSM – DTM* data which however might not widely be available for European cities at our target spatial resolution of at least 10 m. The European data layer digital elevation model (DEM)[1] is for example provided at a spatial resolution of only 25m. From the viewpoint of the data basis, the most stable method is actually the Geoclimate ML method using OSM input data. OSM data may however vary in accuracy and quality due to its crowd source approach. If we can overcome the numerical issues of executing the Geoclimate ML model on large cities, we can obtain a good base line estimation of building heights or number of stories per building.

We now have derived a building stock model according to the LoD 1, e.g., each building is approximated as a building block with volume given by ground area and height. As a next step in our task, we will assign the building volumes with construction year and classify according to the building types as defined by the Episcope data base. Combining all the input layers finally allows us to estimate energy performance, building material composition and additional measures.

i) Processing workflow

The overall progress for UC4, as of December 2024, is presented in Figure 42. Explanation of the data sources and processes are provided below.

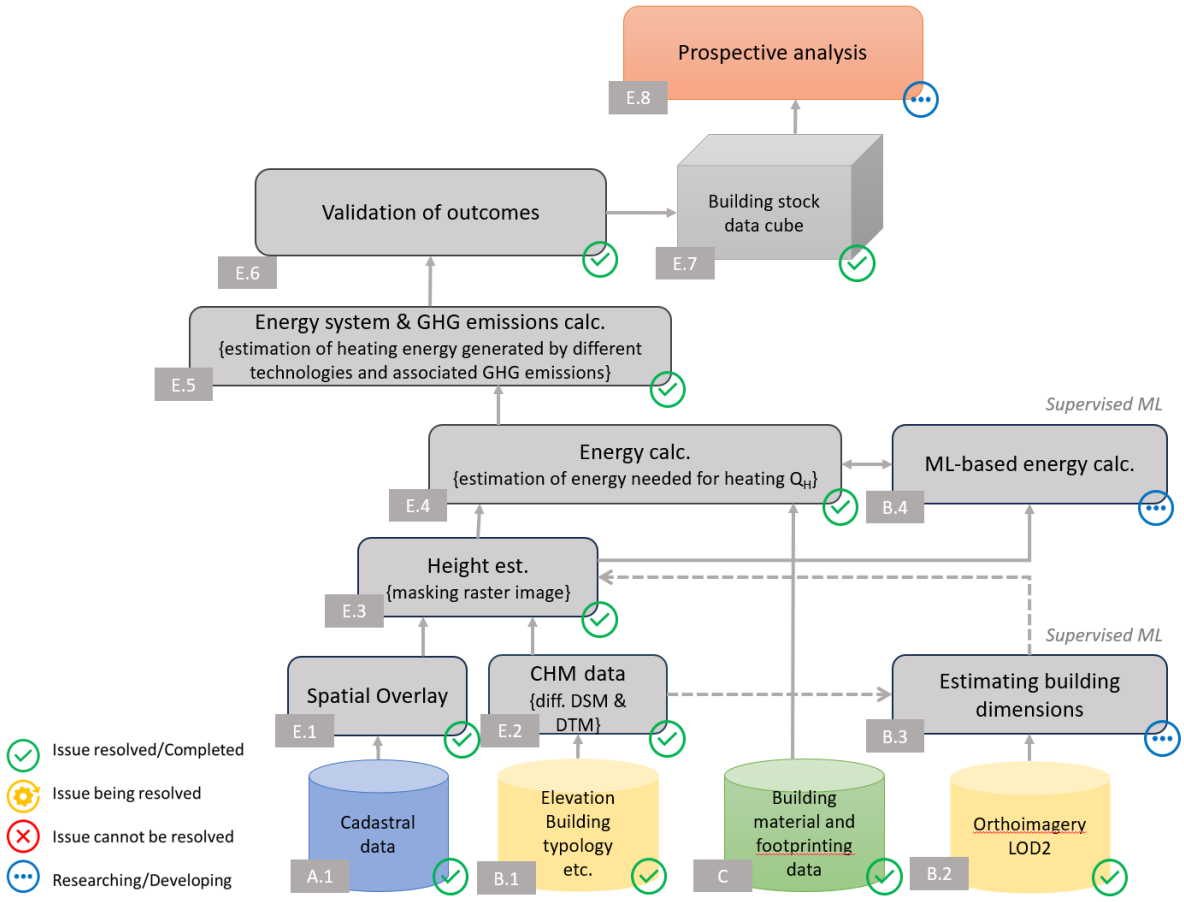


Figure 42 : UC4 data analysis and processing workflow

[1] https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1

Data sources

Ground-based data

A.1. Cadastral data: Here a combination of community-based data, and local data are considered. Open Street Map (OSM) is used to extract building geometries, while local data containing building information like age and functionality is collected from municipalities. From the OSM data only polygons are extracted, meaning that, point and multi-polygon geometries have been excluded. In addition, A set of point data from the municipality of Oslo is collected. The spatial data present a static information about the building cadastre in Oslo, like construction year, building type, number of floors, etc.

Remote sensing data

B.1. Elevation and building typology: Data representing Digital Surface Model (DSM) and Digital Terrain Model (DTM) are used here to estimate building height and subsequently their geometries.

B.2. Orthophoto imagery: It was of interest to test the feasibility of using orthophoto imagery and LOD2 to estimate building dimensions. This was done, as an alternative way to estimate building height.

B.3. Estimating building dimensions: In a carried-out work, the possibility of estimating height of buildings in Vienna was tested. Here, a randomly selected cropped orthophoto images presenting building rooftops are used in conjunction with LOD2 data to train ML models and later test the supervised model on other buildings. It is also of interest to combine some to further develop the trained ML model to design the geometry of building rooftops by using canopy height data as an additional data source.

B.4. Machine learning-based energy calculations: In addition to the EPISCOPE/TABULA model, various machine learning (ML) models are tested to estimate the energy performance of residential buildings. The most effective ML model is then compared against the EPISCOPE/TABULA results to assess the extent of differences between the two approaches.

Tabular data

C. Building archetype information (like environmental footprint per mass of building and heat transfer coefficient of different building elements) is collected from different sources. TABULA/EPISCOPE datasheet is the main source of information here. It contains information about architype building information in certain countries in Europe with their corresponding climate information. In addition, other sources of information are used to estimate in-use mass of materials to estimate availability of secondary construction materials from buildings and estimation of embodied environmental impacts associated with them.

Processing work

E.1. Spatial overlay: Cadastral data are spatially merged here. It might happen a building has a several purpose in its use (like, a building might have grocery store underneath, or coffee shop etc.). In the overlaying process, UC4 exclude building that might have multipurpose/functionality to ensure that the energy balance calculation do not underestimate energy needed for heating.

E.2. Canopy Height Model (CHM): Canopy Height Model contains the height difference between DSM and DTM. However, before this, mosaic of DTM and DSM is created, as the attained raster images are not in one peace.

E.3. Height estimation: Each building geometry is masked over CHM data, and the average height attained from each masked image is stored. The height value is later used to estimate the volume of each residential building and calculate the shaded wall between residential buildings.

E.4. Energy demand calculation: This block estimates the energy needed for each residential building

based on the energy calculation formula provided by TABULA/EPISCOPE[1]. The estimations are based on three archetypical scenarios for each residential building: 1) the as-built design, 2) intermediate renovation, and 3) advance renovation. For newly constructed buildings, the as-built design is compatible with intermediate renovated or advance renovated building. Figure 47 presents the as-built energy demand for space heating for residential buildings in Oslo.

E.5. Energy systems and embodied carbon in building materials: After identifying the demanded energy per building unit, the model further investigates potential energy systems needed to deliver the demanded energy. Here, the suggested energy systems developed by EPISCOPE/TABULA is used. In addition, the model estimates potential secondary building materials and associated GHG emissions from the in-service building materials. Here, a short list of in-service building materials is covered which are materials in roof, wall, floor, window, and door.

E.6. Validation: Based on the outcomes of the model, the energy estimates are cross checked with the claimed/calculated energy performance from official agencies in difference countries. This is necessary to ensure that the estimated energy demand

E.7. Building stock data cube: The outputted modelling results are quarried so that they can represent different information in different raster layers.

E.8. Prospective analysis in conjunction with the renovation-wave roadmap: This building block uses the estimated energy performance of buildings to perform optimization modelling, aiming to identify energy retrofitting priorities based on specific criteria, such as renovation rate, the number of buildings in a neighbourhood, and the scale of retrofitting required. The analysis is designed to align with the broader goals of the Renovation-Wave initiative, ensuring that retrofitting strategies contribute effectively to meeting energy efficiency and climate objectives.

ii) ML for rooftop height estimation

Viewing a satellite image of a rooftop can give insight on its shape, its type and its height. In this task, we have trained an ML model to discover these patterns using the extracted rooftop images for Vienna together with their estimated heights. For this, we have implemented three models, the first is a regression model that estimates the height value from the input image, the other two are classification models that classify the rooftop images into 4 and 2 classes, respectively.

Feature extraction:

To predict the rooftop heights, we first need feature extraction from images using Convolutional Neural Networks (CNNs). CNNs are a type of neural network architecture designed to process inputs as matrices instead of flat vectors, making them ideal for learning from images. This approach allows the model to recognize connections between neighbouring regions in the image during feature extraction. Each layer applies operations that reduce the size of the matrix while preserving essential patterns. Finally, the last layer transforms the reduced matrix into a vector representing the key features of the image. These features can then be used for tasks like classification, regression, or object detection. Training CNNs can be time consuming and require advanced hardware resources. Luckly, nowadays, many pretrained models are available for use. One of which is VGG16, a 16-layer CNN that was trained on millions of images from the ImageNet database.[2] We hence used VGG16 for features extraction of the images and then trained regression and classification models for the final output (rooftop heights).

[1] Diefenbach et al. (2013). TABULA Calculation Method – Energy Use for Heating and Domestic Hot Water. URL: www.building-typology.eu

[2] https://www.image-net.org/download.php

Regression model:

For predicting the rooftop height values from images, we have used multiple models where XGboost has proven to be the best. XGboost is an efficient tree-based regression model that showed efficiency in many real problems. The model takes as input the CNN extracted features and returns the height value. It was trained and tests on over 26,798 images with 20% dedicated to testing. The results are presented in the scatter plot in Figure 43. The model showed acceptable results, reaching Mean Squared Error (MSE) of 1.29, Root Mean Squared Error (RMSE) of 1.14 and an R^2 Score of 0.57. This is much better than a dummy model that always predicts the mean, with MSE = 3.01, RMSE 1.73 and R^2 Score = -0.0002.

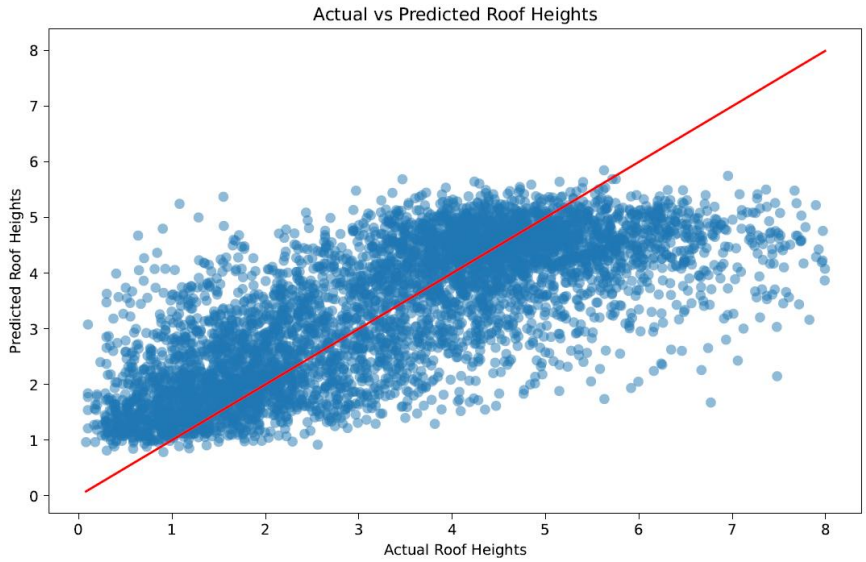


Figure 43 : Actual vs Predicted rooftop heights

Explainable AI:

To understand the patterns learned by our model, we have used the explainable AI tool LIME. LIME (for Local Interpretable Model-Agnostic Explanations) is a generic explainable AI approach that returns the most important features used by a model for predicting on a specific instance. On an image where our model almost correctly estimates the rooftop height of around 4.5 meters, LIME returned the image in Figure 44 highlighting the most important pixels in yellow. We can clearly see that the model learned the triangle shape and the slope as important features for estimating the height, which is consistent and a proof that our model has learned some patterns.

Classification models:

In some real word situations, the user is not interested in predicting the exact rooftop values, but rather, on classifying the rooftops into high or low, etc. We hence implemented classification models in addition to the regression model. For classifying the images into 4 (and then 2) classes, we have split the data among ranges to have a balanced distribution between classes. The height ranges for each class are presented in Table 9 (ranges of 4 classes and 2 classes are separated with a black row). We have tested multiple approaches for classifying the images based on their height and using CNN's extracted features. LightGBM (for Light Gradient-Boosting Machine) has yielded the best results (accuracies on binary classification across multiple models are presented in Table 10). LightGBM is a tree-based ML classifier that is similar to XGboost, but less complex and prioritizes exploring the depth of trees (leaf-wise growth strategy). On four-classes classification, LightGBM, with 1,000 estimators,

achieves an accuracy of 57.47% with training testing split of 80%-20%. This accuracy can be further improved with more data, higher quality images and more accurate estimation of the reference height, but it is already better than a 25% accuracy of a random classifier. On the other hand, LightGBM achieves an accuracy of 86% on binary classification (distinguishing between high and low rooftops). These results are very promising and support our hypothesis on the relevance of satellite images of a rooftops to their height. For more details on the results, please refer to the confusion matrices for four-classes and binary classifications in Figure 45 and Figure 46 respectively.

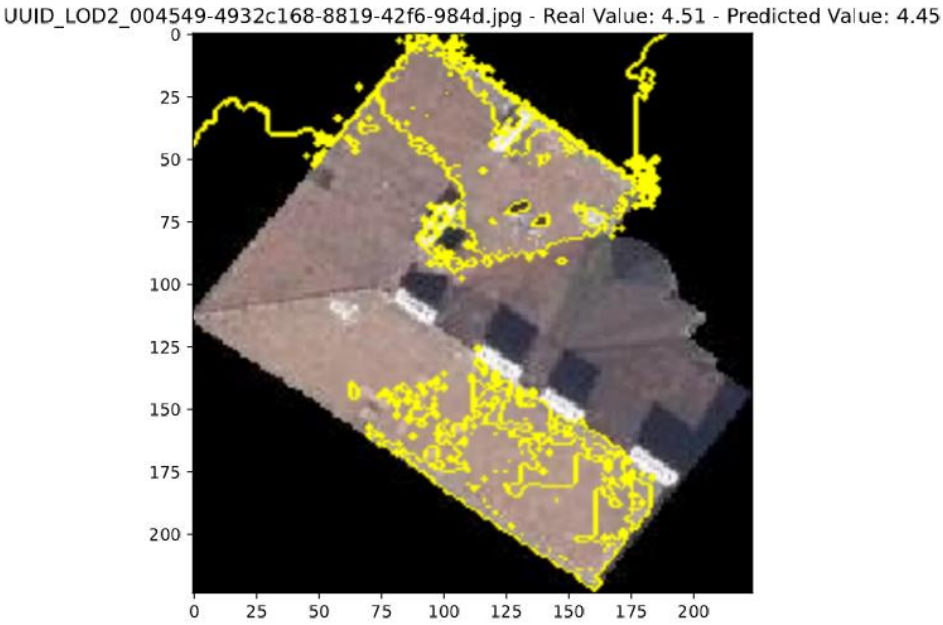


Figure 44 : LIME for explaining rooftop height estimation of our model on a specific image.

Table 9: Rooftop heights range in meters for each class in the data

Class	Range
Class 1	[0, 1.79]
Class 2	]1.79, 3.37]
Class 3	]3.37, 4.59]
Class 4	]4.59, 8]
Class 1	[0, 3.37]
Class 2	]3.37, 8]

Table 10: Accuracy of different model on binary classification of the images.

XGboost	Fandom Forest	Neural Networks	LightGBM
84.78%	83.77%	83.13%	**86.01%**

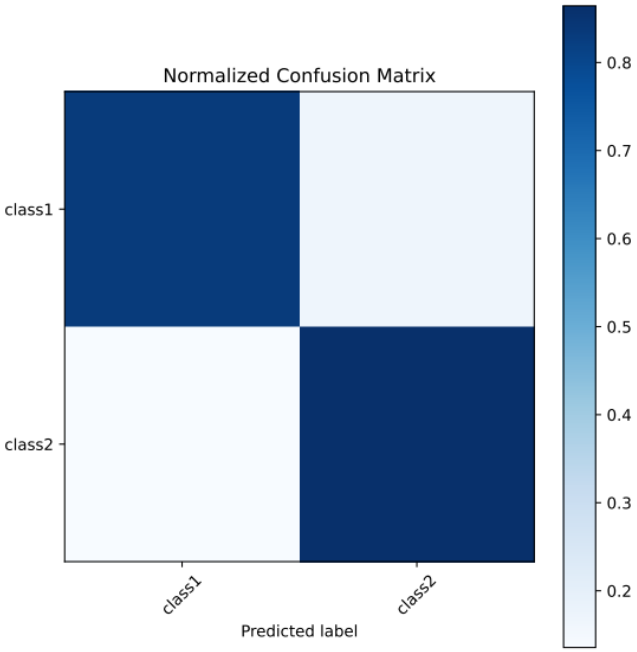


Figure 45 : Confusion matrix on binary classification using LightGBM

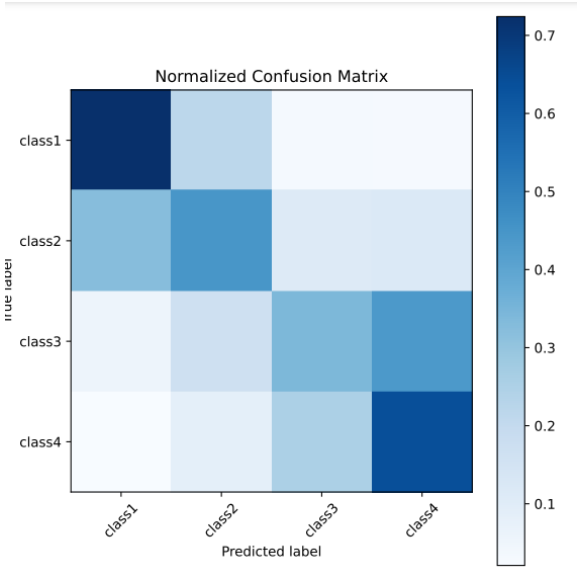


Figure 46 : Confusion matrix on a four-classes classification using LightGBM

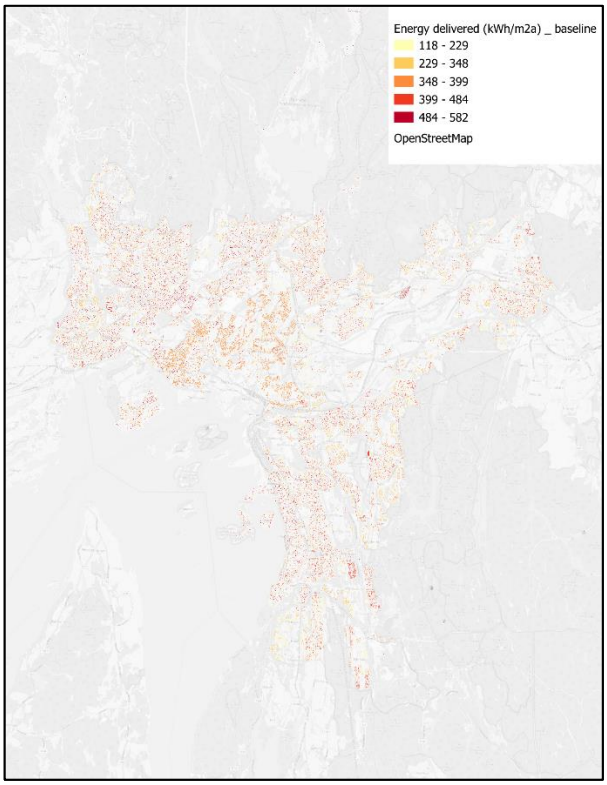
iii) Energy calculation for Oslo, reporting the model and the plot.

The process begins by extracting relevant building attributes from a shapefile, such as geometry, construction year, and building type. The geometry is used to calculate key building dimensions, including roof, floor, wall, and window areas, while the construction year is matched with the EPISCOPE/TABULA dataset to retrieve specific thermal and structural properties. Additionally, the building type (e.g., single-family house, rowhouse, multi-family house) is used to classify buildings into typologies, ensuring consistency with predefined energy performance characteristics.

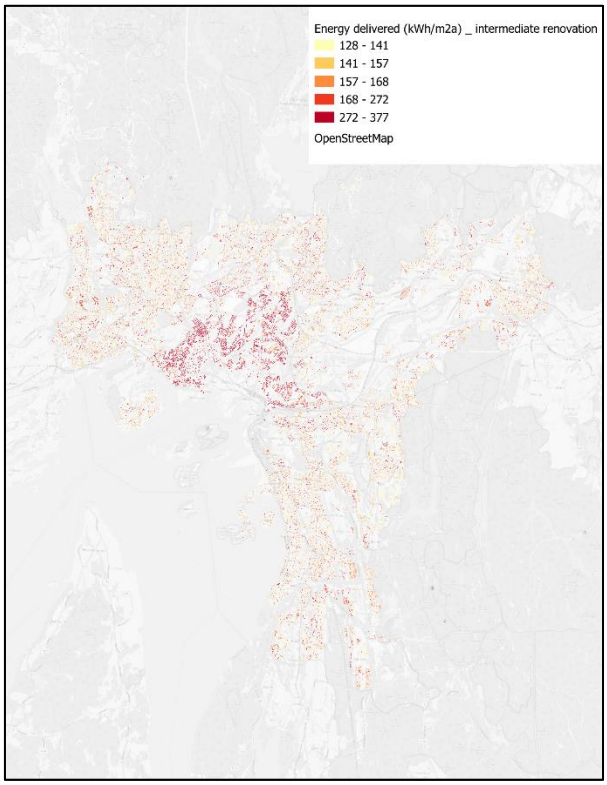
Based on the estimated heights, adjacent wall areas, and building properties (such as number of floors, age, residential type, and location), energy demand for space heating and domestic hot water was calculated for three scenarios: as-built, medium renovation, and advanced renovation. The assumptions for these scenarios were based on the EPISCOPE/TABULA project, which used expert insights and survey data to assign values for variables such as thermal transmittance of materials, heating degree days, internal heat sources, ventilation systems, and solar irradiation.

The energy demand estimates incorporate heat transfer losses through walls, roofs, windows, and floors using U-values and material-specific transmission coefficients. Ventilation heat losses are also included, considering air exchange rates and conditioned air volume. Solar and internal heat gains offset some heating requirements, with solar gains determined by building orientation, shading, and window properties, and internal gains derived from occupant activity and appliances. The results were calculated for the city of Oslo, though validation against observed energy performance is ongoing. Validation will utilize data from self-assessments conducted by Oslo property owners between 2009 and 2023, ensuring that the energy demand estimates align with real-world performance.

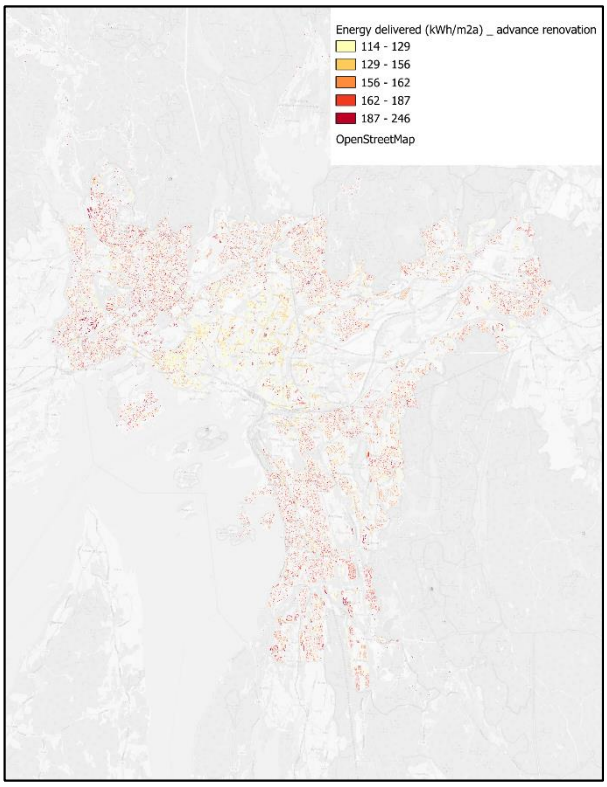
This systematic approach integrates building typologies, geometric properties, and environmental data to produce a robust methodology for estimating residential building energy performance. It not only accounts for physical characteristics and adjacency factors but also ensures that the modelled scenarios reflect realistic conditions for different renovation levels.



(a)



(b)



(c)

Figure 47 : Presentation of the estimated energy delivery to the residential buildings in Oslo for three different systems: (a) as-built energy, (b) mild energy renovation, and (c) deep energy renovation

i) Validation and outlook

The subplot figure in Figure 47 demonstrates the correlation between predicted energy performance (FAIRiCUBE – EPISCOPE/TABULA) and self-reported true values (ENOVA) across various years from 2010 to 2021. The blue dots represent individual data points, while the red line depicts a perfect correlation (ideal 1:1 agreement). Despite adjustments to refine the prediction model, significant scatter is observed, with notable deviations from the diagonal line. This indicates a considerable mismatch between the predicted and self-reported energy performance values, even after excluding extreme outliers with self-declared energy consumption exceeding 600 kWh/m². The MSE values, ranging from 8,718 to 14,335 kWh²/m², highlight the magnitude of the errors, while negative R^2 values across all years confirm that the model predictions perform worse than simply using the mean of the true values.

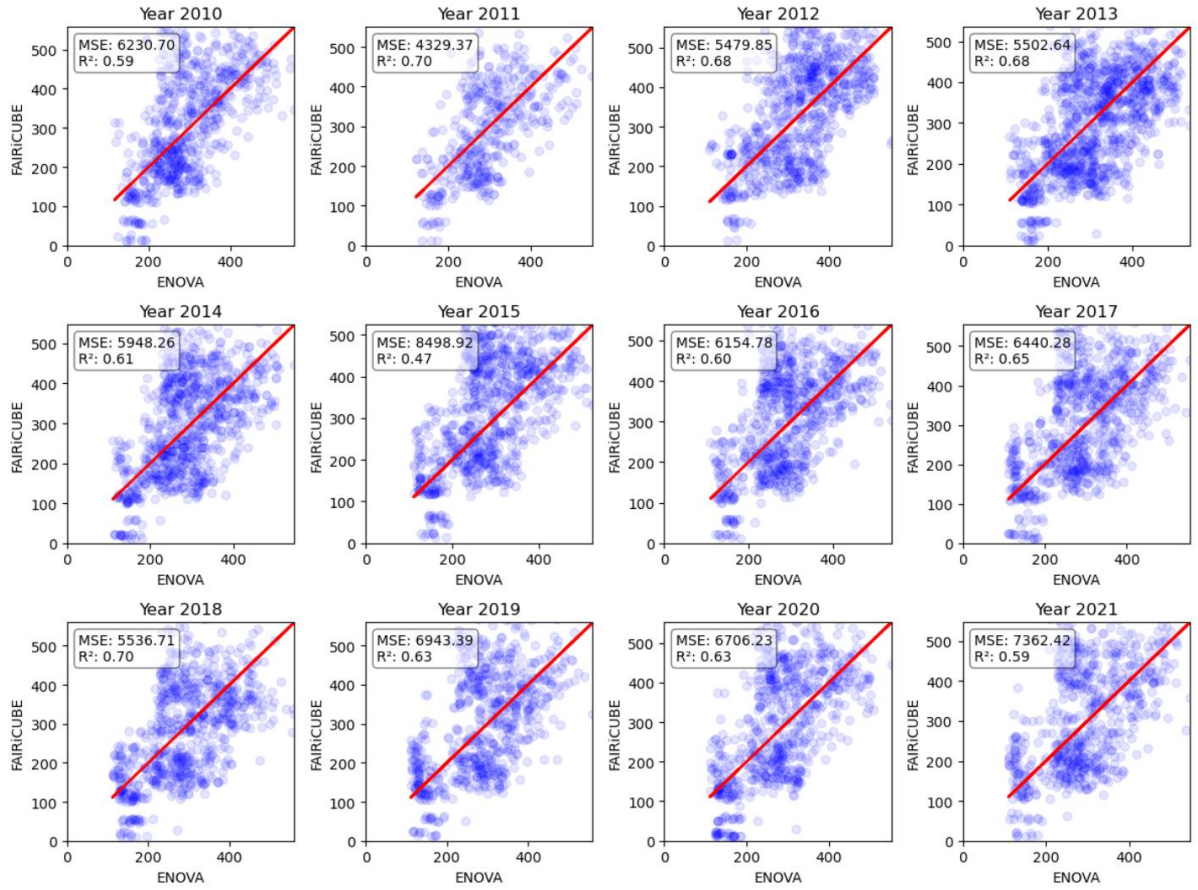


Figure 48 : Comparison of predicted (FAIRiCUBE – EPISCOPE/TABULA) vs. observed (ENOVA) energy performance for residential buildings (2010–2021) with MSE and R^2 values.

In the analysis, 90% of Z-min height was used as the primary variable for height estimation after testing various proportions of building height. Z-min represents the height from the ground to the roof eave, excluding structural elements like additional architectural features above the roof. This adjustment was motivated by the fact that the ENOVA energy calculation model incorporates the internal height of each floor to estimate energy demand, making the slab height at each floor unnecessary for accurate calculations. By focusing on 90% of Z-min, we were able to align the height variable more closely with the parameters used in the ENOVA model, yielding comparatively better results across years. However, even with this refinement, the scatter in the data underscores the challenges of capturing energy performance solely based on building height.

The limitations of the EPISCOPE/TABULA model become evident in this context, as its energy performance predictions are solely dependent on three variables: building dimensions, construction year, and residential building type. While this simplicity offers flexibility and ease of use, it also exposes significant weaknesses in estimating energy performance accurately. The exclusion of other critical factors, such as occupant behaviour and energy efficiency upgrades (retrofitting history), means that the model struggles to provide precise predictions. This lack of granularity underscores the importance of incorporating additional parameters to improve the accuracy and reliability of energy performance estimation in future models.

Based on the progress of UC4, the following activities are expected to take place: (1) identification of the driving forces in energy retrofitting, (2) stochastic modelling of identified drivers on energy retrofitting. Statistical modelling will focus on identifying the key factors influencing the renovation rate of residential buildings, such as income levels, energy prices, age, and education of residents in a neighbourhood. These factors will be evaluated using findings from other scientific studies to determine their importance and potential influence. After identifying the most significant factors, a stochastic model will be developed to simulate the effects of different actions and scenarios, helping to better understand the dynamics and their impact on energy retrofitting.

UC5 Validation of Phytosociological Methods through Occurrence Cubes

This use validates the traditional methods applied in phytosociology and the European Habitat classification system, while developing a new approach to characterize and classify plant communities and predict their distribution in localities with favourable conditions. To achieve this, occurrences of plant taxa and vegetation communities, categorized by habitat types, are integrated with environmental data from Earth Observation (EO) sources and analysed through Machine Learning techniques, such as ensemble modelling, to predict distribution of selected taxon in Europe.

i) Processing and ML workflow

Figure 49 summarizes the processing and ML workflow followed for the UC5. As explained previously in the Deliverable 3.1, a selection of European habitats from the EUNIS classification was chosen to test our method. As a starting point to build the models, the UC5 is focusing on the EUNIS Habitat

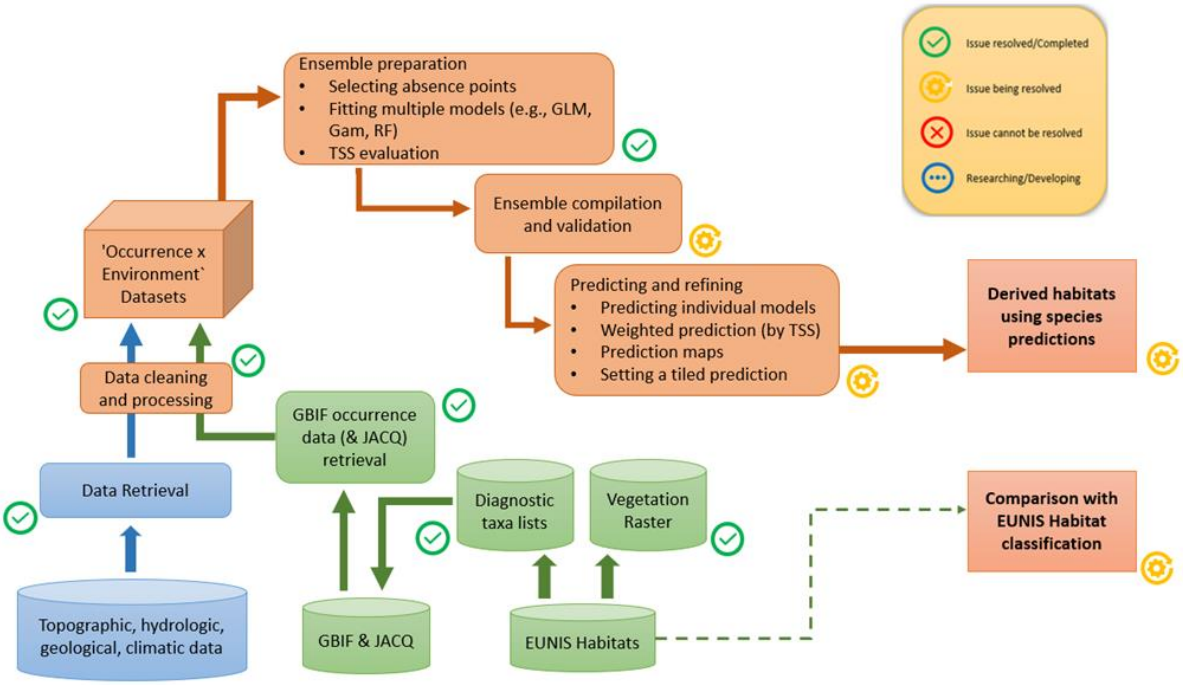


Figure 49 : UC4 data analysis and processing workflow

S22 and its diagnostic taxa. Occurrence data for the diagnostic taxa and the environmental data were retrieved from GBIF and Earth Observation sources respectively.

ii) Processing of Occurrence data

Despite the availability of large and heterogeneous datasets in GBIF, which adhere to established standards, occurrence data from the Information System still pose several challenges. For example, the validation of data records originating from citizen science projects and identification apps is often unreliable from a scientific perspective, or records of living specimens from zoos or botanical gardens may be incorrectly recorded, without the possibility of verification. As a result, multiple steps of data cleaning are required, and even then, such data should be used cautiously, as not all issues can be fully resolved through cleaning or filtering. In our Use Case, after an initial exploration of the occurrence data (Deliverable 3.1), we decided to filter those occurrences recorded outside their typical geographical distributions. These were coming from datasets such as iNaturalist and Pl@ntNet where validation from experts was rarely confirmed. For this reason, we excluded those records outside the typical geographical distributions using the EuroPlusMed Plantbase[1] and the FloraVeg portal[2].

Despite the problems with iNaturalist and Pl@ntNet, we chose to include data from these databases which were recorded in countries listed in the Plantbase. This was necessary because, in case of total exclusion of those, we would have lost a great portion of data. Additionally, for the initial run of our model, we decided to filter out observations marked as preserved specimens, as these might instead represent specimens collected in botanical gardens. Lastly, the data were cleaned to address missing values and duplicates, coordinate reference systems (CRS) were checked for uniformity, and occurrences with location uncertainties greater than 500 meters were excluded. Figure 50 shows the final number of occurrence data obtained for each diagnostic taxon of the habitat S22 after the cleaning and filtering steps of the original GBIF datasets.

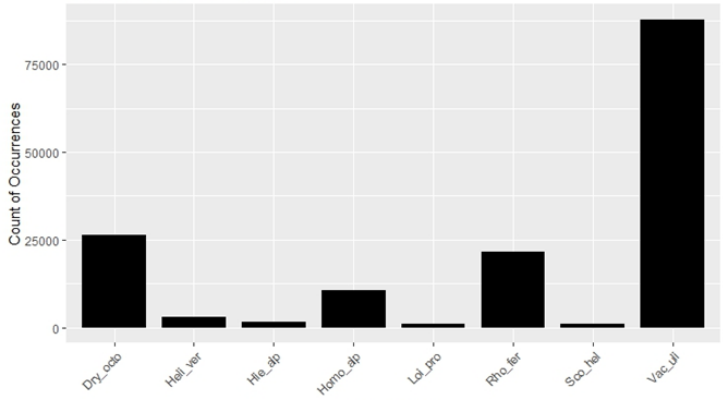


Figure 50 : Number of occurrences of the diagnostic species for the Habitat S22 after cleaning and filtering steps from the raw GBIF datasets. Abbreviations stand for: *Dryas octopetala* (Dry_octo), *Helictochloa versicolor* (Heli_ver), *Hieracium alpinum* (Hie_alp), *Homogyne alpina* (Homo_alp), *Loiseleuria procumbens* (Loi_pro), *Rhododendron ferrugineum* (Rho_fer), *Scorzoneroides helvetica* (Sco_hel), *Vaccinium uliginosum* (Vac_uli).

[1] Euro+Med PlantBase - Preview of the new data portal | Euro+Med-Plantbase
[2] FloraVeg.EU

Figure 51 shows the occurrence data visualization over the raster map of the EUNIS habitat S22 (EEA). Dots in different colours symbolize the diagnostic taxa of the habitat. As we were hypothesizing, many occurrences are present outside of the habitat areas or, especially in southern regions, almost none of the diagnostic taxa are recorded.

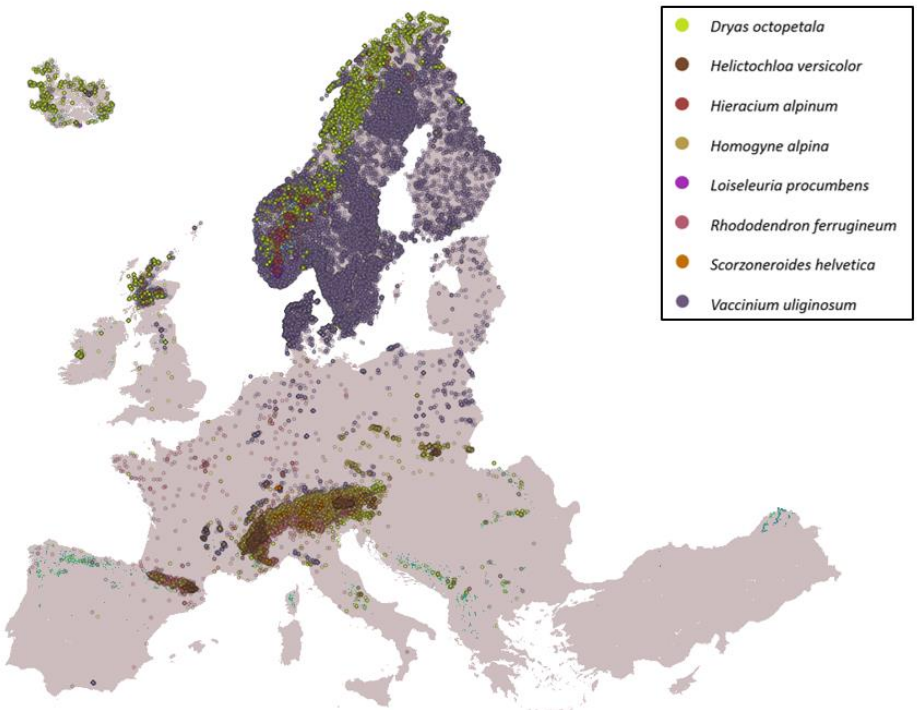


Figure 51 : Data visualization of occurrences of diagnostic taxa of the Habitat S22 over the raster map of the EUNIS habitat (EEA).

Finally, in order to run the individual models and the ensemble model, we derived pseudo-absence data 1-to-1 to the number of presence data, through the disc method with a buffer of 10 km diameter. We needed to use such data, which are artificial absence data generated from specific geographic locations that are assumed to be negative samples, since true absence data are unavailable in GBIF for our taxa list. The 'disc' radius is set to a diameter sufficient to avoid biases arising from spatial overlap between presence and pseudo-absence data, consequently improving the accuracy of the models[1]. Table 11 shows the number of pseudo-absence data obtained through the Disc method, corresponding to the number of presence data.

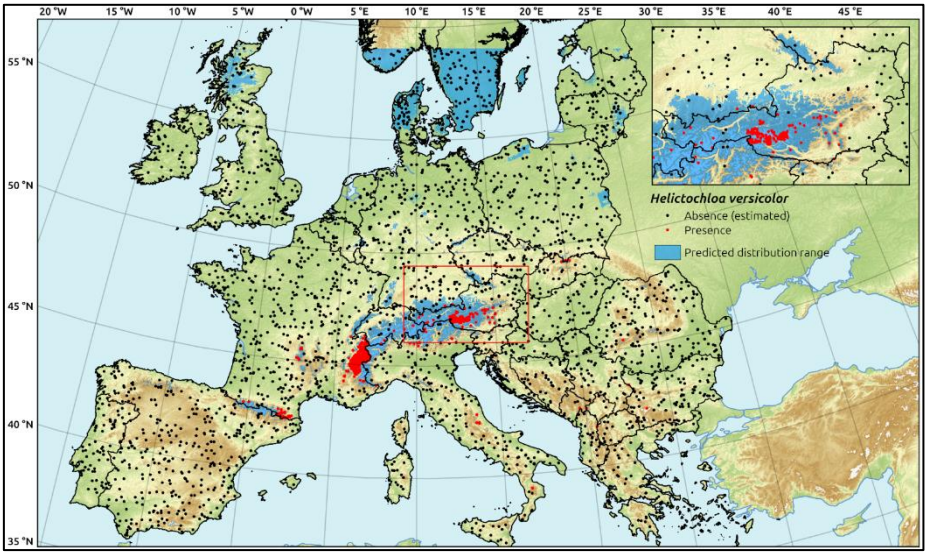
In Figure 52 the predicted distribution ranges (blue areas) for four of the diagnostic species of the habitat S22 are depicted (a, *Helictochloa versicolor*; b, *Homogyne alpina*; c, *Scorzoneroides helvetica*; d) *Vaccinium uliginosum*). Presence and pseudo-absence data are also shown, respectively in red and black dots. It should be noticed how the number of pseudo-absences increase the more presence data are included in the predictions, e.g. *Vaccinium uliginosum* (d).

[1] Xiao, Xiong, Wang., Jing, Liu. (2023). 3. Determining representative pseudo-absences for invasive plant distribution modeling based on geographic similarity. Frontiers in Ecology and Evolution, doi: 10.3389/fevo.2023.1193602

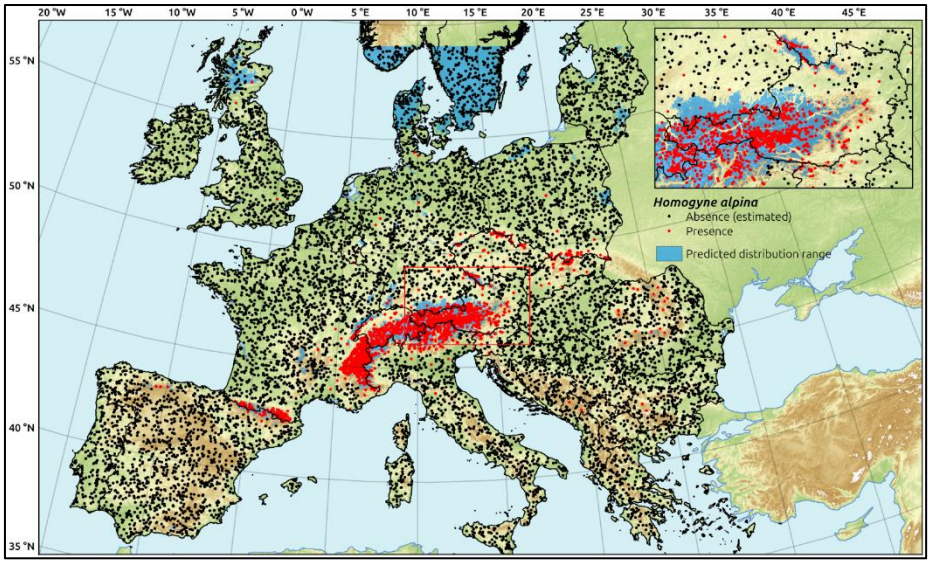
Table 11: Number of presence and pseudo-absence data used to run individual models for the ensemble model.

Taxon	Presence data	Pseudo-absence data
Dryas octopetala	25388	25388
Helictochloa versicolor	3057	3057
Hieracium alpinum	1167	1167
Homogyne alpina	10158	10158
Loiseleuria procumbens	779	779
Rhododendron ferrugineum	21453	21453
Scorzoneroides helvetica	1095	1095
Vaccinium uliginosum	83051	83051

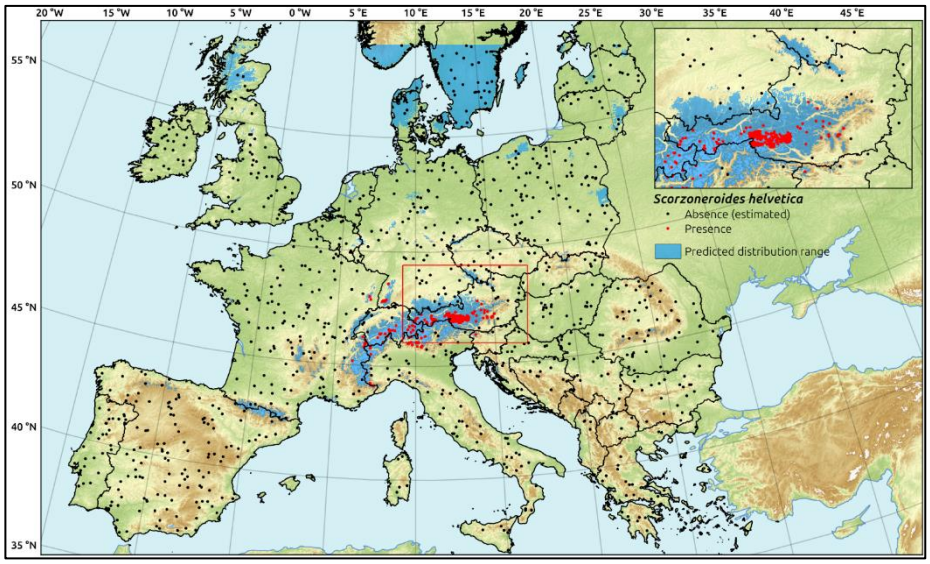
a)



b)



c)



d)

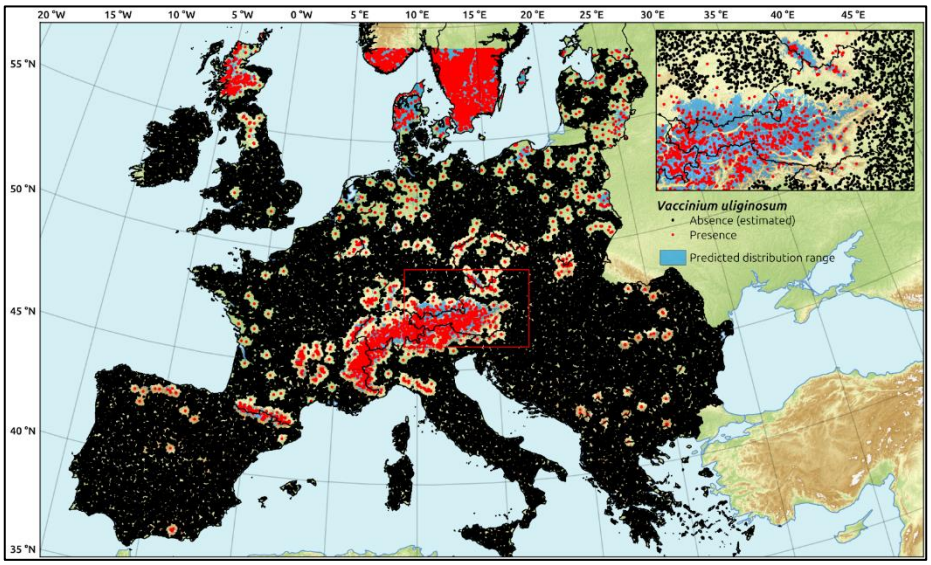


Figure 52 : Geographical distribution of four species of the habitat S22 (a, *Helictochloa versicolor*; b, *Homogyne alpina*; c, *Scorzoneroides helvetica*; d, *Vaccinium uliginosum*). Red dots indicate presence data obtained from GBIF after cleaning and processing steps. Black dots indicate pseudo-absences data obtained through the disc method. Areas in blue correspond to the predicted distribution range.

iii) Processing of environmental data

In addition to the elevation variable, we derived topographical indices from the Digital Elevation Model (DEM) to gain a more comprehensive understanding of the role of topography in taxa distribution. The topographical indices included the terrain ruggedness index (TRI), which indicates landscape roughness; and the heat load index (HLI), which measures the potential heat load based on topography. TRI is computed by applying focal statistics to assess elevation differences from local slopes within a specified neighbourhood around each pixel in the DEM. HLI is determined through trigonometric functions of slope and aspect, estimating solar radiation exposure based on terrain orientation.

A second step in environmental dataset preparation was to reproject the datasets to a resolution of 1 km. Although all the datasets originally had a common resolution of 100 m, we decided to reproject them to 900m to reduce the high processing resource demands associated with a 100 m resolution. Once taxa and environmental datasets were completed, these were combined in a final dataset, in which each occurrence is combined with values on environmental predictors with corresponding coordinates.

iv) Modelling steps

The species distribution modelling (SDM) strategy for this use case involved implementing an ensemble model based on multiple individual models employing regression and machine learning methods. Specifically, the individual models included were Generalized Linear Model (GLM), Generalized Additive Model (GAM) and Random Forest (RF). Such approach has demonstrated that combining multiple individual models produces more accurate predictions than a single model alone[1]. For each individual model, distribution probabilities and binary assessment of presence and pseudo-absences were derived. Predictions resulted from the individual models were then weighted by True Skill Statistics (TSS) scores from 10-fold independent cross-validation and combined into the ensemble model. In this way, the ensemble is based on weighted probabilities resulting in distribution maps which are then converted into a binary assessment used to derive the TSS values. Once the ensemble models are assessed and the distribution ranges are predicted, the last step is to compare our habitat areas obtained with the taxa occurrences with the EUNIS habitats through aggregation method where any occurrence within the 1 km grid is considered as occurrence for the whole grid.

[1] K., Manjusha., Kavya, Jeevan., Shalu, George., Nadirsha, P.S., Nawab., Mukesh, Lal, Das. Anbazhagi, Muthukumar., M., Muthukumar. (2024). Ensemble species distribution model of threatened Cycas L. species of Kannur district and Kerala, India.

3 Summary

The deliverable report outlines the machine learning strategies for the FAIRiCUBE use cases following data ingestion, formulation of the scientific research questions (deliverable D2.2) and the results from the exploratory data analysis (deliverable D3.2). In the following, we list the main points for each use case:

UC1: Urban Adaptation to Climate Change
This use case focuses on harmonizing diverse datasets into structured data cubes and developing a toolkit for analysis and presentation to urban decision makers. At the European level, the strategy involves identifying cities with similar characteristics and analysing factors influencing climate adaptation capacity. At the local level, specialized "city cubes" are used for managing invasive plant species in Luxembourg City and supporting Vienna city initiatives. The machine learning approach includes clustering cities based on land use and socioeconomic data, using algorithms like k-means, Mean-Shift, and Agglomerative Hierarchical Clustering (AHC).

UC2: Agriculture and Biodiversity Nexus
The strategy here is to evaluate and quantify the correlation between agriculture and biodiversity using causal machine learning (CML). The approach involves collecting data on biodiversity, environmental factors, and agricultural practices, and applying CML techniques to identify causal relationships. The workflow includes species distribution modelling using MaxEnt species suitability modelling, and causal modelling to validate and assess the robustness of expected causal factors. The final goal is to provide insights into how agricultural practices impact biodiversity.

UC3: Biodiversity Occurrence Cubes – Drosophila Landscape Genomics
This use case leverages the massive collection of DNA-sequenced data of Drosophila melanogaster populations. The machine learning strategy involves gap filling of missing information in genomic data using clustering and deep learning methods. Initial clustering is performed using k-means, followed by more advanced methods like Variational Autoencoder (VAE) and Generative Adversarial Networks (GAN) for improved accuracy. The goal is to enrich the dataset and reveal further insights into the genetic variation of Drosophila populations.

UC4: Spatial and Temporal Assessment of Neighbourhood Building Stock
The focus here is on estimating building heights and energy performance using various methods. The strategy includes using Open Street Map (OSM) data, random forest algorithms, and digital elevation models (DSM and DTM) to estimate building heights. The estimated heights are then used to calculate building volumes and energy performance. The workflow involves data preprocessing, feature extraction, and machine learning-based energy calculations. The final goal is to create a building stock model that can be used to estimate energy performance, climate gas emissions, and building material composition.

UC5: Validation of Phytosociological Methods through Occurrence Cubes
This use case validates traditional phytosociological methods and develop a new approach to characterize and classify plant communities. The strategy involves integrating occurrence data of plant taxa with environmental data from Earth Observation (EO) sources and applying machine learning techniques like ensemble modelling. The workflow includes data cleaning, processing, and modelling steps to predict the distribution of selected taxa. The final goal is to compare the predicted habitat areas with the EUNIS habitat classification.